March 28, 2012

Attn: Prof. David Sheinberg
Dear Chair and Faculty Search Committee:

My name is Zhe (Sage) Chen, a senior postdoc currently working at the Neuroscience Statistics Research Laboratory (directed by Prof. Emery Brown) at MIT/MGH/Harvard Medical School.

I received my formal PhD education in Electrical and Computer Engineering, but I have conducted intensive post-doc training in bioengineering and neuroengineering. My research areas are highly interdisciplinary, and my research interests focus on **Neural Signal Processing, Neural Engineering, Computational Neuroscience,** and **Computational Statistics and Machine Learning.** In collaboration with experimental neuroscientists, I have been applying novel computational approaches to study functions of neural circuits and their link to behavior. In the past 5 years, I have been assistive in supervising dozens of undergraduate/graduate students for various research projects undertaken in the hosted labs. I have teaching experiences in both undergraduate and graduate-level courses in traditional electrical engineering and other related fields (e.g., Bayesian estimation, Statistics in Neuroscience).

I am interested in applying for a tenure-track faculty position in Computational Neuroscience at the Department of Neuroscience at Brown University. Here I enclose my application package that includes

1) Cover Letter
2) Comprehensive CV with a complete publication list
3) Current and Future Research Projects & Proposals
4) Statement of Research and Teaching Philosophy
5) Contact information of three References
6) Three representative journal publications

Shall you have any question about me or the application material, please let me know by email. Thanks!

Best regards

Zhe (Sage) Chen, PhD
zhechen@mit.edu
Office Phone: 617-324-1882

Curriculum Vitae

Date Prepared: March 28, 2012

Name: Zhe Chen

Office Address: Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, 43 Vassar Street, Bldg. 46-6057, Cambridge, MA 02139

Department of Anesthesia, Critical Care and Pain Medicine, Massachusetts General Hospital, 55 Fruit Street, Jackson 4, Boston, MA 02114

Home Address: 112 Marlborough Street, Apt. 3B, Boston, MA 02116

Work Phone: 617-324-1882

Work E-Mail: zhechen@mit.edu zhe.sage.chen@gmail.com

Work FAX: 617-726-8410

Education

1997 B.Sc. Electrical Engineering, Ocean University of China, Qingdao, China
2000 M.E. Electrical Engineering, Ocean University of China, Qingdao, China
2005 Ph.D. Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada (Supervisor: University Prof. Simon Haykin, PhD)

Postdoctoral Training

06/2005-02/2007 Research Scientist RIKEN Brain Science Institute, Wako, Japan
03/2007- Senior Research Fellow MGH-HMS-MIT (Mentor: Prof. Emery N. Brown, PhD, MD)

Academic Appointments

01/2000-12/2004	Research Assistant	Dept of ECE	McMaster University
01/2005-06/2005	Research Associate	Dept of ECE	McMaster University
06/2005-02/2007	Research Scientist		RIKEN Brain Science Institute
03/2007-	Research Fellow	Dept of Anesthesia and C.C.	Harvard Medical School, Boston, MA
03/2007-	Research Affiliate, Dept of Brain and Cog. Sci.		Mass. Institute of Technology, MA
01/2012-	Research Associate, MIT Intelligence Initiative		Mass. Institute of Technology, MA

Appointments at Hospitals/Affiliated Institutions

03/2007- Research Engineer Dept of Anesthesia and C.C. Mass. General Hospital, Boston, MA

Professional Committee Services

2008 Program Committee and Area Chair, Seventh Mexican International Conference on Artificial Intelligence (MICAI'08), October 27-31, 2008, Mexico City.

2009 Program Committee, Eighth Mexican International Conference on Artificial Intelligence (MICAI'08), Nov. 9-13, 2009, Guanajuato.

2010 Program Committee, Ninth Mexican International Conference on Artificial Intelligence (MICAI'10), November 8-13, 2010, Pachuca, Mexico.

2009 Program Committee, First Bernstein Conference on Computational Neuroscience, Oct. 2009, Frankfurt, Germany.

2009 Special Session Co-organizer and Session Chair, "Signal Processing for Neural Spike Trains", IEEE ICASSP, April 24, 2009, Taipei.

2010 Special Session Co-organizer and Session Chair, "Multivariate and Multimodal Analysis of Brain Signals: Methods and Applications", IEEE ICASSP, March 16, 2010, Dallas, TX.

2011 Special Session Co-organizer and Session Chair (with Dr. ShiNung Ching) on "Computational Methods for Modeling Sleep and Anesthesia", EMBC'11, Aug 30-Sept 3, Boston, MA.

2011 Program Committee, Tenth Mexican International Conference on Artificial Intelligence (MICAI'11), Nov 26-Dec 4, 2011, Puebla, Mexico.

2012 Invited Session Co-organizer (with Drs. Todd Coleman and Rui Ma) on "Neural Decoding: New Paradigms and Open Challenges", EMBC'12, Aug 28-Sept 1, San Diego, CA.

2012- Editorial Board Member of ***Journal of Biological Research (HongKong)***

Professional Societies

2008 - Member, Society for Neuroscience
2009 - Member, Biomedical Engineering Society (BMES)
2007-2010 Member, Institute of Electrical and Electronics Engineers (IEEE)
2010 - Senior Member, IEEE (elected)

Grant Review Activities

2006 Dutch Research Council (Technology Foundation STW)

Editorial Activities

- Editorial Board

  Journal of Biological Research (HongKong)
  (http://www.ghrnet.org/index.php/JBR/about/editorialTeam)

- Guest Editor

  Special Issue on "Signal Processing for Neural Spike Trains", *Journal of Computational Intelligence and Neuroscience*, 2010.

  Research Topic on "Engineering Approaches to Study Cardiovascular Physiology: Modeling, Estimation, and Signal Processing", *Frontiers in Computational Physiology and Medicine*, 2012.

- Book Reviewer

  2009 Wiley Book Series on Adaptive Systems
  2009 Artech House Series on Biomedical Engineering

2010 Springer Book Series on Statistics

- Ad hoc Reviewer

2002- IEEE Transactions on Signal Processing
2002- IEEE Signal Processing Letters
2004 Proceedings of the IEEE
2003- EURASIP Journal of Applied Signal Processing
2005 IEEE Transactions on Image Processing
2006- IEEE Transactions on Biomedical Engineering
2007- IEEE Transactions on Neural Systems & Rehabilitation Engineering
2002- IEEE Transactions on Neural Networks
2005- Neural Computation
2003- Neurocomputing
2006- Computational Intelligence and Neuroscience
2007- Cognitive Neurodynamics
2007- Journal of Computational Neuroscience
2007- Journal of Neurophysiology
2007- Journal of Neural Engineering
2007- Journal of Neuroscience Methods
2011- Network: Computation in Neural Systems
2009- Annals of Applied Statistics
2009- Annals of Biomedical Engineering
2008- Journal of Biomedical Science and Engineering
2008 International Journal of Tomography & Statistics
2008 AEUE International Journal of Electronics and Communications

Awards and Honors

2000 YUDAFU Fellowship for Most Outstanding Graduates, Ocean Univ. China
2001 Entrance Scholarship, McMaster University
2001-2002 Clifton W. Sherman Scholarship in Sciences and Engineering, McMaster University
2002,2004 GSA Travel Grants, McMaster University
2002 IEEE Neural Networks Society Walter Karplus Student Summer Research Award
2008,2010 NSF Travel Grant for Statistical Analysis of Neuronal Data (SAND) Workshop
2009 Partners in Excellence Award, Massachusetts General Hospital
2010 *Marquis Who'sWho in America* (elected)
2010 Senior Member of IEEE (elected)
2011 Included in *Dictionary of International Biography*
2011 Elected in ``2000 Outstanding Intellectuals of the 21st Century", International Biographical Center, Cambridge, England.
2012 Early Career Award at Mathematical Biosciences Institute, Ohio State University

Report of Funded and Unfunded Projects

Funding Information

- *Past*

2002 Sole Investigator IEEE Neural Networks Society Walter Karplus Student Summer Research Award $6500

Proportion Adaptation for Echo Cancellation

The major goal of this summer project was to develop new adaptive echo cancellation algorithm for Voice-IP telephone communications. The project won an "outstanding" rank in the final evaluation.

2007-2010 Principal Researcher NHLBI RO1HL084502 $1,250,000

Point Process Models of Human Heart Beat Interval Dynamics (PI: Dr. Riccardo Barbieri)

The major goal of this project was to implement and validate a statistical point process model of cardiovascular control in order to provide new definitions of heart rate and heart rate variability that could have important implications for research studies of cardiovascular and autonomic regulation and for heart rate monitoring in clinical settings.

- *Current*

2010- Researcher NIH DP1-OD003646 $2,500,000

Understand General Anesthesia via a Systems Neuroscience Approach (PI: Prof. Emery Brown)

The major goal of this project is to use various statistical modeling methods to characterize the electrophysiological data or behavior data recorded from human subjects or animals during general anesthesia.

2012-2013 Principal Investigator MBI Early Career Award $7,000/month × 9 months

Various research projects related to mathematical neuroscience.

Current Unfunded Projects

2010- Co-Principal Investigator

Transductive Neural Decoding Approach for Inferring Rat Hippocampal Population Neuronal Codes

In collaboration with Prof. Matthew Wilson (MIT), we (i) develop spike sorting-free algorithm for Bayesian decoding, as applied to rat hippocampal CA1 recordings; (ii) addition, develop a Bayesian algorithm for discovering spatial topology represented by the population neuronal codes; (iii) apply the analysis to data during periods of sleep.

2011- Principal Researcher (NIH RO1 grant proposal) $2,500,000

Statistical Analysis of Spontaneous Miniature Postsynaptic Currents (PI: Prof. Emery Brown)

In collaboration with Prof. Emery Brown (MIT), Prof. Martha Constantine-Paton (MIT), and Dr. Marnie Phillips (George Washington University), we develop statistical tools and signal-processing algorithms for unified feature analysis of miniature postsynaptic currents using intracellular sliced recordings and whole-cell patch clamp technique.

2012-2013 Group Member (MIT Intelligence Initiative Proposal for Seed Funding) $100,000

Life-long Mapping

Working with Prof. Matt Wilson (MIT-BCS) and Prof. John Leonard (MIT CSAIL), we are investigating the connections between SLAM (simultaneous localization and mapping) formulation in mobile robotics, neurophysiological spatial representations and the development of computational algorithms that may relate the two approaches, with immediate exploration of the application of the newly developed Bayesian HMM approach (Chen et al., *J. Compu Neurosci*, 2012) for the analysis of neural spatial representations.

Report of Local Teaching and Training

Teaching of Students in Courses

1998 Signals & Systems, Ocean University of China
2000 Neural Networks Theory and Applications, Ocean University of China
2003 Neural Networks, McMaster University, Canada
2004 Adaptive Filter Theory and Sequential Bayesian Estimation, McMaster University, Canada
2005 Adaptive Filters, Saitama Institute of Technology, Japan
2011 Statistics in Neuroscience, Massachusetts Institute of Technology (Course 9.07), USA

Formally Supervised Trainees

- *Undergraduate and Graduate Students*

2004-2005 Research Preceptor. "Adaptive hearing systems". **Kevin Kan.** Master student, Communications Research Lab, McMaster University.
2004-2005 Research Preceptor. "Particle filtering and state-space estimation". **Ulas Gunturkun.** PhD student, Communications Research Lab, McMaster University.
2006 Research Preceptor. "Portable EEG-based motor-imagery brain-computer interfaces". **Wei Wu**, visiting Master student, Tsinghua University.
2008 Research Preceptor. "Spiking modeling for hippocampal place cells", **Tobias Denninger**, visiting Master student at MIT.
2008 Technical Advisor. "Spike sorting data anlyaiss". **Jessica Chemali.** Visiting undergrad student at MIT
2008 Technical Advisor. "Point process modeling for auditory nervous cells". **Andrea Trevin.** Visiting Master student from Univ. of Illinois.
2008 Technical Advisor. "Neural decoding". **Uma Bhushan**. Visiting Master student at MIT.
2008-2010 Technical Advisor. "System identification and estimation, and point process modeling for spike train data". **Iahn Cajigas.** Harvard-MIT HST PhD/MD student.

2008-2010 Research Preceptor. "Machine learning algorithms for multi-channel EEG signal discrimination analysis". **Wei Wu**. Visiting PhD student, MIT.
2008-2010 Technical Advisor. "Causality analysis between multiple neural spike trains". **Feng Chen**. Visiting PhD student, MIT.
2009-2011 Technical Advisor. "Neural encoding and decoding with GLM point process model". **Demba Ba**, PhD student, Dept. EECS, MIT.
2009 Technical Advisor. "EMG data analysis using a filtered point process model". **Pavitra Krishnaswamy,** PhD student, HST, MIT.
2010-2011 Technical Advisor. "Maximum likelihood estimation of mPSC data". **Jen Gong**, Undergraduate student, Dept. Applied Mathematics, Harvard University.
2011 Technical Advisor. "Variational EM algorithm for state-space model". **Patrick Stokes,** PhD student, HST, MIT.
2011 Technical Advisor. "Statistical modeling of rise time of mPSC events". **Amanda Du**, Undergraduate student, Statistics Department, Boston University.
2011-2012 Technical Advisor. "Decoding the rat's hippocampal population codes". **Stuart Layton,** PhD student, Dept. BCS, MIT.
2012 Technical Advisor. "Deconvolution of cortisol time series data". **Rose Faghih**, PhD student, Dept. EECS, MIT

- *Medical Students/Residents/Fellows/Post Docs*

2009 Technical Advisor. "Analysis of multi-unit activity of rat somatosensory cortex". **Kevin Wong.** Massachusetts General Hospital
2009- Technical Advisor. "Assessment of functional connectivity of neuronal ensemble neurons from cat primary motor cortex". **David Putrino**. Massachusetts General Hospital
2010- Technical Advisor. "Characterization of sleep-awake transition of rat's sleep". **Francisco Flores**. Massachusetts General Hospital

Report of Regional, National and International Invited Teaching and Presentations

Regional, National and International Invited Presentations and Courses

- *Regional (Local)*

2008 "Probabilistic models for estimating neuronal 'UP/DOWN' states", MIT Picower Institute Seminar, Cambridge, MA, February 21.
2008 (**Invited)** "State-space models and estimation for neural data", Department of Biostatistics, Harvard School of Public Health, Boston, Septempter 10.
2009 (**Invited)** "A point process framework to assess heartbeat dynamics and autonomic cardiovascular control", MIT Lab for Computational Physiology, Cambridge, MA, March 16.
2009 (**Invited)** "Decision making and behavior learning: statistical vs. reinforcement learning approaches", MIT Graybiel Lab, Department of Brain and Cognitive Sciences, March 25.
2011 (**Invited)** "Inferring functional connectivity between ensemble neurons with sparse spiking data", Dept. Mathematics and Statistics, Boston University, Boston, MA, February 16.

2011 "Inferring population codes in rat hippocampus with a hidden Markov model", MIT Wilson Lab, Department of Brain and Cognitive Sciences, May 31.
2011 (**Invited)** "Inferring population codes in rat hippocampus using a hidden Markov model", Children's Hospital/Harvard Medical School, September 14.
2011 (**Invited)** "Inferring population codes in rat hippocampus via transductive neural decoding", MIT Bizzi Lab, Cambridge, MA, December 9.
2012 "Uncovering spatial topology represented by rat hippocampal population neuronal codes in navigation", MIT Picower Institute Seminar, Cambridge, MA, February 23.
2012 "Uncovering embedded spatial topology represented by rat hippocampal population neuronal codes in navigation", Boston CRC Meeting, Cambridge, MA, March 6.

- *National*

2007 (**Invited)** "Learning, regularization, and kernels", Department of Psychology, University of Michigan, Ann Arbor, MI, May 2.
2009 "Efficient spike encoding for mapping visual receptive fields", Computational and System Neuroscience (COSYNE), Salt Lake City, UT, Feb. 28.
2009 (**Invited)** "Precise mapping of visual receptive field by tomographic reconstruction", Dept. Electrical and Computer Engineering, University of Minnesota, April.
2009 "Linear and nonlinear quantification of respiratory sinus arrhythmia during propofol general anesthesia", Annual IEEE-EMBS Conference, Minneapolis, MN, September 5.
2010 "Hierarchical Bayesian modeling of inter-trial variability and variational Bayesian learning of common spatial patterns from multichannel EEG", IEEE ICASSP, Dallas, TX, March 16.
2010 (**Invited)** "Precise mapping of visual receptive field by tomographic reconstruction", Dept. Statistics and Center for Theoretical Neuroscience, Columbia University, NYC, July 23.
2011 "Assessing neuronal interactions of cell assemblies during general anesthesia", Annual IEEE-EMBS Conference, Boston, MA, September 1.
2011 "Instantaneous assessment of autonomic cardiovascular control during general anesthesia", Annual IEEE-EMBS Conference, Boston, MA, September 3.
2012 (**Invited)** "Decoding and uncovering rat hippocampal ensemble neuronal codes", Dept. Electrical and Computer Engineering, University of California, San Diego, March 22.

- *International*

2003 "A tutorial of particle filters", Department of Signals, Systems and Radiocommunications, Universidad Politecnica de Madrid, Spain, December 15,
2004 (**Invited)** "Stochastic correlative learning and particle filtering", Universidade Federal da Bahia, Brazil, October 8.
2004 (**Invited)** "Stochastic optimization approaches for correlation-based learning", Max-Planck Institute for Biological Cybernetics, Tubingen, Germany, November 30.
2005 (**Invited)** "Monte Carlo approaches for Bayesian inference and optimization", Deutsche Telekom T-Laboratories, Berlin, Germany, January 15.
2005 (**Invited)** "Sequential Monte Carlo inference and neural network-based black-box modeling", ZENON Environmental Inc., Oakville, Canada, January 28.

2005 (**Invited)** "Improved particle filtering schemes for tracking and communications", Dept. Electrical and Electronic Engineering, Imperial College London, London, United Kingdom, July 7.

2006 (**Invited)** "Adaptive filters", Department of Electronics and Information Engineering, Saitama Institute Technology, Japan, July 4.

2006 "Contrast functions of non-circular and circular sources separation in complex-valued ICA", International Joint Conf Neural Networks, Vancouver, BC, Canada, July 17.

2006 (**Invited)** "Biomedical applications with signal processing, statistics, and machine learning tools", Dept. Electrical Engineering, Fudan University, Shanghai, China, September 7.

2007 (**Invited)** "Signal processing and statistics: New roles in neuroscience and biomedical research", Communications Research Lab, McMaster University, Hamilton, Canada, July 4.

2007 "An empirical quantitative EEG analysis for evaluating clinical brain death", Annual IEEE-EMBS Conference, Lyon, France, August 25.

2008 "Assessment of hippocampal and autonomic neural activity by point process Models. 30th Annual IEEE-EMBS Conference, Vancouver, BC, Canada, August 20.

2008 "Characterizing nonlinear heartbeat dynamics within a point process framework", 30th Annual IEEE-EMBS Conference, Vancouver, BC, Canada, August 21.

2008 "A point process approach to assess dynamic baroreflex gain," Computers in Cardiology, Bologna, Italy, September 17.

2008 **Invited Workshop Lecturer** "Modeling neuronal multi-unit activity and neural dynamics", IEEE Control and Decision Conference, Cancun, Mexico, December 8.

2009 "Assessment of baroreflex control of heart rate during general anesthesia using a point process method", IEEE ICASSP, Taipei, Taiwan, April 21.

2009 (**Invited)** "Precise mapping of visual receptive field by tomographic reconstruction", Dept. Computer Science & Engineering, Shanghai Jiaotong University, Shanghai, China, May 12.

2009 (**Invited)** "Precise mapping of visual receptive field by tomographic reconstruction", Institute of Neuroscience, Chinese Academy of Sciences, Shanghai, China, May 15.

2009 (**Invited)** "Signal processing in neuroscience and bioengineering", Dept. Electrical Engineering, Ocean University of China, Qingdao, China, May 20.

2009 (**Invited)** "Precise mapping of visual receptive field by tomographic reconstruction", Center for Advanced Studies, Tsinghua University, Beijing, China, June 3.

2009 (**Invited)** "Point process modeling and inference in neuroscience and bioengineering", Dept. Biomedical Engineering, Institute of Neural Engineering & Tsinghua-Johns Hopkins Biomedical Engineering Center, Tsinghua University, Beijing, China, June 4.

2009 (**Invited)** "Point process modeling and inference in neuroscience and bioengineering", Dept. Information Science, School of Mathematical Sciences, Peking University, Beijing, China, June 5.

2010 "A differential autoregressive modeling approach within a point process framework for non-stationary heartbeat intervals analysis", Annual IEEE-EMBS Conference, Buenos Aires, Argentina, September 1.

Report of Technological and Other Scientific Innovations

- Japanese Patent Pending (with J. Cao, G. Hori, and A. Cichocki): "EEG-based pretest system for brain death diagnosis".

Report of Scholarship

Publications

- **Peer Reviewed Publications in print or other media**

Original Research Reports

1. **Chen Z**, Feng. TJ. An image segmentation approach based on wavelet and fractal features extraction. *Chinese Journal of Image & Graphics (A),* 4(12), 1072-1077, 1999. [in Chinese]
2. **Chen Z**, Feng. TJ. Research development and prospects on wavelet neural networks. *Journal of Ocean University of Qingdao*, 29(4), 663-668, 1999. [in Chinese]
3. Feng. TJ, **Chen Z**, Gu, FF. A fuzzy adaptive algorithm for learning parameters of BP networks. *Journal of Ocean University of Qingdao*, 30(1), 137-141, 2000. [in Chinese]
4. **Chen Z**, Feng. TJ, Chen G. A kind of BP algorithm-based wavelet neural network. *Journal of Ocean University of Qingdao*, 31(1), 122-128, 2000. [in Chinese]
5. **Chen Z**, Feng. TJ. A review: the research advances on combination of wavelet analysis and neural networks. *Chinese Journal of Electronics*, 22(3), 496-504, 2000. [in Chinese]
6. Feng. TJ, **Chen Z**, Xiong JS. A study on internal decision pattern of MLP networks. *Journal of Data Acquisition and Processing,* 15(4), 408-412, 2001. [in Chinese]
7. **Chen Z**, Feng. TJ, Sun Q. Wavelet neural networks for time series analysis and state space reconstruction. *Chinese Journal of Computer Research and Development*, 38(5), 591-596, 2002 [in Chinese]
8. **Chen Z**, Haykin S. On different facets of regularization theory. *Neural Computation*, 14: 2791-2846, 2002. PubMed PMID: 12487794
9. Haykin S, Huber K, **Chen Z**. Bayesian sequential state estimation for MIMO wireless communications. *Proceedings of the IEEE*, 92: 439-454, 2002.
10. Haykin S, **Chen Z**, Becker S. Stochastic correlative learning algorithms. *IEEE Transactions on Signal Processing*, 52: 2200-2209, 2004.
11. Haykin S, **Chen Z**. The cocktail party problem. *Neural Computation*, 17: 1875-1902, 2005. PubMed PMID: 15992485
12. **Chen Z**. Stochastic correlative firing for figure-ground segregation. *Biolgical Cybernetics*, 92: 192-198, 2005. PubMed PMID: 15750867
13. **Chen Z**, Becker S, Bondy J, Bruce I, Haykin S. A novel model-based hearing compensation design using a gradient-free optimization method. *Neural Computation*, 17: 2648-2671, 2005. PudMed PMID: 16212766
14. Papadelis C, **Chen Z**, Kourtidou-Papadeli C, Bamidis PD, Chouvarda I, Bekiaris A, Maglaveras N. Monitoring sleepiness with on-board electrophysiological recordings for preventing sleep-deprived traffic accidents. *Clinical Neurophysiology,* 118: 1906-1922, 2007. PubMed PMID: 17652020

15. **Chen Z,** Ohara S, Cao J, Vialatte F, Lenz FA, Cichocki A. Statistical modelling and analysis of laser-evoked potentials of electrocorticogram recordings from awake humans. *Computational Intelligence and Neuroscience*, Online Article ID 10479, 2007. PudMed PMID: 18369410

16. **Chen Z**, Cao J, Cao Y, Zhang Y, Gu F, Zhu G, Hong Z, Wang B, Cichocki A. An empirical EEG analysis in brain death diagnosis for adults. *Cognitive Neurodynamics*, 2: 257-271, 2008. PudMed PMID: 19003489

17. **Chen Z**, Brown E, Barbieri R. Assessment of autonomic control and respiratory sinus arrhythmia using point process models of human heart beat dynamics. *IEEE Transactions on Biomed Engineering* 2009 Jul;56(7):1791-802. Epub 2009 Mar 4. PubMed PMID: 19272971

18. **Chen Z**, Vijayan S, Barbieri R, Wilson MA, Brown EN. Discrete- and continuous-time probabilistic models and algorithms for inferring neuronal UP and DOWN states. *Neural Computation* 2009 Jul;21(7):1797-1862. Epub Mar 26. PubMed PMID:19323637

19. **Chen Z**, Brown E, Barbieri R. Characterizing nonlinear heartbeat dynamics within a point process framework. *IEEE Transactions on Biomedical Engineering* 2010 Jun;57(6):1335-1347. Epub 2010 Feb 17. PubMed PMID: 20172783

20. **Chen Z**, Putrino D, Ghosh S, Barbieri R, Brown EN. Statistical inference for assessing functional connectivity of neuronal ensembles with sparse spiking data. *IEEE Transactions on Neural Systems and Rehabilitation Engineering.* 2011, 19(2): 121-135. Epub 2010 Oct 11. PubMed PMID: 20937583

21. **Chen Z**, Purdon PL, Harrell G, Pierce ET, Walsh J, Brown EN, Barbieri R. Dynamic assessment of baroreflex control of heart rate during induction of propofol anesthesia using a point process method. *Annals of Biomedical Engineering* 2011, 39(1): 260-276. Epub 2010 Oct 13. PubMed PMID: 20945159

22. Wu W, **Chen Z**, Gao S, Brown EN. A hierarchical Bayesian approach for learning spatio-temporal decomposition of multichannel EEG. *NeuroImage*, 2011, 56(4): 1929-1945. Epub 2011 March 21 PubMed PMID: 21420499.

23. Putrino D*, **Chen Z***, Ghosh S, Brown EN. Motor cortical networks for skilled movements have dynamic properties that are related to accurate reaching. *Neural Plasticity*, Article ID 413543, 2011. (* Co-first author) PubMed PMID: 22007332

24. Kodituwakku S, Lazar SW, Indic P, **Chen Z**, Brown EN, Barbieri R. Point process time- frequency analysis of dynamic breathing patterns during meditation practice. *Medical and Biological Engineering and Computing*, 2012, 50: 261-275. Epub 2012, Feb. 21. PubMed PMID: 22350435.

25. **Chen Z**, Purdon P, Brown EN, Barbieri R. A unified point process probabilistic framework to assess heartbeat dynamics and autonomic cardiovascular control. *Frontiers in Computational Physiology and Medicine*, Epub 2012, Feb. 1, vol. 3, Article 4. PubMed PMID: 22375120.

26. **Chen Z**, Kloosterman F, Brown EN, Wilson MA. Uncovering spatial topology represented by rat hippocampal population neuronal codes. *Journal of Computational Neuroscience*, Epub 2012, Feb. 4. Online First DOI: 10.1007/s10827-012-0384-x. PubMed PMID: 22307459

27. Pipa G*, **Chen Z***, Neuenschwander S, Lima B, Brown EN. Mapping of visual receptive fields by tomographic reconstruction. *Neural Computation*, 2012 (*Co-first author)

Journal Articles being cited #: 201 (ISI Web of Knowledge), 380 (Google Scholar)

Submitted Original Reports

1. Kloosterman F, Layton S, **Chen Z**, Wilson MA. Bayesian decoding of unsorted spikes in the rat hippocampus. *Journal of Neurophysiology*.
2. **Chen Z**, Shimazaki H, Phillips MA, Constantine-Paton M, Brown EN. Mutual information estimation using a nonparametric copula approach. *Journal of Neuroscience Methods*.
3. Phillips MA, Bolton AD, Amico S, Kussius C, **Chen Z**, Brown EN, Popescu GK, Constantine-Paton M. Subunit-specific gating of NMDA receptors is independent of NR2 intracellular domain identity. *Frontiers in Cellular and Molecular Neuroscience*.
4. Wu W, **Chen Z**, Brown EN, Gao S. Robust probabilistic CSP algorithm for EEG analysis. *IEEE Trans. Biomedical Engineering*.

Peer-reviewed Conference Proceedings

1. **Chen Z**, Feng TJ, Houkes Z. Texture segmentation based on wavelet decomposition and Kohonen network for remotely sensed images. *Proc. 1999 IEEE Int. Conf. Syst., Man and Cybern*. 1999: 816-821.
2. **Chen Z**, Feng TJ, Meng QC. The application of wavelet neural network for time series prediction and system modeling based on multiresolution learning. *Proc. 1999 IEEE Int. Conf. Syst., Man and Cybern*. 1999: 425-430.
3. Meng QC, Feng TJ, **Chen Z**. Genetic algorithms encoding study and a sufficient convergence condition of GAs. *Proc. 1999 IEEE Int. Conf. Syst., Man and Cybern*. 1999: 649-652.
4. Meng QC, Feng TJ, **Chen Z**. Path planning strategy in multi-robot system, case study: Soccer robot. *Proc. Chinese Intelligent Automation Conference*, 1999, 11-15.
5. **Chen Z**, Feng TJ, Houkes Z. Mexican hat wavelet and its application in edge detection. *Proc. Int. Forum on Multimedia Image Processing*, 2000, Honolulu, HI.
6. **Chen Z**, Feng TJ, Houkes Z. Incorporating a priori knowledge into initialized weights for neural classifier. *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Networks* (IJCNN'00), 2000: 291-296.
7. **Chen Z**, Haykin S. A new view of regularization theory. *Proc. IEEE Conf. Syst., Man and Cybern*. 2001: 1642-1647.
8. **Chen Z**, Becker S, Haykin S. Theory of Monte Carlo sampling-based ALOPEX algorithms for neural networks. *Proc IEEE Int Conf Acoust Speech Signal Process* (ICASSP'04), 2004: 501-504.
9. **Chen Z**, de C. Lima, AC. A new neural equalizer for decision-feedback equalization. *Proc. IEEE Workshop on Machine Learning for Signal Processing*. 2004: 675-684.
10. **Chen Z**, Kirubarajan T, Morelande M. Improved particle filtering schemes for target tracking. *Proc IEEE Int Conf Acoust Speech Signal Process* (ICASSP'05), 2005: 145-148.
11. Ma J, **Chen Z**, Amari SI. Analysis of feasible solutions of the ICA problem under the one-bit-matching condition. *Proc. 6th Int. Conf. Independent Component Analysis and Blind Signal Separation* (ICA'06), Lecture Notes in Computer Science 3889. 2006: 838-845.

12. **Chen Z**, Cichocki A, Rutkowski TM. Constrained non-negative matrix factorization method for EEG analysis in early detection of Alzheimer disease. *Proc IEEE Int Conf Acoust Speech Signal Process* ICASSP, 2006: 893-896.

13. **Chen Z**, Ma J. Contrast functions of non-circular and circular sources separation in complex-valued ICA. *Proc. IEEE Int. Joint Conf. Neural Networks* (IJCNN'06), 2006: 1192-1199.

14. **Chen Z**, Cao J. An empirical quantitative EEG analysis for evaluating clinical brain death. *Conf Proc IEEE Eng Med Biol Soc.* 2007: 3880-3883. PubMed PMID:18002846.

15. Cao J, **Chen Z**. ICA and complexity measures of EEG analysis in brain death determination. *Proc. 1st Int. Conf. Cognitive Neurodynamics* (ICCN'07), Lecture Notes in Computer Science (*Advances in Cognitive Neurodynamics,* Chap. 121), 2007: 699-703.

16. Jankovic M, Martinez P, **Chen Z**, Cichocki A. Modified modulated Hebb-Oja learning rule: A method for biologically plausible principal component analysis. *Proc. 14th International Conference on Neural Information Processing* (ICONIP2007). Lecture Notes in Computer Science 4984. 2007: 527-536.

17. Sole-Casals J, Vialatte F, **Chen Z**, Cichocki A. Investigation of ICA algorithms for feature extraction of EEG signals in discrimination of Alzheimer disease. *Proc. 1st Int. Conf. Bio-inspired Systems and Signal Processing* (BIOSIGNALS'08) 2008: 232-235.

18. **Chen Z**, Brown EN, Barbieri R. A study of probabilistic models for characterizing human heart beat dynamics in autonomic blockade control. *Proc IEEE Int Conf Acoust Speech Signal Process* ICASSP 1-4244-1484-9/08: 481-484, 2008 March 31. PubMed PMID: 19593392. PubMed Central PMCID: PMC2707847.

19. **Chen Z**, Brown EN, Barbieri R. Characterizing nonlinear heartbeat dynamics within a point process framework. *Conf Proc IEEE Eng Med Biol Soc.* 2008; 978-1-4244-1815-2/08: 2781-2784. PubMed PMID: 19163282. PubMed Central PMCID: PMC2644067

20. Barbieri R, **Chen Z**, Brown EN. Assessment of hippocampal and autonomic neural activity by point process models. *Conf Proc IEEE Eng Med Biol Soc.* 2008; 978-1-4244-1815-2/08: 3679. PubMed PMID: 19163509. PubMed Central PMCID: PMC2652877.

21. **Chen Z**, Brown E, Barbieri R. A point process approach to assess dynamic baroreflex gain. *Comput Cardiol.* 2008 Sep 14;35:805-808. PubMed PMID: 19756137. PubMed Central PMCID: PMC2676855.

22. Sole-Casals J, Vialatte F, Cichocki A, **Chen Z**. Coherency and sharpness measures by using ICA algorithms: an investigation for Alzheimer's disease discrimination. *Proc. 2nd Int. Conf. Bio-inspired Systems and Signal Processing* (BIOSIGNALS'09) 2009: 468-475.

23. **Chen Z**, Purdon PL, Pierce ET, Harrell PG, Brown EN, Barbieri R. Assessment of baroreflex control of heart rate during general anesthesia using a point process method. *Proc IEEE Int Conf Acoust Speech Signal Process.* ICASSP 978-1-4244-2354-5/09: 333-336, 2009. PubMed PMID: 20473342. PubMed Central PMCID: PMC2867254.

24. **Chen Z**, Brown EN, Barbieri R. A unified point process framework for assessing heartbeat dynamics and cardiovascular control. *Proc. IEEE 35th Northeast Bioengineering Conf.* 2009.

25. **Chen Z**, Purdon PL, Pierce ET, Harrell PG, Walsh J, Salazar AF, Tavares CL, Brown EN, Barbieri R. Linear and nonlinear quantification of respiratory sinus arrhythmia during propofol general

anesthesia. *Proc IEEE Eng Med Biol Soc.* 2009; 978-1-4244-3296-7/09: 5336-5339. PubMed PMID: 19963899. PubMed Central PMCID: PMC2804255.

26. Wu W, **Chen Z**, Gao S, Brown EN. A probabilistic framework for learning robust common spatial patterns. *Proc IEEE Eng Med Biol Soc.* 2009: 4658-4661. PubMed PMID: 19963618.
27. **Chen Z**, Putrino DF, Ba D, Ghosh S, Barbieri R, Brown EN. A regularized point process generalized linear model for assessing the functional connectivity in the cat motor cortex. *Proc IEEE Eng Med Biol Soc.* 2009; 978-1-4244-3296-7/09: 5006-5009. PubMed PMID: 19965032. PubMed Central PMCID: PMC2822661.
28. **Chen Z,** Kloosterman F, Wilson MA, Brown EN. Variational Bayesian inference for point process generalized linear models in neural spike train analysis. *Proc IEEE Int Conf Acoust Speech Signal Process.* ICASSP 2010; 2086-2089.
29. Wei W, **Chen Z**, Gao S, Brown EN. Hierarchical Bayesian modeling of inter-trial variability and variational Bayesian learning of common spatial patterns from multichannel EEG. *Proc IEEE Int Conf Acoust Speech Signal Process.* ICASSP 2010; 501-504.
30. **Chen Z**, Purdon PL, Brown EN, Barbieri R. A differential autoregressive modeling approach within a point process framework for non-stationary heartbeat intervals analysis. *Proc IEEE Eng Med Biol Soc.* 2010; 3567-3570. PubMed PMID: 21096829.
31. **Chen Z**, Citi L, Purdon PL, Brown EN, Barbieri R. Instantaneous assessment of autonomic cardiovascular control during general anesthesia. *Proc IEEE EMBC*, 2011; 8444-8447. PubMed PMID: 22256307.
32. **Chen Z**, Vijayan S, Ching S, Hale G, Flores F, Wilson MA, Brown EN. Assessing neuronal interactions of cell assemblies during general anesthesia. *Proc IEEE EMBC*, 2011; 4175-4178. PubMed PMID: 22255259.
33. **Chen Z**, Kloosterman F, Layton S, Wilson MA. Transductive neural decoding for unsorted neuronal spikes of rat hippocampus. *IEEE EMBC* (invited session), 2012.

- **Semi-peer reviewed scientific or medical publications/materials in print or other media**

Books, Chapters, and Editorials

1. **Chen Z**, Gay SL, Haykin S. Proportionate adaptation: new paradigms in adaptive filters. In S. Haykin & B. Widrow (eds): *Least-Mean-Square Adaptive Filters* (pp. 293-334), Wiley, 2003.
2. Haykin S, **Chen Z**. The machine cocktail party problem. In S. Haykin et al. (eds): *New Directions in Statistical Signal Processing: From Systems to Brain* (pp. 51-75) MIT Press, 2006.
3. **Chen Z**, Haykin S, Eggermont, JJ Becker, S. *Correlative Learning: A Basis for Brain and Adaptive Systems* (Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications and Control). New York: Wiley, 2007.
4. Cao J, **Chen Z.** Advanced EEG signal processing in brain death diagnosis. In D. P. Mandic et al. (eds): *Signal Processing Techniques for Knowledge Extraction and Information Fusion* (pp. 275-298), Springer, 2008.

5. Berger TW, **Chen Z**, Cichocki A, Oweiss KG, Quian Quiroga R, Thakor NV. Editorial: Signal processing for neural spike trains. *Computational Intelligence and Neuroscience*, Volume 2010.
6. **Chen Z,** Barbieri R, Brown EN. State-space modeling of neural spike train and behavioral data. Chapter 6 In K. G. Oweiss (ed): *Statistical Signal Processing for Neuroscience and Neurotechnology*. Elsevier. pp. 175-218, 2010.
7. **Chen Z.** Signal processing for neuroscience: an introductory reading. English introductory chapter for annotated version of *Statistical Signal Processing for Neuroscience and Neurotechnology*. Science Press, China Science Publishing Group, 2012.

Theses and Dissertations

1. **Chen Z.** "Some studies of theory and applications of feedforward neural networks", Master thesis, Department of Electrical of Engineering, Ocean University of China, Qingdao, China, 2000.
2. **Chen Z.** "Stochastic approaches for correlation-based learning". Ph.D. dissertation, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada, 2005.

Selected Abstracts, Poster Presentations and Exhibits Presented at Professional Meetings

1. **Chen Z**. The application of wavelet neural network for time series prediction and system modeling based on multiresolution learning. Abstract in Third Conf Cognitive and Neural Systems (ICCNS), 1999, Boston, MA, USA.
2. **Chen Z**. New derivative-free Kalman filtering for training neural networks and Gaussian mixture model. Abstract in Sixth Conf Cognitive and Neural Systems (ICCNS), 2002, Boston, MA, USA.
3. **Chen Z**, Vijayan S, Barbieri R, Wilson MA, Brown EN. Learning probabilistic models for estimating neuronal "UP/DOWN" states. Poster in Fourth Int. Workshop Statistical Analysis of Neuronal Data (SAND4), May 29-31, 2008, Pittsburgh, PA, USA.
4. **Chen Z**, Vijayan S, Barbieri R, Wilson MA, Brown EN. Learning probabilistic models for estimating neuronal "UP/DOWN" states. Abstract in Computational Neuroscience Meeting (CNS'08), July19-24, 2008, Portland, OR, USA.
5. Phillips MA, Bolton A, **Chen Z**, Brown EN, Constantine-Paton M. NR2A and NR2B subunit chimeras for the study of NMDA receptors in developmental plasticity. Abstract in Society for Neuroscience (SfN), Nov. 15-19, 2008, Washington DC, USA.
6. Barbieri R, **Chen Z**, Brown EN. A point process framework to assess cardiovascular functions. Abstract in Society for Neuroscience (SfN), Nov. 15-19, 2008, Washington DC, USA.
7. Pipa G, **Chen Z**, Neuenschwander S, Lima B, Brown EN. A Comparative study of the point process GLM, the Wiener-Voltera filter and spike-triggered average methods in visual receptive field mapping. Abstract in Society for Neuroscience (SfN), Nov. 15-19, 2008, Washington DC, USA.
8. Pipa G, **Chen Z**, Neuenschwander S, Lima B, Brown EN. Efficient spike encoding for mapping visual receptive fields. Computational and Systems Neuroscience (COSYNE). Feb. 2009, Salt Lake City, UT, USA.
9. **Chen Z**, Putrino DF, Barbieri R, Brown EN. Assessing neuronal interactions and functional connectivity with sparse spiking data. Poster in Fifth Int. Workshop Statistical Analysis of Neuronal Data (SAND5), May 20-22, 2010, Pittsburgh, PA, USA.

10. Putrino DF, **Chen Z**, Ghosh S, Brown EN. Alterations in neural spiking rate and spiking associations in the cat motor cortex are related to errors in reaching. Abstract in Society for Neuroscience (SfN), Nov. 13-17, 2010, San Diego, CA, USA.
11. Layton S, Kloosterman F, **Chen Z**, Wilson MA. Bayesian decoding of unsorted spike trains. Computational & Systems Neuroscience (COSYNE), Feb. 24-27, 2011, Salt Lake City, UT, USA.
12. Layton S, Kloosterman F, **Chen Z**, Wilson MA. Bayesian decoding of unsorted spike trains. Abstract in Society for Neuroscience (SfN), Nov. 2011, Washington DC, USA.
13. **Chen Z**, Kloossterman F, Brown EN, Wilson MA. Uncovering embedded spatial topology represented rat hippocampal ensemble codes. Sixth Int. Workshop Statistical Analysis of Neuronal Data (SAND6), May 31-June 2, 2012, Pittsburgh, PA, USA.

Narrative Report

I am a Senior Research Fellow at the Massachusetts General Hospital (MGH), Harvard Medical School (HMS), and a Research Affiliate at the Massachusetts Institute of Technology (MIT). Under the mentorship of Prof. Emery Brown, I am currently involved in several research projects within the Neuroscience Statistics Research Laboratory in the Department of Anesthesia, Critical Care and Pain Medicine at MGH, and in the Department of Brain and Cognitive Sciences at MIT. My research interests include: (1) Neural signal processing and statistical modeling for various types of neuroscience data (e.g., spike trains, local field potentials, EEG, ECoG, intracellular recordings, behavior data, sleep-awake hypnogram data); (2) Development of efficient spike sorting-free decoding algorithm for real-time feedback control (e.g., neuroprosthetics) and efficient algorithm for uncovering spatial topology represented by rat hippocampal ensemble neuronal codes; (3) Biomedical signal processing in cardiovascular control and clinical monitoring. I currently devote 95% of my effort to research and collaborations, and 5% to providing technical advices regarding data analysis for graduate students and fellow postdoctoral associates.

Investigation. My research focuses on the following two areas.

Neural signal processing methods for neuronal data analysis focuses on development of new statistical algorithms and tools to analyze how individual and ensembles of neurons encode information about relevant external stimuli, as well as the neural correlates related to the behavior. In collaboration with experimental neuroscientists (at both MGH/HMS and MIT), I have been actively involved in several research investigations. One project investigated the characterization of neuronal up-down states from the primary somatosensory cortex in behaving rats. One other ongoing project is to develop spike sorting-free Bayesian decoding algorithm for estimating rat's positions based on the spiking activity of ensemble hippocampal place cells. This project may lead to other interesting applications of brain-machine interface (BMI) or neuroprosthetic devicies. Another project is aimed at developing new parametric/nonparametric statistical tools for assessing ensemble neuronal interactions, which have been used for analyzing neuronal data recordings from the cat primary motor cortex as well as the rat hippocampus. In addition, I have designed new statistical model and algorithm to uncover the spatial topology represented by rat hippocampal ensemble neuronal codes, and I have also used temporal difference reinforcement learning theory to model the rat's midbrain dopamine cell activity in a reward-navigation task.

Biomedical signal processing methods to cardiovascular signals focused on developing novel signal processing algorithms for modeling heartbeat dynamics and important cardiovascular/cardiorespiratory function, which produce new quantitative indices that could potentially have important implications for research studies of cardiovascular and autonomic regulation and for heart rate monitoring in clinical settings. In collaboration with Dr. Ricardo Barbieri at MGH/HMS, we have proposed a point process framework for assessing several cardiovascular functions (e.g., heart rate variability, baroreflex sensitivity, and respiratory sinus arrhythmia), and thus far we have successfully applied this framework to a variety of physiological recordings. A most recent investigation has assessed baroreflex control of heart rate during induction of general anesthesia from a dozen of healthy subjects.

Mentoring Experiences. At MIT, I have helped to mentor several visiting graduate students and a few MIT PhD students in their research projects, some of which have led to peer-reviewed publications. In the past three years, I have also been frequently offering technical advices on statistical data analysis for graduate students and postdoctoral fellows within or outside the Neuroscience Statistics Research Laboratory. I was invited by some principal investigators to their labs to present tutorial lectures on statistical analysis on neuronal data. I helped Prof. Emery Brown on lectures for the MIT undergraduate course "Statistics for Neuroscience".

Current and Future Research Projects & Proposals

by Zhe Chen, PhD

Project 1: Bayesian Modeling and Inference for Applications in Neural Engineering and Rehabilitation

Bayesian inference is a powerful toolkit for modeling and estimating the uncertainties of data, given partial knowledge of the system. In the past few years, I have been developing *approximate* fully Bayesian inference paradigms for tackling many statistical problems that arise from neuroscience data analysis [1]-[3]. Previously accomplished research projects involved (i) developing hierarchical Bayesian models for analyzing the inter-trial variability of multi-channel EEG signals for non-invasive brain-machine interface (BMI), (ii) analyzing the spiking trial-variability of single neurons, and (iii) assessing the functional connectivity of ensemble neurons in cortical/subcortical areas from animals or humans during periods of active behavior or general anesthesia.

Currently, I am investigating the Bayesian machinery for several neuroengineering applications. One working project (in collaboration with researchers from Prof. Matthew Wilson's Lab at MIT) aims to uncover the *spatial topology and hidden structural patterns* represented by hippocampal ensemble neuronal codes [4]. The new approach employs a hidden Markov model and an efficient Bayesian inference algorithm. This work is important for exploratory spiking data analysis not only during periods of active navigation but also during periods of sleep, which may provide new insights for studying memory consolidation in the rodent model. Another working project involves developing *spike-sorting free transductive neural decoding* algorithms for invasive BMI [5,6]. Unlike the standard neural encoding/decoding paradigm, the new decoding paradigm requires no spike encoding, thereby sidestepping the need of constructing explicit tuning curves for individual sorted units (and avoiding the error-prone spike sorting step). To date our algorithm has been successfully tested for reconstructing the behaving rat's position based on the spike activity from ensemble hippocampal place cells, yielding a lower decoding error and greater mutual information as compared to the standard spike-sorted Bayesian decoding algorithm. The new decoding paradigm is model-free and uses a "decode-as-you-go" strategy. It also opens new opportunities to investigate neural circuits using a closed-loop biofeedback system combined with external manipulation (e.g., optogenetics or electrical simulation). Some physiological experiments are currently undertaken using the rat hippocampus system. This decoding paradigm can be also adapted to *motor/auditory/visual* cortical spike data, and has potentials for real-time decoding in chronic neuroprosthetic devices, based on either single electrode, stereotrode or tetrode wires. We are currently extending the analysis to primate motor (M1) data during the reach-grasp task [7].

The methodology of Bayesian modeling/inference goes far beyond the applications described here. It is my firm belief that there remain a lot of opportunities for applying Bayesian machinery to human-computer interfaces and other neuroscience data analysis, and it will be one of my research focuses to investigate its horizons in computational neuroscience (such as decision making, behavior modeling and prediction, and optimal experimental design). It also has a broad application for smart engineering system design.

[1] **Chen Z**, Putrino D, Ghosh S, Barbieri R, Brown EN (2011). Statistical inference for assessing functional connectivity of neuronal ensembles with sparse spiking data. *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* 19(2): 121-135.
[2] **Chen Z**, Vijayan S, Ching S, Hale G, Flores F, Wilson MA, Brown EN (2011). Assessing neuronal interactions of cell assemblies during general anesthesia. *Proc IEEE EMBC*, pp. 4175-4178, Boston, MA.
[3] Wu W, **Chen Z**, Gao S, Brown EN (2011). A hierarchical Bayesian approach for learning spatio-temporal decomposition of multichannel EEG. *NeuroImage*, 56(4): 1929-1945.
[4] **Chen Z**, Kloosterman F, Brown EN, Wilson MA (2012). Uncovering hidden spatial topology represented by hippocampal population neuronal codes. *Journal of Computational Neuroscience,* Feb 1. epub.
[5] Kloosterman F, Layton S, **Chen Z**, Wilson MA (2012). Bayesian decoding of unsorted spikes in the rat hippocampus. *Journal of Neurophysiology*, submitted.
[6] **Chen Z**, Kloosterman F, Layton S, Wilson MA (2012). Transductive neural decoding of unsorted neuronal spikes of rat hippocampus. *EMBC'12*, San Diego, CA.
[7] Saleh M, Takahashi K, Hatsopoulos NG (2012). Encoding of coordinated reach and grasp trajectories in primate motor cortex. *Journal of Neuroscience*, 32(4): 1220-1232.

Project 2: Statistical Analysis of Spontaneous Miniature Postsynaptic Currents

Miniature postsynaptic currents (mPSCs) represent postsynaptic responses to the spontaneous, quantal release of neurotransmitter vesicles. Intracellular single-unit recordings over a period of minutes may detect hundreds or even thousands of synaptic events in a single cell. Quantitative assessment of those spontaneous events at either single cell or population levels might help to reveal the underlying neural mechanisms of synaptic plasticity.

The research aims of this project (in collaboration with Prof. Martha Constantine-Paton's Lab at MIT) are to (i) develop detailed statistical characterization of individual salient features of mPSC data (e.g., inter-event intervals, amplitude, rise and decay time, weighted time constant, charge transfer, and background noise activity); (ii) develop likelihood-based inference paradigms and goodness-of-fit assessment, as well as methods for between- and within-group comparisons of the features of mPSCs; (iii) characterize statistical dependency among the features of mPSC and quantify the change of statistical dependency across different experimental conditions; (iv) develop a new filtered point process model and a Bayesian deconvolution algorithm, under which the analysis of all salient features can be conducted within a unified framework. Several likelihood-based and Bayesian methods have been developed for the above statistical analysis. This established statistical analysis paradigm would be applied to spontaneous currents arising from different types of neurotransmitter receptors, excitatory or inhibitory, in a few experimental studies (e.g., analyses of synaptic modification during sensory development) [1]-[3].

Through these studies, we hope to establish a principled approach for analysis of miniature synaptic transmission [4, 5], providing new insights to answer a wide array of important questions in synaptic neurophysiology and developmental neuroscience, which will help understand how synapses in the brain develop, and determine the factors that control this maturation process.

[1] Phillips, MA, Colonnese MT, Goldberg J, Lewis LD, Brown EN, Costantine-Paton M (2011). A synaptic strategy for rapid experience dependent consolidation of convergent visuotopic maps. *Neuron*, 71(4), 710-724.
[2] Phillips, MA, Lewis, LD, Gong J, Constantine-Paton M, Brown EN (2012). Accurate model-based analysis of spontaneous miniature synaptic transmission. *J. Neurophysiol*., under revision.

[3] Phillips MA, Bolton AD, Amico S, Kussius C, **Chen Z**, Brown EN, Popescu GK, and Constantine-Paton M (2012). Subunit-specific gating of NMDA receptors is independent of NR2 intracellular domain identity. *Frontiers in Cellular and Molecular Neuroscience*, submitted.
[4] **Chen Z**, Shimazaki H, Phillips MA, Constantine-Paton M, Brown EN (2012). Mutual information estimation using a nonparametric copula approach. *Journal of Neuroscience Methods*. In preparatin.
[5] **Chen Z**, Phillips MA, Constantine-Paton M, Brown EN. Bayesian deconvolution of noisy miniature excitatory postsynaptic currents using a filtered point process model. In preparation.

Project 3: Computational and Statistical Approaches for Modeling Dynamic State of the Brain/Mind

A fundamental question in neuroscience is to characterize and quantify the dynamic state of the brain. The "state" of a neural system reflects the phase of an active recurrent network, which organizes the internal states of individual neurons into synchronization through recurrent network synaptic activity. The neuronal state dynamics can be either externally driven (such as by sensory stimuli) or internally driven (such as by attention shift). Modeling neural dynamics with intracellular/extracellular recordings (e.g., membrane potentials, spike trains, local field potentials) or multichannel ECoG/EEG recordings would provide insights on understanding of functions of neural circuits, neuromodulation, and brain pathologies/disorders at the systems neuroscience level.

In collaboration with experimentalists, I have been developing and applying novel neural signal processing tools to analyze specific neural dynamics (using recordings from neocortex, hippocampus, VTA, thalamus, or basal ganglia areas) of animals or humans during periods of behavior, anesthesia, or sleep [1]-[6]. Examples include (i) modeling pain-evoked potentials of ECoG from awake humans; (ii) characterizing neurophysiological markers of human sleepiness in real-life driving; (iii) using hidden Markov models for detecting neuronal UP/DOWN states based on multi-unit activity of cortical neurons; (iv) using parametric and nonparametric statistical tools for assessing ensemble neuronal interactions as well as for quantifying spike-field mutual information; (v) using survival analysis approach to characterize sleep-state transitions for either rodents or humans; and (vi) using temporal difference reinforcement learning to model rat's midbrain dopamine neurons in a rewarded spatial navigation task. By virtue of various multi-modal multi-facet neural data analyses, we hope to gain new neurophysiological insights in understanding brain functions as well as their links to behavior.

[1] **Chen Z,** Ohara S, Cao J, Vialatte F, Lenz FA, Cichocki A (2007). Statistical modelling and analysis of laser-evoked potentials of electrocorticogram recordings from awake humans. *Computational Intelligence and Neuroscience*, Article ID 10479.
[2] Papadelis C, **Chen Z**, Kourtidou-Papadeli C, Bamidis PD, Chouvarda I, Bekiaris A, Maglaveras N (2007). Monitoring sleepiness with on-board electrophysiological recordings for preventing sleep-deprived traffic accidents. *Clinical Neurophysiology,* 118: 1906-1922.
[3] **Chen Z**, Vijayan S, Barbieri R, Wilson MA, Brown EN (2009). Discrete- and continuous-time probabilistic models and algorithms for inferring neuronal UP and DOWN states. *Neural Computation*, 21(7):1797-1862.
[4] **Chen Z,** Barbieri R, Brown EN (2010). State-space modeling of neural spike train and behavioral data. *Statistical Signal Processing for Neuroscience and Neurotechnology* (Chapter 6, pp. 175-218), Elsevier.

[5] Putrino D*, **Chen Z***, Ghosh S, Brown EN (2011). Motor cortical networks for skilled movements have dynamic properties that are related to accurate reaching. *Neural Plasticity*, Article ID 413543. (* co-first author)

[6] Pipa G*, **Chen Z***, Neuenschwander S, Lima B, Brown EN. Mapping of visual receptive fields by tomographic reconstruction. *Neural Computation*, 2012 (*Co-first author)

Project 4: Quantitative Assessment of Heartbeat Dynamics and Cardiovascular Control

In collaboration with Prof. Riccardo Barbieri and Prof. Emery Brown (MGH/HMS), over the past few years we have been developing a mathematically rigorous, unified point process framework for quantitative assessment of heartbeat dynamics and cardiovascular functions [1]-[5]. Using recordings of noninvasive clinical recordings of ECG, arterial blood pressure, and lung volume (respiratory measure), we designed stochastic parametric models and algorithms for estimating *instantaneous* physiological indices such as the heart rate variability, baroreflex sensitivity, and respiratory sinus arrhythmia. As expected, these estimates are useful for providing informative indicators in cardiovascular medicine, such as for evaluating cardiovascular diseases (e.g., hypertension, or congestive heart failure), for assessing the depth of general anesthesia or the sensation of pain. Accumulating interests and research efforts have attempted to incorporate these measures to future-generation biomedical devices for clinical monitoring, whereas developing new signal analysis tools remains important to better understand the physiological mechanisms of cardiovascular functions. A pilot study of developing a biofeedback interface for stress control is under consideration based on some informative physiological indices computed for important cardiovascular functions (in addition to skin conductance, muscle tension). Presumably, such *heart-computer interfaces* that can read out and manipulate human emotions may open new opportunities in the future biomedical research [6].

[1] **Chen Z**, Brown EN, Barbieri R (2009). Assessment of autonomic control and respiratory sinus arrhythmia using point process models of human heart beat dynamics. *IEEE Transactions on Biomed Engineering*, 56(7): 1791-802.

[2] **Chen Z**, Brown EN, Barbieri R (2010). Characterizing nonlinear heartbeat dynamics within a point process framework. *IEEE Transactions on Biomedical Engineering*, 57(6): 1335-1347.

[3] **Chen Z**, Purdon PL, Harrell G, Pierce ET, Walsh J, Brown EN, Barbieri R (2011). Dynamic assessment of baroreflex control of heart rate during induction of propofol anesthesia using a point process method. *Annals of Biomedical Engineering,* 39(1): 260-276.

[4] **Chen Z**, Purdon PL, Brown EN, Barbieri R (2012). A unified point process probabilistic framework to assess heartbeat dynamics and autonomic cardiovascular control. *Frontiers in Computational Physiology and Medicine*, vol. 3, Article 4, Feb. 2012.

[5] Kodituwakku S, Lazar SW, Indic P, **Chen Z**, Brown EN, Barbieri R (2012). Point process time-frequency analysis of dynamic breathing patterns during meditation practice. *Medical & Biological Engineering & Computing*, 50:261-275.

[6] Henriques G, et al. (2011). Exploring the effectiveness of a computer-based heart rate variability biofeedback program in reducing anxiety in college students. *Appl. Psychophysiol. Biofeedback*, 36(2):101-112.

To summarize, my research efforts for seeking external funding will focus on (i) Neural Engineering and BMI Systems (*Project 1*), (ii) Modeling and Analysis in Computational Neuroscience (*Projects 2&3*), and (iii) Cardiovascular Signal Processing, Monitoring and Heart-Computer Interface Design (*Project 4*). The research aims of those projects fit the missions of major Government funding agencies (such as the **NIH, NIMH, NIBIB, NSF**) as well as a few other agencies or foundations. It is my goal to maintain a long-term collaboration with experimental physiologists, neuroscientists whom I have been working with. Meanwhile, I would also try to seek new collaborations within the new institution. Finally, in the near future I will also submit a **NSF CAREER** proposal that aims to integrate neuroengineering research and education for future generations of engineers.

Statements of Research and Teaching

Motto

In Pursuit of the Excellence and Innovations of Research, Teaching, and Discovery

Research

The 21st century has witnessed a lot of theoretical and technological advances that reshape our lives and society. Nowadays, the needs for processing/analyzing the exploding data or information have substantially grown. Processing real-life data, because of their inherent nature of *nonlinearity, non-Gaussianity, non-stationarity, mixed-modality, and high-dimensionality*, certainly provides challenges as well as research opportunities for modern signal processing. Specifically, *adaptive* and *intelligent* signal processing has become a growing field that attracts increasing attention in many biomedical or neural engineering applications, such as the analysis of multichannel EEG/MEG/EMG signals, gene expression classification, monitoring or early diagnosis of brain disorder/diseases, and design of human-machine interfaces or neuroprosthetics. It is exactly these research topics that deeply attract me to study and explore the field of signal processing to engage in biomedical research and the promotion of technology. With this philosophy in mind, my research vision and goals are structured as follows:

Research Directions

- Signal processing for man-computer interfacing in rehabilitation and neurotherapy: wired/wireless prosthetic system; mental disease diagnosis; clinical monitoring and evaluation of "state of the mind".
- New paradigms for mixed-modality signal processing (with broad applications in electrical/biomedical/neural engineering).
- Translational/computational neuroscience & neural computation: affective brain-style computing; active perception/sensing; neural plasticity during biofeedback and rehabilitation.

Research Plans

- Establishing a new laboratory (*Laboratory for Neural Signal Processing*) and recruiting undergrad/grad students to form a productive research team; securing research funds from external funding agencies or industry.
- Integrating science & engineering for developing pilot-study protocols that lead to new frontier investigations.
- Establishing cross-disciplinary collaborations; participating in various professional activities (editorial, conference organization, etc.)

Research Goals

- In the short term, establishing innovative research projects that secure steady research funding; supervising graduate students that lead to publications of high-quality peer-reviewed journal articles and conference papers.
- In the medium term, making significant contributions to the signal processing field in bioengineering and neural engineering; translating the research efforts into industrial practice and developing techniques or biomedical devices that impact the society; whenever it is possible, applying for patents or intellectual properties in biomedical technology.
- In the long term, establishing a strong research team and good reputation in the research field; developing new teaching curricula related to biomedical or neural engineering education.

Teaching

Courses offered: undergraduate level: *Probability and Stochastic Processes*, *Linear Systems*, *Signal Processing*, *Introduction on Neural Computation*; graduate level: *Filtering, Estimation and Statistical Inference*, *Machine Learning and Statistics Methods in Neuroengineering*.

Curricula to be developed: As time goes by, developing some new curricula to cater the ever-growing need for bioengineering and biomedicine: *Signal Processing and Statistics in Bioengineering* (nonlinear non-Gaussian system analysis, neural signal processing, human-machine interface, etc.).

"Teaching is an interruption, and so it's the greatest pain in the neck in the world....
...teaching and the students keep life going ...."
— Richard Feynman

Teaching Philosophy

Undergrad

The central task of undergraduate teaching is *knowledge semination*, with the goal to help students deepen the understanding of fundamental concepts in the specialized field. The following points are planned to help achieve this goal:

- develop newly-updated class notes, class-related web resources, software or codes.
- use simple examples to illustrate the underlying idea; explain problems from different perspectives; use computer simulations to help expound the technically difficult content.
- relate the class materials to real-life applications; use multimedia demonstrations to boost students' interests.
- guide students to tackle technical difficulties (including course exercises and assignments); present regular tutorials to help students develop problem-solving skills.
- challenge students with some post-class or summer projects; guide students to develop basic research skills and the ability to conduct independent investigations.

Graduate

The essence of graduate teaching is *knowledge redistribution*, whereas the goal of graduate education and research is *innovation*. With this in mind, teaching will emphasize the knowledge depth and the clarification of key concepts and innovative ideas (Above all, idea matters!). The following points are planned to help achieve this goal:

- develop newly-updated class notes, class-related web resources, reusable softwares or codes; develop class-related reading lists (containing classic and state-of-the-art research papers).
- encourage students to raise questions and pursue their own scientific interests.
- guide students to conduct research-oriented course projects, and guide students to write projected-based conference papers.
- help students develop self-learning and self-teaching skills: students are asked to present important research papers briefly to the class and conduct basic experiments, with emphasis on highlighting the contribution (why is important) and the novelty (what is new).
- help student develop practical skills in oral presentation and paper writing.
- help to bridge the technical gap between different research disciplines (e.g., signal processing and biology, machine learning and neuroscience).

In my own understanding, teaching is the *highest* level of assimilating knowledge. On the other hand, teaching opens many ways to potential research topics — the questions from students are often the sources for new research topics; it stimulates the instructor to *think* and *rethink*; it is also justified by the instructor's effort: the more you prepare for the students, the higher gain you will expect to achieve for the class. I am a deep believer of this philosophy and I am planning to do so in the future.

References

1) Emery N. Brown, M.D., Ph.D. (Fellow of IEEE, ASA, Institute of Medicine)
Professor of Computational Neuroscience and Health Sciences and Technology
Department of Brain and Cognitive Sciences
MIT-Harvard Division of Health Science and Technology
Massachusetts Institute of Technology
77 Massachusetts Avenue, Bldg 46-6079
Cambridge, MA 02139

Warren M. Zapol Professor of Anaesthesia
Harvard Medical School
Department of Anesthesia, Critical Care and Pain Medicine
Massachusetts General Hospital
55 Fruit Street, GRJ 4
Boston, MA 02114

tel: 617 324 1879
fax: 617 324 1884
email: enb@neurostat.mit.edu
http://bcs.mit.edu/people/brown.html

2) Riccardo Barbieri, PhD
Assistant Professor of Anaesthesia
Harvard Medical School
Assistant Biomedical Engineer
Department of Anesthesia, Critical Care and Pain Medicine
Massachusetts General Hospital
55 Fruit Street, GRJ 4
Boston, MA 02114

tel: (617) 724-1061
email: barbieri@neurostat.mit.edu
https://neurostat.mgh.harvard.edu/barbieri/riccardohomepage.htm

3) Matthew A. Wilson, PhD
Sherman Fairchild Professor of Neuroscience and Picower Scholar
Associate Department Head for Education
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
77 Massachusetts Avenue, Bldg 46-5233
Cambridge, MA 02139

Tel: (617)-253-2046

email: mwilson@mit.edu
http://bcs.mit.edu/people/wilson.html

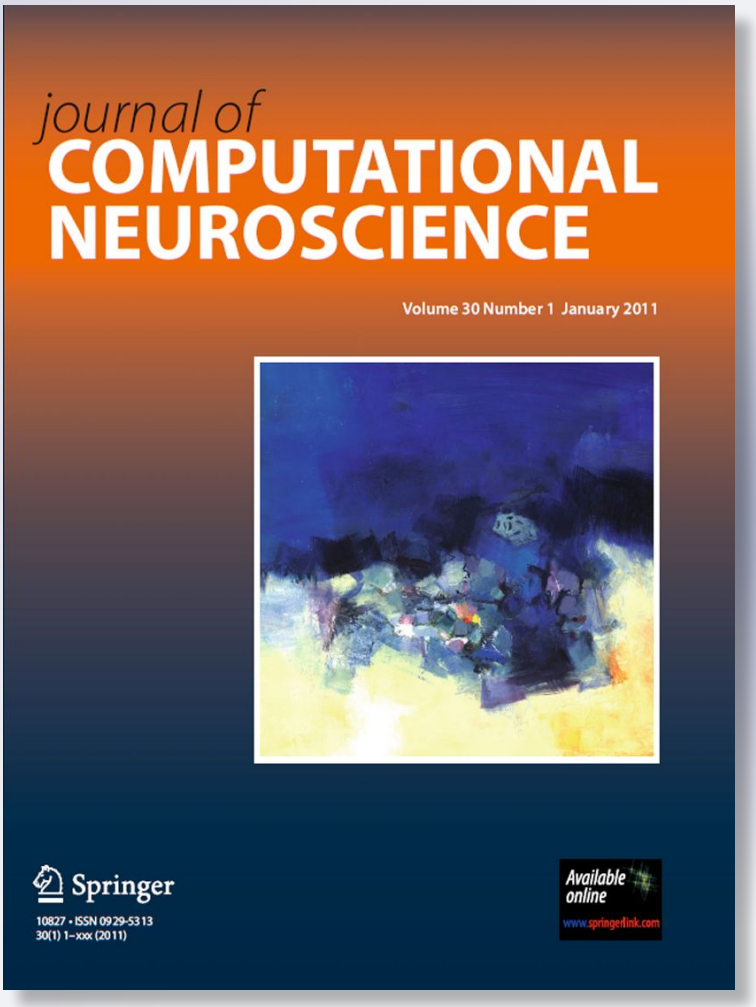
Uncovering spatial topology represented by rat hippocampal population neuronal codes

Zhe Chen, Fabian Kloosterman, Emery N. Brown & Matthew A. Wilson

Journal of Computational Neuroscience

ISSN 0929-5313

J Comput Neurosci
DOI 10.1007/s10827-012-0384-x



Springer

J Comput Neurosci
DOI 10.1007/s10827-012-0384-x

Uncovering spatial topology represented by rat hippocampal population neuronal codes

Zhe Chen · Fabian Kloosterman · Emery N. Brown · Matthew A. Wilson

Received: 27 October 2011 / Revised: 16 January 2012 / Accepted: 23 January 2012

Abstract Hippocampal population codes play an important role in representation of spatial environment and spatial navigation. Uncovering the internal representation of hippocampal population codes will help understand neural mechanisms of the hippocampus. For instance, uncovering the patterns represented by rat hippocampus (CA1) pyramidal cells during periods of either navigation or sleep has been an active research topic over the past decades. However, previous approaches to analyze or decode firing patterns of population neurons all assume the knowledge of the place fields, which are estimated from training data a priori. The question still remains unclear how can we extract information from population neuronal responses either without a priori knowledge or in the presence of finite sampling constraint. Finding the answer to this question would leverage our ability to examine the population neuronal codes under different experimental conditions. Using rat hippocampus as a model system, we attempt to uncover the hidden "spatial topology" represented by the hippocampal population codes. We develop a hidden Markov model (HMM) and a variational Bayesian (VB) inference algorithm to achieve this computational goal, and we apply the analysis to extensive simulation and experimental data. Our empirical results show promising direction for discovering structural patterns of ensemble spike activity during periods of active navigation. This study would also provide useful insights for future exploratory data analysis of population neuronal codes during periods of sleep.

Keywords Hidden Markov model · Expectation-maximization · Variational Bayesian inference · Place cells · Population codes · Spatial topology · Force-based algorithm

Action Editor: Jonathan David Victor

Z. Chen (✉) · E. N. Brown
Neuroscience Statistics Research Lab,
Massachusetts General Hospital, Harvard Medical School,
Boston, MA 02114, USA
e-mail: zhechen@mit.edu

Z. Chen
Department of Brain and Cognitive Sciences,
Massachusetts Institute of Technology, Cambridge,
MA 02139, USA

F. Kloosterman · M. A. Wilson
Department of Brain and Cognitive Sciences
and Picower Institute for Learning and Memory,
Massachusetts Institute of Technology, Cambridge,
MA 02139, USA

E. N. Brown
Department of Brain and Cognitive Sciences
and Harvard-MIT Division of Health and Science
Technology, Massachusetts Institute of Technology,
Cambridge,
MA 02139, USA

1 Introduction

1.1 Motivation

Hippocampal population codes play an important role in representation of spatial environment and spatial navigation (O'Keefe and Nadel 1978; Buzsaki 2006). It is known that the receptive fields of hippocampal pyramidal cells encode information of the position of space, hence those cells are referred to as "place cells" (O'Keefe and Nadel 1978). Using the multielectrode

Published online: 04 February 2012

technique, spiking activity of ensemble hippocampal place cells can be simultaneously recorded from rodents, which enable us to examine the internal representation of the population codes at different behavioral stages (Wilson and McNaughton 1993, 1994). One of the goal in exploratory data analysis is to discover the hidden structures or firing patterns of spiking activity from simultaneously recorded hippocampal population neurons, either during periods of active behavior (Wilson and McNaughton 1993; Harris et al. 2003; Foster and Wilson 2006) or during periods of sleep (Wilson and McNaughton 1994; Louie and Wilson 2001; Lee and Wilson 2002; Ji and Wilson 2007). For instance, finding rodent hippocampus neuronal "replay" (Foster and Wilson 2006; Davidson et al. 2009) or "preplay" patterns (Dragoi and Tonegawa 2011) in cell assemblies during either quiet awakefulness or slow-wake sleep (SWS), as compared to the firing patterns during periods of active navigation, has been an important research topic in recent years (Skaggs and McNaughton 1996; Diba and Buzsaki 2007; Karlsson and Frank 2009). Two types of neuronal codes were used in previous studies. One is based on temporal code, which assumes that the individual cells of neuronal assembly fire in a specific order when the animal navigates in the spatial environment (Lee and Wilson 2002; Ji and Wilson 2007). The other is based on rate code, which assumes that the spiking activity of population cells follows a probabilistic rule (Brown et al. 1998; Zemel et al. 1998; Zhang et al. 1998; Davidson et al. 2009). However, these approaches have some drawbacks. First, all previous approaches rely on the assumption that the receptive fields of population neurons (i.e., place fields of hippocampal pyramidal neurons) are known, which are commonly constructed from empirical training data. This assumption could be problematic since the receptive fields are plastic, thus the empirical internal representation of the stimulus space could change at different stages (e.g., navigation vs. sleep) or at different learning phases (first day vs. second day), or when the shape of the stimulus space changes (Lever et al. 2002; Frank et al. 2004; Wills et al. 2005). The change in hippocampal place-cell representation is known as *remapping*. Second, if the goal of the analysis is to examine the internal representation of the population codes, we shall assume no or little knowledge about the environment (i.e., either linear track, or T-maze, or open field). This is critically important especially when the firing patterns are examined during SWS or REM sleep periods, or the animal has been exposed to multiple distinct spatial environments before the experimental recording. Meanwhile, noticing the fact that knowing the receptive fields of a real environment is not completely necessary for the replay or preplay analysis, since the place fields are only a proxy to examine the relative proximity of spatial position in the environment. Therefore, one could imagine the possibility that population neurons encode an internal representation of the "virtual environment" which could be an abstract representation of the real environment. To our best knowledge, very few study has been done in this area in the literature, except for the work by Curto and Itskov (2008). Specifically, with the same motivation (but completely different methodology) and with no assumption of the hippocampal place fields, Curto and Itskov showed that simply knowing which groups of cells fire together would reveal structure in the stimulus space, which then enables the brain to construct its own internal representations. Put in their words, "*a rather unexplored question is how the output of hippocampal place cells (without access to corresponding place fields) might be used by downstream structures in order to reconstruct position and the underlying space*". In their method, the authors made certain assumptions of the place fields in an open field environment, and identified the cell groups (a group of place cells that collectively fire within a two theta-cyle, or 250 ms time window), and further computed the homology groups and extracted the topological features of the spatial environment, and finally constructed an internal representation of the environment using a graph (that contains a vertex for every cell group and an edge between neighboring cell groups) and a distance metric (that contains distances between any two cell groups).

Motivated by these above-mentioned open questions, we develop a probabilistic generative model and a statistical inference approach to solve the above-mentioned problems. Our approach is different from the method of Curto and Itskov (2008) in terms of the assumptions of place fields and the use of mathematical tools. Finding the internal representation of hippocampal population codes is viewed as an *unsupervised* learning problem. More precisely, we propose a solution based on a hidden Markov model (HMM) and an associated efficient Bayesian inference procedure. Our computational goal is to infer or uncover the spatial topology represented by the hippocampal population neuronal codes in rodent. It shall be pointed out that the term "spatial topology" used here has a narrower meaning than its conventional sense, it is simply referred to the structure of the stimulus space or behavior sequences underlying the hippocampal population neuronal codes. As a byproduct of our estimation procedure, we also recover the receptive fields of hippocampal population neurons with respect to

the virtual environment, which are referred to as the "virtual place fields".

1.2 Overview of methods

Inferring the spatial topology represented by the hippocampal population codes is considered as an inverse problem with missing data (Dabaghian et al. 2008). To our best knowledge, very few study has been found in the literature. In this study, we examine this problem from a computational perspective. From a statistical data analysis viewpoint, the observed data are the spiking activity of hippocampal ensemble place cells, whereas the missing data are the hidden trajectory (in the virtual environment) associated with the firing patterns exhibited by the place cells, as well as the neuronal tuning curves with respect to the spatial environment. The unobserved trajectory is treated as a hidden state, which is assumed to follow a Markovian structure. For simplicity, we also assume that the number of hidden states is finite. To model the dynamical system, HMM is a powerful tool for inferring hidden variables given partially observed data. In the computational neuroscience field, to name a few, HMM (Cappé et al. 2005; Rabiner 1989) has been widely used either for decoding natural stimuli (Jones et al. 2007), or for inferring states of population neurons during periods of SWS (Chen et al. 2009), or for detecting neural-state transition for motor cortical prostheses (Kemere et al. 2008), or for sorting neuronal spikes (Herbst et al. 2008), or for spatial-temporal clustering of neural data (Darmanjian and Principe 2009).

Once the statistical model is determined, two kinds of inference approaches can be considered: one is the maximum likelihood approach (Pawitan 2001), the other is the Bayesian approach (Robert 2001; MacKay 2003; Gelman et al. 2004). Maximum likelihood estimate is asymptotically optimal and invariant, but it is prone to overfitting in the presence of small sample size. In contrast, Bayesian inference imposes priors (e.g., sparsity, invariance) onto the model, and its estimate is more meaningful and efficient; in addition, the uncertainty of the estimate can be represented by the posterior in place of the point estimate. There are various Bayesian inference methods available in the literature (Scott 2002), such as the Markov chain Monte Carlo (MCMC) (Gilks et al. 1995; Rydén 2008), Laplace approximation (MacKay 2003), and variational methods (MacKay 2003; Bishop 2006). Specifically, in contrast to the MCMC methods, variational Bayesian (VB) methods are more computationally appealing, and they have been proposed for learning a number of statistical models (Beal 2003; Bishop 2006; Katahira et al. 2010; Chen et al. 2011; Wu et al. 2011).

Spatial topology is typically visualized by graphs. Force-based algorithms are a class of algorithms for drawing graphs in a way that the nodes of a graph are positioned in two dimensional or three dimensional space so that all the edges are of more or less equal length and there are as few crossing edges as possible (Tollis et al. 1999). The force-based algorithms achieve this by assigning forces amongst the set of edges and the set of nodes; the most straightforward method is to assign forces as if the edges were springs (*Hooke's law*) and the nodes were electrically charged particles (*Coulomb's law*). The entire graph is then simulated in the same fashion as a physical system. The forces are applied to the nodes, pulling them closer together or pushing them further apart. This process is repeated iteratively until the system reaches an equilibrium state (i.e., their relative positions no longer change or change very little from one iteration to the next). The physical interpretation of this equilibrium state is that all the forces are in mechanical equilibrium.

Our computational approach consists of two steps: first, infer the unknown parameters of the HMM using VB inference; second, infer the spatial topology of the animal behavior within the environment based on the parameters of the HMM using a force-based algorithm. The rest of the paper is organized as follows. Section 2 presents the background of the finite-state HMM. Section 3 presents the VB inference algorithm for HMM. Section 4 introduces the force-based algorithm for visualizing the spatial topology. Section 5 presents results from a number of computer simulations and experimental data. Interpretations and implications of these results are discussed in detail. Finally, in Section 6 we present some discussions on important issues and conclude the paper in Section 7.

2 Finite-state hidden Markov model

Let us consider a discrete-time homogenous Markov chain. By discrete time, we assume that the time is evenly discretized into fixed-length intervals, which have time indices $t = 1, \ldots, T$. The standard HMM is characterized by three elements: *transition probability, emission probability*, and *initial state probability*.

- The initial probability of state is denoted by a vector $\boldsymbol{\pi} = \{\pi_i\}$, where $\pi_i = \Pr(S_0 = i)$ $(i = 1, \ldots, m)$. Without loss of generality, we assume that the discrete variable $S_t \in \{1, \ldots, m\}$, and size of the discrete state is $dim\{S\} = m$.

– The m-by-m transition probability matrix is written as

$$\boldsymbol{P} = \begin{pmatrix} P_{11} & P_{12} & \dots & P_{1m} \\ P_{21} & P_{22} & \dots & P_{2m} \\ \vdots & \vdots & \dots & \vdots \\ P_{m1} & P_{m2} & \dots & P_{mm} \end{pmatrix} \tag{1}$$

with P_{ij} corresponding to the transition (conditional) probabilities from state i to state j.

– For c-th cell, the Poisson spike counts $y_{c,t}$ observed at the t-th time bin follows products of exponentiated Poisson distributions (denoted by $\texttt{Poi}$)

$$\begin{aligned} p(y_{c,t}|S_t) &= \prod_{i=1}^{m} p(y_{c,t}|S_t = i)^{S_{t,i}} \\ &= \prod_{i=1}^{m} \texttt{Poi}(y_{c,t}|\lambda_{ic}, S_t = i)^{S_{t,i}} \\ &= \prod_{i=1}^{m} \left(\frac{\exp(-\lambda_{ic})\lambda_{ic}^{y_{c,t}}}{y_{c,t}!} \right)^{S_{t,i}} \end{aligned} \tag{2}$$

where the exponent $S_{t,i}$ denotes a Kronecker delta, i.e., $S_{t,i} = 1$ if and only if $S_t = i$. The $\lambda_{ic} \geq 0$ denotes the rate parameter for cell c at the i-th hidden state. Given all $c = 1, \dots, C$ cells, the emission probability for the i-state is given by $\prod_{c=1}^{C} p(y_{c,t}|S_t = i)^{S_{t,i}}$.

Let $\boldsymbol{\Lambda} = \{\lambda_{ic}\}$ be an m-by-C matrix, and let $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})$ denote all the unknown parameters. Under the assumption of Poisson distribution for spike counts, the observations $\mathbf{y}_t$ at different time indices t are mutually independent, the observed data likelihood is given by

$$\begin{aligned} p(\mathbf{y}_{1:T}|S_{1:T}, \boldsymbol{\theta}) &= \Pr(\mathbf{y}_{1:T}|S_{1:T}, \boldsymbol{\theta}) \\ &= \prod_{t=1}^{T} \prod_{c=1}^{C} \prod_{i=1}^{m} \left(\frac{\exp(-\lambda_{ic})\lambda_{ic}^{y_{c,t}}}{y_{c,t}!} \right)^{S_{t,i}}. \end{aligned} \tag{3}$$

The hidden variables $S_{1:T}$ are treated as the missing data, $\mathbf{y}_{1:T}$ as the observed (incomplete) data, and their combination $\{S_{1:T}, \mathbf{y}_{1:T}\}$ as the complete data, we write the complete data likelihood as

$$\begin{aligned} p(S_{1:T}, \mathbf{y}_{1:T}|\boldsymbol{\theta}) &= p(\mathbf{y}_{1:T}|S_{1:T}, \boldsymbol{\theta}) p(S_{1:T}|\boldsymbol{\theta}) \\ &= \prod_{t=1}^{T} p(\mathbf{y}_t|S_t, \boldsymbol{\theta}) p(S_t|S_{t-1}, \boldsymbol{\theta}). \end{aligned} \tag{4}$$

And the complete data log-likelihood, denoted as $\mathcal{L}$, is derived as (by ignoring the constant)

$$\begin{aligned} \mathcal{L} &= \log p(S_{0:T}, \mathbf{y}_{1:T}|\boldsymbol{\theta}) \\ &= \sum_{t=1}^{T} \sum_{c=1}^{C} \sum_{i=1}^{m} \gamma_t(i) \Big(y_{c,t} \log \lambda_{ic} - \lambda_{ic} \Big) \\ &\quad + \sum_{i=1}^{m} \gamma_1(i) \log \pi_i \\ &\quad + \sum_{t=2}^{T} \sum_{i=1}^{m} \sum_{j=1}^{m} \xi_t(i, j) \log P_{ij}, \end{aligned} \tag{5}$$

where $\gamma_t(i) = \Pr(S_t = i)$ and $\xi_t(i, j) = \Pr(S_{t-1} = i, S_t = j)$.

The maximum likelihood (ML) inference procedure for the standard finite HMM is given by an efficient estimation procedure known as the EM algorithm (Dempster et al. 1977; McLachlan and Krishnan 2008), which is also referred to as the Baum-Welch algorithm (Baum et al. 1970). The EM algorithm iteratively and monotonically maximizes (or increases) the log-likelihood function given the incomplete data. In the E-step, a forward-backward procedure is used to recursively estimate the hidden state posterior probability. In the M-step, based on the sufficient state statistics (estimated from the E-step), the re-estimation procedure is used to estimate the unknown parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})$. For self-contained purpose, the details of the EM algorithm is presented in Appendix A.

In our current application, the hidden state trajectory corresponds to the animal's *directional* position in the track, the number of states m corresponds to the number of bins used for representing the virtual environment, and the m-by-C matrix $\boldsymbol{\Lambda}$ corresponds to the place fields of ensemble neurons, with each row representing one neuronal tuning curve with respect to the m-dimensional state space.

2.1 Practical estimation issues

The above-described estimation procedure is based on ML estimation. In practice, the ML estimation might not be desirable while dealing with large-scale problems in the presence of small size. For the current estimation problem, assuming that the spatial environment is divided into m non-overlapping regions that are represented by m discrete states. Given the observation of spike counts from C neurons within T time intervals, the size of unknown parameters is $\dim(\boldsymbol{\theta}) = m^2 + mC + m$. In a typical experimental protocol of a spatial navigation task, we have $T \ll \dim(\boldsymbol{\theta})$. Therefore, for

an ensemble of $C = 20 \sim 50$ cells and a reasonable size $m = 60 \sim 200$, the parameter space is very large and estimation might be subject to overfitting. On the other hand, since the EM algorithm only searches for the locally optimal solution that are prone to the local optima problem, the initialization of the parameters are important for obtaining for a good solution.

With these practical concerns in mind, it is important to impose certain constraints or priors onto the HMM. In the probabilistic framework, the ML estimation problem is converted into a *maximum a posteriori* (MAP) problem; and the likelihood inference is replaced by the Bayesian inference. The Bayesian estimate is optimal, especially in the presence of small sample size in statistical inference (Gelman et al. 2004). For the HMM, the following three types of Bayesian inference approaches can be considered, with gradually increasing model and computational complexity.

- empirical Bayesian: In this approach, strong structural priors can be imposed onto the HMM, such as the entropic prior (Brand 1999; Brand and Ketnaker 2000). In this case, the MAP solution is straightforward to resolve a modified optimization problem.
- parametric hierarchical Bayesian: In this approach, the parameters of the HMM are assigned with hierarchical priors. The inference algorithm can be based on either MCMC (Scott 2002; Rydén 2008), or ensemble learning (MacKay 1997), or VB-EM (Beal 2003; Ji et al. 2006; McGrory and Titterington 2009).
- nonparametric hierarchical Bayesian: In this approach, the statistical model is treated as a stochastic process with an infinite capacity; a direct extension of the HMM gives rise to the infinite HMM (Beal et al. 2002; Beal 2003). Statistical inference is based on either Gibbs sampling (Teh et al. 2006) or beam sampling (van Gael et al. 2008).

In the case of space navigation task for rodent, due to behavior prior or constraint, it is reasonable to impose a sparsity structure on $\boldsymbol{P}$, which is either diagonal or banded diagonal. With this imposed constraint, the size of unknown variables reduces dramatically, decreasing from quadratic $\mathcal{O}(m^2)$ to linear $\mathcal{O}(m)$ order.

Another important issue for using the HMM is to determine m—the size of hidden states. A naive solution is to empirically choose different values of m, and then conduct model selection based on certain statistical criteria. However, this solution is not necessarily effective since the EM algorithm has the local minimum problem and it is dependent on the initialization of the parameters. Alternatively, the natural solution is to learn all unknown parameters $\boldsymbol{\theta} = \{m, \boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}\}$ from the observed data. In this paper, for the purpose of reducing computational complexity and gaining empirical insights in the first-round investigation, we fix the model size or the number of the hidden states in the inference procedure. We will revisit the model selection issue in Section 6.

3 Variational Bayesian inference for hidden Markov model

In the literature, the VB inference has been used for HMM in various problem settings (MacKay 1997; Beal 2003; Ji et al. 2006; McGrory and Titterington 2009; Katahira et al. 2010). The advantage of VB inference lies in its computational efficiency for Bayesian inference. To avoid model overfitting in the ML estimation, instead of maximizing the log-likelihood function $\log p(\mathbf{y}_{1:T}|\boldsymbol{\theta})$, the objective of VB inference is to maximize the marginal log-likelihood or its lower bound

$$\begin{aligned}
\log p(\mathbf{y}_{1:T}) &= \log \int d\boldsymbol{\pi} \int d\boldsymbol{P} \int d\boldsymbol{\Lambda} \sum_{S_{1:T}} p(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) \\
&\quad \times p(\mathbf{y}_{1:T}, S_{1:T}|\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) \\
&= \log \int d\boldsymbol{\pi} \int d\boldsymbol{P} \int d\boldsymbol{\Lambda} \sum_{S_{1:T}} q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T}) \\
&\quad \times \frac{p(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) p(\mathbf{y}_{1:T}, S_{1:T}|\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})}{q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T})} \\
&\geq \int d\boldsymbol{\pi} \int d\boldsymbol{P} \int d\boldsymbol{\Lambda} \sum_{S_{1:T}} q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T}) \\
&\quad \times \log \frac{p(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) p(\mathbf{y}_{1:T}, S_{1:T}|\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})}{q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T})} \\
&= \Big\langle \log p(\mathbf{y}_{1:T}, S_{1:T}, \boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) \Big\rangle_q \\
&\quad + \mathcal{H}_q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T}) \equiv \mathcal{F}(q) \qquad (6)
\end{aligned}$$

where $q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T})$ is called the variational posterior distribution that approximates the joint posterior of the hidden state and parameter $p(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T}|\mathbf{y}_{1:T})$. The term $\mathcal{H}_q$ of Eq. (6) represents the entropy of the distribution q, and $\mathcal{F}$ is called the free energy (in light of statistical physics).

By assuming a factorial form of variational posterior distribution

$$\begin{aligned}
q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T}) &= q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) q(S_{1:T}) \\
&\approx p(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T}|\mathbf{y}_{1:T}) \qquad (7)
\end{aligned}$$

Eq. (6) can be further simplified as

$$\begin{aligned}\log p(\mathbf{y}_{1:T}) &\geq \log \int d\boldsymbol{\pi} \int d\boldsymbol{P} \int d\boldsymbol{\Lambda} \sum_{S_{1:T}} q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T}) \\ &\quad \times \log \frac{p(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) p(\mathbf{y}_{1:T}, S_{1:T}|\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})}{q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T})} \\ &= \log \int d\boldsymbol{\pi} \int d\boldsymbol{P} \int d\boldsymbol{\Lambda} \sum_{S_{1:T}} q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) \\ &\quad \times \Bigg[\log \frac{p(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})}{q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})} + \sum_{S_{1:T}} q(S_{1:T}) \\ &\quad \times \log \frac{p(\mathbf{y}_{1:T}, S_{1:T}|\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})}{q(S_{1:T})} \Bigg] \\ &\equiv \mathcal{F}(q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}), q(S_{1:T})) \end{aligned} \tag{8}$$

where for notation simplicity we have made the conditional on $\mathbf{y}_{1:T}$ in the variational posteriors $q(\cdot)$ implicit.

To maximize the free energy $\mathcal{F}(q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}), q(S_{1:T}))$, we optimize alternatingly with respect to its arguments $q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})$ and $q(S_{1:T})$, which will be done in the VB-M and VB-E steps, respectively.

3.1 VB-M step

In the VB-M step, taking functional derivatives of $\mathcal{F}$ with respect to $q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})$ yields

$$\begin{aligned}&\log q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) \\ &\quad \propto \log p(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) \big\langle \log p(\mathbf{y}_{1:T}, S_{1:T}|\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) \big\rangle_{q(S_{1:T})} \\ &\quad \propto \log p(\boldsymbol{\pi}) + \log p(\boldsymbol{P}) + \log p(\boldsymbol{\Lambda}) \\ &\quad + \big\langle \log p(S_1|\boldsymbol{\pi}) \big\rangle_{q(S_1)} + \big\langle \log p(S_{2:T}|S_1, \boldsymbol{P}) \big\rangle_{q(S_{1:T})} \\ &\quad + \big\langle \log p(\mathbf{y}_{1:T}|S_{1:T}, \boldsymbol{\Lambda}) \big\rangle_{q(S_{1:T})} \end{aligned} \tag{9}$$

We further impose a factorial form onto the variational posterior of the parameters

$$q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) = q(\boldsymbol{\pi}) q(\boldsymbol{P}) q(\boldsymbol{\Lambda}) \tag{10}$$

To derive individual variational posteriors, we assume appropriate conjugate prior in order to get an analytic form of the posterior.

For the initial state probability $\boldsymbol{\pi}$, we assume a conjugate Dirichlet prior (denoted by $\mathtt{Dir}$):

$$\begin{aligned} p(\boldsymbol{\pi}) &= \mathtt{Dir}\Big(\{\pi_1, \ldots, \pi_m\}|\mathbf{u}^{(\pi)}\Big) \\ &= \frac{\Gamma(u_0^{(\pi)})}{\prod_{i=1}^m \Gamma(u_i^{(\pi)})} \prod_{i=1}^m \pi_i^{u_i^{(\pi)}-1} \end{aligned} \tag{11}$$

where $\mathbf{u}^{(\pi)} = \big[u_1^{(\pi)}, \ldots, u_m^{(\pi)}\big]$, $u_i^{(\pi)} \geq 0$, and $u_0^{(\pi)} = \sum_{i=1}^m u_i^{(\pi)}$ denotes the strength of the Dirichlet distribution. From the Bayes rule, it is inferred that the posterior is also a Dirichlet distribution:

$$q(\boldsymbol{\pi}) = \mathtt{Dir}\Big(\big\{\pi_1, \ldots, \pi_m\big\}\big|\big\{w_1^{(\pi)}, \ldots, w_m^{(\pi)}\big\}\Big) \tag{12}$$

where $w_i^{(\pi)} = u_i^{(\pi)} + q_S(S_1 = i) = u_i^{(\pi)} + \gamma_1(i)$.

Similarly, we can derive the posterior for the transition probability matrix $\boldsymbol{P}$ as the products of posteriors of its row vectors

$$\begin{aligned} q(\boldsymbol{P}) &= \prod_{i=1}^m q(\boldsymbol{P}_i) \\ &= \prod_{i=1}^m \mathtt{Dir}\Big(\{P_{i1}, \ldots, P_{im}\}|\big\{w_{i1}^{(P)}, \ldots, w_{im}^{(P)}\big\}\Big) \end{aligned} \tag{13}$$

where $w_{ij}^{(P)} = u_{ij}^{(P)} + \sum_{t=2}^T q_S(S_{t-1} = i, S_t = j) = u_{ij}^{(P)} + \sum_{t=2}^T \xi_t(i, j)$.

Given the Poisson likelihood for the rate parameters $\boldsymbol{\Lambda} = \{\lambda_{ic}\}$, we assume a conjugate gamma prior (denoted by $\mathtt{Gam}$) for each state i (shared by all cell indices $c = 1, \ldots, C$):

$$\begin{aligned} p(\lambda_{ic}) &= \mathtt{Gam}\Big(\alpha_i^{(\lambda)}, \beta_i^{(\lambda)}\Big) \\ &= \frac{(\beta_i^{(\lambda)})^{\alpha_i^{(\lambda)}}}{\Gamma(\alpha_i^{(\lambda)})} \lambda_{ic}^{\alpha_i^{(\lambda)}-1} e^{-\beta_i^{(\lambda)}\lambda_{ic}} \end{aligned} \tag{14}$$

where $\alpha_i^{(\lambda)} > 0$ and $\beta_i^{(\lambda)} > 0$ are the *shape* and *inverse scale* parameters, respectively.[1] The above gamma distribution has a mean $\alpha_i^{(\lambda)}/\beta_i^{(\lambda)}$ and variance $\alpha_i^{(\lambda)}\big(\beta_i^{(\lambda)}\big)^{-2}$. Correspondingly, the rate parameters follow a gamma posterior

$$\begin{aligned} q(\boldsymbol{\Lambda}) &= \prod_{i=1}^m \prod_{c=1}^C q(\lambda_{ic}) \\ &= \prod_{i=1}^m \prod_{c=1}^C \mathtt{Gam}\Big(C\alpha_i^{(\lambda)} + \sum_{t=1}^T y_{c,t}\gamma_t(i), C\beta_i^{(\lambda)} + l_i\Big) \end{aligned} \tag{15}$$

where $l_i = \sum_{t=1}^T \gamma_t(i)$.

[1] The Jefferey's improper prior corresponds to a limiting case of the gamma distribution, with a shape parameter of 0.5 and inverse scale parameter of 0.

3.2 VB-E step

In the VB-E step, the variational joint posterior of the hidden state is given by

$$q(S_{1:T}) = \prod_{i=1}^{m} \pi_i^{\gamma_1(i)} \prod_{t=2}^{T} \prod_{i=1}^{m} \prod_{j=1}^{m} P_{ij}^{\xi_t(i,j)} \times \prod_{t=1}^{T} \prod_{i=1}^{m} \prod_{c=1}^{C} \mathtt{Poi}(y_{c,t}|\lambda_{ic}, S_t = i)^{\gamma_t(i)} \tag{16}$$

Maximizing $\mathcal{F}$ precedes by taking a functional derivative with respect to $q(S_{1:T})$, which yields

$$\begin{aligned} \log q(S_{1:T}) &= \left\langle \log p(S_{1:T}, \mathbf{y}_{1:T}|\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) \right\rangle_{q(\boldsymbol{\pi})q(\boldsymbol{P})q(\boldsymbol{\Lambda})} \\ &\quad - \log Z(\mathbf{y}_{1:T}) \\ &= \sum_{i=1}^{m} \hat{S}_{1,i} \left\langle \log \pi_i \right\rangle_{q(\boldsymbol{\pi})} \\ &\quad + \sum_{t=2}^{T} \sum_{i=1}^{m} \sum_{j=1}^{m} \hat{S}_{t-1,i} \hat{S}_{t-1,j} \left\langle \log P_{ij} \right\rangle_{q(\boldsymbol{P})} \\ &\quad + \sum_{t=1}^{T} \sum_{c=1}^{C} \sum_{i=1}^{m} \hat{S}_{t,i} \left\langle -\lambda_{ic} + y_{c,t} \log \lambda_{ic} \right\rangle_{q(\boldsymbol{\Lambda})} \\ &\quad - \log Z(\mathbf{y}_{1:T}) \end{aligned} \tag{17}$$

where $\hat{S}_{t,i} = \mathbb{E}_{q(S_{1:T})}[S_t = i] = q_S(S_t = i|\mathbf{y}_{1:T}) \equiv \gamma_t(i)$ and $\hat{S}_{t-1,i}\hat{S}_{t,j} = \mathbb{E}_{q(S_{1:T})}[S_{t-1} = i, S_t = j] = q_S(S_{t-1} = i, S_t = j|\mathbf{y}_{1:T}) \equiv \xi_t(i, j)$ will be computed from the forward-backward algorithm (Appendix A). The last term of Eq. (17), $\log Z(\mathbf{y}_{1:T})$, is a normalization constant that is independent of the variational posterior. To compute the first term of Eq. (17), we have

$$\begin{aligned} \left\langle \log \pi_i \right\rangle_{q(\boldsymbol{\pi})} &= \int \mathtt{Dir}(\boldsymbol{\pi}|\mathbf{u}^{(\pi)}) \log \pi_i d\boldsymbol{\pi} \\ &= \psi\left(u_i^{(\pi)}\right) - \psi\left(\sum_{i=1}^{m} u_i^{(\pi)}\right) \end{aligned} \tag{18}$$

where ψ is the *diagamma function*. To compute the second term of Eq. (17), we have

$$\begin{aligned} \left\langle \log P_{ij} \right\rangle_{q(\boldsymbol{P})} &= \int \mathtt{Dir}(P_{ij}|u_{ij}^{(P)}) \log P_{ij} d\boldsymbol{P} \\ &= \psi\left(u_{ij}^{(P)}\right) - \psi\left(\sum_{i=1}^{m} u_{ij}^{(P)}\right) \end{aligned} \tag{19}$$

To compute the third term of Eq. (17), we have

$$\begin{aligned} &\left\langle -\lambda_{ic} + y_{c,t} \log \lambda_{ic} \right\rangle_{q(\boldsymbol{\Lambda})} \\ &= \int \mathtt{Gam}\left(\lambda_{ic} \middle| C\alpha_i^{(\lambda)} + \sum_t y_{c,t}\gamma_t(i),\, C\beta_i^{(\lambda)} + l_i\right) \\ &\quad \times (-\lambda_{ic} + y_{c,t} \log \lambda_{ic}) d\lambda_{ic} \\ &= -\frac{C\alpha_i^{(\lambda)} + \sum_t y_{c,t}\gamma_t(i)}{C\beta_i^{(\lambda)} + l_i} \\ &\quad + y_{c,t}\psi\left(C\alpha_i^{(\lambda)} + \sum_t y_{c,t}\gamma_t(i)\right) \\ &\quad - y_{c,t} \log(C\beta_i^{(\lambda)} + l_i)\Big) \end{aligned} \tag{20}$$

From Eqs. (18) through (20), we update the new initial state probability as

$$\begin{aligned} \tilde{\boldsymbol{\pi}} &= \{\tilde{\pi}_i\} \\ &= \exp\left\langle \log \pi_i \right\rangle_{q(\boldsymbol{\pi})} \\ &= \exp\left(\psi(w_i^{(\pi)}) - \psi\left(\sum_{i=1}^{m} w_i^{(\pi)}\right)\right), \end{aligned} \tag{21}$$

and update the new state transition probability as[2]

$$\begin{aligned} \tilde{\boldsymbol{P}} &= \{\tilde{P}_{ij}\} \\ &= \exp\left\langle \log P_{ij} \right\rangle_{q(\boldsymbol{P})} \\ &= \exp\left(\psi(w_{ij}^{(P)}) - \psi\left(\sum_{i=1}^{m} w_{ij}^{(P)}\right)\right), \end{aligned} \tag{22}$$

and update the new emission probability as

$$\begin{aligned} \Pr(\mathbf{y}_t|\{\tilde{\lambda}_{ic}\}, S_t = i) &= \prod_{c=1}^{C} \exp\left(\left\langle -\lambda_{ic} + y_{c,t} \log \lambda_{ic} \right\rangle_{q(\boldsymbol{\Lambda})}\right) \\ &= \prod_{c=1}^{C} \exp\left(-\frac{C\alpha_i^{(\lambda)} + \sum_t y_{c,t}\gamma_t(i)}{C\beta_i^{(\lambda)} + l_i}\right) \\ &\quad \times \exp\left(y_{c,t}\psi\left(C\alpha_i^{(\lambda)} + \sum_t y_{c,t}\gamma_t(i)\right)\right. \\ &\quad \left. - y_{c,t} \log(C\beta_i^{(\lambda)} + l_i)\right) \end{aligned} \tag{23}$$

The VB-E step further proceeds with the standard forward-backward algorithm using the new parameter $\tilde{\boldsymbol{\theta}} = (\tilde{\boldsymbol{\pi}}, \tilde{\boldsymbol{P}}, \tilde{\boldsymbol{\Lambda}})$ (computed from Eqs. (31) and (34), Appendix A).

[2] Note that the new probabilities are *sub-normalized probabilities* (due to using the geometric mean instead of the standard arithmetic mean), where $\sum_{i=1}^{m} \tilde{\pi}_i \leq 1$ and $\sum_{j=1}^{m} \tilde{P}_{ij} \leq 1$.

3.3 Computation of the lower bound $\mathcal{F}$

Upon completing every iteration of the VB-E step, we compute the free energy shown in Eq. (8), which is rewritten here:

$$\begin{aligned}\mathcal{F}(q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}), q(S_{1:T})) &= \int q(\boldsymbol{\pi}) \log \frac{p(\boldsymbol{\pi})}{q(\boldsymbol{\pi})} d\boldsymbol{\pi} \\ &+ \int q(\boldsymbol{P}) \log \frac{p(\boldsymbol{P})}{q(\boldsymbol{P})} d\boldsymbol{P} \\ &+ \int q(\boldsymbol{\Lambda}) \log \frac{p(\boldsymbol{\Lambda})}{q(\boldsymbol{\Lambda})} d\boldsymbol{\Lambda} \\ &+ \log \tilde{Z}(\mathbf{y}_{1:T}) \\ &\leq \log \tilde{Z}(\mathbf{y}_{1:T}) \end{aligned} \tag{24}$$

where the inequality holds because the non-negativeness of the Kullback-Leibler (KL) divergence. Note that the first term $\int q(\boldsymbol{\pi}) \log \frac{p(\boldsymbol{\pi})}{q(\boldsymbol{\pi})} d\boldsymbol{\pi}$ of Eq. (24) measures the negative KL divergence between the variational posterior $q(\boldsymbol{\pi}) = \texttt{Dir}(\pi_1, \ldots, \pi_m | u_1, \ldots, u_m)$ and prior Dirichlet distribution $p(\boldsymbol{\pi}) = \texttt{Dir}(\pi_1, \ldots, \pi_m | u'_1, \ldots, u'_m)$ for vector $\boldsymbol{\pi}$ (similarly, for each row of the matrix $\boldsymbol{P}$ in the second term of Eq. (24))

$$\begin{aligned} KL_{\texttt{Dir}}(q \| p) &= \log \frac{\Gamma(u_0)}{\Gamma(u'_0)} + \sum_{i=1}^{m} \log \frac{\Gamma(u'_i)}{\Gamma(u_i)} \\ &+ \sum_{i=1}^{m} (u_i - u'_i) \Big(\psi(u_i) - \psi(u'_i) \Big) \end{aligned} \tag{25}$$

The third term $\int q(\boldsymbol{\Lambda}) \log \frac{p(\boldsymbol{\Lambda})}{q(\boldsymbol{\Lambda})} d\boldsymbol{\Lambda}$ of Eq. (24) measures the negative KL divergence between two gamma distributions $q = \texttt{Gam}(\alpha_1, \beta_1)$ and $p = \texttt{Gam}(\alpha_2, \beta_2)$, which can be computed analytically

$$\begin{aligned} KL_{\texttt{Gam}}(q \| p) &= \log \left(\frac{\Gamma(\alpha_2) \beta_1^{\alpha_1}}{\Gamma(\alpha_1) \beta_2^{\alpha_2}} \right) \\ &+ (\alpha_1 - \alpha_2) \Big(\psi(\alpha_1) - \log \beta_1 \Big) \\ &+ \alpha_1 \frac{\beta_2 - \beta_1}{\beta_1} \end{aligned} \tag{26}$$

Furthermore, the last term $\log \tilde{Z}(\mathbf{y}_{1:T})$ of Eq. (24) is the new normalization constant that can be estimated from the forward-backward algorithm (Eq. (35), Appendix A); it also corresponds to the estimated marginal log-likelihood of the data (Eq. (36), Appendix A).

3.4 Initialization of priors and hyperparameters

The purpose of conjugate priors is to make the VB inference more tractable. However, the hyperparameters of these priors are designed by user, depending on the user's belief on the data. The priors can be either very informative or very uninformative. In that sense, the conjugate prior is still quite general. Obviously, a highly structured solution will require a very specific prior for the desirable solution.

In the previous subsection, the hyperparameters are assumed to be known or set by the user. In our problem, the hyperparameters are set according to the prior knowledge as follows.

- We set $u_1^{(\pi)} = u_2^{(\pi)} = \cdots = u_m^{(\pi)} = 1/m$, which corresponds to a uniform distribution. If the hyperparameter is smaller than $1/m$, it implies that the solution favors a specific initial state, rather than a uniform solution.
- We set $\left[u_{i1}^{(P)}, u_{i2}^{(P)}, \ldots, u_{im}^{(P)}\right] = \alpha^{(P)}[1/m, 1/m, \ldots, 1/m]$, where $\alpha^{(P)}$ denotes the concentration parameter. Values of the concentration parameter above 1 prefer variates that are dense, evenly-distributed distributions (i.e. all probabilities returned are similar to each other). Values of the concentration parameter below 1 prefer sparse distributions (i.e., most of the probabilities returned will be close to 0, and the vast majority of the mass will be concentrated in a few probabilities). We set $\alpha^{(P)} = 0.3$.
- We set $\alpha_i^{(\lambda)} = \beta_i^{(\lambda)} = 0.0001$, and the initial mean of the λ_{ic} is set to be the overall mean firing rate of neuron c, i.e. $\frac{1}{T} \sum_{t=1}^{T} y_{c,t}$.

Alternatively, the hyperparameters $\alpha_i^{(\lambda)}$ and $\beta_i^{(\lambda)}$ can be optimized iteratively by maximizing the log-likelihood or marginal log-likelihood (Appendix C). However, no closed-form solution exists for these hyperparameters.

4 Visualization of spatial topology via force-based algorithm

Spatial topology is a mathematical abstraction of the real environment space. Spatial topology reflects the geometrical structure and spatial relations that are invariant or unaffected by the continuous change of shape or size of figure. For a rodent spatial navigation task in a two-dimensional space, the spatial topology determines the animal's natural behavior. Figure 1 shows a few example experimental spatial topology commonly used in navigation tasks. As seen in Fig. 1, the physical shapes of the experimental tracks (top row) can be converted into the equivalent spatial topology (bottom row) by considering the directional factor; two tracks with physically different shapes (e.g., T-maze vs.

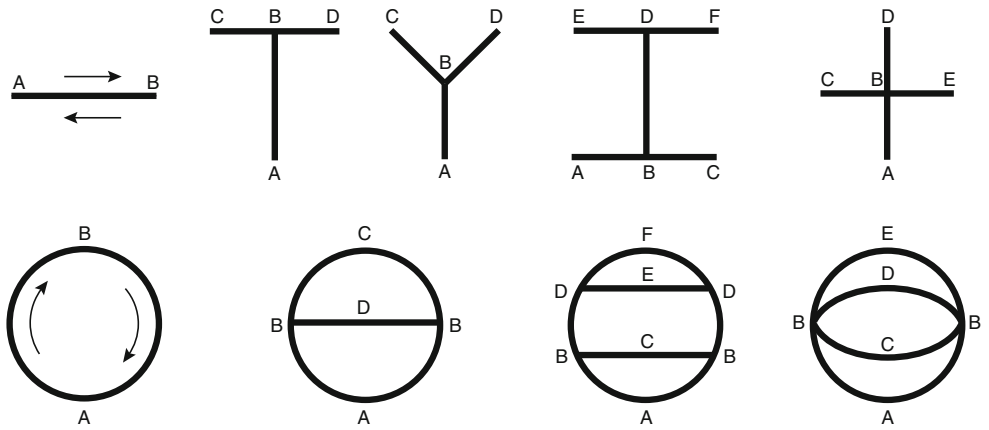


Fig. 1 Examples of experimental space topology explored by animal (for *left* to *right*: linear track, T-maze, Y-maze, H-maze, cross maze). The *arrow* indicates the traveling direction. The first row shows the physical shape of the track, and the second row shows the corresponding (equivalent) topology by considering the bidirectional factor. Note that the T-maze and Y-maze have two bifurcation points (each with 2 possible choices), the H-maze has four bifurcation points (each with 2 possible choices), and the cross-maze has two bifurcation points (each with 3 possible choices)

Y-maze) share the same spatial topology. The bifurcations of the track increases the complexity of the topology since it introduces multiple pathways connecting one point in the track to another. In representing the space, the spatial environments are often binned and linearized. Note that the linearization strategy is non-unique, and there are multiple ways to discretize and represent the same space.

The inference outcomes of the HMM consist of the estimated state trajectory, the state transition probability matrix, and the tuning curves (place fields) of the population neurons. Particularly, the estimated state trajectory and transition probability matrix reveal important cues about the spatial environment and the animal behavior in that environment. Since the behavior determines the state transition probability, we will use the transition probability matrix to infer the spatial topology of the environment. Specifically, we would like to draw a graph that consists of multiple nodes representing the hidden states, the edges between the graph represents the link between the spatial locations coded by the hidden states, whereas the strengths of the edges reflects the values of transition probability between two states, which not only depends on the spatial topology but also the animal's actual behavior. For instance, for the same spatial topology such as a linear track, the state transition matrix will be different between a regular back-and-forth navigation without turns and a navigation with frequent turns inside the track. A non-stop navigation between two end points will induce a shifted diagonal-like structure in the state transition matrix, whereas frequent stops and turns inside the track will induce many nonzero off-diagonal elements in the transition matrix.

A direct way to visualize the spatial topology is to draw graphs. A graph displays the geometrical relationship between distinct nodes or different objects via edges. There are various graph-drawing methods, most of which rely on certain distance metrics. In general, a high transition probability implies a short distance between two nodes in the graph. Meanwhile, for the aesthetic reason, it is preferred that all the edges are proportional in length, and there are as few crossing edges as possible. For instance, the classical or nonclassical multidimensional scaling (MDS) methods (Cox and Cox 2001; Borg and Groenen 2005), which are originally used in information visualization for exploring similarities or dissimilarities in data, can be used here for visualizing the relationship between nodes based on a selected distance metric. To do that, we can transform the transition probability matrix into a symmetric distance (or dissimilarity) matrix. However, the choice of transformation is rather ad hoc, and from our practical experiences the visualization of the graph is less satisfactory and the results are more difficult to interpret (results not shown).

Another type of popular graph drawing methods is based on the force-based algorithm (Tollis et al. 1999). Typically, this type of algorithm is motivated from physics, whereas the nodes are viewed as particles, and the graph is treated as physical (mechanical or electrical) system. At the end of the completing the graph drawing, the total kinetic energy is minimized and the system reaches an equilibrium state. Some

publicly available softwares, for instance the Tulip (http://tulip.labri.fr/) and Gephi© (http://gephi.org), provide interfaces to draw aesthetically satisfactory graphs with various levels of user control. The typical force-based algorithms are generally considered to have a $\mathcal{O}(m^3)$ running time, where m is the number of nodes of the graph. It shall be emphasized out that the force-based algorithm is based on iterative optimization, which also has the poor local minimum problem; thus the graph drawing outcome also depends on the initial condition.

In addition to the available public resources, we have also written our own custom MATLAB© (MathWorks, Natick, MA) programs to visualize the spatial topology in either two-dimensional (2D) or three-dimensional (3D) space. The only input for the program is the estimated (with or without thresholding) state transition matrix and the algorithmic convergence criterion (a default value is also set). The pseudocode for the force-based algorithm is given below (Algorithm 1).[3]

Algorithm 1 Pseudocode for the force-based algorithm

```
Initialize node velocities to (0, 0), initialize node positions randomly.
while non-convergence (i.e., total kinetic energy is greater than desired value) do
    Set total kinetic energy to 0. // running sum of total kinetic energy over all particles
    for each node
        net-force = (0, 0) // running sum of total force on this particular node
        for each other node
            net-force = net-force + Coulomb-repulsion (this node, other node )
        next node
        for each spring connected to this node
            net-force = net-force + Hooke-attraction (this node, spring )
        next spring
    // without damping, it moves forever
        this node.velocity = (this node.velocity + timestep × net-force) × damping
        this node.position = this node.position + timestep × this node.velocity
        total kinetic energy = total kinetic energy + this node.mass × (this node.velocity)^2
    next node
end while
```

[3]Online resource: http://en.wikipedia.org/wiki/Force-based_algorithms_(graph_drawing)

5 Results

5.1 Computer simulations

We have done a variety of computer simulations to verify our analysis. The setup of the simulated experiments is listed in Table 1. In all simulations, we assume that the animal is always in the RUN-mode (i.e., stop periods are excluded), with a 250 ms temporal bin size (about two theta-cycle). For the sake of simulation simplicity, we also assume that the animal runs multiple laps, the running trajectory at each lap is identical (i.e., the overall trajectory is periodic). In addition, the spike activity of ensemble neurons is drawn from a Poisson distribution based on the real tuning curves constructed from experimental tracks.

Due to the presence of local maxima, in each experimental condition we run the VB-EM algorithm multiple times, each with different random initializations. We examine and select the results with the free energy criterion. The solution associated with the higher free energy is more likely to be a better solution. However, the free energy criterion alone may not be sufficient, quantitative assessment of the estimated solutions is also necessary.

5.1.1 Quantitative assessment

For computer simulations, we propose two quantitative indices to measure the quality of the estimation results. The first index measures the quality of estimated trajectory. It is noted that because of the permutation ambiguity of the state ID, two correct trajectories may exhibit different forms after remapping the state ID. For that reason, we first compute the occupancy time (OT) of each state and then sort these values denoted by a vector $\mathbf{OT}$. We then compute the difference between the two vectors

$$D_1 = \frac{\|\mathbf{OT}_{\text{true}} - \mathbf{OT}_{\text{est}}\|}{T/m}, \tag{27}$$

where $\|\cdot\|$ denotes the L_1 norm of the vector, and the denominator T/m denotes the averaged occupancy time of m states within the total T time bins. The index D_1 is aimed to check the consistency between two state trajectories: when two trajectories are perfectly consistent (upon permutation), $D_1 = 0$. In the presence of the state ID ambiguity, provided that each state has a different occupancy number, after sorting the OT, the consistency of two trajectories can be checked without explicit state remapping.

The second index measures the similarity between the true and estimated state transition matrices. Again,

Table 1 Summary of all computer simulations

No.	Environment	m	C	T (laps × bin/lap)	Remark
1-1	Linear track (bidirectional)	62	21	1,240 (20 × 62)	31 bins per direction, without turns
1-2	Linear track (bidirectional)	62	21	3,720 (20 × 186)	With turns in both directions
2-1	T-maze	86	21	5,070 (15 × 338)	Two bifurcations
2-2	T-maze	86	35	4,240 (20 × 212)	Two bifurcations, with turns
3-1	Linear track A + T-maze	86	21	8,790	A is part of of T, multiple transitions
3-2	Linear track A + linear track B	86	21	1,075	A and B are gated, one transition

due to permutation ambiguity of the state ID, it is difficult (if not impossible by an exhaustive search) to compare all possible permutations. To illustrate this point, let's consider a simple example shown in Fig. 2. Suppose that we have two matrices, each of them have two rows that have more than one (here, say two) dominant nonzero off-diagonal entries. All entries of the matrix are nonnegative, and each row entries sum to 1. As illustrated in Fig. 2, we denote the row entries as (a_1, a_2) and (b_1, b_2) in the first matrix (say, the true matrix), and denote the row entries as (c_1, c_2) and (d_1, d_2) in the second matrix (say, the estimated matrix), among which a_1, b_1, c_1, d_1 are the one-column-right-shifted diagonal elements. Given the permutation ambiguity, there are two possibilities to compute the matrix row deviation (MRD): one is $MRD_1 = |a_1 - c_1| + |a_2 - c_2| + |b_1 - d_1| + |b_2 - d_2|$, the other is $MRD_2 = |a_1 - d_1| + |a_2 - d_2| + |b_1 - c_1| + |b_2 - c_2|$. Obviously, the one with the smallest MRD value would be a more desirable solution; namely, using the row permutation option associated with the smallest MRD, the estimated matrix will be more similar to the true matrix (at least for the two rows under consideration).

The cartoon example in Fig. 2 is only aimed to illustrate the situation when considering two matrix rows that have two dominant nonzero off-diagonal entries. In a more general setting, when considering to compare n rows in the m-by-m true matrix (where $n \geq 2$ and $n \ll m$) that have two dominant nonzero off-diagonal entries,[4] there will be $n!$ (the factorial of n) permutation possibilities and we would need to compute a total of $n!$ MRD values. From which, we define the second index as

$$D_2(n) = \min_k \{MRD_k\}, \quad k = 1, \ldots, n! \tag{28}$$

For the sake of computational simplicity, we have ignored the comparison among the remaining $(m - n)$ rows between two matrices. In our case (when only two dominant nonzero off-diagonal elements in each row are considered), it can be shown that $0 \leq D_2(n) \leq n$. In all computer simulations conducted below, we only consider either $n = 2$ or $n = 3$. Since we can only consider a low dimensionality of n, D_2 shall be combined with D_1 as an additional criterion to assess the solution. Alternatively, we can also use the continuity of state trajectory sequences to compare a large chuck of (more than 3) rows between two matrices.

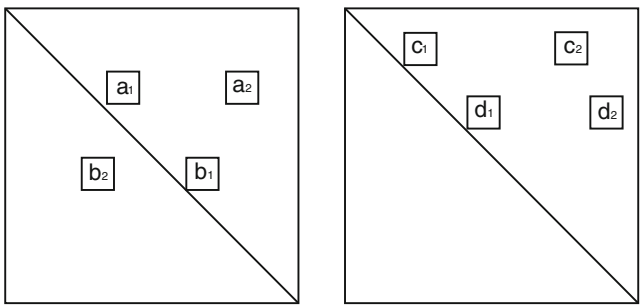


Fig. 2 Cartoon illustration for comparing two matrix-rows that have two dominant off-diagonal elements, where a_1, b_1, c_1, d_1 are the one-column-right-shifted diagonal elements (i.e., column index = row index + 1)

5.1.2 Simulated linear track

In the first simulation scenario, the simulated animal's behavior follows a non-stop back-and-forth navigation (i.e., animal moves from one end of the track to the other end, then returns and the motion repeats). The animal stops nowhere in the middle track (neither at the ends of the track) and makes no turn inside the track. In this case, we would know in advance that the state transition matrix $\boldsymbol{P}$ would have a shifted-diagonal structure. In inference, to impose a linear-track topology preference, in parameter initialization we first set $\boldsymbol{P}$ to have a dominant shifted-diagonal structure; we further add small positive values randomly into the elements of $\boldsymbol{P}$ (to allow other possible state jumps to account for animal's behavior turns inside the track).

[4] In this paper, we only consider this situation. More generally, if there are more than two (say l) dominant nonzero off-diagonal entries, we have to consider not only row permutation but also column permutation, there will be a total of $n! \times (l-1)!$ permutation possibilities.

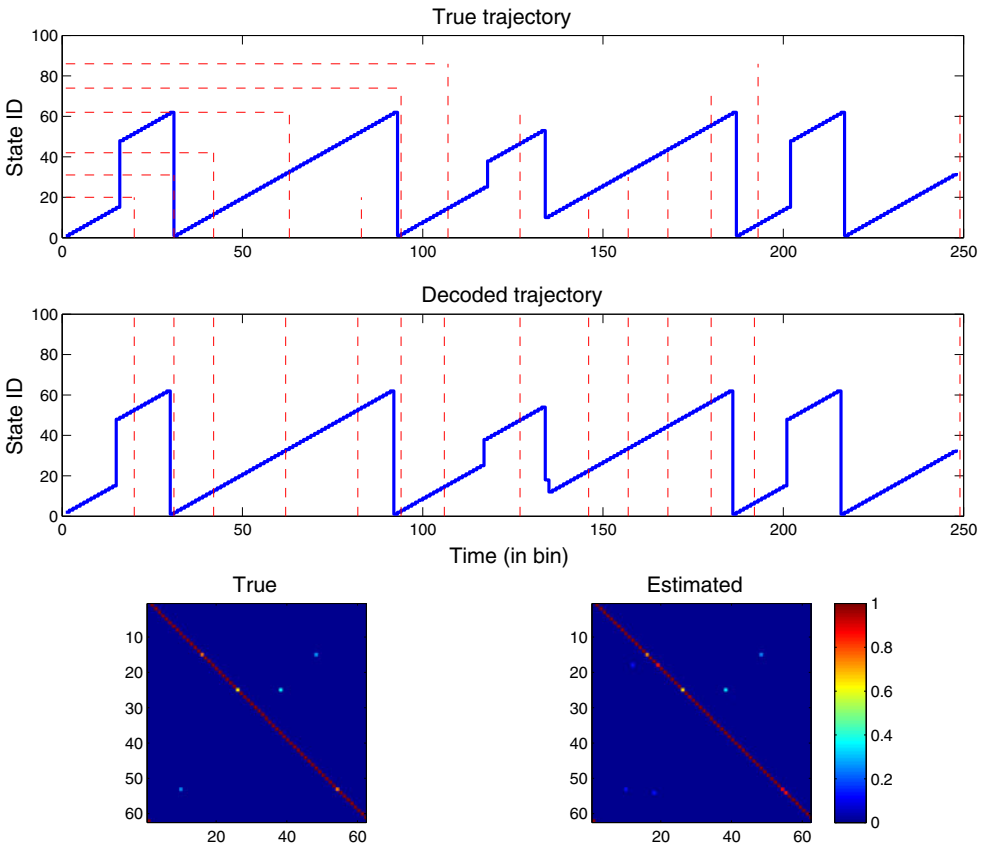


Fig. 3 One illustrated estimation result from the linear track (Simulation 1–2): the true trajectory in one lap (*top*) and the corresponding estimated trajectory (*middle*). State 1-31 represents the left–to–right positions inside the linear track, and state 32-62 represents right–to–left positions inside the track. The color-coded true (*bottom left*) and estimated (*bottom right*) transition matrices are qualitatively and quantitatively similar. Note that the transition matrix has a shifted-diagonal structure. Quantitative indices: $D_1 = 0.1333$, $D_2(2) = 0.0407$, $D_2(3) = 0.0479$

Each row of $\boldsymbol{P}$ is normalized such that the sum of the entries is 1. In practice, we found that this initialization strategy is very effective and leads to fast algorithmic convergence and good estimation performance.[5]

In the second simulation scenario, the animal still navigates in the same linear track environment, but the animal's behavior is different from the first case in that it now makes a few turns in the middle of the track. For instance, the state sequence in one lap to account for the animal's behavior and moving direction is $[1:15, -15:-1, 1:31, -31:-1, 1:25, -25:-10, 10:31, -31:-1]$ (where the negative sign indicates the reverse direction). The simulated true trajectory in one lap and the decoded trajectory obtained from VB-HMM are shown in Fig. 2. As comparison, two trajectories are very similar, so are the true and estimated transition matrices (Fig. 3). As expected, due to behavioral turns at certain state locations (e.g., state 15, 25), there are more than one nonzero elements in a few rows of the transition matrix, indicating the presence of a shortcut between two non-neighboring states. Furthermore, when comparing with the true tuning curves of 21 cells, it is found that the estimated tuning curves have a faithful resemblance (Fig. 4). It is also noted that the VB algorithm is capable of decoding state trajectory accurately despite the fact that many cells have multiple-peak place fields.

It shall be pointed out that there are many equivalent solutions (due to the singularity of latent probabilistic model and the ambiguity of state permutation). In other words, even the decoded state sequence trajectory appears different from the true one, but the solution is actually consistent with the true one after remapping state ID (this can be confirmed by visual inspection or quantitative evaluation). Figure 5 shows such an example. In Fig. 5, the quantitative metrics are $D_1 = 0.2$, $D_2(2) = 0.0292$, $D_2(3) = 0.0363$, as compared to $D_1 = 0.1333$, $D_2(2) = 0.0407$, $D_2(3) = 0.0479$ in Fig. 3.

Notes: At this point, it is worth mentioning several important observations from computer simulations:

– Given a sensible initialization, the VB-HMM algorithm converges very fast, typically within less than 20 iterations. Our algorithm also produces much a

[5]Note that this trick attempts to impose a structural prior. In contrast, a completely random $\boldsymbol{P}$ will cause a slow convergence and a poor solution.

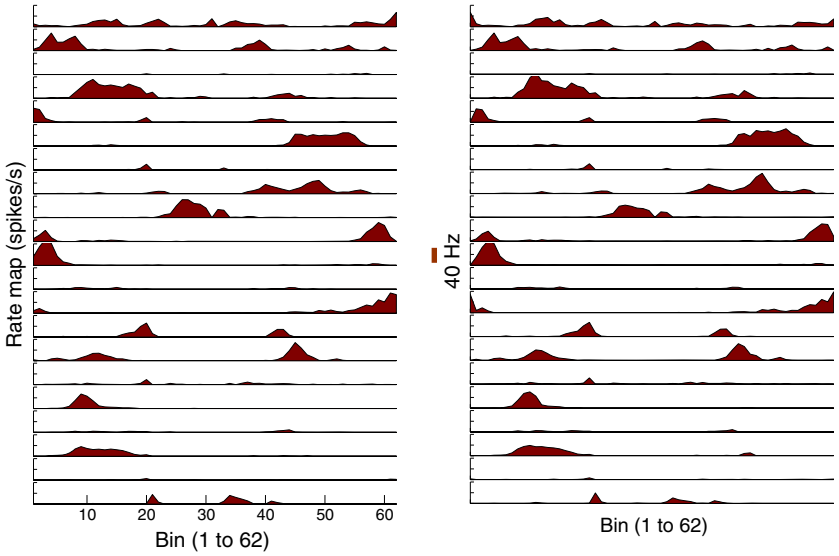


Fig. 4 The simulated (*left*) and estimated (*right*) tuning curves of 21 neurons from the same result shown in Fig. 3 (Simulation 1-2). The full length of the vertical bar marks the firing rate scale of 40 Hz

better solution than the standard EM-HMM algorithm. Without the imposed prior information (as in the VB), the decoded state trajectory obtained from the EM algorithm is rarely comparable to the true trajectory (result not shown). In addition, the estimated transition matrix from the EM algorithm lacks the sparsity structure (data not shown). This is because without the constraints, the solution space is too large, and the EM algorithm is prone to being stuck in poor local maxima.

– As illustrated in Figs. 3 and 5, there are non-unique but equally satisfactory solutions for the decoded trajectory (because of permutation ambiguity). Even the decoded trajectory may appear different from the true trajectory at the first sight, the solution can be consistent and valid upon state

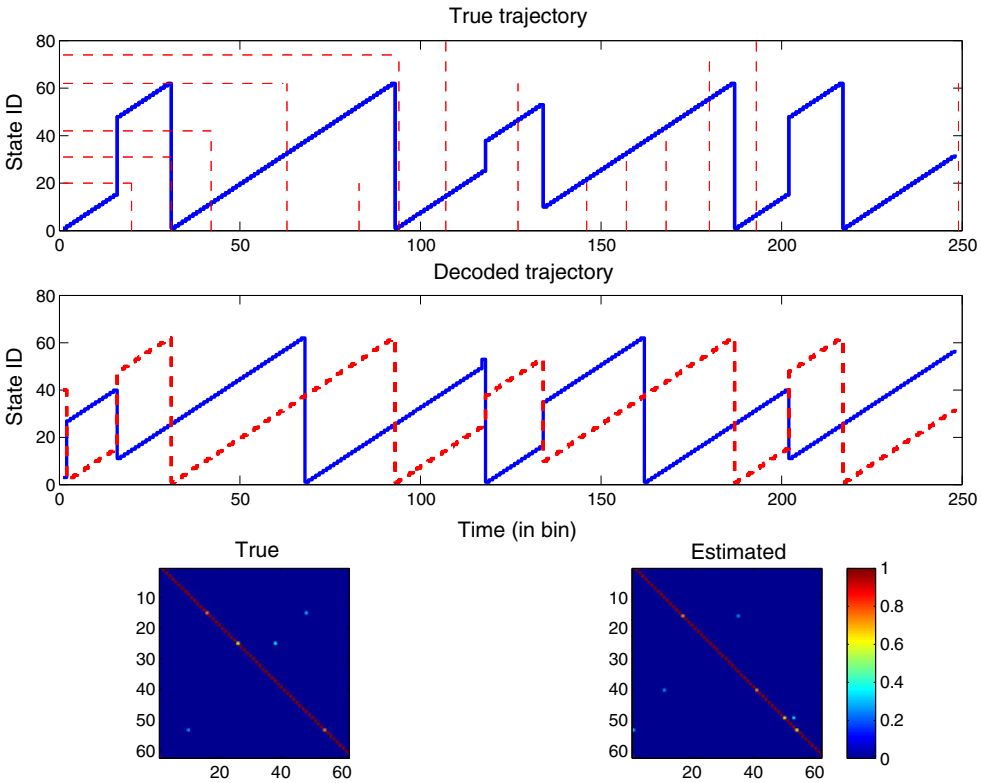


Fig. 5 In comparison with the results illustrated in Fig. 3, another correct estimation result from the trajectory in a linear track (Simulation 1-2). Note that the the raw (*top panel*) and remapped (*second panel, dashed line*) state trajectories will become nearly identical upon state ID remapping (using the following ID map: [1:25]→[38:62], [26:62]→[1:37]). Also note that the true (*bottom left*) and estimated (*bottom right*) transition matrices will become nearly identical upon state ID remapping. Quantitative indices: $D_1 = 0.1333$, $D_2(2) = 0.0560$, $D_2(3) = 0.093$

ID remapping. The proposed quantitative measures D_1 and D_2 provide a hint about the quality of the estimation. However, it shall be noted that the D_1 value also depends on the distribution of the actual state occupancy time. Because of the data dependency, the comparison of the D_1 value only makes sense among the same simulation experiment.

– Typically, a small D_1 value is accompanied by a small D_2 value and a large free energy $\mathcal{F}$. However, the reverse statement is not always true. In other words, sometimes a large free energy may be associated with relatively large D_1 and D_2 values, or sometimes even when D_2 and the negative free energy $-\mathcal{F}$ is small, the D_1 value can be large. To show that, we have conducted 50 independent Monte Carlo simulations for Simulation 1-2, and the statistics of D_1, D_2 and $\mathcal{F}$ are shown in Fig. 6. Note that these results consist of all solutions with different degrees of performance (both failures and successes). In our observations, a "qualitatively good" solution is often accompanied with lower values of D_1 and D_2, in combination of reasonably high value of free energy; a "qualitative bad" solution is often accompanied with a low free energy, a high D_2 value. In this specific Monte Carlo experiment, the conservative failure rate estimate is around 10–14% (5–7 cases).

5.1.3 Simulated T-maze

Linear track is the simplest spatial topology among all rat navigation tasks. Next, a slightly more complex spatial topology—T-maze, is considered. In the first simulation scenario, the animal navigates in a T-maze (Fig. 7, leftmost panel). In each lap, the animal makes random exploration to the left or right arms of the maze. In this case, the animal makes no turn in the middle of two arms. Based on the same initialization setup as before, the VB-HMM algorithm is capable of producing very accurate estimation results. One of the estimation results from this simulation is illustrated in Fig. 7. As seen, the estimated state transition probability matrix is very similar to the true one ($D_2(2) = 0.015$). Note that the transition matrix has two bifurcation points at states 20 and 42, and there is also a break point at state 62. In addition, the estimated state sequence trajectory is also consistent with the true one ($D_1 = 0$), and so are the estimated tuning curves (not shown).

In the second simulation scenario, the animal is assumed to navigate in the same T-maze environment. However, the animal makes regular turns inside the two arms of the maze. Using the VB-HMM algorithm, one of the estimation results from this simulation is illustrated in Fig. 8. By inspection, the estimated state transition matrix and state sequence trajectory are also consistent, achieving excellent quantitative metrics: $D_1 = 0$, $D_2(2) = 0.024$, $D_2(3) = 0.038$.

5.1.4 Combined environments

We further examine the scenario with two combined spatial environments, which is not uncommon in some rodent navigation protocols. In the first simulation scenario, we consider one linear track A and one T-maze, where the linear track A is part of the T-maze (i.e., one arm of the T-maze). The linear track A is represented by 62 states, and the T-maze is represented by 86 states. Therefore, the combined environment is also represented by 86 states. In each lap, the animal first explores the linear track (state 1–62), and then the animal is exposed to the complete environment (state 1–86). Although the task is more challenging than the first two scenarios, the VB-HMM algorithm still performs quite well. One of the estimated results is illustrated in the left panel of Fig. 9. By inspection, we see the estimated state sequence is consistent with the true one,

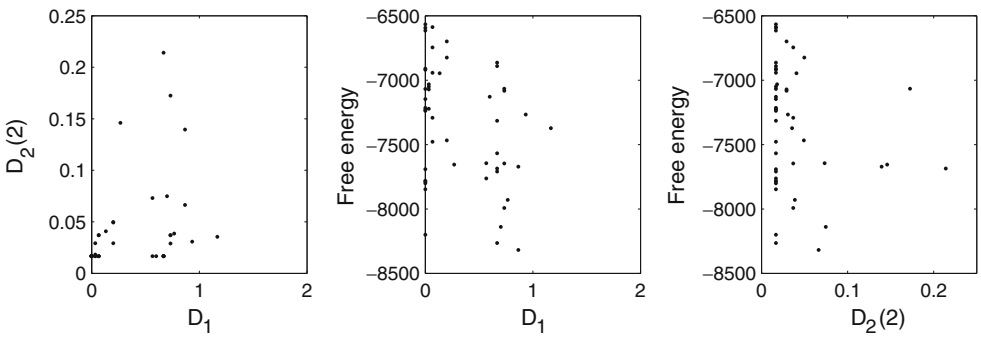


Fig. 6 Scatter plots of statistics of D_1, $D_2(2)$, and free energy F. Statistics are obtained from 50 independent Monte Carlo simulations (Simulation 1-2)

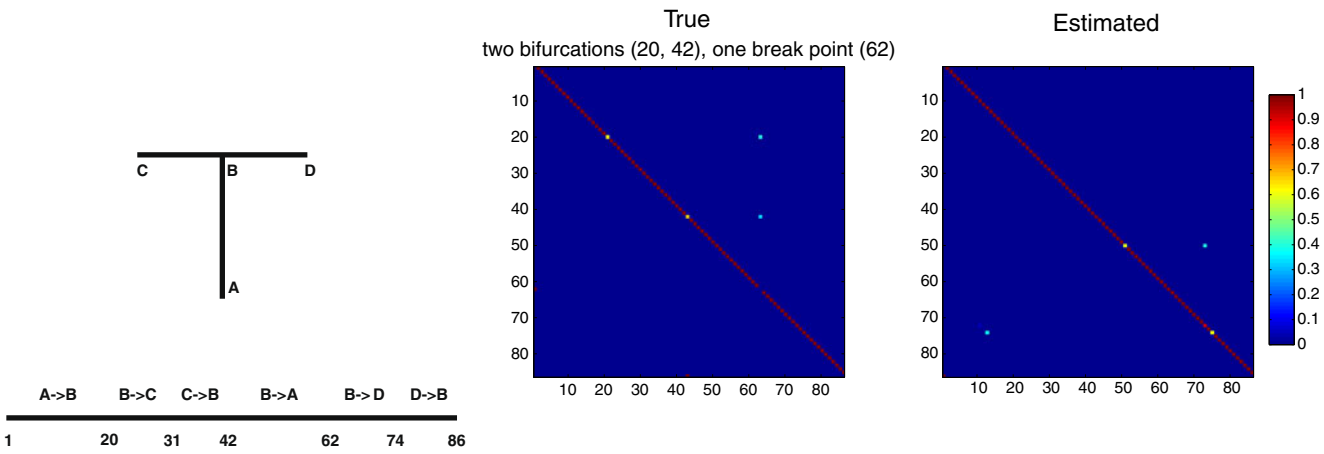


Fig. 7 *Left* Illustration of linearization of a simulated T-maze. Due to the bidirectional factor of the place field, a total of $43 \times 2 = 86$ states represents the 43 bins. Linearized bin assignment: $A \rightarrow B$: 1:20, $B \rightarrow C$: 21:31, $C \rightarrow B$: 32:42, $B \rightarrow A$: 43:62, $B \rightarrow D$: 63:74, $D \rightarrow B$: 75:86. One illustrated result from the simulated T-maze (Simulation 2-1): color-coded true (*middle*) and estimated (*right*) transition probability matrices, and $D_2(2) = 0.015$. The comparison of the true and estimated trajectories are not shown here

this is also confirmed by $D_1 = 0.783$ and the statistics of the respective sorted state occupancy time (right panel, Fig. 9). The estimated transition probability matrix has a relatively similar structure as the one shown in Fig. 7 (data not shown, $D_2(2) = 0.698$).

In the second simulation scenario, we consider the combination of two linear tracks (A and B) without overlapping region between each other. Two linear tracks are represented by 43 states, resulting a total of 86 states with the bidirectional factor. In the first

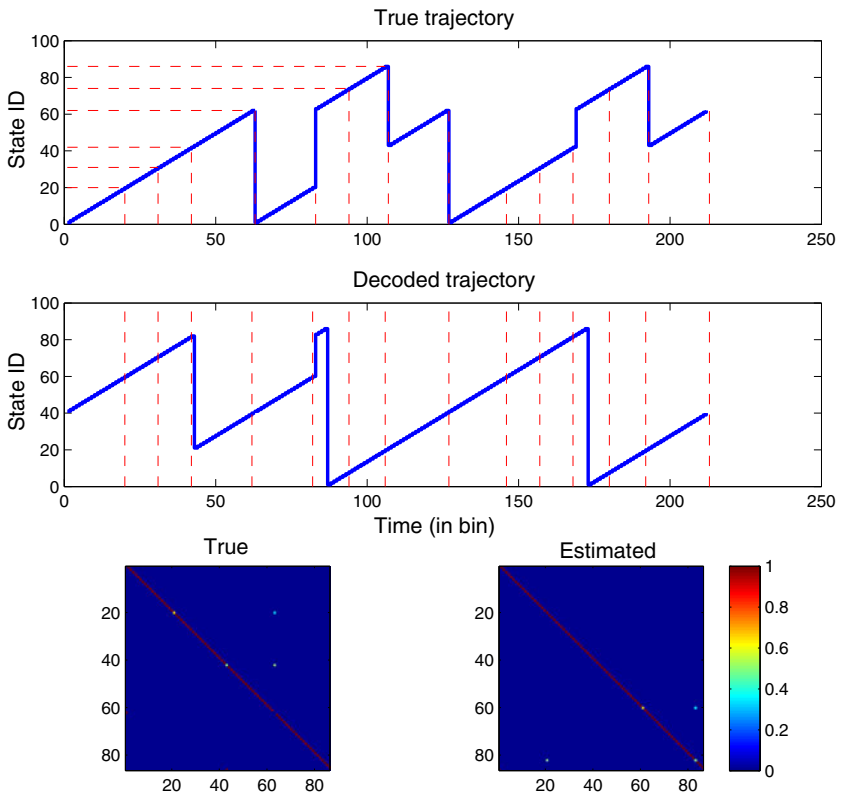


Fig. 8 One illustrated snapshot from the simulated T-maze (Simulation 2-2). *Top* Comparison of the true and estimated trajectories. *Bottom* Comparison of the true (*left*) and estimated transition matrices (*right*). Quantitative indices: $D_1 = 0$, $D_2(2) = 0.024$, $D_2(3) = 0.038$

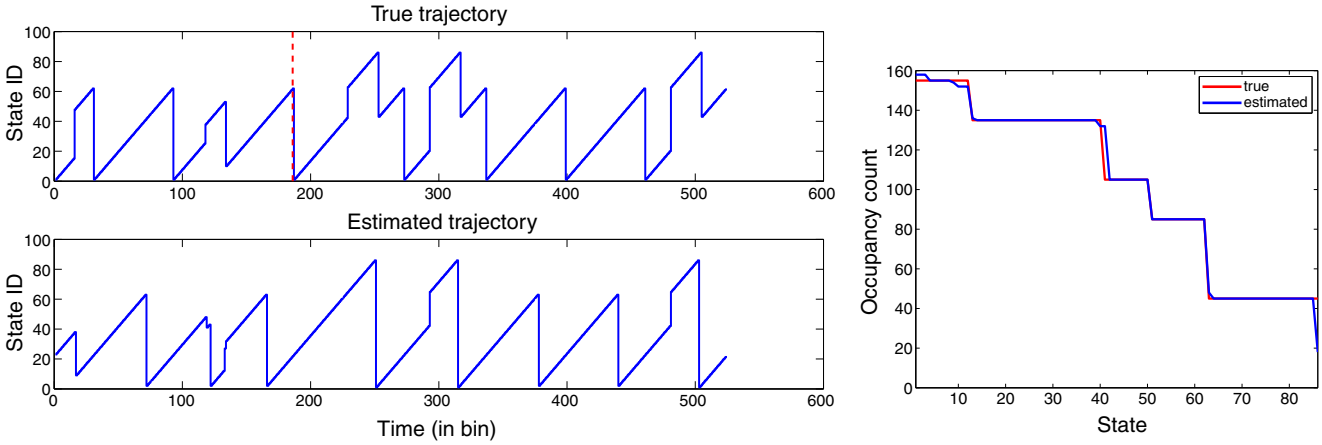


Fig. 9 One illustrated snapshot from two combined environments (Simulation 3-1): the comparison of the snapshots of the true (*left top*) and estimated (*left bottom*) trajectories, with the transition marked by a dashed line. In this case, two environments have overlapping regions. From the true and estimated state trajectories, we can compare the statistics of the state occupancy time (*right*) and obtain $D_1 = 0.783$

865 temporal bins, the animal first explores the linear track A (state 1–62), and then the gate between two tracks opens and closes behind once the animal moves to the linear track B (state 63–86) and explores in the remaining 147 temporal bins. Therefore, there is only one single transition chaining two environments. To our little surprise, the VB-HMM algorithm is still capable of recovering the behavior trajectory. One of

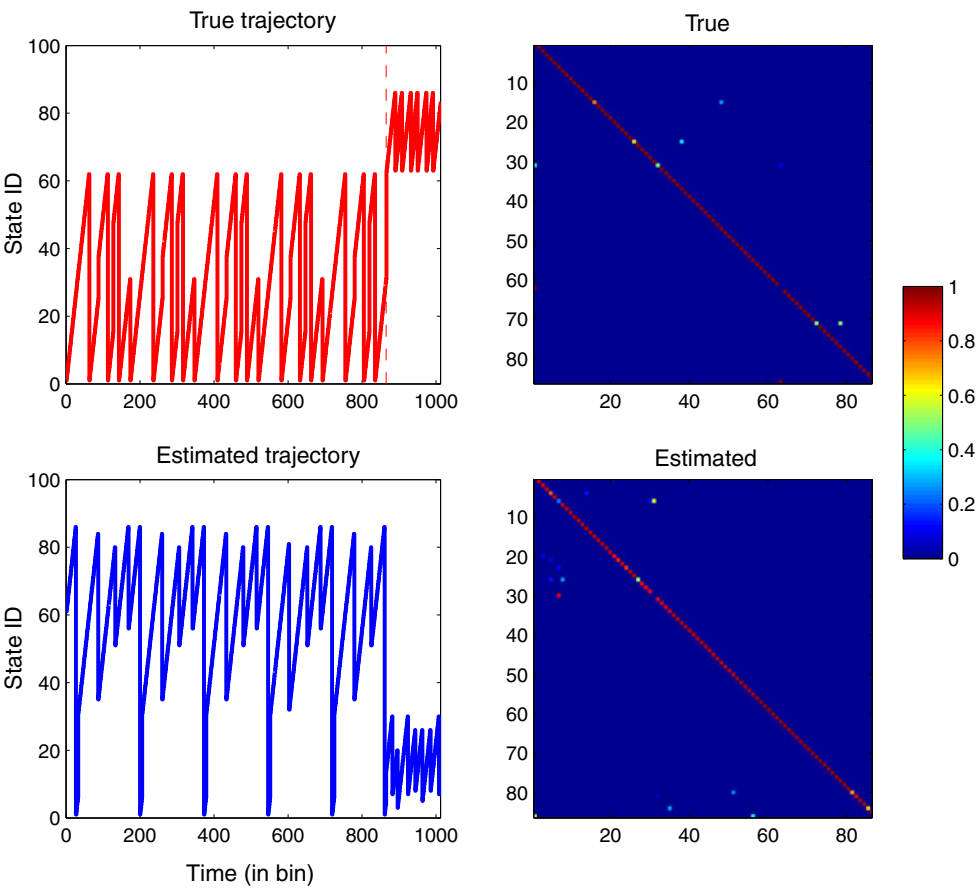


Fig. 10 One illustrated result from two combined environments (Simulation 3-2): the comparison of the snapshots of the true (*top left*) and estimated (*bottom left*) trajectories. Note that two environments have no overlapping region, the one-time transition between A and B ($31 \rightarrow 63$) is marked by a *dashed line*. Linear track A has state ID 1–31 (forward direction) and 32–62 (reverse direction); linear track B has state ID 63–74 (forward direction) and 75–86 (reverse direction). Quantitative indices: $D_1 = 3.04$, $D_2(2) = 0.313$, $D_2(3) = 0.445$

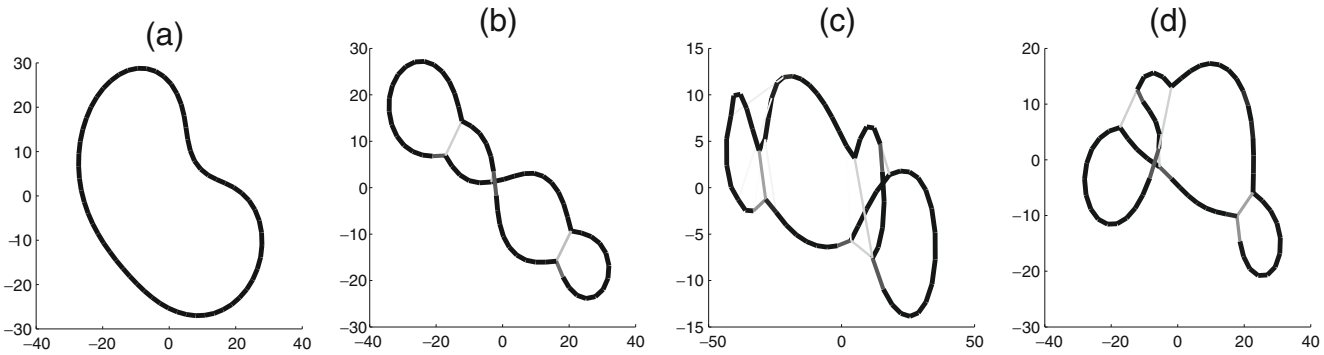


Fig. 11 (**a**) The graph inferred from the estimated transition matrix from a simulated linear track without behavioral turns (Simulation 1-1, $m = 62$). (**b**)–(**d**) The graphs inferred from a simulated linear track with behavioral turns (Simulation 1-2, $m = 62$): from the true transition matrix (**b**), the estimated transition matrix (**c**) and the estimated transition matrix followed by 0.05 thresholding (**d**)

the estimated results are illustrated in Fig. 10. As seen, even with a sample size as small as 1012 bins, our approach can decode the state trajectory rather reliably ($D_1 = 3.04$) and produce a reasonably good estimate of the state transition matrix ($D_2(2) = 0.313$, $D_2(3) = 0.445$).

5.1.5 Interpretation of graphs

The topological graph reveals important information about the spatial topology of the environment as well as the animal's behavior. Take a look at the examples of the inferred graphs shown in Fig. 11, Fig. 11(a) is a graph obtained from Simulation 1-1, Fig. 11(b) is the inferred graph from the true state transition matrix used in Simulation 1-2, Fig. 11(c) and (d) are the inferred graphs from the estimated state transition matrix in the same simulation, without and with thresholding, respectively.

We would like to point out a few important facts in interpretation of the graphs:

- The end-to-end navigation behavior (i.e., no turns) in the linear track and T-maze environment will have simpler graphs (shown in Fig. 1, bottom left two graphs). This example is perfectly illustrated in the graph of Fig. 11(a), which is inferred from Simulation 1-1.
- Whenever there are navigation turns inside the track, shortcuts will be created in the graph. The number of locations at which the turns occur determines the number of shortcuts. This can be perfectly illustrated in the graph of Fig. 11(b): in addition to the "8"-figure topology (solid string in dark color), two shortcuts (weak edges in light color) are created.
- In the T-maze, since there are bifurcation points (e.g., states 20 and 42 in Simulation 2-2) as well as

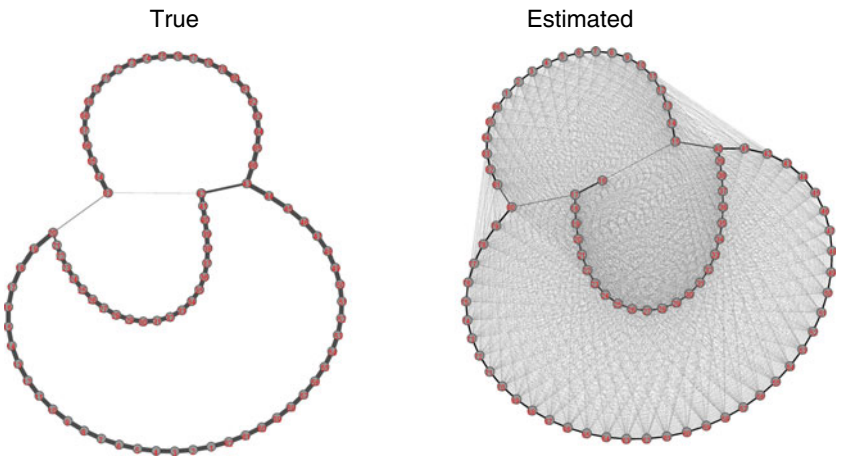


Fig. 12 The graphs inferred from the true estimation matrix (*left*) and the estimated transition matrix without thresholding (*right*) in a simulated T-maze (Simulation 2-2, $m = 86$). The nodes represent the states, and the edges represent the strengths between the nodes. Notice the similar spatial topology between these two graphs

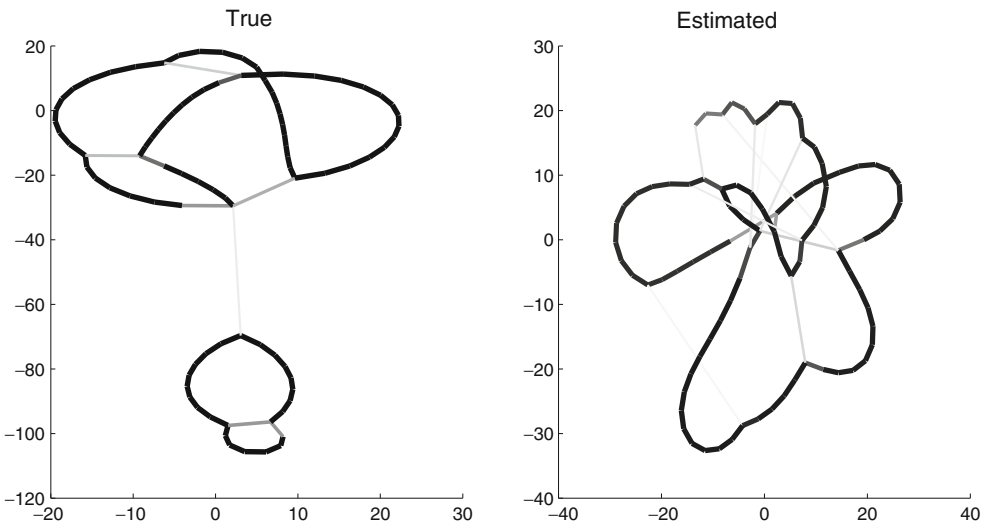


Fig. 13 The graphs inferred from the true transition matrix (*left*) and the estimated transition matrix with 0.01 thresholding (*right*) in Simulation 3-2 ($m = 86$)

Table 2 Summary of experimental data

Environment	Selected m	C	T	Remark
3.1-m linear track	$60 \sim 80$	30	1,138	~4.7-min run period extracted from 30-min recording
T-maze	$70 \sim 80$	39	952	~4-min run period extracted from 16-min recording

All data use a 250-ms bin size. The run-only data are obtained with a 0.15 m/s velocity filter

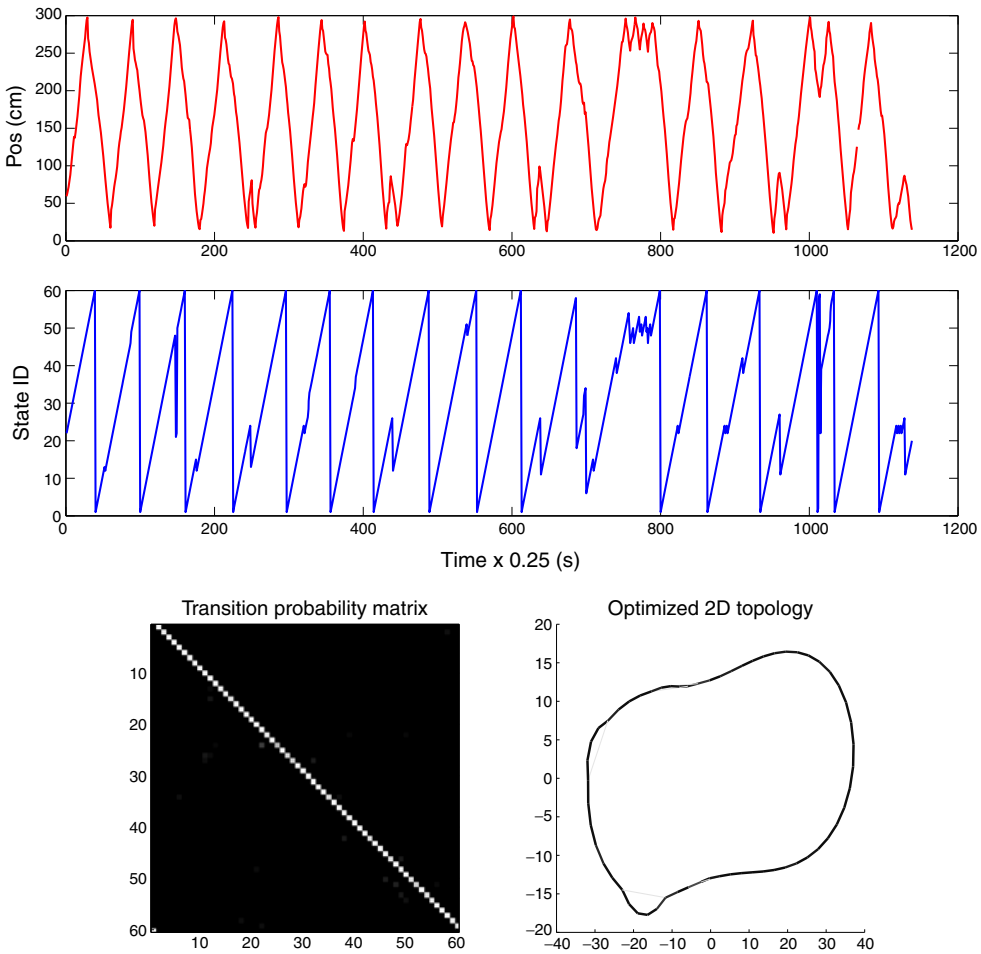


Fig. 14 One illustrated estimation result from the experimental linear track ($m = 60$): true position during the run period (*top panel*) and the decoded state sequence (*middle panel*). The estimated transition probability matrix (*bottom left*) and the inferred 2D graph (*bottom right*) are also shown. Note that there are three weak edges or shortcuts appended to the well-connected closed-loop

a break point (i.e., discontinuity), the end points of the inferred graph are not connected (left panel of Fig. 12). However, there are two shortcuts due to the existence of behavior turns in the simulation.

– In the combination of two environments (Simulation 3-2), the graph inferred from the true transition matrix (Fig. 13, left panel) consists of two separate yet weakly linked loops, each of them has its own shortcuts. The large loop on the top of the left panel (Fig. 13) represents the state space 1–62, whereas the small loop represents the state space 63–86, and these two loops are linked by a week edge (states 62 and 63).

All of these facts are observed from a "ground truth" graph inferred from a true state transition matrix. In statistical inference, the statistical estimation error in the state transition matrix will inevitably make the graph interpretation more difficult (e.g, Fig. 13, right panel). In practice, we found that thresholding small probabilities before using the force-based algorithm will improve the graph presentation (e.g., Fig. 11(c) vs. (d)).

5.2 Experimental data

Given the successful estimation results from the extensive computer simulations, we further apply our analysis to two experimental data sets. The statistics of experimental data are summarized in Table 2.

5.2.1 Linear track

In the first experimental protocol, the rat navigated in a 3.1-m linear track environment. To exclude the rat's pause or stop periods inside the track, we apply a velocity filter (15 cm/s) to obtain run-only periods of recording in one session. A total of 30 putative

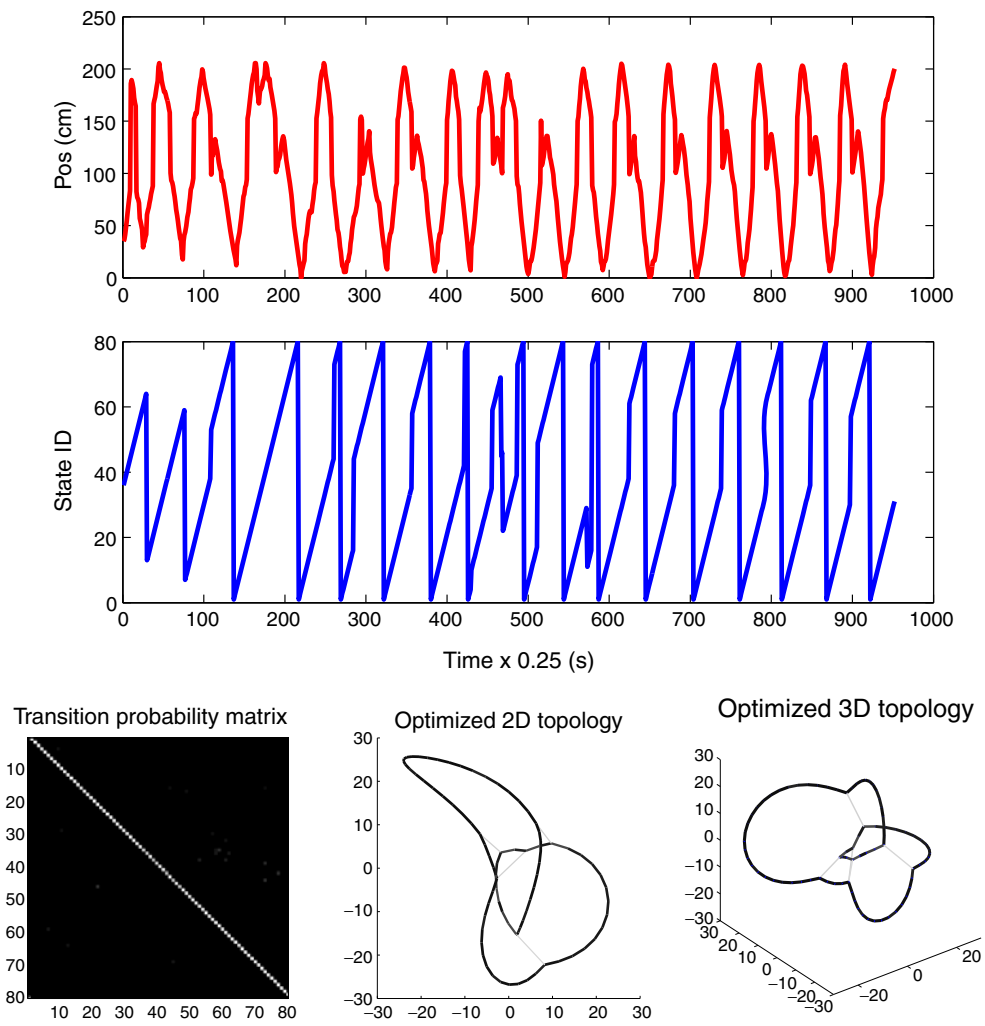


Fig. 15 One illustrated estimation result from the experimental T-maze ($m = 80$): true position during run period (*top panel*) and the decoded state sequence (*middle panel*). The estimated transition probability matrix (*bottom left*) and the inferred 2D (*bottom middle*) and 3D graphs (*bottom right*) are also shown. The 3D graph is simply another perspective to visualize the topological graph

pyramidal cells were simultaneously recorded from the rat hippocampal CA1 area based on multiple tectrodes. Spikes are sorted and binned with a 250-ms window, and the spike count statistics of the ensemble neurons are obtained.

We run the VB-HMM algorithm more than 50 times (each with independent random initialization) and select the result associated with the highest free energy. A range of model size of $m = 60 \sim 80$ has been tested. The estimated trajectory, the transition probability matrix, and the optimized 2D topology from the best result are illustrated in Fig. 14. The spatial topology is optimized with the force-based algorithm using the estimated transition matrix (upon 0.01 thresholding). In a closer examination of the graph, the basic topology appears to be a closed-loop circle, reflecting the nature of the back-and-forth navigation inside the linear track (Fig. 14, first panel). In addition, there are two or three weak edges within the closed-loop circle, implying there are shortcuts between the states. This is also consistent with the rat's behavior: the rat make turns at two specific locations: one is around 70 cm and other two are around 200 and 250 cm.

5.2.2 T-maze

In the second experimental environment, the rat navigated in a T-maze (as illustrated in the left panel of Fig. 2). After linearization, the environment is about 200 cm in length. A total of 39 putative pyramidal cells were simultaneously recorded from the rat hippocampal CA1 area based on multiple tectrodes. Spikes are sorted and binned with a 250-ms window, and the spike count statistics of the ensemble neurons are obtained. Again, a velocity filter is applied to extract about 4-min run periods from a total of 16-min recording.

Similarly, we run the algorithm more than 50 times (each with independent random initialization) and select the best result associated with the highest free energy. A range of model size of $m = 70 \sim 80$ has been tested. The estimated trajectory, estimated transition probability matrix, and the optimized 2D and 3D topologies from the best result are illustrated in Fig. 15. Specifically, the 3D graph is simply another perspective to visualize the topological graph (using the same force-based algorithm). As seen, the spatial topology inferred from the T-maze is more complex than that obtained from the linear track, the presence of twisted loop and weak edges make the inferred graph more difficult to interpret. This result is not too surprising, since in comparison to the simulated T-maze, the animal's behavior is more versatile and the real data length is about 4∼5 times shorter; which all make the inference task more challenging.

6 Discussion

6.1 Model selection and local maximum

For the finite m-state HMM, an important issue in statistical inference is to choose the model size m. Various model selection studies have been conducted in the HMM literature (Scott 2002; Cappé et al. 2005; Rydén 2008). In this paper, we focus on highlighting the methodology of VB inference and uncovering the spatial topology. For this reason, we have either selected the true model size (as in computer simulations) or empirically selected the model size (as in experimental data). Detailed comparison of results from using various model sizes is beyond the scope of current paper. Despite that, in order to illustrate the important issue of model selection, we use one example to illustrate how different model sizes will affect the estimation results. In the example of Simulation 1-2, the true model size is 62, we have also conducted inference using either a smaller ($m = 50$) or a larger ($m = 80$) model size. Two selected results are shown in Fig. 16. As seen, when the model size is insufficient, mistakes will be found in the inferred results (Fig. 16, left panels); when the model size is too large, redundant states are often found (Fig. 16, right panels).

In the terms of the optimized free energy function, a larger model size is typically accompanied with a greater free energy value. However, the local maximum problem would make direct comparison of different model sizes nontrivial. To illustrate this point, we run Monte Carlo experiments using different model sizes and compare their statistics (Fig. 17). Three important observations are noteworthy from Fig. 17: (i) When the right model size is selected, the greater free energy value is achieved upon convergence; in contrast, models with either too large or too small size will have lower free energy values. (ii) Compared to the small model size ($m = 50$), the large model size ($m = 80$) has a slightly more spread-out free energy distribution, whereas its best performance is slightly better. (iii) In the case of using a large model size, there often appear many redundant states—namely, not all state IDs are used in trajectory decoding. The actually used states are referred to the *effective* states. A commonly observed phenomenon when selecting a large model size is that most of 'good' performance is only obtained when the effective state size is around 62, but not always vice

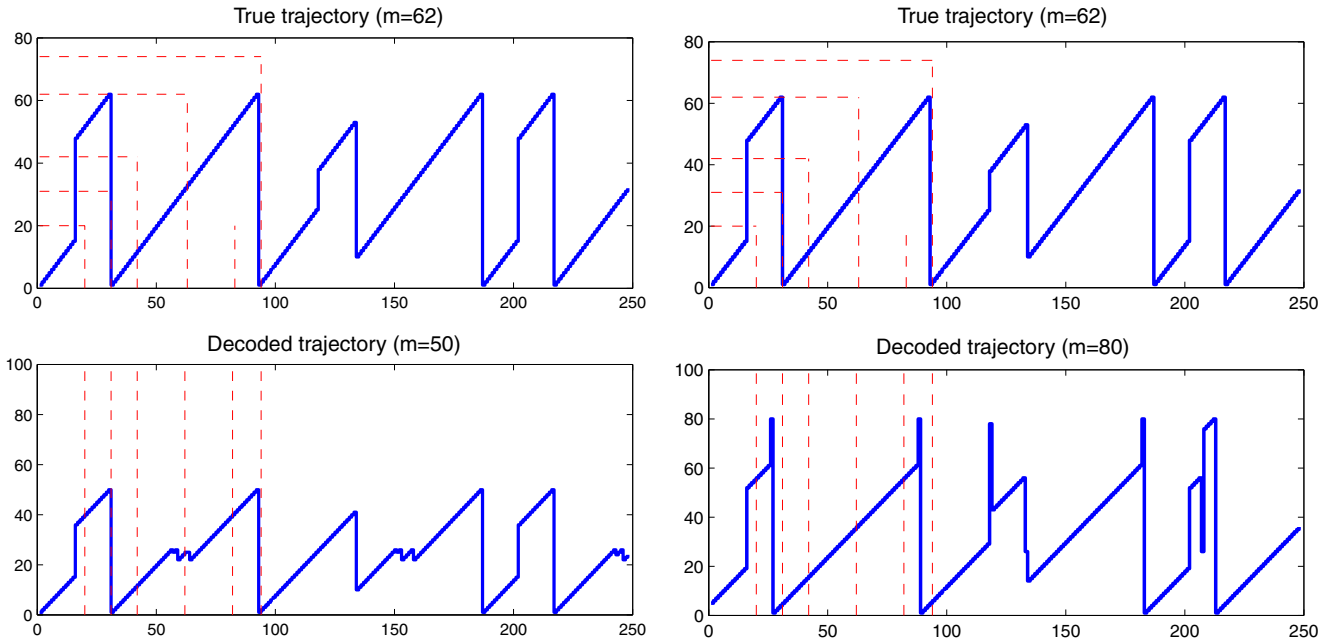


Fig. 16 Illustration of the estimated trajectory where the model size mismatch (Simulation 1-2). Underestimation (*left*): $m = 50$, $D_1 = 8$, $D_2(2) = 0.087$, $D_2(3) = 0.093$. Overestimation (*right*): $m = 80$, $D_1 = 4$, $D_2(2) = 0.017$, $D_2(3) = 0.182$

versa (namely, the good performance is not guaranteed when the effective state size is around 62).

In practice, one can use the free energy or the Bayesian deviance information criterion (DIC) as a guiding principle for model selection. Specifically, the DIC is defined as the sum of the expected deviance and the model complexity measure p_D (McGrory and Titterington 2009):

$$\begin{aligned} DIC &= \mathbb{E}_{p(\boldsymbol{\theta}|\mathbf{y})}\Big[-2\log p(\mathbf{y}|\boldsymbol{\theta})\Big] + p_D \\ &\approx -2\log p(\mathbf{y}|\tilde{\boldsymbol{\theta}}) - 2\int q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \log \frac{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} d\boldsymbol{\theta} \\ &\quad + 2\log \frac{q_{\boldsymbol{\theta}}(\tilde{\boldsymbol{\theta}})}{p(\tilde{\boldsymbol{\theta}})} \end{aligned} \tag{29}$$

where $\tilde{\boldsymbol{\theta}}$ denotes the posterior mean computed with respect to the variational posterior $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, and $p(\mathbf{y}|\tilde{\boldsymbol{\theta}})$ can be computed from the forward-backward algorithm (Appendix A).

We can also consider an alternative HMM. The infinite HMM is a nonparametric Bayesian extension of the HMM with an infinite number of hidden states (Beal et al. 2002). The key difference in hierarchical Bayesian modeling of the infinite HMM from the finite HMM is to treat the priors in the context of stochastic process. Recall that the prior used for the state transition matrix follows a Dirichlet distribution (van Gael et al. 2008):

$$\boldsymbol{P}_j \sim \texttt{Dir}(\alpha\boldsymbol{\beta}) = \frac{\Gamma(\sum_i \alpha\beta_i)}{\prod_i \Gamma(\alpha\beta_i)} \prod_{i=1}^{m} (P_{ij})^{\alpha\beta_i - 1}$$

$$\boldsymbol{\beta} \sim \texttt{Dir}(\gamma/m, \ldots, \gamma/m)$$

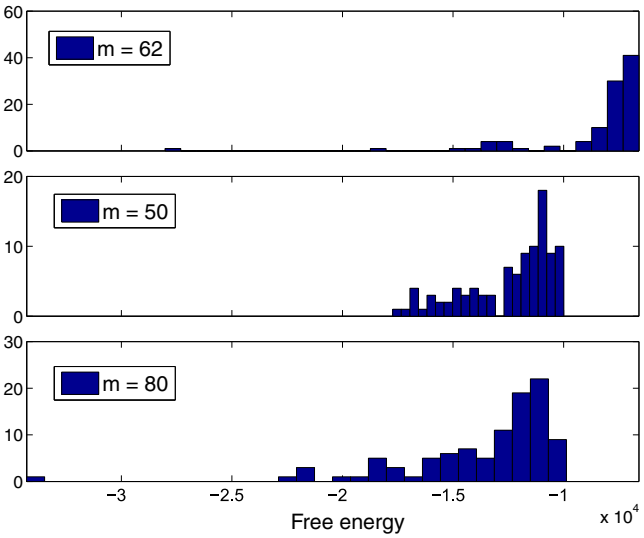


Fig. 17 Illustration of the distribution statistics of the converged free energy value based on independent random initializations. In each setup (from Simulation 1-2), 100 Monte Carlo experiments are conducted. As expected, when the selected model order matches the true model size ($m = 62$), a higher free energy value is typically achieved

where $\boldsymbol{P}_j$ denotes the j-th row of the transition matrix $\boldsymbol{P}$, and $\boldsymbol{\beta}$ are the shared prior parameters. The infinite-dimensional generalization of the Dirichlet distribution is a Dirichlet process. As $m \to \infty$, the hierarchical prior approaches a hierarchical Dirichlet process (HDP) (Teh et al. 2006). A HDP is a set of Dirichlet processes (DPs) coupled through a shared random based measure G_0 which is itself drawn from a DP. The concentration parameter $\alpha > 0$ governs the variability of the base measure, with small α implying greater variability. However, learning an infinite HMM would require a large amount of data samples, it may not be very practical in dealing with the experimental data in our current problem.

In learning latent probabilistic models, it is well known that the iterative EM and VB-EM algorithms are subject to the local maximum problem during optimization. The value of the local maximum highly depends on the initial conditions of the parameters or priors. Therefore, for every experimental data set, multiple runs of the iterative algorithm is a common practice. Meanwhile, the local maximum problem can be alleviated by using the so-called *deterministic annealing* (DA) procedure (see Appendix B for details). The key idea of the DA is to optimize a modified free energy function using an annealing parameter (in the analogy of the inverse temperature). As the inference process continues, the temperature is lowered and the annealing parameter is increased, it is expected (with higher probability) that the VB-EM algorithm can escape from the local maximum and approach a better solution. As a trade-off, the DA version of the algorithm is computationally slower and the result is also sensitive to the choice of the annealing parameter. Our observation of using DA in our current experiment is that the DA computation is very slow and it is wiser to spend the CPU resource in running the standard algorithm a few more times. Another possible solution is to use MCMC methods for *exact* Bayesian inference (in opposition to *approximate* Bayesian inference in VB). In this case, the VB-EM would be replaced by a Monte Carlo EM algorithm (McLachlan and Krishnan 2008). The inference principle remains similar: in the E-step, run the forward-backward algorithm, in the Monte Carlo M-step, run the Gibbs sampler for estimating the unknown parameters (using the same conjugate priors). Finally, the posteriors of the parameters would be represented by simulated Monte Carlo samples. However, as we have discussed earlier, the MCMC methods are more computationally expensive and a large memory space is required for storing samples for the parameter $\boldsymbol{\Lambda}$ of size m-by-C.

6.2 Extension with a dummy state

Thus far, we have only considered the spiking activity within the periods of active behavior. In principle, this can also be extended to periods of sleep or quiet wakefulness (although the temporal bin size needs to be adjusted). However, because of the distinct neural mechanisms of hippocampal circuitry between periods of behavior and periods of sleep or quiet awakefulness, it is important to analyze these periods separately. Currently, we use a velocity criterion to segment the run and stop periods in behaving animals. The stop epochs have been excluded in the analysis.

Alternatively, the stop epochs can be treated as an observed indicator variable and included in the analysis. In this case, we use a dummy or NULL state (without loss of generality, the augmented $(m+1)$-st state) to represent the situation in the presence of either non-RUN period or missing data (e.g., no recording is available between two independent episodes or between the change of experimental conditions). Since the $(m+1)$-st state is not hidden (i.e.,, being observable via the velocity filter), the inference of the HMM can be adapted to accommodate this scenario. Basically, the transition probability $P_{i,m+1}$ represents the conditional probability from state i to STOP, and the transition probability $P_{m+1,i}$ represents the conditional probability from STOP to state i, where the i-th state represents the i-th location in the virtual environment. The inference algorithm still remains similar, except for a slight modification of the forward-backward algorithm employed in the VB-E step.

For illustration, we apply the augmented HMM to the experimental data in the linear track. In the experimental linear track example, we include all non-RUN periods into our analysis, which consist of many RUN→STOP and STOP→RUN transitions. As expected, the estimated transition probability matrix has a shifted-diagonal substructure (as before) plus an additional column that reflects the RUN→STOP behavior (Fig. 18, left panel) . The non-sparse patterns in the last column reflects the stop behavior from various spatial locations. Applying the force-based algorithm to the shifted-diagonal substructure of the transition probability matrix (by excluding the last row and the last column) yields the spatial topology shown in the right panel of Fig. 18. Note that the loose ends of the graph are due to systematic stop behavior at the ends of the track. Therefore, the inferred graph reveals not only important cues about the spatial topology, but also important information about the animal's behavior.

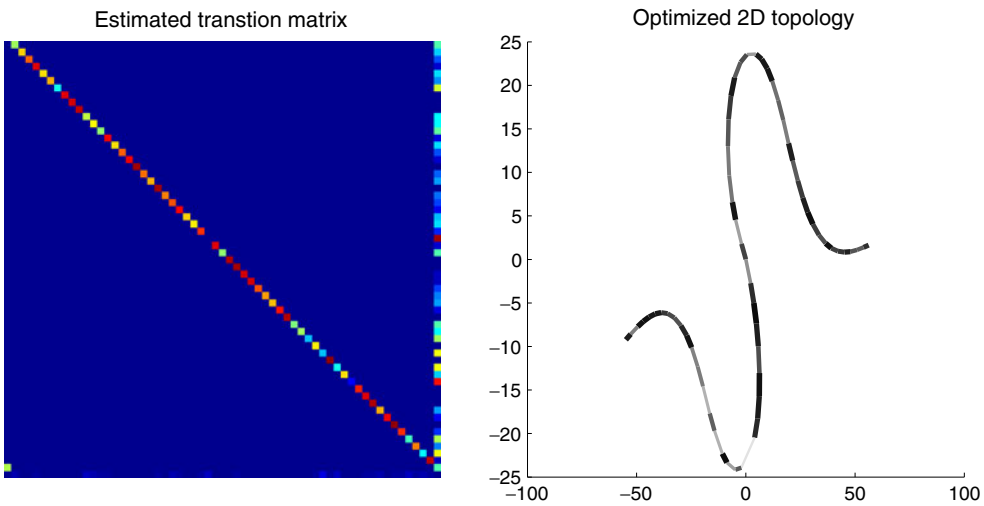


Fig. 18 Illustration of the estimated transition matrix in the presence of dummy state (*left*) and the inferred graph (*right*). The results are obtained from the experimental linear track data (using $m = 60$). Note that in the *left panel*, the non-sparse pattern of the 61st column implies frequent stop behavior from various spatial locations. Also note that in the *right panel*, the graph is drawn excluding the dummy state (i.e., based on the 60×60 submatrix of $\boldsymbol{P}$), which gives rise to a spatial topology without the closed loop

6.3 About the Markovian assumption

In this paper, we have assumed that the latent state process, which represents the rat's position combined with the directionality, follows a first-order Markovian process. This assumption is reasonable while using a relatively large temporal bin size (here, 250 ms). In reality, this assumption might not be completely valid. For example, there could be a high-order Markovian dependence in terms of motion, or there could be a non-Markovian or semi-Markovian behavior. Nevertheless, modeling these situations would require a large amount of data for fitting a more complex statistical model, which is beyond the scope of the current paper.

Also note that, based on the decoded state trajectory, one can estimate the high-order transition probability. For instance, the second-order transition probability, represented by a 3D tensor $\boldsymbol{P}^{(2)} = \{P^{(2)}_{ijk}\}$ (where $\sum_j \sum_k P^{(2)}_{ijk} = 1$), reveals information about a 3-bit state sequence $i \to j \to k$. Imaginably, at the bifurcation point (denoted by state j), we will see two (or more) dominant values $P^{(2)}_{ijk}$ and $P^{(2)}_{ijl}$ ($l \neq k$). These high-order statistics would be even more important when navigating in an open field environment.

6.4 Extension to non-Poissonian firing model

In Eq. (2), we have assumed that all neurons follow a pure Poisson spiking model. However, this assumption can be extended to other non-Poisson firing models, such as the Gamma distribution (which will be associated with a conjugate prior with four hyperparameters). Also, we may introduce individual neuronal firing history or ensemble neuronal firing activity as an observed covariate and characterize the neuronal firing within a generalized linear model (GLM) framework (Truccolo et al. 2005), the regression coefficients of the GLM can be estimated with a VB approach (Chen et al. 2011) in the VB-M step.

6.5 Identifiability

A model is said to be identifiable if it is theoretically possible to learn the true value of this model's underlying parameter after obtaining an infinite number of samples from it, which is also equivalent to saying that different values of the parameter would generate different probability distributions of the observable samples. In our statistical model, the unknown variables $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})$ are estimated by optimizing the free energy (Eq. (24)) assuming a factorial form of the posterior distribution (Eq. (10)). Due to non-convexity of the objective function, there might be many equivalent solutions in the joint space of $(\boldsymbol{P}, \boldsymbol{\Lambda})$ (permutation). This issue, in combination with the large dimensionality of $\boldsymbol{\theta}$ and small sample size, makes the task of statistical inference very challenging.

6.6 Assessment criterion

In computer simulations, in addition to visual inspection, we assess the quality of the estimation via two quantitative metrics: D_1 and D_2. However, visual inspection would become difficult when dealing with experimental data associated with complex behavior, or when selecting varying model sizes (since different m values would induce different state reconstruction results). Because of the state permutation ambiguity, it is important to check the consistency between two solutions.

From an information coding perspective, the HMM can be viewed as trying to represent or remap a continuous space $\mathcal{S}$ with a finite discrete alphabet $\mathcal{A}$ using a code book: $\mathcal{S} = f(\mathcal{A})$. The criterion for the consistency is to assure a *one-to-one* mapping between $\mathcal{S}$ and $\mathcal{A}$: (i) Any element in $\mathcal{S}$ is not simultaneously represented by A_i and A_j $(i \neq j)$; (ii) The same A_i does not represent two or more distinct regions in $\mathcal{S}$ (except for neighboring regions, since two neighboring regions can be combined into one by a merging operation). In addition, the binning strategy may be very flexible, A_i and A_j can encode two regions with different amounts of spatial coverage. Although it is easy to state the consistency principle, a practical quantitative evaluation of the estimated result is nontrivial, especially in the absence of ground truth for the experimental data. This issue requires further investigation.

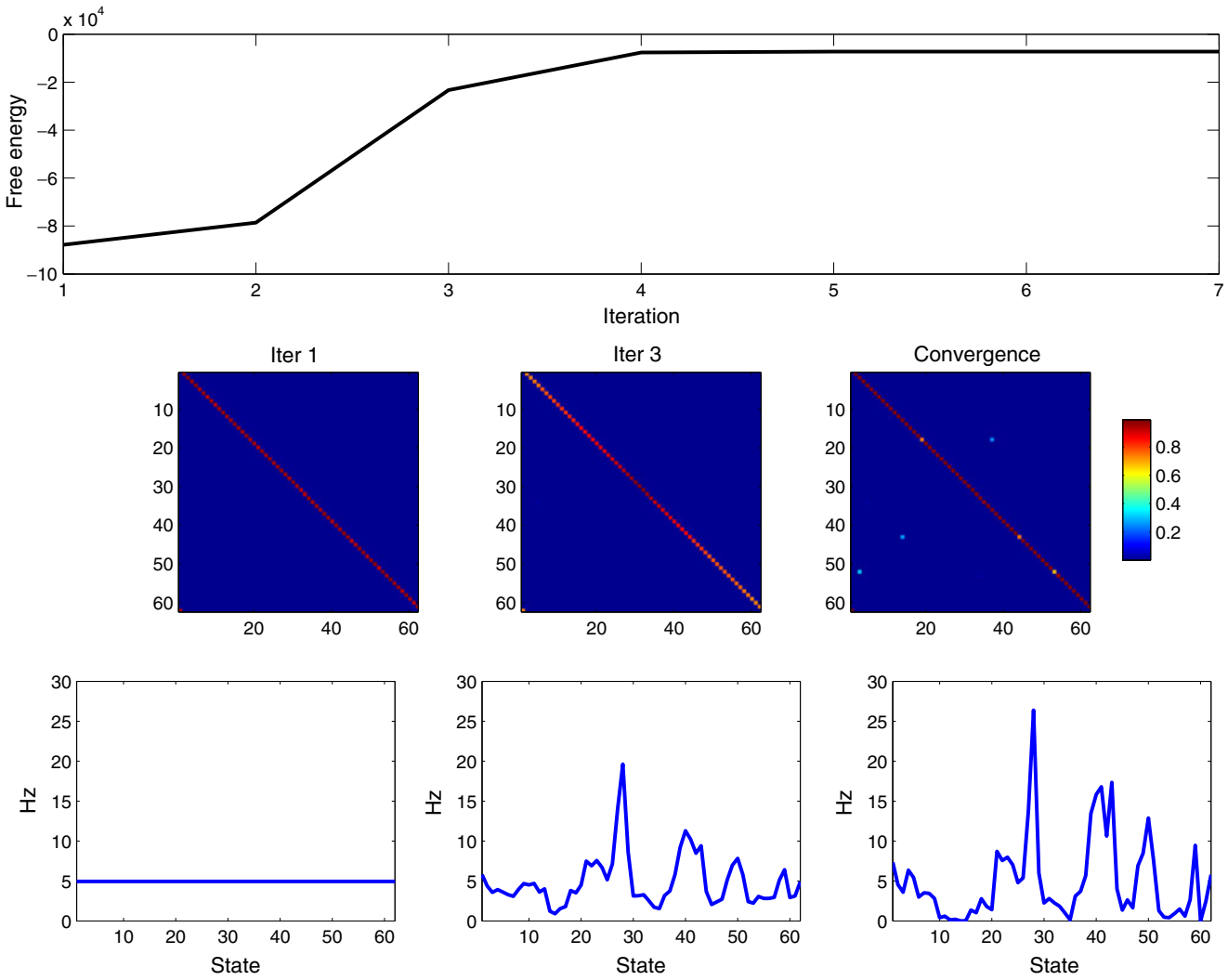


Fig. 19 Illustration of the algorithmic convergence and stability (Simulation 2-2). At different stages (1st and 3rd iterations, and final convergence), the free energy (*top row*), the estimated state transition matrix (*middle row*), and the tuning curve of one neuron (*bottom row*) are shown

6.7 Computational issue and computer software

Depending on the data size and the initial condition, the convergence speed of the VB-HMM algorithm is fast, typically less than 30 iterations. During the inference process, we can monitor the learning curves of the free energy as well as the estimated parameters, see Fig. 19 for a simulation illustration. The algorithm can handle a large number of neurons with large sample size (in Simulation 3-1, $T = 8{,}790$ corresponds to about 36 minutes with a 250 ms bin size). However, the number of states could become very large when considering a hypothetically complex spatial environment, exploiting the sparse structure of the state-transition matrix would be important in the presence of small sample size.

All software implementations are done in MATLAB©. Custom-written codes on the VB-HMM and the force-based algorithm used in this paper will be made available upon request.

7 Conclusion

In conclusion, we have used the rat hippocampus as a model system to uncover the "spatial topology" represented by the population codes. With the help of graph illustration, we develop a HMM and a VB inference algorithm to achieve this computational goal. Our empirical results from both extensive computer simulations and experimental data have shown a promising direction in uncovering the structural patterns of ensemble spike activity during the periods of active navigation. Since the spatial topology graph is just the proxy of the state-transition matrix, our proposed approach can also be extended to other model systems with the interest of characterizing the behavior-related transition probability (Jones et al. 2007; Kemere et al. 2008).

Our study provides important insights for future direction in exploratory data analysis of population neuronal codes. In addition to further investigation of some technical issues (e.g., selecting optimal temporal window and model size, model extension), we are planning to apply the same methodological analysis to the other rodent data recorded in more complex spatial environments (e.g., H-maze and open field), which will pose more challenges for interpreting the trajectories and graphs. The same exploratory analysis can also be applied to spiking data of ensemble neurons recorded during periods of sleep (Wilson and McNaughton 1994; Louie and Wilson 2001; Lee and Wilson 2002; Ji and Wilson 2007) or during "preplay analysis" without prior exposure of spatial environment (Dragoi and Tonegawa 2011). Other challenges can also arise due to the complex dynamics and multiple functional representations of the hippocampal place cells, as reported in the literature (Wood et al. 2000; Jackson and Redish 2007). Incorporating new neurophysiological findings in space representation within the rat hippocampal circuitry, such as the spiking activity from head-direction cells and entorhinal cortical cells (McNaughton et al. 2006), would further enrich the model and pave the way for a deeper understanding of hippocampal neural mechanisms.

Acknowledgements The work was supported in part by the NIH Grant DP1-OD003646 (E.N.B.) and MH061976 (M.A.W.). We thank Stuart Layton for providing the linear track experimental data used in the paper.

Appendix A: EM and Viterbi algorithms

The goal of ML inference is to maximize the log-likelihood function (Eq. (5)) based on missing data. In each full iteration, the EM algorithm (Dempster et al. 1977; McLachlan and Krishnan 2008) iteratively maximizes the so-called Q-function

$$\begin{aligned} Q(\boldsymbol{\theta}^{\text{new}}|\boldsymbol{\theta}^{\text{old}}) &= \mathbb{E}\Big[\log p(\hat{S}_{1:T}, \mathbf{y}_{1:T}|\boldsymbol{\theta})\Big|\boldsymbol{\theta}^{\text{old}}\Big] \\ &= \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{c=1}^{C}\sum_{i=1}^{m}\gamma_t(i)\Big(y_{c,t}\log\hat{\lambda}_{ic} - \hat{\lambda}_{ic}\Big) \\ &\quad + \sum_{i=1}^{m}\gamma_1(i)\log\hat{\pi}_i \\ &\quad + \sum_{t=2}^{T}\sum_{i=1}^{m}\sum_{j=1}^{m}\xi_t(i,j)\log\hat{P}_{ij}\Big|\boldsymbol{\theta}^{\text{old}}\Big], \end{aligned} \tag{30}$$

and the new $\boldsymbol{\theta}^{\text{new}}$ is obtained by maximizing the incomplete data likelihood conditional on the old parameters $\boldsymbol{\theta}^{\text{old}}$; and the iterative optimization procedure continues until the algorithm ultimately converges to a local maximum or a stationary point.

E-step: forward-backward algorithm In the E-step, the major task of the forward-backward procedure is to compute the state conditional marginal probabilities:

$$\begin{aligned} \Pr(S_t = i|\mathbf{y}_{1:T}, \boldsymbol{\theta}) &= \frac{\Pr(\mathbf{y}_{1:T}, S_t = i|\boldsymbol{\theta})}{\Pr(\mathbf{y}_{1:T}|\boldsymbol{\theta})} \\ &= \frac{\Pr(\mathbf{y}_{1:T}, S_t = i|\boldsymbol{\theta})}{\sum_{l=1}^{m}\Pr(\mathbf{y}_{1:T}, S_t = l|\boldsymbol{\theta})} \end{aligned} \tag{31}$$

as well as the state conditional joint probabilities:

$$\begin{aligned}&\Pr(S_{t-1}=i, S_t=j|\mathbf{y}_{1:T}, \boldsymbol{\theta})\\ &= \frac{\Pr(\mathbf{y}_{1:T}, S_{t-1}=i, S_t=j|\boldsymbol{\theta})}{\Pr(\mathbf{y}_{1:T}|\boldsymbol{\theta})}\\ &= \frac{\Pr(\mathbf{y}_{1:T}, S_{t-1}=i, S_t=j|\boldsymbol{\theta})}{\sum_{l=1}^{m}\sum_{n=1}^{m}\Pr(\mathbf{y}_{1:T}, S_{t-1}=l, S_t=n|\boldsymbol{\theta})}.\end{aligned} \tag{32}$$

To make the notation simple, in the derivation below we will let the conditional $\boldsymbol{\theta}$ be implicit in the equation.

To estimate Eqs. (7) and (8), we first factorize the joint probability as

$$\begin{aligned}\Pr(\mathbf{y}_{1:T}, S_t=l) &= \Pr(\mathbf{y}_{1:t}, S_k=l)\Pr(\mathbf{y}_{t+1:T}|\mathbf{y}_{1:t}, S_t=l)\\ &= \Pr(\mathbf{y}_{1:t}, S_t=l)\Pr(\mathbf{y}_{t+1:T}|S_t=l)\\ &\equiv a_t(l)b_t(l) \quad \text{for } l=1,\ldots,m\end{aligned} \tag{33}$$

where

$$\begin{aligned}a_1(l) &= \pi_l \Pr(\mathbf{y}_1|S_1=l)\\ a_t(l) &= \Pr(\mathbf{y}_{1:t}, S_t=l) \quad \text{for } t=2,\ldots,T\\ b_t(l) &= \Pr(\mathbf{y}_{t+1:T}|S_t=l) \quad \text{for } t=1,\ldots,T-1\\ b_T(l) &= 1\end{aligned}$$

and the forward and backward messages $a_t(l)$ and $b_t(l)$ can be computed recursively along the time index t (Rabiner 1989)

$$\begin{aligned}a_t(l) &= \sum_i a_{t-1}(i)P_{il}\Pr(\mathbf{y}_t|S_t=l)\\ b_t(l) &= \sum_i b_{t+1}(i)P_{li}\Pr(\mathbf{y}_{t+1}|S_{t+1}=i),\end{aligned}$$

where P_{il} denotes the transition probability from state i to l.

Given $\{a_t(\cdot), b_t(\cdot)\}$, the state posterior conditional joint probability (Eq. (8)) is determined by

$$\begin{aligned}&\Pr(S_{t-1}=i, S_t=j|\mathbf{y}_{1:T})\\ &\quad\propto a_t(i)P_{ij}\Pr(\mathbf{y}_{t+1}|S_{t+1}=j)b_{t+1}(j).\end{aligned} \tag{34}$$

In light of Eq. (9), the observed (incomplete) data likelihood is computed as

$$\begin{aligned}p(\mathbf{y}_{1:T}) &= \sum_{l=1}^{m}\Pr(\mathbf{y}_{1:T}, S_t=l)\\ &= \sum_{l=1}^{m}a_t(l)b_t(l) = \sum_{l=1}^{m}a_T(l).\end{aligned} \tag{35}$$

Alternatively, the incomplete data likelihood is given by

$$\begin{aligned}p(\mathbf{y}_{1:T}) &= \prod_{t=1}^{T}p(\mathbf{y}_t|\mathbf{y}_{t-1})\\ &= \prod_{t=1}^{T}\zeta_t(\mathbf{y}_t) = Z(\mathbf{y}_{1:T})\end{aligned} \tag{36}$$

where $\zeta_t(\mathbf{y}_t) \equiv p(\mathbf{y}_t|\mathbf{y}_{t-1})$ is a normalization constant; $Z(\mathbf{y}_{1:T})$ is also called the marginal likelihood.

From Eqs. (9) and (11), the state posterior conditional marginal probability (Eq. (7)) is determined by the Bayes' rule

$$\begin{aligned}\Pr(S_t=i|\mathbf{y}_{1:T}) &= \frac{\Pr(\mathbf{y}_{1:T}, S_t=i)}{p(\mathbf{y}_{1:T})}\\ &= \frac{a_t(i)b_t(i)}{\sum_{l=1}^{m}a_t(l)b_t(l)} \propto a_t(i)b_t(i).\end{aligned} \tag{37}$$

Equations (10) and (13) are the sufficient statistics computed from the E-step (to be used in the M-step).

In the term of the computational overhead for the m-state HMM, the above-described forward-backward procedure requires an order of computational complexity $\mathcal{O}(m^2T)$ and memory storage $\mathcal{O}(mT)$.

M-step: re-estimation In the M-step, we update the unknown parameters by setting the partial derivatives of the Q-function to zeros: $\frac{\partial Q(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$, from which we may derive either closed-form or iterative solutions.

Let $\xi_t(i,j) = \Pr(S_{t-1}=i, S_t=j|\mathbf{y}_{1:T}, \boldsymbol{\theta})$ and $\gamma_t(i) = \Pr(S_t=i|\mathbf{y}_{1:T}, \boldsymbol{\theta})$ denote, respectively, the state posterior conditional marginal and joint probabilities (which are the sufficient statistics for the complete data log-likelihood). From the E-step, we may obtain

$$\begin{aligned}\gamma_t(i) &= \frac{a_t(i)b_t(i)}{\sum_{l=1}^{m}a_t(l)b_t(l)}\\ &= \sum_j \xi_t(j,i) = \sum_j \xi_{t+1}(i,j).\end{aligned} \tag{38}$$

The transition probability estimates are given by the Baum's re-estimation procedure

$$\begin{aligned}\hat{P}_{ij} &= \frac{\sum_{t=2}^{T}\xi_t(i,j)}{\sum_{t=2}^{T}\sum_{j=1}^{m}\xi_t(i,j)}\\ &= \frac{\sum_{t=2}^{T}\xi_t(i,j)}{\sum_{t=2}^{T}\gamma_t(i)}.\end{aligned} \tag{39}$$

And the rate parameter estimates $\boldsymbol{\Lambda} = \{\lambda_{ic}\}$ are given by solving $\frac{\partial Q}{\partial \lambda_{ic}} = 0$ from Eq. (6)[6]

$$\hat{\lambda}_{ic} = \frac{\sum_{t=1}^{T} y_{c,t}\gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{40}$$

Finally, the convergence of the EM algorithm is monitored by the incremental changes of the log-likelihood as well as the parameters. If the quantity of the absolute change or relative change of the log-likelihood is smaller than a desirable value, the EM algorithm is terminated.

Viterbi algorithm Upon estimating parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda})$, we can run the Viterbi algorithm (Viterbi 1967) for decoding the *most likely* state sequences. The Viterbi algorithm is a dynamical programming method (Bellman 1957) that uses the "Viterbi path" to discover the single most likely explanation for the observations. Specifically, the MAP estimate $\hat{S}_t$ at time t is

$$\hat{S}_t^{\text{MAP}} = \arg \max_{i \in \{1,\ldots,m\}} \gamma_t(i) \quad 1 \le t \le T. \tag{41}$$

The computational overhead of the forward Viterbi algorithm has an overall time complexity $\mathcal{O}(m^2 T)$ and space complexity $\mathcal{O}(mT)$.

Appendix B: Deterministic annealing

For the discrete m-state HMM, there are exponential numbers (i.e., $\mathcal{O}(2^m)$) of local maxima. The local maximum problem is particularly severe when the transition matrix $\boldsymbol{P}$ is sparse (many zero elements) or there are equal state emission probabilities for distinct states. This phenomenon is known as the "singularity" of the objective function (Amari et al. 2003; Watanabe 2009), which is omnipresent in many estimation problems of probabilistic models and artificial neural network models. In order to alleviate the local maximum problem, the so-called *deterministic annealing* (DA) procedure was proposed for several latent probabilistic models, such as the mixture models and HMM (Beal 2003; Katahira et al. 2008).

[6]To avoid numerical problem we set $\hat{\lambda}_{ic} = 0$ if the denominator is 0 or nearly 0.

The key idea of DA-VB is to modify the original free energy function (Eq. (6)) by introducing an annealing parameter ρ:

$$\begin{aligned}\mathcal{F}(q) &= \Big\langle \log p(\mathbf{y}_{1:T}, S_{1:T}, \boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) \Big\rangle_q + \frac{1}{\rho}\mathcal{H}_q(\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}, S_{1:T}) \\ &= \Big\langle \log p(\mathbf{y}_{1:T}, S_{1:T}, \boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{\Lambda}) \Big\rangle_{q(\boldsymbol{\theta})q(S_{1:T})} \\ &\quad + \frac{1}{\rho}\mathcal{H}_{q(S_{1:T})}(S_{1:T}) + \frac{1}{\rho}\mathcal{H}_{q(\boldsymbol{\theta})}(\boldsymbol{\theta})\end{aligned} \tag{42}$$

where $\rho = 1/\mathcal{T}$ can be viewed as an inverse temperature parameter. The annealing procedure gradually lowers the temperature (or increases ρ) during the inference process, hoping to escape from local maxima and ultimately to reach the global maximum with a higher probability.

Consequently, the new state posterior probabilities will be recomputed from the VB-E step:

$$\tilde{\gamma}_t(i) = \frac{\gamma_t(i)^\rho}{\sum_{j=1}^{m} \gamma_t(j)^\rho} \tag{43}$$

$$\tilde{\xi}_t(i, j) = \frac{\xi_t(i, j)^\rho}{\sum_{l=1}^{m}\sum_{n=1}^{m} \xi_t(l, n)^\rho} \tag{44}$$

whereas in the VB-M step, we have the following new update equations (used for Eqs. (12), (13) and (15)):

$$w_i^{(\pi)} = \rho\Big(u_i^{(\pi)} + \tilde{\gamma}_1(i) - 1\Big) + 1 \tag{45}$$

$$w_{ij}^{(P)} = \rho\Big(u_{ij}^{(P)} + \sum_{t=2}^{T} \tilde{\xi}_t(i, j) - 1\Big) + 1 \tag{46}$$

$$q(\lambda_{ic}) = \mathtt{Gam}\Big(C\alpha_i^{(\lambda)} + \sum_{t=1}^{T} y_{c,t}\tilde{\gamma}_t(i), C\beta_i^{(\lambda)} + \sum_{t=1}^{T} \tilde{\gamma}_t(i)\Big) \tag{47}$$

Note that when the annealing parameter $\rho = 1$, the standard VB-EM algorithm (Sections 3.1 and 3.2) is recovered.

Appendix C: Optimizing hyperparameters

In light of Eq. (23), taking the derivatives of the logarithm of Eq. (23) with respect to $\alpha_i^{(\lambda)}$ and $\beta_i^{(\lambda)}$ and setting them to zeros yields

$$0 = \sum_{c=1}^{C} -\frac{C + \sum_t y_{c,t}\gamma_t(i)}{C\beta_i^{(\lambda)} + l_i} + C y_{c,t}\psi'\Big(C\alpha_i^{(\lambda)} + \sum_t y_{c,t}\gamma_t(i)\Big) \tag{48}$$

$$0 = \sum_{c=1}^{C} \frac{C\alpha_i^{(\lambda)} + \sum_t y_{c,t}\gamma_t(i)}{(C\beta_i^{(\lambda)} + l_i)^2} - \frac{y_{c,t}}{C\beta_i^{(\lambda)} + l_i} \tag{49}$$

There is no closed-form solution to these two equations. However, solving these two fixed-point equations using a gradient or Newton-type algorithm within each VB-M step would increase the marginal log-likelihood or free energy.

References

Amari, S., Ozeki, T., & Park, H-Y. (2003). Learning and inference in hierarchical models with singularities. *Systems and Computers in Japan, 34*(7), 34–42.

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics, 41*(1), 164–171.

Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. PhD. Thesis, Gatsby Computational Neuroscience Unit, University College London.

Beal, M. J., Ghahramani, Z., & Rasmussen, C. E. (2002). *The infinite hidden Markov model. Advances in neural information processing systems* (Vol. 14). Cambridge, MA: MIT Press.

Bellman, R. (1957). *Dynamic programming*. Boston: Princeton University Press.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Borg, I., & Groenen, P. (2005). *Modern multidimensional scaling: Theory and applications* (2nd ed.). New York: Springer.

Brand, M. (1999). Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation, 11*, 1155–1182.

Brand, M., & Ketnaker, V. (2000). Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 844–851.

Brown, E. N., Frank, L. M., Tang, D., Quirk, M. C., & Wilson, M. A. (1998). A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *Journal of Neuroscience, 18*, 7411–7425.

Buzsaki, G. (2006). *Rhythms of the brain*. London, UK: Oxford University Press.

Cappé, O., Moulines, E., & Ryden, T. (2005). *Inference in hidden Markov models*. New York: Springer.

Chen, Z., Vijayan, S., Barbieri, R., Wilson, M. A., & Brown, E. N. (2009). Discrete- and continuous-time probabilistic models and algorithms for inferring neuronal UP and DOWN states. *Neural Computation, 21*(7), 1797–1862.

Chen, Z., Putrino, D., Ghosh, S., Barbieri, R. & Brown, E. N. (2011). Statistical inference for assessing neuronal interactions and functional connectivity with sparse spiking data. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 19*(2), 121–135.

Cox, T. F., & Cox, M. A. A. (2001). *Multidimensional scaling*. London, UK: Chapman and Hall.

Curto, C., & Itskov, V. (2008). Cell groups reveal structure of stimulus space. *PLoS Computational Biology, 4*, e1000205.

Dabaghian, Y., Cohn, A. G., & Frank, L. (2008). Topological coding in hippocampus. Online paper. arXiva:q-bio/0702052v1.

Darmanjian, S., & Principe, J. C. (2009). Spatial-temporal clustering of neural data using linked-mixtures of hidden Markov models. *EURASIP Journal on Advances in Signal Processing*, 2009, Article ID 892461.

Davidson, T. J., Kloosterman, F., & Wilson, M. A. (2009). Hippocampal replay of extended experience. *Neuron, 63*, 497–507.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39*(1), 1–38.

Diba, K., & Buzsaki, G. (2007). Forward and reverse hippocampal place-cell sequences during ripples. *Nature Neuroscience, 10*, 1241–1242.

Dragoi, G., & Tonegawa, S. (2011). Preplay of future place cell sequences by hippocampal cellular assemblies. *Nature, 469*, 397–401.

Frank, L. M., Stanley, G. B., & Brown, E. N. (2004) Hippocampal plasticity across multiple days of exposure to novel environments. *Journal of Neuroscience, 24*, 7681–7689.

Foster, D. J., & Wilson, M. A. (2006). Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature, 440*, 680–683.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian data analysis* (2nd ed.). London, UK: Chapman & Hall/CRC.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (Eds.) (1995). *Markov chain Monte Carlo in practice*. London, UK: Chapman & Hall/CRC.

Harris, K., Csicsvari, J., Hirase, H., Dragoi, G., & Buzsaki, G. (2003). Organization of cell assemblies in the hippocampus. *Nature, 424*, 552–556.

Herbst, J. A., Gammeter, S., Ferrero, D., & Hahnloser, R. H. R. (2008). Spike sorting with hidden Markov models. *Journal of Neuroscience Methods, 174*, 126–134.

Jackson, J., & Redish, A. D. (2007). Network dynamics of hippocampal cell-assembliesresemble multiple spatial maps within single tasks. *Hippocampus, 17*, 1209–1229.

Ji, D., & Wilson, M. A. (2007). Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature Neuroscience, 10*, 100–107.

Ji, S., Krishnapuram, B., & Carin, L. (2006). Variational Bayes for continuous hidden Markov models and its application to active learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(4), 522–532.

Jones, L. M., Fontanini, A., Sadacca, B. F., Miller, P., & Katz, D. B. (2007). Natural stimuli evoke dynamic sequences of states in sensory cortical ensembles. *Proceedings of the National Academy of Sciences, USA, 104*, 18772–18777.

Karlsson, M. P., & Frank, L. M. (2009). Awake replay of remote experiences in the hippocampus. *Nature Neuroscience, 12*, 913–918.

Katahira, K., Nishikawa, J., Okanoya, K., & Okada, M. (2010). Extracting state transition dynamics from multiple spike trains using hidden Markov models with correlated Poisson distribution. *Neural Computation, 22*, 2369–2389.

Katahira, K., Watanabe, K., & Okada, M. (2008). Deterministic annealing variant of variational Bayes method. *Journal of Physics: Conference Series, 95*, 012015.

Kemere, C., Santhanam, G., Yu, B. M., Afshar, A., Ryu, S. I., Meng, T. H., et al. (2008). Detecting neural-state transitions using hidden Markov models for motor cortical prostheses. *Journal of Neurophysiology, 100*(4), 2441–2452.

Lee, A. K., & Wilson, M. A. (2002). Memory of sequential experience in the hippocampus during slow wave sleep. *Neuron, 36*, 1183–1194.

Lever, C., Wills, T., Cacucci, F., Burgess, N., & O'Keefe, J. (2002). Long-term plasticity in hippocampal place-cell representation of environmental geometry. *Nature, 416*, 90–94.

Louie, K., & Wilson, M. A. (2001). Temporally structured REM sleep replay of awake hippocampal ensemble activity. *Neuron, 29*, 145–156.

Mackay, D. J. C. (1997). Ensemble learning for hidden Markov models. Technical Report, Cavendish Laboratory, Cambridge University, UK.

MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge, UK: Cambridge University Press.

McGrory, C. A., & Titterington, D. M. (2009). Variational Bayesian analysis for hidden Markov models. *Australian & New Zealand Journal of Statistics, 51*(2), 227–244.

McLachlan, G. J., & Krishnan, T. (2008). *The EM algorithm and extensions* (2nd ed.). New York: Wiley.

McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I., & Moser, M. B. (2006). Path integration and the neural basis of the 'cognitive map'. *Nature Reviews. Neuroscience, 7*, 663–678.

O'Keefe, J., & Nadel, N. (1978). *The hippocampus as a cognitive map*. New York: Oxford University Press.

Pawitan, Y. (2001). *In all likelihood: Statistical modelling and inference using likelihood*. New York: Oxford University Press.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257–286.

Robert, C. P. (2001). *The Bayesian choice–A decision-theoretic motivation* (2nd ed.). New York: Springer.

Rydén, T. (2008). EM versus Markov chain Monte Carlo for estimation of hidden Markov models: A computational perspective. *Bayesian Analysis, 3*(4), 659–688.

Scott, S. L. (2002). Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association, 97*, 337–351.

Skaggs, W. E., & McNaughton, B. L. (1996). Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science, 271*, 1870–1873.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association, 101*, 1566–1581.

Tollis, I. G., di Battista, G., Eades, P., & Tamassia, R. (1999). *Graph drawing: Algorithms for the visualization of graphs*. Englewood Cliffs, NJ: Prentice Hall.

Truccolo, W., Eden, U. T., Fellow, M., Donoghue, J. P., & Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble and covariate effects. *Journal of Neurophysiology, 93*, 1074–1089.

van Gael, J., Saatci, Y., Teh, Y. W., & Ghahramani, Z. (2008). Beam sampling for the infinite hidden Markov model. In *Proc. 25th int. conf. machine learning*, Helsinki, Finland.

Viterbi, J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory, 13*, 260–269.

Watanabe, S. (2009). *Algebraic geometry and statistical learning theory*. Cambridge, MA: Cambridge University Press.

Wills, T., Lever, C., Cacucci, F., Burgess, N., & O'Keefe, J. (2005). Attractor dynamics in the hippocampal representation of the local environment. *Science, 308*, 873–876.

Wilson, M. A., & McNaughton, B. L. (1993). Dynamics of the hippocampal ensemble code for space. *Science, 261*, 1055–1058.

Wilson, M. A., & McNaughton, B. L. (1994). Reactivation of hippocampal ensemble memories during sleep. *Science, 265*, 676–679.

Wood, E. R., Dudchenko, P. A., Robitsek, R. J., & Eichenbaum, H. (2000). Hippocampal neurons encode information about different types of memory episodes occurring in the same location. *Neuron, 27*, 623–633.

Wu, W., Chen, Z., Gao, S., & Brown, E. N. (2011). A hierarchical Bayesian approach for learning spatio-temporal decomposition of multichannel EEG. *NeuroImage, 56*(4), 1929–1945.

Zemel, R. S., Dayan, P., & Pouget, A. (1998). Probabilistic interpretation of population codes. *Neural Computation, 10*, 403–430.

Zhang, K., Ginzburg, I., McNaughton, B. L., & Sejnowski, T. J. (1998). Interpreting neuronal population activity by reconstruction: Unified framework with application to hippocampal place cells. *Journal of Neurophysiology, 79*, 1017–1044.

Statistical Inference for Assessing Functional Connectivity of Neuronal Ensembles With Sparse Spiking Data

Zhe Chen, *Senior Member, IEEE*, David F. Putrino, Soumya Ghosh, Riccardo Barbieri, *Senior Member, IEEE*, and Emery N. Brown, *Fellow, IEEE*

***Abstract*—The ability to accurately infer functional connectivity between ensemble neurons using experimentally acquired spike train data is currently an important research objective in computational neuroscience. Point process generalized linear models and maximum likelihood estimation have been proposed as effective methods for the identification of spiking dependency between neurons. However, unfavorable experimental conditions occasionally results in insufficient data collection due to factors such as low neuronal firing rates or brief recording periods, and in these cases, the standard maximum likelihood estimate becomes unreliable. The present studies compares the performance of different statistical inference procedures when applied to the estimation of functional connectivity in neuronal assemblies with sparse spiking data. Four inference methods were compared: maximum likelihood estimation, penalized maximum likelihood estimation, using either ℓ_2 or ℓ_1 regularization, and hierarchical Bayesian estimation based on a variational Bayes algorithm. Algorithmic performances were compared using well-established goodness-of-fit measures in benchmark simulation studies, and the hierarchical Bayesian approach performed favorably when compared with the other algorithms, and this approach was then successfully applied to real spiking data recorded from the cat motor cortex. The identification of spiking dependencies in physiologically acquired data was encouraging, since their sparse nature would have previously precluded them from successful analysis using traditional methods.**

Manuscript received March 10, 2010; revised June 01, 2010, August 13, 2010; accepted October 02, 2010. Date of publication October 11, 2010; date of current version April 08, 2011. This work was supported by the National Institutes of Health (NIH) under Grant DP1-OD003646, Grant R01-DA015644, and Grant R01-HL084502.

Z. Chen is with the Neuroscience Statistics Research Laboratory, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA, and also with the Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: zhechen@neurostat.mit.edu).

D. F. Putrino is with the Neuroscience Statistics Research Laboratory, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA, and also with the Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: trinod01@neurostat.mit.edu).

S. Ghosh is with the Centre for Neuromuscular and Neurological Disorders, University of Western Australia, QEII Medical Centre, Nedlands, Western Australia 6009, Australia (e-mail: sghosh@cyllene.uwa.edu.au).

R. Barbieri is with the Neuroscience Statistics Research Laboratory, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA, and also with Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: barbieri@neurostat.mit.edu).

E. N. Brown is with the Neuroscience Statistics Research Laboratory, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114 USA, and also with the Harvard-MIT Division of Health Science and Technology, and the Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: enb@neurostat.mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNSRE.2010.2086079

***Index Terms*—Conjugate gradient, interior-point method, functional connectivity, maximum likelihood estimate (MLE), neuronal interactions, penalized maximum likelihood, point process generalized linear model, variational Bayes, ℓ_2 regularization, ℓ_1 regularization.**

I. Introduction

IDENTIFYING the functional connectivity of a neuronal system using simultaneously recorded neural spike trains has provided valuable implications for understanding the system from a statistical perspective [6], [25]. Statistical modeling of neuronal data has been used for establishing statistical associations or causality between neurons, finding spatiotemporal correlations, or studying the functional connectivity in neuronal networks [10], [3], [50], [32], [31], [41], [14]. This analysis has many functional applications such as neural decoding, and assisting attempts to understand the collective dynamics of coordinated spiking cortical networks [49]. A statistical tool for analyzing multiple spike trains is the theory of random point processes. Statistical inference for point process observations often starts with a certain class of statistical (parametric or nonparametric) model, followed by parameter estimation using a statistical (either maximum likelihood or Bayesian) inference procedure [42], [5], [47]. To date, a number of statistical tools and models have been used to identify functional connectivity between ensemble neurons. The cross-correlogram and joint peri-stimulus time histogram (JPSTH) are standard (and possibly simplest) nonparametric methods for analyzing the interactions between pairwise neurons [35], [20], [1]. However, these tools have serious drawbacks: correlation-based analysis is limited to second-order spike count statistics, which is inadequate for neuronal spike trains. Further, these methods are nonparametric and there is no model validation or goodness-of-fit tests for the data. Recently, point process generalized linear models (GLMs) have been widely used for characterizing functional (spiking) dependence among ensemble neurons [10], [32], [47]. Specifically, the spiking probability of a particular neuron may be modeled as a function of the spiking history of concurrently recorded ensemble neurons (and possibly, a function of the input of the other stimuli as well), and the corresponding parameters of the point process GLM are inferred by maximum likelihood estimation. Bayesian inference has also been recently proposed for GLM inference on neural spike train data [36], [39], [48], [21], [42], [9]. Various approximation procedures have been developed, based on either Laplace approximation, expectation

propagation (EP), Markov chain Monte Carlo (MCMC) sampling, or variational approximation.

To date, spike train modeling based upon the point process GLM has been used as a statistical tool in an effective manner to infer functional connections between groups of simultaneously recorded neurons [32]. However, physiological experiments that aim to record neural activity from awake, behaving animals are technically difficult to perform, and at times, the quantities of data that are collected during recording periods can be less than ideal. In addition, neurons that are recorded from animals or human subjects in anesthetized state often fire at very low spiking rates. Most traditional methods that analyze pairwise relationships between neurons simply cannot be used to reliably infer interactions when neural firing rates are low, or the number of trials of a behavioral task that is being studied are low. This presents a rather important problem in the field of neuroscience, as many sets of carefully acquired experimental data are then considered unusable for analysis (i.e., because insufficient trials of a difficult behavioral task were acquired, or recorded neurons are firing at an extremely low rate). Theoretically, the more recently developed model-based methods have the ability to reliably perform this analysis, but their effectiveness at being applied to this problem has not been examined. The present paper evaluates the ability of two different approaches to address this problem based on the point process GLM framework: the first one is penalized maximum likelihood estimation that uses ℓ_2 or ℓ_1 regularization, which aims to improve the generalization of the model while reducing the variance of the estimate; the second is hierarchical Bayesian estimation, which uses an efficient variational approximation technique that allows deterministic inference (without resorting to random MCMC sampling). The statistical algorithms under investigation are all capable of handling large-scale problems, but the present paper focuses on relatively small-scale data sets. The current paper focuses on the investigation of different inference algorithms rather than on different modeling paradigms for characterizing functional connectivity. It is worth noting that, in addition to the point process GLMs, other statistical models such as the maximum entropy model [38], [40] and the dynamical Bayesian network [15], are also useful complementary tools for inferring the spiking dependence among ensemble neurons.

II. Point Process Generalized Linear Model

A point process is a stochastic process with 0 and 1 observations [5], [12]. Let $c = 1, \ldots, C$ denote the index of a multivariate (C-dimensional) point process. For the cth point process, let $\boldsymbol{y}_{1:T}^c = (y_1^c, \ldots, y_T^c)$ denote the observed response variables during a (discretized) time interval $[1, T]$, where y_t^c is an indicator variable that equals to 1 if there is a spike at time t and 0 otherwise. Therefore, multiple neural spike train data are completely characterized by a multivariate point process $\{\boldsymbol{y}_{1:T}^c\}_{c=1}^C$. Mathematical backgrounds on the point process theory can be found in [12] and [7].

A. Exponential Family and Generalized Linear Models

In the framework of GLM [29], we assume that the observations $\{y_{1:T}\}$ follow an exponential family distribution with the form

$$p_\theta(y_t \,|\, \theta_t) = \exp(y_t\theta_t - b(\theta_t) + c(y_t)) \tag{1}$$

TABLE I
Examples of Exponential Family in a Canonical Form

prob. dist.	link func.	θ	$b(\theta)$	$\dot{b}(\theta)$	$\ddot{b}(\theta)$
Bernoulli$(1,\pi)$	logit	$\log\frac{\pi}{1-\pi}$	$\log(1+e^\theta)$	π	$1-\pi$
Poisson(λ)	log	$\log\lambda$	$\exp(\theta)$	λ	λ

where θ denotes the canonical parameter, and $c(y_t)$ is a normalizing constant. Assume that $b(\theta_t)$ is twice differentiable, then $\mu_t \equiv \mathbb{E}_{y|\theta}[y_t] = \dot{b}(\theta_t) = (\partial b(\theta_t))/(\partial\theta_t)$, $\mathrm{Var}[y_t] = \ddot{b}(\theta_t) = (\partial^2 b(\theta_t))/(\partial\theta_t\partial\theta_t^\top)$ (where $^\top$ denotes the transpose). Moreover, the mean μ_t is related to the linear predictor via a link function g

$$g(\mu_t) = \eta_t = \boldsymbol{\beta}\boldsymbol{x}_t \tag{2}$$

where $\boldsymbol{x}_t$ denotes the input covariate at time t. Using a canonical link function, the natural parameter relates to the linear predictor by $\theta_t = \eta_t = \boldsymbol{\beta}\boldsymbol{x}_t$, Table I lists two probability distributions of exponential family (in a canonical form) for modeling point process data. In the case of Bernoulli distribution, the link function is a logit function ($\mathrm{logit}(\pi) = \log(\pi/(1-\pi))$; in the case of Poisson distribution, the link function is a log function. Consequently, the point process GLMs based on either logistic regression or Poisson regression can be used to model neural spike trains [47]. The difference between these two models is that in Poisson regression, the generalized "rate" (or conditional intensity function) λ is estimated, whereas in logistic regression, the spiking probability π is directly estimated. When the bin size Δ of the spike trains is sufficiently small, we can approximate $\pi = \lambda\Delta$ and the difference of using these two models is small. In the present paper, we use the (Binomial) logistic regression GLM for the illustration purpose.

To model the spike train point process data, we use the following logistic regression model with the logit link function.[1] Specifically, let c be the index of target neuron, and let $i = 1, \ldots C$ be the indices of trigger neurons. The Bernoulli (binomial) logistic regression GLM is written as

$$\mathrm{logit}(\pi_t) = \boldsymbol{\beta}_c\boldsymbol{x}_t = \sum_{j=0}^{d}\beta_j^c x_{j,t} = \beta_0^c + \sum_{i=1}^{C}\sum_{k=1}^{K}\beta_{i,k}^c x_{i,t-k} \tag{3}$$

where $\dim(\boldsymbol{\beta}_c) = d+1$ (where $d = C \times K$) denotes total number of parameters in the augmented parameter vector $\boldsymbol{\beta}_c = \{\beta_0^c, \beta_{i,k}^c\}$, and $\boldsymbol{x}(t) = \{x_0, x_{i,t-k}\}$. Here, $x_0 \equiv 1$ and $x_{i,t-k}$ denotes the spike count from neuron i at the kth time-lag history window. The spike count is nonnegative, therefore $x_{i,t-k} \geq 0$. Alternatively, we can rewrite (3) as

$$\pi_t = \frac{\exp(\boldsymbol{\beta}_c\boldsymbol{x}_t)}{1+\exp(\boldsymbol{\beta}_c\boldsymbol{x}_t)} = \frac{\exp(\beta_0^c + \sum_{j=1}^d \beta_j^c x_{j,t})}{1+\exp(\beta_0^c + \sum_{j=1}^d \beta_j^c x_{j,t})} \tag{4}$$

which yields the probability of a spiking event at time t. It is seen from (4) that the spiking probability π_t is a logistic sigmoid function of $\boldsymbol{\beta}_c\boldsymbol{x}(t)$; when the linear regressor $\boldsymbol{\beta}_c\boldsymbol{x}(t) = 0$,

[1] In practice, the value of K (or d) needs to be determined using a statistical procedure, such as the Akaike information criterion (AIC) or Bayesian information criterion (BIC).

$\pi_t = 0.5$. Note that $\boldsymbol{\beta}_c \boldsymbol{x}(t) = 0$ defines a $(d+1)$-dimensional hyperplane that determines the decision favoring either $\pi_t > 0.5$ or $\pi_t < 0.5$.

Equation (3) essentially defines a spiking probability model for neuron c based on its own spiking history, and that of the other neurons in the ensemble. It has been shown that such a simple spiking model is a powerful tool for the inference of functional connectivity among ensemble neurons [32], and in predicting single neuronal spikes based on collective population neuronal dynamics [49]. Here, $\exp(\beta_0^c)$ can be interpreted as the baseline firing probability of neuron c. Depending on the algebraic (positive or negative) sign of coefficient $\beta_{i,k}^c$, $\exp(\beta_{i,k}^c)$ can be viewed as a dimensionless nonnegative "gain" factor that influences the firing probability of neuron c from another neuron i at the preceding kth time lag. Therefore, a negative value of $\beta_{i,k}^c$ will strengthen the inhibitory effect and move π_t towards the negative side of the hyperplane; a positive value of $\beta_{i,k}^c$ will enhance the excitatory effect, and thereby moving π_t towards the positive side of the hyperplance. In our paper, two neurons are said to be functionally connected if any of their pairwise connections is nonzero (or the statistical estimate is significantly different from zero).

For the cth spike train point process data, we can write down the log-likelihood function

$$L(\boldsymbol{\beta}_c) = \sum_{t=1}^{T} \left[y_t^c \log \pi_t(\boldsymbol{\beta}_c) + (1 - y_t^c) \log \pi_t(1 - \boldsymbol{\beta}_c) \right]. \tag{5}$$

Let $\boldsymbol{\theta} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_C\}$ be the ensemble parameter vector, where $\dim(\boldsymbol{\theta}) = C(1+d)$. By assuming that the spike trains of ensemble neurons are mutually *conditionally independent*, the network log-likelihood of C-dimensional spike train data is written as [32]

$$L(\boldsymbol{\theta}) = \sum_{c=1}^{C} L(\boldsymbol{\beta}_c). \tag{6}$$

Note that the index c is uncoupled from each other in the network log-likelihood function, which implies that we can optimize the function $L(\boldsymbol{\beta}_c)$ separately for individual spike train observations $\boldsymbol{y}_{1:T}^c$. For simplicity, from now on we will drop off the index c at notations y_t^c and $\boldsymbol{\beta}_c$ when no confusion occurs.

III. Maximum Likelihood Estimation and Regularization

The objective of the standard maximum likelihood estimation is to maximize (6) given all spike train data. It is known that when the data sample is sufficiently large, the maximum likelihood estimate (MLE) is asymptotically unbiased, consistent, and efficient. However, when the number of samples is small, or the neural spiking rate is small (i.e., the number of "1"s in the observation $\boldsymbol{y}$ is sparse), many empirical observations have indicated that MLE produces either wrong or unreliable estimates. The error in the MLE is typically related to two factors: bias and variance, and thus the ultimate goal of statistical estimation is to produce an unbiased minimum variance estimate. The issue of bias and large variance becomes more severe when a data set with a small sample size is encountered, and the size of the parameter space is relatively large. One way to reduce variance is through regularization, which aims to improve the generalization ability of the model (on new data) while fitting finite observed training data. The idea of regularization is to impose certain prior knowledge (such as sparsity) or physiologically plausible constraint (such as temporal smoothness) on the parameters [46], [23]. Furthermore, regularization can be interpreted as imposing a prior on the parameter space from an empirical Bayesian perspective, and the penalized log-likelihood will be interpreted as the log posterior density of the parameters [42], [8]. Therefore, penalized maximum likelihood estimation seeks to maximize a regularized log-likelihood function, which consists of a log-likelihood function plus a penalty function weighted by a regularization parameter. The resultant penalized MLE can be viewed as a maximum *a posteriori* (MAP) estimate.

A. ℓ_2 Regularization

First, let us consider the following penalized log-likelihood function using ℓ_2-regularization:

$$L_2(\boldsymbol{\beta}) = L(\boldsymbol{\beta}) - \rho \boldsymbol{\beta}^\top \boldsymbol{Q} \boldsymbol{\beta} \tag{7}$$

where $\rho > 0$ denotes the regularization parameter, and $\boldsymbol{Q}$ denotes a user-defined positive semidefinite matrix. The use of the quadratic term $\boldsymbol{\beta}^\top \boldsymbol{Q} \boldsymbol{\beta}$ brings to the name of ℓ_2 regularization. Different choices of matrix $\boldsymbol{Q}$ lead to different regularization solutions (see Appendix A for more discussions on the choices of matrix $\boldsymbol{Q}$). As a special case when $\boldsymbol{Q} = \mathbf{I}$ (identity matrix), the standard "ridge regression" is recovered

$$L_2(\boldsymbol{\beta}) = L(\boldsymbol{\beta}) - \rho \|\boldsymbol{\beta}\|_2. \tag{8}$$

Note that (7) and (8) are concave function of the parameter vector $\boldsymbol{\beta}$, and minimizing the negative penalized log-likelihood estimation is a convex optimization problem.

Once the regularization parameter ρ is determined (e.g., via cross-validation or regularization path), the optimization problem reduces to maximize a concave function of $\boldsymbol{\beta}$. A standard approach to minimize a convex function is through the Newton method. Specifically, let $\mathbf{H}(\boldsymbol{\beta})$ and $\mathbf{g}(\boldsymbol{\beta})$ denote the Hessian matrix and gradient vector of the parameter vector $\boldsymbol{\beta}$ computed from (7), respectively. Denote $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T] \in \mathbb{R}^{T \times (d+1)}$ and $\boldsymbol{y} = [y_1, \ldots, y_T] \in \mathbb{R}^T$, the iterative Newton update equation (at the nth iteration) is given by [16], [34]

$$\begin{aligned} \boldsymbol{\beta}_{n+1} &= \boldsymbol{\beta}_n - \mathbf{H}^{-1}(\boldsymbol{\beta}_n) \mathbf{g}(\boldsymbol{\beta}_n) \\ &= \boldsymbol{\beta}_n + \left[\boldsymbol{X}^\top \boldsymbol{W}(\boldsymbol{\beta}_n) \boldsymbol{X} + \rho \boldsymbol{Q} \right]^{-1} \boldsymbol{X}^\top (\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{\beta}_n)) \end{aligned} \tag{9}$$

where $\hat{\boldsymbol{y}}(\boldsymbol{\beta}_n) = [\hat{\pi}_1(\boldsymbol{\beta}_n), \ldots, \hat{\pi}_T(\boldsymbol{\beta}_n)]$, $\boldsymbol{W}(\boldsymbol{\beta}) = \mathrm{diag}\{w_1, \ldots, w_T\}$ is a $T \times T$ diagonal weighting matrix, with diagonal entry $w_t = \pi_t(\boldsymbol{\beta}_n)(1 - \pi_t(\boldsymbol{\beta}_n))$. Equation (9) can also be formulated as iteratively solving a linear quadratic system

$$[\boldsymbol{X}^\top \boldsymbol{W}(\boldsymbol{\beta}_n) \boldsymbol{X} + \rho \boldsymbol{Q}] \boldsymbol{\beta}_{n+1} = \boldsymbol{X}^\top \boldsymbol{W}(\boldsymbol{\beta}_n) \boldsymbol{b} \tag{10}$$

where $\boldsymbol{b} = \boldsymbol{X}\boldsymbol{\beta}_n + \boldsymbol{W}^{-1}(\boldsymbol{\beta}_n)(\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{\beta}_n))$. For such a convex optimization problem, efficient iterative algorithms such as the iteratively reweighted least squares (IRWLS) [34] or conjugate gradient (CG) [28] can be used. For a large-scale data, or a large-size parameter estimation problem, the CG method presents a more computationally efficient solution. The CG algorithm is known to be highly efficient [with a linear complexity proportional to $\dim(\boldsymbol{\beta})$], especially when the matrix $\boldsymbol{X}^\top \boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X}$ is sparse [28]. Since the optimization problem is convex, the solution (from either IRWLS or CG) will identify the global optimum. The convergence criterion is determined such that the iteration stops when the log-likelihood change in two subsequent updates is less than 10^{-4}.

B. ℓ_1 Regularization

Another popular regularization scheme is through ℓ_1-regularization, which penalizes the ℓ_1 norm of the solution [44]. Unlike ℓ_2 regularization, ℓ_1 regularization favors the sparse solution (i.e., many coefficients in $\hat{\boldsymbol{\beta}}$ are zeros). From a decision-theoretic perspective, ℓ_2-norm is a result of penalizing the mean of a Gaussian prior of the unknown variables, while an ℓ_1 norm penalizes the median of a Laplace prior, which has heavier tails in its distribution shape. Specifically, the penalized log-likelihood function with ℓ_1 norm regularization is written as

$$L_1(\boldsymbol{\beta}) = L(\boldsymbol{\beta}) - \rho\|\boldsymbol{\beta}\|_1 \tag{11}$$

which is a concave function of $\boldsymbol{\beta}$, but is not twice differentiable with respect to $\boldsymbol{\beta}$ (therefore, the Hessian matrix cannot be computed). Recently, many convex optimization procedures have been proposed for the ℓ_1 regularized GLM [45], [26], [27], [33], [37], [17]. Although individual algorithms differ in their own implementations, the common optimization goal is to seek a sparse solution that simultaneously satisfies the data fitting constraints. We shall briefly describe one efficient and state-of-the-art algorithm based on an interior-point method [27], which will be used for benchmark comparison in Section V.

The interior-point method for maximizing $L_1(\boldsymbol{\beta})$ in (11) aims to solve an equivalent optimization problem [27]

$$\begin{aligned} \text{minimize} \quad & -L(\boldsymbol{\beta}) + \rho\mathbf{1}^\top\boldsymbol{u} \\ \text{subject to} \quad & -u_j \le \beta_j \le u_j, j = 1, \ldots, d \end{aligned}$$

with variables $\boldsymbol{u} \in \mathbb{R}^d$. The logarithmic barrier for the bound constraints $-u_j \le \beta_j \le u_j$ is

$$\Phi(\boldsymbol{\beta}, \boldsymbol{u}) = -\sum_{j=1}^{d} \log\left(u_j^2 - \beta_j^2\right) \tag{12}$$

with domain $\mathrm{dom}\Phi = \{(\boldsymbol{\beta}, \boldsymbol{u}) \in \mathbb{R}^{d+1} \times \mathbb{R}^d \| |\beta_j| < u_j, j = 1, \ldots, d\}$. The new weighted objective function augmented by the logarithmic barrier is further written as [27]

$$E(\boldsymbol{\beta}, \boldsymbol{u}) = -\kappa L(\boldsymbol{\beta}) + \kappa\rho\mathbf{1}^\top\boldsymbol{u} + \Phi(\boldsymbol{\beta}, \boldsymbol{u}) \tag{13}$$

where $\kappa > 0$ is a scalar parameter that defines the central path of a curve of $E(\boldsymbol{\beta}, \boldsymbol{u})$. The new function defined in (13) is smooth and strictly convex, and it can be optimized using the Newton or CG method. Increasing the values of κ leads to a sequence of points on the central path, which ultimately leads to a suboptimal estimate $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{u}})$ [27].

C. Identifying Functional Connectivity

Upon completing the standard or penalized likelihood inference, we obtain the parameter estimate $\hat{\boldsymbol{\beta}}$. Let

$$\boldsymbol{\Sigma} \approx I(\boldsymbol{\beta})^{-1} = -\mathbb{E}\left[\frac{\partial^2 L}{\partial\hat{\boldsymbol{\beta}}\partial\hat{\boldsymbol{\beta}}^\top}\right]^{-1} \tag{14}$$

denote the inverse of the negative Hessian matrix of the log-likelihood estimated from the ensemble samples. From the property of MLE it is known that $\boldsymbol{\Sigma}$ approximates the inverse of the Fisher information matrix $I(\boldsymbol{\beta})$; in addition, under the regularity condition and large sample assumption, the MLE $\hat{\boldsymbol{\beta}}$ *asymptotically* follows a multivariate Gaussian distribution [34], [5], with the mean as the true parameter $\boldsymbol{\beta}$ and the covariance matrix given in (14): $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$, from which we can further derive the 95% Wald confidence bounds of each element in $\boldsymbol{\beta}$ as $\hat{\beta}_j \pm 1.96\Sigma_{jj}^{1/2}$. Provided any of the coefficients are significantly different from zero, or their 95% Wald confidence intervals are not overlapping with 0, we conclude that the "directional connection" (at a certain time lag) between the trigger neuron(s) to target neuron is either excitatory (positive) or inhibitory (negative).

To estimate the matrix $\boldsymbol{\Sigma}$ in (14), in the case of standard maximum likelihood estimation for the GLM, upon convergence we can derive that $\boldsymbol{\Sigma} = (\boldsymbol{X}^\top\boldsymbol{W}(\boldsymbol{\beta})\boldsymbol{X})^{-1}$; in the case of ℓ_2 penalized maximum likelihood (PML), we have $\boldsymbol{\Sigma} = (\boldsymbol{X}^\top\boldsymbol{W}\boldsymbol{X} + \rho\boldsymbol{Q})^{-1}$: the derivation follows a regularized IRWLS algorithm. In the case of ℓ_1-PML, let $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_-)$ and $\boldsymbol{X} = (\mathbf{1}, \boldsymbol{X}_-)$, upon convergence for the augmented vector $(\beta_0, \boldsymbol{\beta}_-, \boldsymbol{u})$, the Hessian matrix computed from the objective function (13) is given by [27]

$$\mathbf{H} = \begin{bmatrix} \kappa\mathbf{1}^\top\boldsymbol{W}\mathbf{1} & \kappa\mathbf{1}^\top\boldsymbol{W}\boldsymbol{X}_- & \mathbf{0} \\ \kappa\boldsymbol{X}_-^\top\boldsymbol{W}\mathbf{1} & \kappa\boldsymbol{X}_-^\top\boldsymbol{W}\boldsymbol{X}_- + \boldsymbol{D}_1 & \boldsymbol{D}_2 \\ \mathbf{0} & \boldsymbol{D}_2 & \boldsymbol{D}_1 \end{bmatrix} \tag{15}$$

where $\mathbf{H} \in \mathbb{R}^{(2d+1)\times(2d+1)}$, $\boldsymbol{D}_1 = \mathrm{diag}\{(2(u_1^2 + \beta_1^2))/((u_1^2 - \beta_1^2)^2), \ldots, (2(u_d^2 + \beta_d^2))/((u_d^2 - \beta_d^2)^2)\}$, and $\boldsymbol{D}_2 = \mathrm{diag}\{(-4u_1\beta_1)/((u_1^2 - \beta_1^2)^2), \ldots, (-4u_d\beta_d)/((u_d^2 - \beta_d^2)^2)\}$. In light of the Schur complement, we obtain

$$\boldsymbol{\Sigma} = \begin{bmatrix} \kappa\mathbf{1}^\top\boldsymbol{W}\mathbf{1} & \kappa\mathbf{1}^\top\boldsymbol{W}\boldsymbol{X}_- \\ \kappa\boldsymbol{X}_-^\top\boldsymbol{W}\mathbf{1} & \kappa\boldsymbol{X}_-^\top\boldsymbol{W}\boldsymbol{X}_- + \boldsymbol{D}_3 \end{bmatrix}^{-1} \tag{16}$$

where $\boldsymbol{D}_3 = \boldsymbol{D}_1 - \boldsymbol{D}_2\boldsymbol{D}_1^{-1}\boldsymbol{D}_2$.

To quantify the connectivity among C neurons, from (3) we define the mean connectivity ratio as follows:

$$\text{ratio} = \frac{1}{K}\sum_{c=1}^{C}\sum_{i=1}^{C}\sum_{k=1}^{K}\frac{\#\{|\beta_{i,k}^c| \gg 0\}}{C(C-1)} \tag{17}$$

where $\#\{|\beta_{i,k}^c| \gg 0\}$ denotes that the number of the coefficient $\beta_{i,k}^c$ whose statistical estimates are significantly nonzero. Note that the spiking dependence is directional and asymmetric in our statistical model, the spiking dependence between A→B and B→A is not necessarily the same. Therefore, for a total of

C ensemble neurons, there are possibly $(C^2 - C)$ directions (excluding C self-connection coefficients) between all neuron pairs.

IV. Bayesian Inference and Variational Bayes Method

In addition to the maximum likelihood estimation, another appealing statistical inference tool is Bayesian estimation. The goal of Bayesian inference is to estimate the parameter posterior $p(\boldsymbol{\beta}|\boldsymbol{y})$ given a specific parameter prior $p(\boldsymbol{\beta})$. Normally, because the posterior is analytically nontrackable, we will need to resort to strategies for approximation. These methods include the Laplace approximation for log-posterior [18], [2], expectation propagation (EP) for moment matching [39], [21], and MCMC sampling [36], [18]. In comparison amongst these approximation methods, the Laplace and EP approximations are less accurate (especially when the posterior has multiple modes or the mode is not near the majority of the probability mass); MCMC methods are more general, but have high computational demands, and experience difficulties with assessing the convergence of Markov chains. As an alternative Bayesian inference procedure, variational Bayesian (VB) methods attempt to maximize the lower bound of the marginal likelihood (also known as *evidence*) or the marginal log-likelihood [2]. Unlike Laplace and EP approximation, MCMC and VB methods allow for a fully hierarchical Bayesian inference. Furthermore, the VB method is deterministic, and thereby more computationally efficient, while MCMC methods are prohibitive for large-scaled problems and require careful convergence diagnosis. Here we use the hierarchical variational Bayesian (HVB) algorithm for inferring the parameters in the point process GLM [9].[2]

Specifically, let $\boldsymbol{\alpha}$ denote the hyperparameter set, and we can derive

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{x}) &= \log \int\int p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\beta})p(\boldsymbol{\beta}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})\mathrm{d}\boldsymbol{\beta}\mathrm{d}\boldsymbol{\alpha} \\ &\geq \int\int q(\boldsymbol{\beta},\boldsymbol{\alpha})\log\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\beta})p(\boldsymbol{\beta}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{q(\boldsymbol{\beta},\boldsymbol{\alpha})}\mathrm{d}\boldsymbol{\beta}\mathrm{d}\boldsymbol{\alpha} \equiv \tilde{L}\end{aligned} \tag{18}$$

where $p(\boldsymbol{\beta}|\boldsymbol{\alpha})$ denotes the prior distribution of $\boldsymbol{\beta}$, specified by the hyperparameter $\boldsymbol{\alpha}$. The variational distribution has a factorial form such that $q(\boldsymbol{\beta},\boldsymbol{\alpha}) = q(\boldsymbol{\beta})q(\boldsymbol{\alpha})$, which attempts to approximate the posterior $p(\boldsymbol{\beta},\boldsymbol{\alpha}|\boldsymbol{y})$. This approximation leads to an analytical posterior form if the distributions are conjugate-exponential. The use of hyperparameters within the hierarchical Bayesian estimation framework provides a modeling advantage compared to the empirical Bayesian approach, since the hierarchical Bayesian modeling employs a fully Bayesian inference procedure that makes the parameter estimate less sensitive to the fixed prior (as in the empirical Bayesian approaches). It is emphasized that the variational log-likelihood $\tilde{L}$ is indeed a *functional*—the function of two variational distributions (or pdfs) $q(\boldsymbol{\beta})$ and $q(\boldsymbol{\alpha})$.

[2]The Bayesian logistic regression algorithm [19], which uses a cyclic coordinate descent algorithm for either Gaussian or Laplace prior, is de facto an empirical Bayes algorithm, since the hyperparameter's probability distribution was not modeled and the hyperparameter was selected by heuristics.

A variational approximation algorithm for logistic regression has been developed in the field of machine learning [24], and it can be easily extended to the Bayesian setting [2]. The basic idea of variational approximation is to derive a variational lower bound for the marginal log-likelihood function. However, the hyperparameters used in [24] are fixed a priori, so their model is empirical Bayesian. Here, we extend the model with hierarchical Bayesian modeling using *automatic relevance determination* (ARD) [30] for the purpose of variable selection. Such a fully Bayesian inference integrated with ARD allows us to design a separate prior for each element β_j in the vector $\boldsymbol{\beta}$ and to set a conjugate prior $p(\boldsymbol{\alpha})$ for the hyperparameters using a common gamma hyperprior. Our prior distributions are set up as follows:

$$\begin{aligned}&p(\boldsymbol{\beta}|\boldsymbol{\alpha}) \\ &\quad\sim \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}_0, \boldsymbol{A}^{-1}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)^\top \boldsymbol{A}(\boldsymbol{\beta}-\boldsymbol{\mu}_0)\right) \\ &p(\boldsymbol{\alpha}) \\ &\quad= \prod_{j=0}^{d}\Gamma(\alpha_j|a_0,b_0)\end{aligned}$$

where $\boldsymbol{A} = \mathrm{diag}\{\boldsymbol{\alpha}\} \equiv \mathrm{diag}\{\alpha_0,\ldots,\alpha_d\}$ (a non-ARD prior is equivalent to setting $\boldsymbol{A} = \alpha\mathbf{I}$, where α is a global hyperparameter), and $\mathrm{Gamma}(\alpha_j|a_0,b_0) = (1)/(\Gamma(a_0))b_0^{a_0}\alpha_j^{a_0-1}e^{-b_0\alpha_j}$. Here, we assume that the mean hyperparameter is fixed (e.g., $\boldsymbol{\mu}_0 = \mathbf{0}$).

Let $\boldsymbol{\xi} = \{\xi_t\}$ denote the data-dependent variational parameters (that are dependent on the input variables $\{\boldsymbol{x}_t\}$). In light of the variational approximation principle [24], one can derive a tight lower bound for the logistic regression likelihood, which will be used in the VB inference. Specifically, applying the VB inference yields the variational posteriors $q(\boldsymbol{\beta}|\boldsymbol{y})$ and $q(\boldsymbol{\alpha}|\boldsymbol{y})$

$$\begin{aligned}\log q(\boldsymbol{\beta}|\boldsymbol{y}) &= \log\tilde{p}(\boldsymbol{\beta},\boldsymbol{\xi}) + \mathbb{E}_{q(\boldsymbol{\alpha})}[\log p(\boldsymbol{\beta}|\boldsymbol{\alpha})] \\ &= \log\mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}_T,\boldsymbol{\Sigma}_T), \qquad (19) \\ \log q(\boldsymbol{\alpha}|\boldsymbol{y}) &= \mathbb{E}_{q(\boldsymbol{\beta})}[\log p(\boldsymbol{\beta}|\boldsymbol{\alpha})] + \log p(\boldsymbol{\alpha}) \\ &= \log\left\{\prod_{j=0}^{d}\mathrm{Gamma}(\alpha_j|a_T,b_{j,T})\right\} \qquad (20)\end{aligned}$$

which follow from updates from conjugate priors and posteriors for the exponential family (Gaussian and Gamma distributions). The term $\tilde{p}(\boldsymbol{\beta},\boldsymbol{\xi})$ appearing in (19) denotes the variational likelihood bound for logistic regression

$$\begin{aligned}\log p(\boldsymbol{\beta},\boldsymbol{\xi}) &\geq \log\tilde{p}(\boldsymbol{\beta},\boldsymbol{\xi}) \\ &= \sum_{t=1}^{T}\left(\log\sigma(\xi_t) - \frac{\xi_t}{2} + \phi(\xi_t)\xi_t^2\right) \\ &\quad - [\boldsymbol{\beta}^\top(\phi(\xi_t)\boldsymbol{x}_t\boldsymbol{x}_t^\top)\boldsymbol{\beta}] + \boldsymbol{\beta}^\top\left(\boldsymbol{x}_t y_t - \frac{\boldsymbol{x}_t}{2}\right)\end{aligned} \tag{21}$$

where $\sigma(\cdot)$ is a logistic sigmoid function. The variational likelihood bound at the right-hand side of (21) has a quadratic form in terms of $\boldsymbol{\beta}$, and therefore it can be approximated by a Gaussian

likelihood (which results in (19). The other terms appearing in (19) through (21) are defined by

$$\begin{aligned}
\xi_t &= \sqrt{\boldsymbol{x}_t^\top(\boldsymbol{\Sigma}_T + \boldsymbol{\mu}_T\boldsymbol{\mu}_T^\top)\boldsymbol{x}_t} \\
\phi(\xi_t) &= \frac{\tanh(\xi_t/2)}{4\xi_t} \\
\boldsymbol{\Sigma}_T^{-1} &= \mathbb{E}_{q(\boldsymbol{\alpha})}[\boldsymbol{A}] + 2\sum_{t=1}^{T}\phi(\xi_t)\boldsymbol{x}_t\boldsymbol{x}_t^\top \\
\boldsymbol{\mu}_T &= \boldsymbol{\Sigma}_T\left(\mathbb{E}_{q(\boldsymbol{\alpha})}[\boldsymbol{A}]\boldsymbol{\mu}_0 + \sum_{t=1}^{T}(y_t - 0.5)\boldsymbol{x}_t\right) \\
\mathbb{E}_{q(\boldsymbol{\alpha})}[\boldsymbol{A}] &= \mathrm{diag}\{a_T/b_{j,T}\} \equiv \boldsymbol{A}_T \\
a_T &= a_0 + 0.5 \\
b_{j,T} &= b_0 + 0.5[(\boldsymbol{\mu}_T)_j^2 + (\boldsymbol{\Sigma}_T)_{jj}]
\end{aligned}$$

where the subscript T in the updated parameters represents the fact that the parameters and hyperparameters are updated after passing a total of T samples. Finally, we can derive the variational lower bound of marginal log-likelihood (Appendix B)

$$\begin{aligned}
\tilde{L} = \frac{1}{2}\Bigg\{&\boldsymbol{\mu}_T^\top\boldsymbol{\Sigma}_T^{-1}\boldsymbol{\mu}_T + \log|\boldsymbol{\Sigma}_T| \\
&+ \sum_{t=1}^{T}\left(2\log\sigma(\xi_t) - \xi_t + 2\phi(\xi_t)\xi_t^2\right)\Bigg\} \\
&+ \sum_{j=0}^{d}\Bigg\{-\log\Gamma(a_0) + a_0\log b_0 - b_0\frac{a_T}{b_{j,T}} \\
&-a_T\log b_{j,T} + \log\Gamma(a_T) + a_T\Bigg\}. \qquad (22)
\end{aligned}$$

The VB inference alternatingly updates (19) and (20) to monotonically increase $\tilde{L}$. The criterion for algorithmic convergence is set until the consecutive change of (22) is sufficiently small (say 10^{-4}). Upon completing the VB inference, the confidence bounds of the estimates can be derived from the posterior mean and the posterior variance [2].

It is noted that due to the assumed factorial form of posterior distribution, the variance of the estimates is relatively underestimated [2]. However, this will have little effect on the identification result. While using the ARD for variable selection, a nonsignificant coefficient is said to be pruned if its mean and variance estimates are both small (close to 0). Therefore, even if the variance is slightly underestimated, provided that the mean estimate value is relatively large (or the solution is nonsparse), it will not change the inferred result.

V. Results

Data simulations were used to compare the performance of different statistical inference procedures: 1) the standard ML method, 2) penalized ML (PML) with ℓ_2 regularization, 3) PML with ℓ_1 regularization, and 4) hierarchical VB (HVB) method. A summary of these methods is given in Table II. Based on the performance of these methods with the simulated data, the optimal statistical inference method was also applied to real spike train data. All of the custom statistical inference algorithms were written in MATLAB (MathWorks, Natick, MA) and can be accessed online.[3] The software of the ℓ_1 regularized logistic regression [27] was accessible online.[4]

TABLE II
Comparison of Statistical Inference Methods and Algorithms

Inference method	Algorithm	Free parameter(s)
maximum likelihood (ML)	IRWLS, CG	none
ℓ_2 penalized ML	IRWLS, CG	ρ
ℓ_1 penalized ML	interior-point method	ρ, κ
hierarchical Bayesian	variational Bayes	hyperpriors a_0, b_0

A. Goodness-of-Fit and Performance Metrics

The goodness-of-fit of the point process models estimated from all algorithms is evaluated based on the Time-Rescaling Theorem and Kolmogorov–Smirnov (KS) test [4], [5].[5] Assuming that a univariate point process specified by J discrete events: $0 < u_1 < \ldots < u_J < T$, defines the random variables $z_j = \sum_{\tau=u_{j-1}}^{u_j}\pi_\tau$ for $j = 1, 2, \ldots, J-1$. Thus, the random variables z_js are independent, and unit-mean exponentially distributed. By variable of transformation $v_j = 1 - \exp(-z_j)$, then v_js are independent, uniformly distributed within the region [0, 1]. Let $r_j = F^{-1}(v_j)$ (where $F(\cdot)$ denotes the cumulative distribution function (cdf) of the standard Gaussian distribution), then r_js will be independent standard Gaussian random variables. Furthermore, the standard KS test is used to compare the cdf of v_j against that of the random variables uniformly distributed within [0, 1], and the KS statistic measures the maximum deviation of the empirical cdf from the uniform cdf. The KS statistics will be computed for both simulated and real spike trains.

In simulation studies, in addition to the KS test, we also compute the misidentification error rate, which is the sum of the false positive (FP) and false negative (FN) rates. By false positive, it is meant that the true connection coefficient (only known in simulations) is zero, but its statistical estimate from the algorithm is mistakenly identified as being significantly nonzero. By false negative, it is meant that the true connection coefficient is nonzero, but the statistical estimate from the algorithm is mistakenly identified as being zero. In simulation studies, we also compute the mean-squared error (MSE) and the normalized MSE (NMSE) of the estimate, which are defined as

$$\begin{aligned}
\mathrm{MSE} &= \frac{1}{C}\sum_{c=1}^{C}||\boldsymbol{\beta}_c - \hat{\boldsymbol{\beta}}_c||_2, \\
\mathrm{NMSE} &= \frac{1}{C}\sum_{c=1}^{C}\frac{||\boldsymbol{\beta}_c - \hat{\boldsymbol{\beta}}_c||_2}{||\boldsymbol{\beta}_c - \overline{\beta_c}||_2}
\end{aligned}$$

where $\hat{\boldsymbol{\beta}}_c$ denotes the estimate of the true (simulated) $\boldsymbol{\beta}_c$, and $\overline{\beta_c}$ denotes the mean value of the vector $\boldsymbol{\beta}_c$.

Above all, we like to compare the bias and variance of the estimates computed from different statistical inference algorithms.

[3]Available online http://neurostat.mit.edu/software.

[4]Available online http://www.stanford.edu/~boyd/l1_logreg/

[5]In the case of fitting low-firing rate neural spike trains, when the lengths of inter-spike intervals are comparable to the length of the observation window, a modified KS test that considers censoring can be used [51].

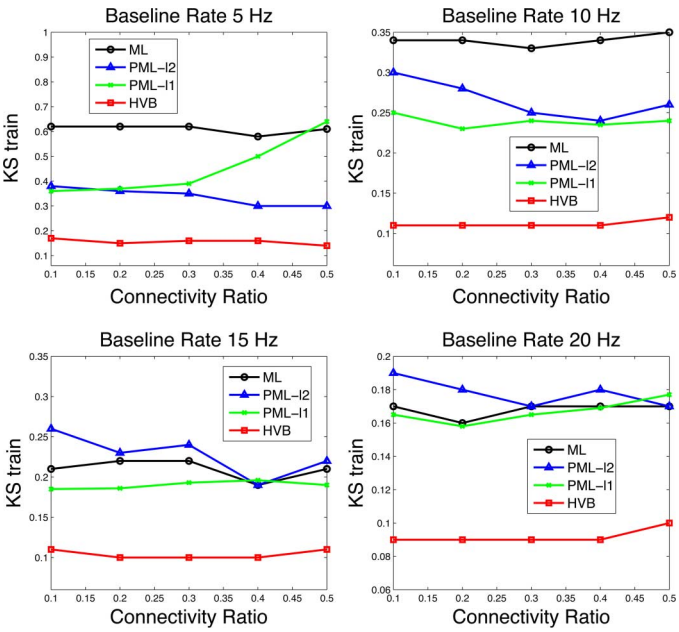


Fig. 1. Comparison of algorithms on KS statistics for the training set.

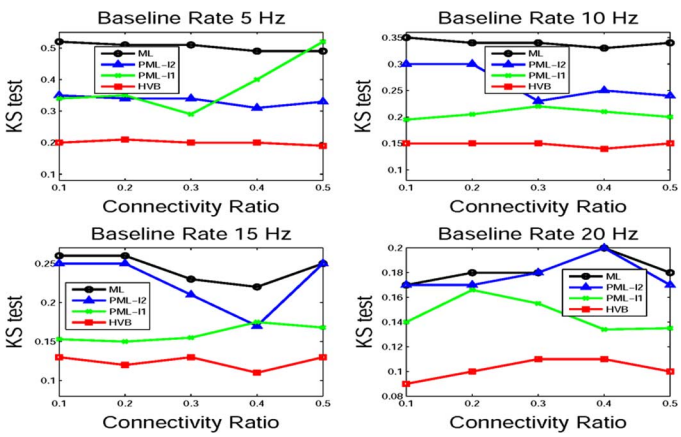


Fig. 2. Comparison of algorithms on KS statistics for the training set.

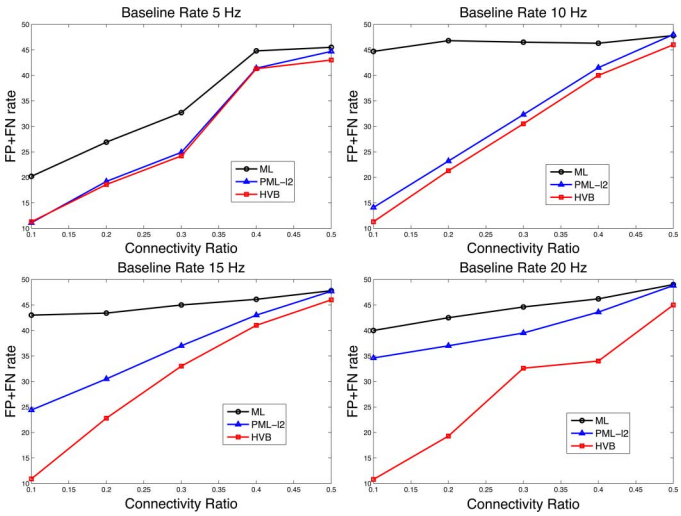


Fig. 3. Comparison of algorithms on misidentification (FP+FN) error rate.

Roughly speaking, MSE and NMSE provide two measures of the estimate's bias. The KS statistics on the testing or validation data as well as the misidentification rate provide essential information about the estimate's variance. As far as functional connectivity is concerned in the present study, the FP and FN rates are particularly relevant. Ideally, a low misidentification error rate and a lower KS statistics would be the most important criterion for choosing the best algorithm.

B. Simulation Studies

With the simulation data, a systematic investigation of algorithmic performance was conducted under different conditions. We considered a relatively small network that consisted of 10 simulated neurons with varying connectivity ratios (five scales: 0.1, 0.2, 0.3, 0.4, 0.5). All neurons are assumed to have roughly equal baseline firing rates (four scales: 5, 10, 15, 20 spikes/s), and the nonzero history-dependent coefficients were uniformly randomly generated between $[-h, h]$ (where h was set to be inversely proportional to the baseline firing rate). To create a small spike train data set with short recordings, we simulated multivariate spike trains under the same condition for eight independent trials, each trial lasting 1000 ms. The input covariates consisted of spike counts from previous temporal windows (equal size of 5 ms) up to 80 ms from all 10 neurons. Thus, there were 16 spike counts from each neuron, totaling $d = 16 \times 10 = 160$ covariates ($\dim(\boldsymbol{\beta}_c) = 161$ because of an additional baseline shift parameter). Under each condition, we simulated 20 Monte Carlo runs with varying random seeds. In each condition, we also generated an independent set of trials, which are reserved for testing (or cross-validation). The KS statistics for fitting the training (KStrain) and testing (KStest) data were both computed.

Since the simulated coefficients are random and have no smoothness structure (which may also be the case in real biological systems), we only used the standard ℓ_2 regularization by setting $\boldsymbol{Q} = \mathbf{I}$. The optimal regularization parameters were chosen by cross-validation or leave-one-out procedure. In the ℓ_1 regularization, there are two options for choosing regularization parameters, one based on fixed value followed by cross-validation, and the other based on regularization path [27]. The regularization parameter that resulted in the best KS statistic during cross-validation was selected. We run experiments for 20 Monte Carlo runs using random generated data from 10 virtual neurons and compared the algorithmic performance. The averaged results of the simulation study are summarized in Figs. 1–4. Note that all mean statistics are averaged over 10 neurons and 20 independent Monte Carlo runs.

The following observations summarize our findings.

1) Testing the algorithms began with the training data, and the KS statistics for each algorithm's performance using the same training data under the different baseline firing rate conditions are displayed in Fig. 1. Across all algorithms, differences in KS statistic were only mildly affected by the varying connectivity ratio in this case. As expected, the KS statistics do decrease as the baseline firing rates of the virtual neurons increase (since more spikes were generated). Interestingly, when the baseline rate was at 5

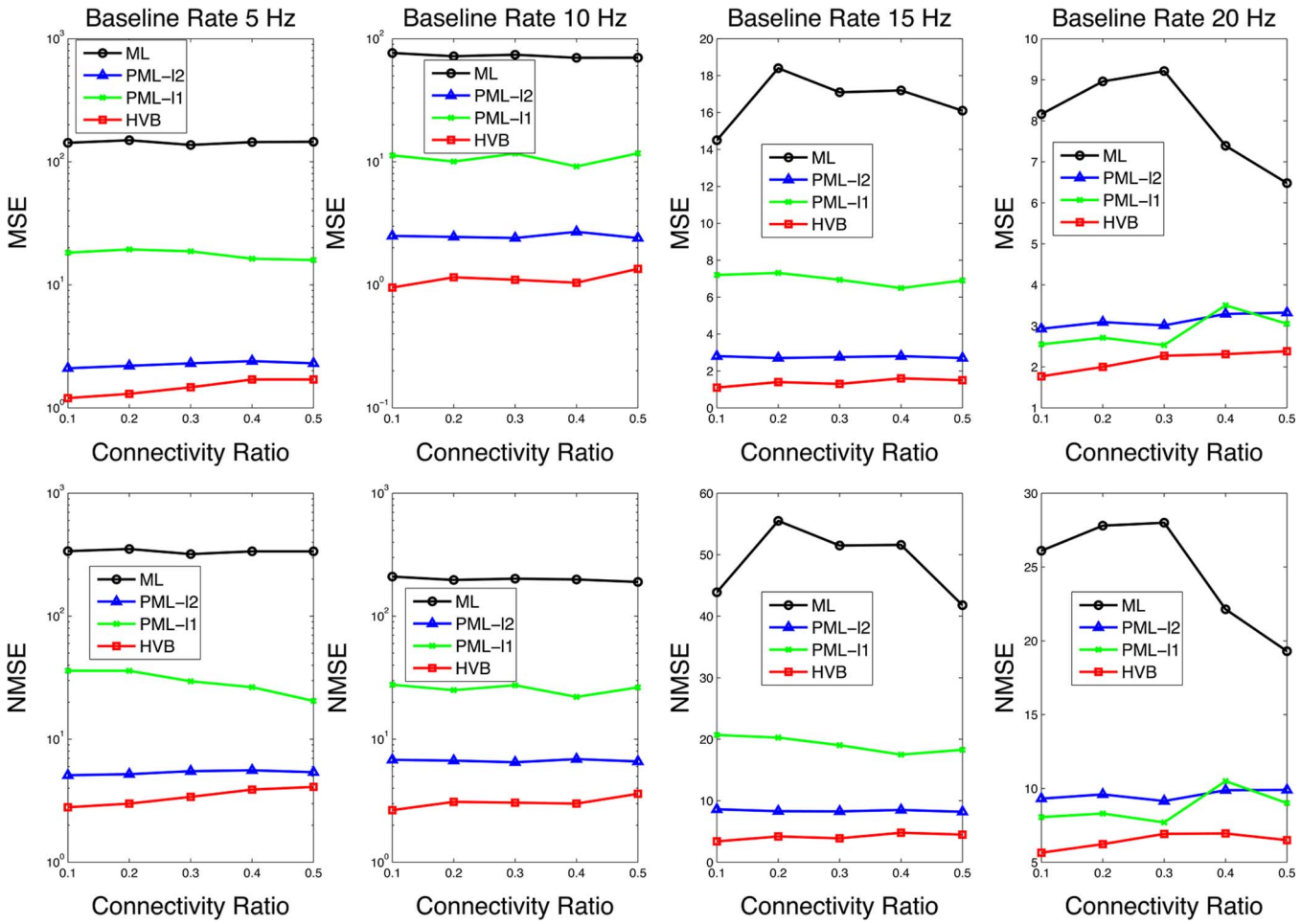


Fig. 4. Comparison of algorithms on MSE and RMSE statistics. Note that the y-axis in the first two columns are in log-scale.

or 10 Hz, the standard ML algorithm had the highest KS statistics in the training data, but when the baseline rate was increased (at 15 and 20 Hz), its performance became comparable to that of the other two penalized ML algorithms. In all cases, the HVB algorithm performed better, and the majority of the neuron fits fall within the 95% confidence bounds. The HVB's superior KS statistics are more apparent with higher firing rate than with lower firing rates. Because all neurons have similar baseline firing rate, the KS scores from all neurons are also very close in value.

2) The algorithms were then applied to the testing data, and the KS statistics were computed for testing the generalization ability on unseen data from each estimate (Fig. 2). The ML and HVB algorithms seem to be more robust regardless of the connectivity ratio, while the KS statistics of two penalized ML algorithms appeared to be affected by changes of the connectivity ratio. However, no general trend was observed. Again, the HVB algorithm exhibited the best performance in all of the tested conditions, followed by the ℓ_1-PML algorithm. As a general observation, the two PML algorithms and HVB algorithm all showed noticeable advantages over the standard ML inference, but this was expected, as it was anticipated that the ML estimate would lose its desirable asymptotic properties in a "small data set" scenario. However, while the two PML algorithms clearly outperform the standard ML inference only under low baseline firing rate conditions, the HVB has showed significantly improved performances regardless of the baseline rate. The results also confirm that when the firing rate is high, and the number of "1s" in the point process observations is large, the KS statistics from all (standard or regularized) ML algorithms are similar for both training and testing data sets. Summarizing Figs. 1 and 2, we have seen HVB has better performance than ML and PML methods in both fitting accuracy (training data) and generalization (testing data), under varying connectivity ratios and varying baseline firing rates. Specifically, the advantage of HVB is most prominent in the low connectivity ratio and/or the medium-to-high firing condition.

3) We next considered how the misidentification (FP+FN) error rate changes as the connectivity ratio increases under different baseline firing rate conditions (Fig. 3). Once again, the overall best performance was produced by use of the HVB algorithm, followed by the ℓ_2-PML algorithm, and finally the standard ML algorithm (the ℓ_1 PML algorithm was excluded from the comparison here because the codes provided in [27] do not produce the confidence bound of the estimates). Under the low baseline firing rate (5 Hz), the PML and HVB had almost identical misidentification error rates, however, as the baseline

firing rate gradually increased, the gap between the PML and HVB algorithms also increased. In our simulations, it was also found that there is an inverse relationship between FP and FN error: an algorithm that had a low FP error rate typically had a relatively greater FN error. A relationship also existed between connectivity ratio and (FP+FN) error rate that occurred regardless of the baseline firing rate, as the connectivity ratio increases to 0.5, the (FP+FN) error rate reaches an asymptotic level for all tested algorithms. This implies that there is a "bottleneck" limit for identifying the true functional connectivity for all algorithms. Therefore, in the worst case, the (FP+FN) error rate can be close to a random guess (50%). The high FN error rate may be due to the fact that many of the "weak connections" in the simulations are mistakenly identified as not showing connections.[6] In the case where a sparse solution is favored (i.e., in ℓ_1-PML and HVB algorithms), weak connectivity coefficients will be pushed towards zero, while only the stronger connections (which matter more) will reach significance. However, as seen in simulations, at about 10–15 Hz baseline rate and with 0.3 connectivity ratio, the misidentification error rate is still relatively high $(\sim 30\%)$ even for the HVB algorithm. Therefore, it remains a challenging task to identify the weak connections (especially weak negative connections where most misidentification errors occur) in statistical inference.

4) Our final comparison using the simulated data involved measuring the bias associated with each algorithm's estimate using MSE and NMSE metrics (Fig. 4). The PML and HVB approaches showed much better performance in MSE and NMSE than the standard ML method, however, the HVB algorithm again performed the best. In most of the testing conditions (except for 20 Hz baseline rate), ℓ_2-PML is better than ℓ_1-PML. This is because ℓ_1 regularization favors a sparse solution, which forces many small or weak connection coefficients' estimates towards zero, possibly causing an increase in the bias of the estimate. As also expected, ℓ_1 regularization had a smaller variance than ℓ_2 regularization (see Fig. 2 on KStest statistics).
5) Overall, the HVB method achieved the best performance in all categories: KS statistics (in both training and testing spike train data sets), MSE and NMSE, as well as the misidentification error rate. Moreover, in terms of computational speed (to achieve algorithmic convergence), it is observed that the standard ML and ℓ_2-PML algorithms have the fastest convergence, while the ℓ_1-PML and HVB algorithms have comparable convergence rates, roughly 3–6 longer (depending on specific simulation) than the other two algorithms. In our experiments, all algorithms converged in all simulated conditions. Generally, when the connectivity ratio is higher or the spiking rate is lower, the convergence speed of ℓ_1-PML is slower; but the convergence speed of HVB remains very similar regardless of the connectivity ratio and spiking rate. Furthermore, when N-fold cross-validation was used to select the suboptimal regularization parameter in the ℓ_1 and ℓ_2 PML algorithms, the total N-run computational cost and time could be much greater when compared with the single-run HVB algorithm. Interestingly, ℓ_2 PML achieved similar performance to HVB in the misidentification error, especially in the low firing rate condition. However, comparing the KStrain and KStest statistics and MSE/NMSE indicates that HVB has significantly better performance. This might suggest that PML is effective in finding a "yes/no" solution, but not an accurate solution; in statistical jargon, the PML methods have a good "discrimination" but a relatively poor "calibration." However, the discrimination ability of PML gradually decreases as the firing rate increases.

Additional issues of interest are discussed below.

1) Data Size Analysis: The effect of the data size on algorithmic performance has been examined. This was accomplished by keeping the size of the parameter space intact, while doubling or tripling the size of the training data set, and comparing the performance of the different algorithms. To illustrate the method, we have used a medium (0.3) connectivity ratio, and computed the KS statistics (only on testing data), misidentification error rate, and MSE for all tested algorithms. The results are summarized in Table III, and mixed results amongst the different algorithms are evident. For the standard ML method, increasing data size improves the KS statistic and MSE, but not necessarily the misidentification error rate. For the penalized ML methods, increasing data size either mildly improves or does not change the MSE or KS statistic, and has a mixed effect on misidentification error rate. For the HVB method, increasing data size improves the KS statistic but has very little effect on the MSE and misidentification error rate. These observations suggest that the results obtained using the HVB method are rather robust with regard to the data size, which is appealing for the small data set problem.

2) Sensitivity Analysis: Except for the standard ML algorithm, the other three algorithms have some additional free parameters (see the last column in Table II). The regularization parameter ρ needs to be selected from cross-validation. The κ parameter is set in a way that it is gradually increased in the interior-point method in a systematic way [27]. In HVB, the two hyperprior parameters a_0 and b_0 control the shape of the gamma prior (which influences the sparsity of the solution). A close examination using simulation data further revealed that the KS statistics are somewhat insensitive to the choices of a_0 and b_0, although their values may change the respective FP or FN error rate. However, given a wide range of values for (a_0, b_0), the total (FP+FN) error rate remains roughly stable. This suggests that changing the hyperprior parameters of the HVB algorithm will potentially change the desirable sparsity of the solution, which will further affect the trade-off between the FP and FN error. As an illustration, Fig. 5 presents the Monte Carlo averaged (across 10 independent runs) performances on the KS statistics

[6]The seemingly high misidentification error was partially due to the setup of our simulations. While simulating the connection weights from a uniform distribution, we counted all weak connections (even very small values) as evidence of true connectivity. In the case of high connectivity ratio, there will be proportionally more weak connections. Consequently, the tested algorithms often failed to detect the "weak connections," thereby causing a high FN error (see Fig. 3 and Table III). As expected, if in simulations we only maintain the strong connections, then the resultant misidentification error rate would significantly reduce.

TABLE III
PERFORMANCE COMPARISON UNDER DIFFERENT DATA SIZE (WITH A FIXED CONNECTIVITY RATIO OF 0.3) IN SIMULATIONS. MEAN STATISTICS IN THE TABLE ARE AVERAGED OVER 10 MONTE CARLO RUNS. DATA SIZE "1 ×" REPRESENTS THE STANDARD SETUP DESCRIBED IN THE TEXT

Data size	1×				2×				4×			
	KStest statistics											
	5 Hz	10 Hz	15 Hz	20 Hz	5 Hz	10 Hz	15 Hz	20 Hz	5 Hz	10 Hz	15 Hz	20 Hz
ML	0.510	0.345	0.238	0.184	0.254	0.153	0.115	0.090	0.120	0.082	0.060	0.055
ℓ_2-PML	0.345	0.232	0.215	0.182	0.243	0.146	0.108	0.088	0.119	0.082	0.062	0.053
ℓ_1-PML	0.295	0.220	0.165	0.154	0.224	0.135	0.098	0.085	0.120	0.082	0.067	0.050
HVB	0.202	0.153	0.135	0.119	0.140	0.115	0.086	0.065	0.092	0.070	0.060	0.048
	MSE											
	5 Hz	10 Hz	15 Hz	20 Hz	5 Hz	10 Hz	15 Hz	20 Hz	5 Hz	10 Hz	15 Hz	20 Hz
ML	137.2	74.1	17.1	7.4	54	10.5	5.5	4.1	20.4	5.3	3.4	2.4
ℓ_2-PML	2.3	2.4	2.7	3.0	2.2	2.5	2.6	2.6	2.4	2.3	2.2	2.2
ℓ_1-PML	18.7	11.7	6.9	2.5	15.6	7.6	4.3	3.1	12.9	5.0	3.2	2.3
HVB	1.5	1.1	1.3	1.8	1.3	1.1	1.2	1.1	1.4	1.0	1.0	0.9
	FP+FN error rate											
	5 Hz	10 Hz	15 Hz	20 Hz	5 Hz	10 Hz	15 Hz	20 Hz	5 Hz	10 Hz	15 Hz	20 Hz
ML	32.7	46.5	45.0	44.6	38.9	43.1	42.4	44.0	41.7	44.5	41.0	42.5
ℓ_2-PML	24.9	32.3	37.0	39.5	32.3	35.3	38.9	45.9	31.9	41.0	41.5	45.5
HVB	24.2	30.5	33.0	32.6	32.2	29.5	27.7	33.4	31.1	36.8	28.4	32.2

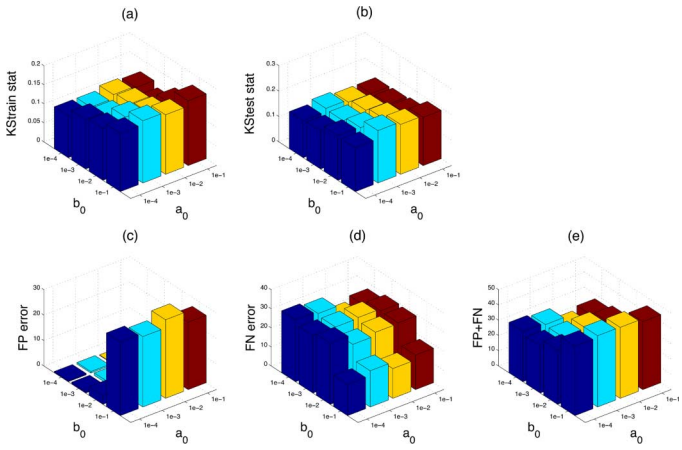


Fig. 5. Comparative performances (mean statistics averaged over 10 Monte Carlo runs) on the KS statistics (a), (b), FP error (c), FN error (d), and (FP+FN) error (e) with varying values of hyperprior parameters a_0 and b_0 in the HVB algorithm.

and misidentification error by varying different values of a_0 and b_0 in the HVB algorithm. In these simulations, the connectivity ratio was chosen to be 0.3 and the baseline firing rate was fixed at 10 Hz. As seen from Fig. 5, the performance of the HVB algorithm is relatively robust to a variety range of the hyperprior parameters. In this example, according to the averaged KStrain statistics (minimum 0.110), the optimal set up is $(a_0, b_0) = (10^{-3}, 10^{-3})$; according to the averaged KStest statistics (minimum 0.146), the optimal setup is $(a_0, b_0) = (10^{-4}, 10^{-4})$; according to the averaged (FP+FN) error rate (minimum 24.5%), the optimal setup is $(a_0, b_0) = (10^{-2}, 10^{-4})$. In practice, we have found that the range $(a_0, b_0) \in [10^{-4}, 10^{-2}]$ consistently achieved good performance.

Overall, it was found in our simulations (with various setup conditions) that in the presence of small connectivity ratio, high spiking rate and a large number of spiking data, all tested algorithms produce similar KS statistics and misidentification errors. In the other conditions, the HVB algorithm always has an obviously superior margin.

C. Real Spike Train Analysis

The best statistical inference method, the HVB algorithm, was then applied to real spike train data. This experimental data featured groups of neurons that were simultaneously recorded from the primary motor cortex (M1) of an awake, behaving cat sitting quietly in a Faraday cage. The experimental protocol for data acquisition, and the behavioral paradigm has been previously reported in detail [22]. In the current study, real neural data is used purely for demonstration purposes, and thus we used recordings from only one animal during a period of motor inactivity (i.e., baseline firing activity) so that neural firing rates were as low as possible (to purposely create sparse spiking data sets). Specific physiological details of the data used for this analysis are provided in Table IV.

Three independent recording sessions were used in this analysis. The first, second and third data sets consist of 13, 15 and 15 neurons, and 18, 18 and 17 trials, respectively, and each trial was 3000 ms in length. The M1 neurons in these data sets were classified as either regular-spiking (RS) or fast-spiking (FS) neurons based upon extracellular firing features. Many of the neurons in these data sets had very low firing rates during the trial periods (Table IV), and the short data recordings and low firing rate fit the two key criteria of the "small data problem." In Fig. 6, we show representative spike rasters and inter-spike interval (ISI) histograms from one RS and one FS neuron. To estimate the functional connectivity amongst the ensemble neurons, we have assumed that the spiking dependence among the cells remain unchanged across different trials.

We binned the spike trains with 1 ms temporal resolution, and obtained multivariate point process observations. From the observed ISI histograms, we chose the spiking history up to 100 ms

TABLE IV
SUMMARY OF REAL SPIKE TRAIN DATA FROM ENSEMBLE M1 NEURONS (DURING THE BASELINE PERIOD)

Dataset	# trials	# neurons (RS+FS)	min/median firing rate (Hz)	# neurons with firing rate below 10 Hz	fraction of neuronal interactions RS-RS	RS-FS	FS-FS
1	18	13 (6+7)	2.7 / 13.0	6	13/36	67/84	49/49
2	18	15 (9+6)	1.0 / 8.7	9	50/81	83/108	33/36
3	17	15 (11+4)	0.7 / 5.4	12	36/121	61/88	16/16

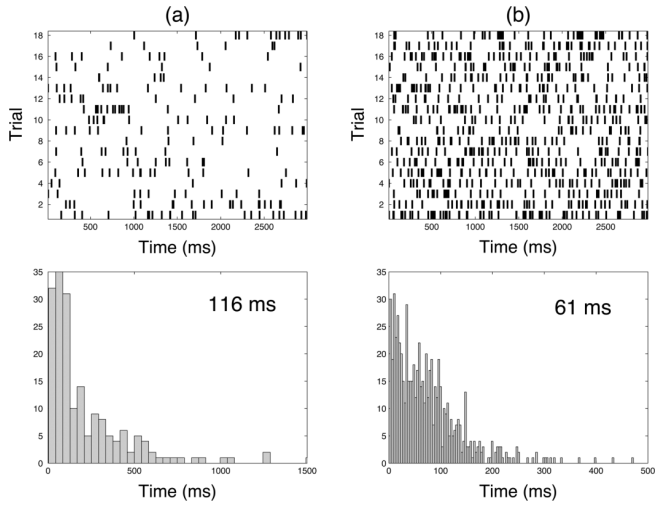


Fig. 6. Spike rasters and inter-spike interval (ISI) histograms of two representative M1 neurons (Dataset-1): (a) a RS neuron (mean firing rate: 3.7 Hz), (b) a FS neuron (mean firing rate: 13 Hz). The legends in the ISI histograms denote the median ISI values. One-sample KS test indicated that these two ISI samples are not exponential distributed ($P > 0.05$).

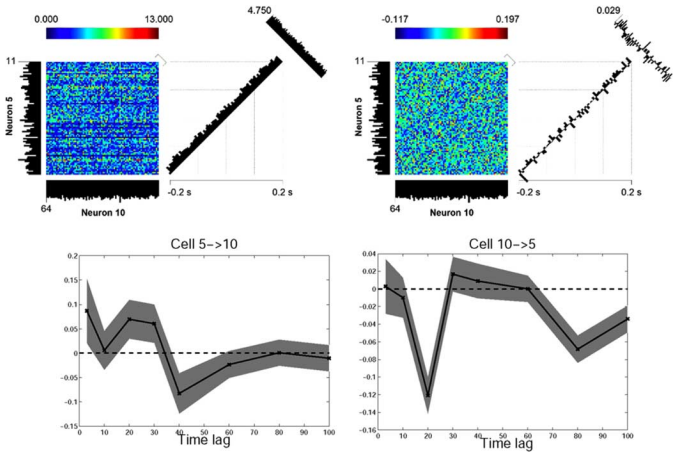


Fig. 7. Illustrations of pairwise neuronal spiking dependence between one RS neuron (#5) and one FS neuron (#10), using the raw (top left) and corrected (top right) joint peri-stimulus time histograms (JPSTHs) and the inferred bi-directional (bottom) spiking dependence based on the HVB algorithm. Here, A→B means that neuron A is a trigger cell, and neuron B is the target cell; the shaded area indicates the 95% confidence intervals of the estimates. An positive/negative value of the estimate implies an excitatory/inhibitory effect on the spiking of the target from the trigger cell. The corrected JPSTH (5 ms bin) did not detect any significant interactions between two neurons as the correlation coefficient is 0.029. In contrast, a significant excitatory/inhibitory RS→FS effect and a significant inhibitory FS→RS effect were detected by our method.

and selected eight history bins (unit: ms) with the following setup:

$$[1 \sim 3, 4 \sim 10, 11 \sim 20, 21 \sim 30, \\ 31 \sim 40, 41 \sim 60, 61 \sim 80, 81 \sim 100].$$

The first spiking history window is chosen to capture the refractory period of the spiking property. For a total of C neurons, the size of parameters for fitting each neuron's spike train recordings is $\dim(\boldsymbol{\beta}) = 8C + 1$. For the present problem, $\dim(\boldsymbol{\beta}) = 105$ (for $C = 13$) or $\dim(\boldsymbol{\beta}) = 121$ (for $C = 15$). For the inference of the HVB algorithm, the initial parameters were set as follows: $a_0 = b_0 = 10^{-3}$, and $\boldsymbol{\mu}_0 = \mathbf{0}$, although it was found that the results are insensitive to these values. Upon fitting all 43 neurons, 30 neurons' KS plots (Dataset-1: 8/13; Dataset-2: 12/15; Dataset-3: 10/15) are completely within the 95% confidence bounds, and 41 neurons' KS plots are within the 90% confidence bounds. These highly accurate fits indicate that the statistical model (3) can satisfactorily characterize the spiking activity of individual neurons. Using the HVB algorithm, the inferred (averaged) network connectivity ratios from 3 data sets are 0.27, 0.21, and 0.13, respectively. Amongst all three data sets, it was found that the fraction of neural interactions is the highest amongst the FS-FS cell-pairing, followed by FS-RS and RS-RS groups (Table IV), which supports previous studies investigating the network properties of M1 neurons [8]. The relatively lower connectivity ratio in Dataset-3 is due to a lower ratio of FS/RS neurons (Table IV). These results suggest that during periods of motor inactivity, FS neurons (inhibitory interneurons) are involved in more functional connectivity than RS neurons (pyramidal cells) in M1. Furthermore, a close examination of the neural interactions inferred by the HVB algorithm also reveals that many FS neurons have inhibitory effects on the spiking activity of RS neurons. Fig. 7 illustrates an example of such spiking dependence between one FS and one RS neuron—note that the inhibitory spiking dependence at a fine timescale was not detectable by a more traditional method (the JPSTH). Finally, the strengths of (absolute value of $\beta^c_{i,k}$ averaged over K time lags) neuronal interactions inferred from 3 data sets are shown in Fig. 8. In each plot, the size of the circle is proportional to the relative (normalized) strength respective to all neurons (including self-interactions).

VI. DISCUSSION AND CONCLUSION

Inferring functional connectivity of a neuronal network using simultaneously recorded spike trains is an important task in computational neuroscience, and the point process GLM provides a principled framework for statistical estimation. However, in addition to employing a favorable statistical model, the selection of an appropriate statistical inference algorithm is also a crucial undertaking. The present study aimed to solve a problem that has been present in the practice of experimental

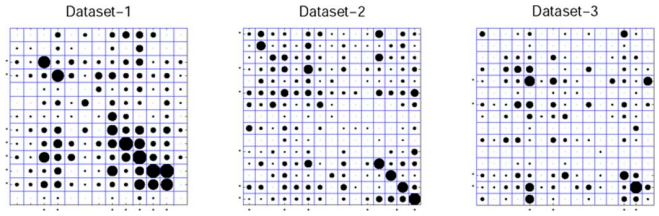


Fig. 8. Visualization of the strengths of neuronal interactions inferred from M1 neuronal assemblies during the baseline period. In each plot, the size of the circle is proportional to the relative strength (normalized such that the maximum strength is 1) respective to all neurons (including self-interactions); the symbol $*$ along the axes in each plot marks the FS neuron.

brain recording for many years: the reliability of sparse data sets for analysis [43]. Thus, this paper investigates several statistical inference procedures, while also applying these methods to the sparse spiking data. Many sets of experimental data are not appropriate for analysis of spiking dependence either because they feature neurons that are spiking at very low frequencies, or because during a difficult behavioral experiment, too few trials of the desired behavior are secured. Essentially, improving the reliability of the statistical estimates was the focus of our study, and simulated spike train data were used to compare different statistical inference algorithms. The four algorithms that were tested were the standard ML method, the ℓ_2 and ℓ_1 PML methods, and the HVB method. Systematic investigations were conducted to compare the performance of the algorithms (in terms of the KS statistics, MSE, and misidentification error) under different conditions (with varying baseline firing rate and network connectivity ratio). From the Monte Carlo estimation results we conclude that 1) the HVB algorithm performed the best amongst all tested algorithms, and 2) regularization is very important for the maximum likelihood estimation, especially in the presence of neurons with low firing rate. As an illustration, we apply the HVB algorithm to real spike train recordings from ensemble M1 neurons.

The hierarchical Bayesian method has been shown to be a powerful tool for statistical inference [18], whose applications have gone far beyond the point process GLM. The VB inference is appealing for Bayesian inference from the perspective of computational efficiency, with the goal of maximizing the lower bound of the marginal log-likelihood of observed data. The ARD principle employed in the Bayesian inference procedure provides a natural way for variable selection in that redundant or insignificant features will be shown to have smaller weights (close to zeros), thereby favoring a sparse solution. The sparsity of the solution is controlled by the hyperprior parameters, which can be set to be noninformative. Finally, the full posteriors of the parameters can be obtained, which can be used to compute a predictive distribution of unseen data [2].

The framework of the point process GLM and the HVB method provide a new way to investigate the neural interactions of ensemble neurons using simultaneously recorded spike trains. The point process GLM using the collective neuronal firing history has been shown to be effective in predicting single neuron spikes from humans and monkeys [49], which has potential applications for neural prosthetic devices. In addition, our proposed methodology can be used for assessing neural interactions. Since the point process GLM using a network likelihood function enabled us to assess spiking dependencies in populations of simultaneously recorded neurons, our approach is favorable when compared with traditional techniques (e.g., cross-correlation or JPSTH), as it may be used to examine functional connectivity as it occurs in multiple neurons simultaneously, compared with only being able to perform pairwise analysis. This is appealing for examining neural interactions at different regions of the brains, or for conducting quantitative comparison during different behaviors or task performances. The findings of our study indicates that the proposed HVB method provides a satisfactory solution to the "sparse spiking data problem" faced by many neuroscience researchers, and this method appears to outperform other contemporary statistical inference procedures in the assessment of functional connectivity in sets of spike train data where sparsity is not an issue.

Similar to other contemporary statistical approaches, our method for inferring the functional connectivity of ensemble neurons relies on certain statistical assumptions. For example, the stationarity of neuronal data during short durations of trials as well as across trials. While this assumption is not always valid between trials, the nonstationarity issue across trials can be addressed by considering a random-effects GLM [11], and maximum likelihood and Bayesian inference procedures can be developed accordingly.

Appendix A
Design of Desirable Positive-Definite Matrix $\boldsymbol{Q}$

Assuming that the spiking history dependent coefficients change smoothly between the neighboring windows), we may impose a "local smoothness" constraint on the parameters. Heuristically, when the parameter sequences $\{\beta_{c,k}\}$ are temporally smooth for any index c, the local variance will be relatively small. Let $\bar{\beta}_{c,k}$ denote the corresponding short-term exponentially weighted average of $\beta_{c,k}$

$$\bar{\beta}_{c,k} = \gamma\bar{\beta}_{c,k-1} + (1-\gamma)\beta_{c,k} \tag{23}$$

where $0 < \gamma < 1$ is a forgetting factor that determines the range of local smoothness. The role of γ is to act like a low-pass filter: the smaller the value of γ, the more emphasis is placed on the $\beta_{c,k}$, and a smaller smoothing effect emerges. Let us define a new quadratic penalty function from (23)

$$\sum_c \sum_k (\beta_{c,k} - \bar{\beta}_{c,k})^2 = \sum_c \sum_k \gamma(\beta_{c,k} - \bar{\beta}_{c,k-1})^2 \tag{24}$$

which penalizes the local variance of $\{\beta_{c,k}\}$. Let $\bar{\boldsymbol{\beta}}_c$ denote the short-term average vector for the corresponding parameter $\boldsymbol{\beta}_c = [\beta_{c,1}, \ldots, \beta_{c,K}]$, then we further have

$$\|\boldsymbol{\beta}_c - \bar{\boldsymbol{\beta}}_c\|^2 = \|\boldsymbol{\beta}_c - \mathbf{S}\boldsymbol{\beta}_c\|^2 \tag{25}$$

where a smoothing matrix $\mathbf{S}$ is introduced to represent $\bar{\boldsymbol{\beta}}_c$ in terms of $\boldsymbol{\beta}_c$. Note that the exponentially moving average $\bar{\beta}_{c,k}$ can be viewed as a *convolution product* between the sequences

$\{\beta_{c,k}\}$ and a template. Suppose the template vector has an exponential-decay property with length 4, such that $\text{template} = [(1-\gamma), \gamma(1-\gamma), \gamma^2(1-\gamma), \gamma^3(1-\gamma)]$. Note that the convolution smoothing operation can also be conveniently expressed as a matrix product operation: $\bar{\boldsymbol{\beta}}_c = \mathbf{S}\boldsymbol{\beta}_c$, where $\mathbf{S}$ is a Toeplitz matrix with the right-shifted template appearing at each row given as follows:

$$\mathbf{S} = \begin{bmatrix} 1-\gamma & 0 & 0 & 0 & \cdots & 0 \\ \gamma(1-\gamma) & 1-\gamma & 0 & 0 & \cdots & 0 \\ \gamma^2(1-\gamma) & \gamma(1-\gamma) & 1-\gamma & 0 & \cdots & 0 \\ \gamma^3(1-\gamma) & \gamma^2(1-\gamma) & \gamma(1-\gamma) & 1-\gamma & 0 & \cdots \\ 0 & \gamma^3(1-\gamma) & \gamma^2(1-\gamma) & \gamma(1-\gamma) & 1-\gamma & \cdots \\ & & \cdots & 0 & \ddots & \vdots \end{bmatrix}.$$

Finally, we obtain the regularization matrix $\boldsymbol{Q} = \mathbf{P}^\top\mathbf{P}$, where the matrix $\mathbf{P}$ has a block-Toeplitz structure

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{P}_2 & \cdots & 0 \\ 0 & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{P}_C \end{bmatrix} = \begin{bmatrix} \mathbf{I}-\mathbf{S} & 0 & \cdots & 0 \\ 0 & \mathbf{I}-\mathbf{S} & \cdots & 0 \\ 0 & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{I}-\mathbf{S} \end{bmatrix}. \tag{26}$$

where $\dim(\mathbf{I}-\mathbf{S}) = K, \dim(\boldsymbol{Q}) = KC$, and the number of blocks is equal to C. It is worth commenting that our smoothing operator can be seen as an extension of the contingent smoothness operator [46], where the term $(\beta_{c,k} - \bar{\beta}_{c,k})^2$ in (24) is replaced by $(\beta_{c,k} - \beta_{c,k-1})^2$ (i.e., the local mean $\bar{\beta}_{c,k}$ is replaced by its intermediate neighbor $\beta_{c,k-1}$ without using any moving averaging). Nevertheless, our regularization operator is more general and also accommodates [23] as a special case. Like ours, the regularization matrix $\boldsymbol{Q}$ in [23] also has a block-Toeplitz structure. Note that when $\gamma = 1$, $\mathbf{S}$ will be an all-zeros matrix, $\mathbf{P}$ and $\boldsymbol{Q}$ will become identity matrices, and our smoothed regularization will reduce to the standard "ridge regularization;" on the other hand, when $\gamma = 0$, $\mathbf{S}$ will be an identity matrix, $\mathbf{P}$ and $\boldsymbol{Q}$ will become all-zeros matrices, therefore no regularization is imposed. Hence, our approach $(0 < \gamma < 1)$ can be viewed as a *quantitative* choice between two extrema of no regularization $(\gamma = 0)$ and ridge regularization $(\gamma = 1)$.

As already mentioned, we may use the contingent smoothness operator as in [46], in which $\{\beta_{c,k}\}_{k=1}^K$ is viewed as a curve, and the first-order derivative of the curve is approximated by $\beta_{c,k} - \beta_{c,k-1}$. If we penalize the norm of the first-order derivative, the objective function is then written as

$$L_2(\boldsymbol{\beta}_c) = L(\boldsymbol{\beta}_c) - \rho \sum_k \|\beta_{c,k} - \beta_{c,k-1}\|^2 \tag{27}$$

and the ith block $(i = 1, \ldots, C)$ of the block-diagonal matrix $\mathbf{P}$ in (27) is derived as

$$\mathbf{P}_i = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & & 0 & 1 & -1 \end{bmatrix}$$

which has a banded-diagonal structure.

Appendix B
Derivation of (22)

From (18) and (21), we can derive the variational lower bound of the marginal log-likelihood (for notation simplicity, the dependence of $\boldsymbol{y}$ in all posteriors is made implicit)

$$\begin{aligned} \tilde{L} = {} & \mathbb{E}_{q(\boldsymbol{\beta})}[\log \tilde{p}(\boldsymbol{y}|\boldsymbol{\beta})] + \mathbb{E}_{q(\boldsymbol{\beta})q(\boldsymbol{\alpha})}[\log p(\boldsymbol{\beta}|\boldsymbol{\alpha})] \\ & - \mathbb{E}_{q(\boldsymbol{\beta})}[\log q(\boldsymbol{\beta})] \\ & + \mathbb{E}_{q(\boldsymbol{\alpha})}[\log p(\boldsymbol{\alpha})] - \mathbb{E}_{q(\boldsymbol{\alpha})}[\log q(\boldsymbol{\alpha})] \end{aligned} \tag{28}$$

where the individual terms in (28) are given by

$$\begin{aligned} & \mathbb{E}_{q(\boldsymbol{\beta})}[\log \tilde{p}(\boldsymbol{y}|\boldsymbol{\beta})] \\ & \quad = \frac{1}{2}\boldsymbol{\mu}_T^\top \boldsymbol{\Sigma}_T^{-1}\boldsymbol{\mu}_T - \frac{d+1}{2} + \frac{1}{2}\left(\boldsymbol{\mu}_T^\top \boldsymbol{A}_T \boldsymbol{\mu}_T + \operatorname{tr}(\boldsymbol{\Sigma}_T \boldsymbol{A}_T)\right) \\ & \qquad + \frac{1}{2}\sum_{t=1}^{T}\left(2\log\sigma(\xi_t) - \xi_t + 2\phi(\xi_t)\xi_t^2\right) \end{aligned} \tag{29}$$

$$\begin{aligned} & \mathbb{E}_{q(\boldsymbol{\beta})q(\boldsymbol{\alpha})}[\log p(\boldsymbol{\beta}|\boldsymbol{\alpha})] \\ & \quad = -\frac{d+1}{2}\log(2\pi) + \sum_{j=0}^{d}\frac{1}{2}\left(\psi(a_T) - \log b_{j,T}\right) \\ & \qquad - \frac{1}{2}\left(\boldsymbol{\mu}_T^\top \boldsymbol{A}_T \boldsymbol{\mu}_T + \operatorname{tr}(\boldsymbol{\Sigma}_T \boldsymbol{A}_T)\right) \end{aligned} \tag{30}$$

$$\begin{aligned} & \mathbb{E}_{q(\boldsymbol{\beta})}[\log q(\boldsymbol{\beta})] \\ & \quad = -\frac{d+1}{2}\left(1 + \log(2\pi)\right) + \frac{1}{2}\log|\boldsymbol{\Sigma}_T| \end{aligned} \tag{31}$$

$$\begin{aligned} & \mathbb{E}_{q(\boldsymbol{\alpha})}[\log p(\boldsymbol{\alpha})] \\ & \quad = \sum_{j=0}^{d}\Bigg(a_0 \log b_0 - \log\Gamma(a_0) + (a_0 - 1)(\psi(a_T) \\ & \qquad - \log b_{j,T}) - b_0\frac{a_T}{b_{j,T}}\Bigg) \end{aligned} \tag{32}$$

$$\begin{aligned} & \mathbb{E}_{q(\boldsymbol{\alpha})}[\log q(\boldsymbol{\alpha})] \\ & \quad = \sum_{j=0}^{d}\left((a_T - 1)\psi(a_T) - \log\Gamma(a_T) + \log b_{j,T} - a_T\right) \end{aligned} \tag{33}$$

where $\Gamma(\cdot)$ denotes the gamma function, and $\psi(\cdot)$ denotes the digamma function, which is defined as the logarithmic derivative of the gamma function. Other notations have been defined earlier following (21). Summarizing (29) through (33) yields (22). The pseudocode of the HVB algorithm is given below.

Algorithm 1 The HVB pseudocode

Initialize the hyperprior parameters a_0 and b_0, and set the initial parameter value to 0 (i.e., $\boldsymbol{\beta} = \mathbf{0}$).

while convergence criteria not met **do**

Evaluate the data-dependent variational parameters $\boldsymbol{\xi} = \{\xi_t\}$ for each data points $\boldsymbol{x}_t$ $(t = 1, \ldots, T)$.

Update the parameter variational posterior mean $\boldsymbol{\mu}_T$ and variance $\boldsymbol{\Sigma}_T$.

Update the noise precision hyperparameter $\mathbb{E}_{q(\boldsymbol{\alpha})}[\boldsymbol{A}]$.

Update the hyperprior parameters a_T and $b_{j,T}$ $(j = 0, \ldots, d)$.

Compute the variational lower bound of marginal log-likelihood $\tilde{L}$ (22).

end while

end

REFERENCES

[1] A. Aertsen, G. Gerstein, M. K. Habib, and G. Palm, "Dynamics of neuronal firing correlation: Modulation of 'effective connectivity'," *J. Neurophysiol.*, vol. 61, pp. 900–917, 1989.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[3] D. R. Brillinger and E. P. Villa, "Assessing connections in networks of biological neurons," in *The Practice of Data Analysis: Essays in Honor of John W. Turkey*. Princeton, NJ: Princeton Univ. Press, 1997, pp. 77–92.

[4] E. N. Brown, R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank, "The time-rescaling theorem and its application to neural spike data analysis," *Neural Computat.*, vol. 14, no. 2, pp. 325–346, 2002.

[5] E. N. Brown, R. Barbieri, U. T. Eden, and L. M. Frank, "Likelihood methods for neural data analysis," in *Computational Neuroscience: A Comprehensive Approach*, J. Feng, Ed. Boca Raton, FL: CRC Press, 2003, pp. 253–286.

[6] E. N. Brown, R. E. Kass, and P. P. Mitra, "Multiple neural spike train data analysis: State-of -the-art and future challenges," *Nat. Neurosci.*, vol. 7, no. 5, pp. 456–461, 2004.

[7] E. N. Brown, "Theory of point processes for neural systems," in *Methods and Models in Neurophysics*, C. C. Chow, Ed. *et al.* New York: Elsevier, 2005, pp. 691–727.

[8] Z. Chen, D. F. Putrino, D. E. Ba, S. Ghosh, R. Barbieri, and E. N. Brown, "A regularized point process generalized linear model for assessing the functional connectivity in the cat motor cortex," in *Proc. IEEE Eng. Med. Biol. Soc. Conf.*, Minneapolis, MN, 2009, pp. 5006–5009.

[9] Z. Chen, F. Kloosterman, M. A. Wilson, and E. N. Brown, "Variational Bayesian inference for point process generalized linear models in neural spike trains analysis," in *Proc. IEEE ICASSP'10*, Dallas, TX, 2010, pp. 2086–2089.

[10] E. Chornoboy, L. Schramm, and A. Karr, "Maximum likelihood identication of neural point process systems," *Biol. Cybern.*, vol. 59, pp. 265–275, 1988.

[11] G. Czanner, U. T. Eden, S. Wirth, M. Yanike, W. A. Suzuki, and E. N. Brown, "Analysis of between-trial and within-trial neural spiking dynamics," *J. Neurophy.*, vol. 99, pp. 2672–2693, 2008.

[12] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes*, 2nd ed. New York: Springer, 2003.

[13] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least-angle regression," *Ann. Stat.*, vol. 32, no. 2, pp. 407–499, 2004.

[14] S. Eldawlatly, R. Jin, and K. Oweiss, "Identifying functional connectivity in large scale neural ensemble recordings: A multiscale data mining approach," *Neural Computat.*, vol. 21, pp. 450–477, 2009.

[15] S. Eldawlatly, Y. Zhou, R. Jin, and K. Oweiss, "On the use of dynamic Bayesian networks in reconstructing functional neuronal networks from spike train ensembles," *Neural Computat.*, vol. 22, pp. 158–189, 2010.

[16] L. Fahrmeir and G. Tutz, *Multivariate Statistical Modelling Based on Generalized Linear Models*, 2nd ed. New York: Springer, 2001.

[17] J. Friedman, T. Hastie, and R. Tibshirani, "Regularized paths for generalized linear models via coordinate descent," *J. Stat. Software*, vol. 33, no. 1, 2010.

[18] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. Boca Raton, FL: Chapman & Hall/CRC, 2004.

[19] A. Genkin, D. D. Lewis, and D. Madigan, Large-scale Bayesian logistic regression for text categorization Rutgers Univ., New Brunswick, NJ, Tech. Rep., 2004 [Online]. Available: http://www.stat.rutgers.edu/~madigan/BBR/

[20] G. L. Gerstein and D. H. Perkel, "Simultaneous recorded trains of action potentials: Analysis and functional interpretation," *Science*, vol. 164, pp. 828–830, 1969.

[21] S. Gerwinn, J. H. Macke, M. Seeger, and M. Bethge, "Bayesian inference for spiking neuron models with a sparsity prior," in *Adv. Neural Info. Proc. Syst. (NIPS)*, J. C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 20, 2008, pp. 529–536.

[22] S. Ghosh, D. F. Putrino, B. Burro, and A. Ring, "Patterns of spatio-temporal correlations in the neural activity of the cat motor cortex during trained forelimb movements," *Somatosensory Motor Res.*, vol. 26, pp. 31–49, 2009.

[23] M. Hebiri, "Regularization with the smooth-lasso procedure," [Online]. Available: http://arxiv.org/abs/0803.0668 2008

[24] T. S. Jaakkola and M. I. Jordan, "Bayesian parameter estimation via variational methods," *Stat. Comput.*, vol. 10, pp. 25–37, 2000.

[25] R. E. Kass, V. Ventura, and E. N. Brown, "Statistical issues in the analysis of neuronal data," *J. Neurophysiol.*, vol. 94, pp. 8–25, 2005.

[26] B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 957–968, Jun. 2005.

[27] K. Koh, S.-J. Kim, and S. Boyd, "An interior-point method for large-scale ℓ_1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 8, pp. 1519–1555, 2007.

[28] P. Komarek, "Logistic Regression for Data Minding and High-Dimensional Classification," Ph.D. Thesis, Carnegie Mellon University, , 2004.

[29] P. McCullagh and J. Nelder, *Generalized Linear Models*, 2nd ed. London, U.K.: Chapman Hall, 1989.

[30] R. Neal, *Bayesian Learning for Neural Networks*. New York: Springer, 1996.

[31] D. Nykamp, "A mathematical framework for inferring connectivity in probabilistic neuronal networks," *Math. Biosci.*, vol. 205, pp. 204–251, 2007.

[32] M. Okatan, M. A. Wilson, and E. N. Brown, "Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity," *Neural Computat.*, vol. 17, pp. 1927–1961, 2005.

[33] M. Y. Park and T. Hastie, "An ℓ_1 regularization-path algorithm for generalized linear models," *J. R. Stat. Soc. B*, vol. 69, no. 4, pp. 659–677, 2007.

[34] Y. Pawitan, *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. New York: Oxford Univ. Press, 2001.

[35] D. H. Perkel, G. L. Gerstein, and G. P. Moore, "Neuronal spike trains and stochastic point processes. II. Simultaneous spike trains," *Biophy. J.*, vol. 7, pp. 419–440, 1967.

[36] F. Rigat, M. de Gunst, and J. van Pelt, "Bayesian modelling and analysis of spatio-temporal neuronal networks," *Bayesian Anal.*, vol. 1, no. 4, pp. 733–764, 2006.

[37] M. Schmidt, G. Fung, and R. Rosaless, Optimization methods for ℓ_1-regularization Dept. Comput. Sci., Univ. Wisconsin, Tech. Rep., 2009 [Online]. Available: http://pages.cs.wisc.edu/~gfung/GeneralL1/, URL:

[38] E. Schneidman, M. Berry, R. Segev, and W. Bialek, "Weak pair-wise correlations imply strongly correlated network states in a neural population," *Nature*, vol. 440, pp. 10007–10212, 2006.

[39] M. Seeger, S. Gerwinn, and M. Bethge, "Bayesian inference for sparse generalized linear models," in *Proc. ECML'07*, 2007, pp. 298–309.

[40] J. Shlens *et al.*, "The structure of multi-neuron firing patterns in primate retina," *J. Neurosci.*, vol. 26, pp. 8254–8266, 2006.

[41] I. H. Stevenson, J. M. Rebesco, L. E. Miller, and K. P. Körding, "Inferring functional connections between neurons," *Curr. Opin. Neurobiol.*, vol. 18, pp. 582–588, 2008.

[42] I. H. Stevenson, J. M. Rebesco, N. G. Hatsopoulos, Z. Haga, L. E. Miller, and K. P. Körding, "Bayesian inference of functional connectivity and network structure from spikes," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 17, no. 3, pp. 203–213, Jun. 2009.
[43] G. Strangman, "Detecting synchronous cell assemblies with limited data and overlapping assemblies," *Neural Computat.*, vol. 9, pp. 51–76, 1997.
[44] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. R. Stat. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.
[45] R. Tibshirani, "The Lasso for variable selection in the Cox model," *Stat. Med.*, vol. 16, pp. 385–395, 1997.
[46] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused Lasso," *J. R. Stat. Soc. B*, vol. 67, pp. 91–108, 2005.
[47] W. Truccolo, U. T. Eden, M. Fellow, J. D. Donoghue, and E. N. Brown, "A point process framework for relating neural spiking activity to spiking history, neural ensemble and covariate effects," *J. Neurophysiol.*, vol. 93, pp. 1074–1089, 2005.
[48] W. Truccolo and J. D. Donoghue, "Nonparametric modeling of neural point processes via stochastic gradient boosting regression," *Neural Computat.*, vol. 19, no. 3, pp. 672–705, 2007.
[49] W. Truccolo, L. R. Hochberg, and J. P. Donoghue, "Collective dynamics in human and monkey sensorimotor cortex: Predicting single neuron spikes," *Nat. Neurosci.*, vol. 13, pp. 105–111, 2010.
[50] K. J. Utikal, "A new method for detecting neural interconnectivity," *Biol. Cybern.*, vol. 76, pp. 459–470, 1997.
[51] M. C. Wiener, "An adjustment to the time-rescaling method for application to short-trial spike train data," *Neural Computat.*, vol. 15, pp. 2565–2576, 2003.

Zhe Chen (S'99–M'09–SM'10) received the Ph.D. degree in electrical and computer engineering from McMaster University, Canada, in 2005.

From 2001 to 2004, he worked as a research assistant in Adaptive Systems Laboratory (directed by Prof. S. Haykin) at McMaster University. During the summer in 2002, he worked as a summer intern in Bell Laboratories, Lucent Technologies, Murray Hill, NJ. After receiving the Ph.D. degree, he joined RIKEN Brain Science Institute in June 2005 and worked as a research scientist in the Laboratory of Advanced Brain Signal Processing (headed by Prof. Shun-ichi Amari and Prof. A. Cichocki). Since March 2007 he has been working in the Neuroscience Statistics Research Laboratory (directed by Prof. E. Brown) at Massachusetts General Hospital, Harvard Medical School as a Harvard Research Fellow. He is also a Research Affiliate in the Department of Brain and Cognitive Sciences at Massachusetts Institute of Technology (MIT). His main research interests include neural signal processing, neural and caridovascular engineering, machine learning, Bayesian modeling, and computational neuroscience. He is the leading author of the book "Correlative Learning: A Basis for Brain and Adaptive Systems" (Wiley, 2007). Since 2009 he became an Associate Editor of *Computational Intelligence and Neuroscience* and served as a Guest Editor for the special issue "Signal Processing for Neural Spike Trains."

Dr. Chen has received a number of scholarships and awards, including the 2002 IEEE Walter Karplus Student Summer Research Award from the Computational Intelligence Society. He is a member of Biomedical Engineering Society and the Society for Neuroscience.

David F. Putrino was born in Perth, Western Australia, on September 13, 1983. He received the B.Sc. degree in physiotherapy (with First Class Honors) from Curtin University of Technology, Perth, in 2004, and the Ph.D. degree in neuroscience from the University of Western Australia, Perth, in 2008.

For the periods of 2004-2008, he held clinical positions as a Physical Therapist in both the hospital and private practice setting, as well as an academic position at Curtin University as a Lecturer of Neuroanatomy. Following the completion of the Ph.D. degree, he accepted a post-doctoral fellowship, in Boston, MA, where he remained for two years studying the statistical modeling of neural data from 2009 to 2010, before accepting his present position as a Post-Doctoral Fellow at New York University where he is studying the role of the posterior parietal cortex in the production of voluntary reaching movements.

Dr. Putrino is a member of the Society for Neuroscience.

Soumya Ghosh received the M.B.B.S. degree from Madras University, Tamil Nadu, India, in 1980, and the Ph.D. degree in neuroscience from the Australian National University, Canberra, Australia, in 1988.

His major field of study is neural control of movement. He is Clinical Associate Professor at the Centre for Neuromuscular and Neurological Disorders, University of Western Australia. His past appointments include Senior Lecturer at Curtin University, and Lecturer at Queensland University. His current and past research interests include cortical control of movement and anatomy of cortical connections.

Dr. Ghosh is a member of the Australian Neuroscience Society, Australian and New Zealand Association of Neurology, Society for Neuroscience, and American Academy of Neurology.

Riccardo Barbieri (M'00–SM'08) was born in Rome, Italy, in 1967. He received the M.S. degree in electrical engineering from the University of Rome ''La Sapienza," Rome, Italy, in 1992, and the Ph.D. degree in biomedical engineering from Boston University, Boston, MA, in 1998.

He is currently Assistant Professor of Anaesthesia at Harvard Medical School, Massachusetts General Hospital and Research Affiliate at Massachusetts Institute of Technology. His main research interests are in the development of signal processing algorithms for analysis of biological systems. He is currently focusing his studies on application of multivariate and statistical models to characterize heart rate and heart rate variability as related to cardiovascular control dynamics, and on computational modeling of neural information encoding.

Emery N. Brown (M'01–SM'06–F'08) received the B.A. degree from Harvard College, the M.D. degree from Harvard Medical School, and the A.M. and Ph.D. degrees in statistics from Harvard University, Cambridge, MA.

He is presently Professor of Computational Neuroscience and Health Sciences and Technology at Massachusetts Institute of Technology (MIT) and the Warren M. Zapol Professor of Anaesthesia at Harvard Medical School and Massachusetts General Hospital. His research interests are in the study of mechanisms of general anesthesia in humans and in the use point process and state-space methods to develop algorithms for neural signal processing.

Dr. Brown is a Fellow of the American Statistical Association, a Fellow of the American Association for the Advancement of Science, a Fellow of the American Institute of Medical and Biological Engineering, a member of Institute of Medicine of the National Academies, a member of the Association of University Anesthesiologists, and a recipient of a 2007 National Institute of Health (NIH) Director's Pioneer Award.

ARTICLE **Communicated by Martin Nawrot**

Discrete- and Continuous-Time Probabilistic Models and Algorithms for Inferring Neuronal UP and DOWN States

Zhe Chen
zhechen@mit.edu
Neuroscience Statistics Research Laboratory, Department of Anesthesia and Critical Care, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114, U.S.A., and Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Sujith Vijayan
vijayan@post.harvard.edu
Program in Neuroscience, Harvard University, Cambridge, MA 02139, U.S.A., and Picower Institute for Learning and Memory, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Riccardo Barbieri
barbieri@neurostat.mit.edu
Neuroscience Statistics Research Laboratory, Department of Anesthesia and Critical Care, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114, U.S.A.

Matthew A. Wilson
mwilson@mit.edu
Picower Institute for Learning and Memory, RIKEN-MIT Neuroscience Research Center, Department of Brain and Cognitive Sciences and Department of Biology, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Emery N. Brown
enb@neurostat.mit.edu
Neuroscience Statistics Research Laboratory, Department of Anesthesia and Critical Care, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114, U.S.A., Harvard-MIT Division of Health Sciences and Technology, Cambridge, MA 02139, U.S.A., and Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

UP and DOWN states, the periodic fluctuations between increased and decreased spiking activity of a neuronal population, are a fundamental feature of cortical circuits. Understanding UP-DOWN state dynamics is important for understanding how these circuits represent and transmit information in the brain. To date, limited work has been done on

characterizing the stochastic properties of UP-DOWN state dynamics. We present a set of Markov and semi-Markov discrete- and continuous-time probability models for estimating UP and DOWN states from multiunit neural spiking activity. We model multiunit neural spiking activity as a stochastic point process, modulated by the hidden (UP and DOWN) states and the ensemble spiking history. We estimate jointly the hidden states and the model parameters by maximum likelihood using an expectation-maximization (EM) algorithm and a Monte Carlo EM algorithm that uses reversible-jump Markov chain Monte Carlo sampling in the E-step. We apply our models and algorithms in the analysis of both simulated multiunit spiking activity and actual multiunit spiking activity recorded from primary somatosensory cortex in a behaving rat during slow-wave sleep. Our approach provides a statistical characterization of UP-DOWN state dynamics that can serve as a basis for verifying and refining mechanistic descriptions of this process.

1 Introduction

1.1 Neuronal State and Recurrent Networks. The state of the neural system reflects the phase of an active recurrent network, which organizes the internal states of individual neurons into synchronization through recurrent network synaptic activity with balanced excitation and inhibition.[1] The neuronal state dynamics can be externally or internally driven. The externally driven dynamics results from either sensory-driven adaptation or encoding of sensory percept; the internally driven dynamics results from changes in internal factors, such as attention shift. Different levels of neuronal state also bring in the dynamics of state transition. Generally state transitions are network controlled and can be triggered by the activation of single cells, which are reflected by changes in their intracellular membrane conductance.

From a computational modeling point of view, two types of questions arise. First, how do neurons generate, maintain, and transit between different states? Second, given the neuronal (intracellular or extracellular) recordings, how can the neuronal states be estimated? The computational solutions to the first question emphasize the underlying neuronal physiology or neural mechanism, which we call mechanistic models, whereas the solutions to the second question emphasize the representation or interpretation of the data, which we call statistical models. In this article, we are taking the second approach to model a specific phenomenon regarding the neuronal state.

[1]The neuronal state sometimes can refer to a single cell level, during which neurons exhibit different lengths or durations of depolarizing shift (e.g., Fujisawa, Matsuki, & Ikegaya, 2005).

1.2 Neuronal UP and DOWN States. The notion of neuronal UP and DOWN states refers to the observation that neurons have two distinct subthreshold membrane potentials that are relevant for action potential (i.e., spike) generation. A neuron is said to be depolarized (or excited) if its intracelluar membrane potential is above the resting membrane potential threshold (around -70 to -80 mV) and is said to be hyperpolarized (or inhibited) if its membrane potential is below the threshold. When a sufficient level of excitation is reached, a spike is likely to occur. Essentially, membrane potential fluctuations define two states of the neocortex. The DOWN state defines a quiescent period during which little or no activity occurs, whereas the UP state corresponds to an active cortical state with depolarized membrane potentials and action potential firing driven by synaptic input. It was generally believed that the spontaneous UP and DOWN states are generated by a balance of excitatory and inhibitory neurons in recurrent networks (Haider, Duque, Hasentaub, & McCormick, 2006). In recent years, many neurophysiological studies have been reported regarding the neuronal UP and DOWN states, ranging from intracellular or extracellular recordings (e.g., Sanchez-Vives & McCormick, 2000; Haider, Duque, Hasentaub, Yu, & McCormick, 2007). The UP and DOWN states that are characterized by the cortical slow oscillation in intracellular membrane potentials are also reflected in extracellular recordings, such as local field potential (LFP) or electroencephalograph (EEG), single-unit activity or multiunit activity (MUA). In the literature, the UP and DOWN states have been characterized by examining extracellular LFP recordings (Sirota, Csicsvari, Buhl, & Buzsáki, 2003; Battaglia, Sutherland, & McNaughton, 2004; Wolansky, Clement, Peters, Palczak, & Dickson, 2006) in either the somatosensory cortex of anesthetized or awake animals (e.g., Haslinger, Ulbert, Moore, Brown, & Devor, 2006; Luczak, Barthó, Marguet, Buzsáki, & Harris, 2007) or the visual cortex of nonanesthetized animals during sleep (e.g., Ji & Wilson, 2007). Recently, attention has also turned to multiunit spike trains in an attempt to relate spike firing activities with EEG recordings (Ji & Wilson, 2007).

In order to examine the relationship between sleep and memory in rats or animals, simultaneous recordings are often conducted in the neocortex and hippocampus with the goal of studying the cortico-hippocampal circuit and the functional connectivity of these two regions while the animals perform different tasks. It has been reported (e.g., Volgushev, Chauvette, Mukovski, & Timofeev, 2006) that during the slow wave sleep (SWS), which is characterized by 0.5 to 2.0 Hz slow oscillations (Buzsáki, 2006), neorcortical neurons undergo near-synchronous transitions, every second or so, between UP and DOWN states. The process of the alternating switch between the two states appears to be a network phenomenon that originates in the neocortex (Ji & Wilson, 2007; Vijayan, 2007).

The work reported here was driven by the experimental data accumulated in our lab (Ji & Wilson, 2007; Vijayan, 2007). The growing interest in UP and DOWN states in the neuroscience literature motivated us to develop

probabilistic models for the UP and DOWN modulated MUA. Specifically, the UP-DOWN states are modeled as a latent two-state Markovian (or semi-Markovian) process (Battaglia et al., 2004), and the modeling goal is to establish the probability for state transition or the probability density of UP or DOWN state duration and the likelihood model that takes into account both a global hidden state variable and individual history dependence of firing. In comparison with the standard and deterministic threshold-based method, our proposed stochastic models provide a means for representing the uncertainty of state estimation given limited experimental recordings.

1.3 Markov and Semi-Markov Processes and Hidden Markov Models. A stochastic process is said to be Markovian if it satisfies the Markov property; the knowledge of the previous history of states is irrelevant for the current and future states. A Markov chain is a discrete-time Markov process with the Markov property. The Markov process and Markov chain are both "memoryless." A Markov process or Markov chain contains either continuous-valued or finite discrete-valued states. A discrete-state Markov process contains a finite alphabet set (or finite state space), with each element representing a distinct discrete state. The change of the state is called the transition, and the probability of changing from one state to the other is called the transition probability. For the Markov chain, the current state has only finite-order dependence on the previous states. Typically the first-order Markov property is assumed; in this case, the probability of S_{k+1} being in a particular state at time $k+1$, given knowledge of states up to time k, depends on the state S_k at time k, namely, $\Pr(S_{k+1} \mid S_0, S_1, \ldots, S_k) = \Pr(S_{k+1} \mid S_k)$. A semi-Markov process (the Markov renewal process) extends the continuous-time Markov process to the condition that the interoccurrence times are not exponential.

When the state space is not directly observable, a Markov process is called hidden or latent. The so-called hidden Markov process is essentially a probabilistic function of the stochastic process (for a review, see Ephraim & Merhav, 2002). In the discrete-time context, the hidden Markov model (HMM) is a probabilistic model that characterizes the hidden Markov chain. The HMM is a generative model in that its full model $\{\boldsymbol{\pi}, \boldsymbol{P}, \boldsymbol{B}\}$ (where π denotes the initial state probability, $\boldsymbol{P}$ denotes the transition probability, and $\boldsymbol{B}$ denotes the emission probability) completely characterizes the underlying probabilistic structure of the Markov chain. Generally several conditions are assumed in the standard HMM: (1) the transition and emission probabilities are stationary or quasi-stationary; (2) the observations, either continuous or discrete valued, are assumed to be identically and independently distributed (i.i.d.); and (3) the model generally assumes a first- or finite-order Markov property. In the literature, there are several methods to tackle the inference problem in the HMM. One (and maybe the most popular) approach is rooted in maximum likelihood estimation. A particular solution is given by the expectation-maximization (EM) algorithm (Dempster, Laird,

& Rubin, 1977), which attempts to solve the missing data problem in the statistics literature. This turns out to be also equivalent to the Baum-Welch algorithm proposed by Baum and Welch and colleagues (Baum, Petrie, Soules, & Weiss, 1970; Baum, 1972). The Baum-Welch algorithm contains a forward-backward procedure (E-step) and reestimation (M-step), and it iteratively increases the likelihood of the incomplete data until the local maximum or the stationary point of the likelihood function is reached. Another inference method is rooted in Bayesian statistics. The Bayesian inference for HMM defines the prior probability for the unknown parameters (including the number of state) and attempts to estimate their posterior distributions. Since the posterior distribution is usually analytically intractable, a numerical approximation method is also used. The Markov chain Monte Carlo (MCMC) algorithms try to simulate a Markov chain to approach the equilibrium of the posterior distribution. The Metropolis-Hastings algorithm is a general MCMC procedure to simulate a Markov or semi-Markov chain. When the state space is transdimensional (this problem often arises from model selection in statistical data analysis), the reversible-jump MCMC (RJMCMC) methods (Green, 1995; Robert, Rydén, & Titterington, 2000) have also been developed. Due to the development of efficient inference algorithms, HMM and its variants have been widely used in speech recognition, communications, bioinformatics, and many other applications (e.g., Rabiner, 1989; Durbin, Eddy, Krough, & Mitchison, 1998).

1.4 Point Process and Cox Process. A point process is a continuous-time stochastic process with observations being either 0 or 1. Spike trains recorded from either single or multiple neurons are point processes. We will give a brief mathematical background for point process in a later section and refer the reader to Brown (2005) and Brown, Barbieri, Eden, and Frank (2003) for the complete and rigorous mathematical details of point processes in the context of computational neuroscience treatment. An important feature of spike trains is that the point process observations are not independently distributed; in other words, the current observation (either 0 or 1) is influenced by the previous spiking activities. This type of history dependence requires special attention for probabilistic modeling of the point process.

A Cox process is a doubly stochastic process, which defines a generalization of Poisson process (Cox & Isham, 1980; Daley & Vere-Jones, 2002). Specifically, the time-dependent conditional intensity function (CIF), often denoted as λ_t, is a stochastic process by its own.[2] A representative example of the Cox process is the Markov-modulated Poisson process, which has a state-dependent Poisson rate parameter.

[2]The CIF is also known as the *hazard rate function* in survival analysis. The value $\lambda_t \Delta$ measures the probability of a failure or death of an event in $[t, t + \Delta)$ given the process has survived up to time t.

1.5 Overview of Relevant Literature. Hidden Markov processes have a rich history of applications in biology. Tremendous effort has been devoted to modeling ion channels as discrete- or continuous-time Markov chains; several inference algorithms were developed for these models (Chung, Krishnamurthy, & Moore, 1991; Fredkin & Rice, 1992; Ball, Cai, Kadane, & O'Hagan, 1999). However, the observations used in ion-channel modeling are continuous, and the likelihood is often modeled by a gaussian or gaussian mixture distribution.

For discrete observations, Albert (1991) proposed a two-state Markov mixture model of a counting Poisson process and provided a maximum likelihood estimate (MLE) for the parameters. A more efficient forward-backward algorithm was later proposed by Le, Leroux, and Puterman (1992) with the same problem setup. In these two models, the two-state Markov transition probability is assumed to be stationary; although Albert also pointed out the possibility of modeling nonstationarity, no exact algorithm was given. In addition, efficient EM algorithms have been developed for discrete- or continuous-time Markov-modulated point processes (Deng & Mark, 1993; Rydén, 1996; Roberts, Ephraim, & Dieguez, 2006), but applying them to neural spike trains is not straightforward.

In the context of modeling neural spike trains, many authors (e.g., Radons, Becker, Dülfer, & Krüger, 1994; Abeles et al., 1995; Gat, Tishby, & Abeles, 1997; Jones, Fontanini, Sadacca, & Katz, 2007; Achtman et al., 2007; Kemere et al., 2008) used HMM for the purpose of analyzing and classifying the patterns of neural spike trains, but their models are restricted to discrete time and the Markov chain is homogeneous (i.e., the transition probability is stationary). In these studies, the hidden states are discrete, and the spike counts were used as the discrete observations for the likelihood models. Smith and Brown (2003) extended the standard linear state-space model (SSM) with continuous state and observations to an SSM with a continuous state Markov-modulated point process, and an EM algorithm was developed for the hidden state estimation problem. Later the theory was extended to the SSM with mixed continuous, binary, and point process observations (Coleman & Brown, 2006; Prerau et al., 2008; Eden & Brown, 2008), but the latent process was still limited to the continuous-valued state. In a similar context, Danóczy and Hahnloser (2006) also proposed a two-state HMM for detecting the "singing-like" and "awake-like" states of sleeping songbirds with neural spike trains; their model assumes a continuous-time Markov chain (with the assumption of knowing the exact timing of state transitions), and the sojourn time follows an exponential distribution; in addition, the CIF of the point process was assumed to be discrete in their work. All of these restricted assumptions have limited the computational model for analyzing real-world spike trains. Recently, more modeling efforts have been dedicated to estimating the hidden state and parameters using an HMM (or its variants) for estimating the stimulus-response neuronal model (Jones et al., 2007; Escola & Paninski, 2008). Xi and Kass (2008) recently also used

a RJMCMC method to characterize the bursty and nonbursty states from goldfish retinal neurons.

In modeling the hidden semi-Markov processes or semi-Markov chains, in which the sojourn time is no longer exponentially distributed, Guon (2003) developed an EM algorithm for a hidden semi-Markov chain with finite discrete-state sojourn time, but the computational complexity of the EM algorithm is much greater than the conventional HMM.[3]

1.6 Contribution and Outline. In this article, with the goal of estimating the population neuron's UP or DOWN state, we propose discrete-state Markov or semi-Markov probabilistic models for neural spikes trains, which are modeled as doubly stochastic point processes. Specifically, we propose discrete-time and continuous-time SSMs and develop the associated inference algorithms for tackling the joint (state and parameter) estimation problem.

Our contributions have three significant distinctions from the published literature: (1) the point-process observations are not i.i.d. Specifically, the rate parameters or the CIFs of the spike trains are modulated by a latent discrete-state variable and past spiking history. (2) In the continuous-time probabilistic models, the state transition is not necessarily Markovian; in other words, the hidden state is semi-Markovian in the sense that the sojourn time is no longer exponentially distributed. (3) The maximum likelihood inference algorithms are derived for discrete-time and continuous-time probabilistic models for estimating the neuronal UP or DOWN states, and the proposed Monte Carlo EM (MCEM) algorithm is rooted in a RJMCMC sampling method and is well suited for various probabilistic models of the sojourn time.

The rest of the article is organized as follows. In section 2, we present the discrete-time HMM and the EM-based inference algorithm. In section 3, we develop the continuous-time probabilistic Markov and semi-Markov chains and their associated inference algorithms. In section 4, we demonstrate and validate our proposed models and inference algorithms with both simulated and real-world spike train data. We present some discussions in section 5, followed by the conclusion in section 6.

1.7 Notation. In neural spike analysis, we examine spike trains from either single or multiunit activity. Due to digitalized recordings, we assume that the time interval $[0, T]$ of continuous-time neural spike train observations is properly discretized with a small time resolution Δ, so the time indexes are discrete integers within $k \in [1, K]$, such that $k\Delta \in ((k-1)\Delta, k\Delta]$

[3] In the worst case, the complexity is $\mathcal{O}(NT(N+T))$ in time, in contrast to $\mathcal{O}(N^2T))$ for the HMM, where N denotes the number of discrete state and T denotes the total length of sequences.

and $K\Delta = T$. Let $N^c_{t_k} \equiv N^c_k$ denote the counting process for spike train c at time t_k, and let $dN^c_{t_k} \equiv dN^c_k$ denote the indicator variable for 0/1 observation: $dN^c_k = 1$ if there is a spike and 0 otherwise. Other notations are rather straightforward, and we will define them in the proper places. Most notations used in this article are summarized in Table 1.

2 Discrete-Time Markov Modulated Probabilistic State-Space Model

To infer the neuronal UP and DOWN states, in this section we develop a simple, discrete-time Markov modulated state-space model that can be viewed as a variant of the standard HMM applied to spike train analysis. The underlying probabilistic structure is Markovian and homogeneous, and the inference algorithm is efficient in identifying the statistics of the hidden state process. Based on that, in the next section we develop a continuous-time probabilistic model in order to overcome some of limitations imposed by this discrete-time probabilistic model.

2.1 Hidden Markov Model. Let us consider a discrete-time homogeneous Markov chain. By discrete time, we assume that the time is evenly discretized into fixed-length intervals, which have time indices $k = 1, \ldots, K$. The neuronal UP or DOWN state, which is characterized by a latent discrete-time first-order Markov chain, is unobserved (and therefore hidden), and the observed spike trains or the spike counts recorded from the MUA are functionally determined by the hidden state. The standard HMM is characterized by three elements: transition probability, emission probability,[4] and initial state probability (Rabiner, 1989). At the first approximation, we assume that the underlying latent process follows a two-state HMM with stationary transition and emission probabilities.

- The initial probability of state is denoted by a vector $\boldsymbol{\pi} = \{\pi_i\}$, where $\pi_i = \Pr(S_0 = i)$ $(i = 0, 1)$. Without loss of generality, we assume that the amplitude of the hidden state is predefined, and the discrete variable $S_k \in \{0, 1\}$ indicates either a DOWN (0) or UP (1) state.
- The transition probability matrix is written as

$$\boldsymbol{P} = \begin{pmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{pmatrix}, \tag{2.1}$$

  with $P_{01} = 1 - P_{00}$ and $P_{10} = 1 - P_{11}$ corresponding to the transition probabilities from state 0 to state 1 and from state 1 to state 0, respectively.

[4]The term *emission probability* arose from the HMM literature in the context of speech recognition; it refers to the probability of observing a (finite) symbol given a hidden state (finite alphabet).

Table 1: Summary of Notation.

c	index of spike trains $c = 1, 2, \ldots, C$
m	index of simulated Markov chains $m = 1, 2, \ldots, M$
t	continuous-time index $t \in [0, T]$
t_i	spike timing of the ith spike in continuous time
Δ	smallest time bin size
k	discrete-time index $k = 1, 2, \ldots, K$, $K\Delta = T$
y_k	number of counts observed from discrete-time Markov chain, $y_k \in \{0, \mathbb{N}\}$
S_k	discrete-time first-order Markov state, $S_k \in \{1, \ldots, L\}$
S_0	initial Markov state at time 0
$S_{0:T}$, $S_{1:k}$	history of the Markov state from time 0 to T (or 1 to k)
n	number of state jumps within the latent process $S_{0:T}$
l	index of state jumps $l = 1, 2, \ldots, n$
$\{S(t); 0 \le t \le T\}$	realization of hidden Markov process
$\mathcal{S} = (n, \boldsymbol{\tau}, \boldsymbol{\chi})$	triplet that contains all information of continuous-time Markov chain $\{S(t)\}$
$\boldsymbol{\tau} = (\tau_0, \ldots, \tau_n)$	$(n+1)$-length vector of the sojourn times of $\{S(t)\}$
$\boldsymbol{\chi} = (\chi_0, \ldots, \chi_n)$	$(n+1)$-length vector of visited states in the sojourn times of $\{S(t)\}$
$\mathcal{S}^{(0)}$	initial state of MCMC sampler
ν_l	$\nu_0 = 0$, $\nu_l = \sum_{r=0}^{l-1} \tau_r$ $(l = 1, 2, \ldots, n+1)$
$\mathcal{H}_{0:T}$, $\mathcal{H}_{1:K}$	history of point-process observations from time 0 to T (or 1 to k)
$N(t)$, N_k	counting process in continuous and discrete time, $N(t)$, $N_k \in \{0, \mathbb{N}\}$
$dN(t)$, dN_k	indicator of point-process observations, 0 or 1
P_{ij}	transition probability from state i to j for a discrete-time Markov chain, $\sum_j P_{ij} = 1$
q_{ij}	transition rate from state i to j for a continuous-time Markov chain, $\sum_j q_{ij} = 0$
$r_i = q_{ii}$	total transition rate of state i for a continuous-time Markov chain, $r_i = \sum_{j \ne i} q_{ij}$
π_i	initial prior probability $\Pr(S_0 = i)$
$a_k(i)$	forward message of state i at time k
$b_k(i)$	backward message of state i at time k
$\gamma_k(i)$	marginal conditional probability $\Pr(S_k = i \mid \mathcal{H}_{0:T})$
$\xi_k(i, j)$	joint conditional probability $\Pr(S_{k-1} = i, S_k = j \mid \mathcal{H}_{0:T})$
$\mathcal{L}$	log likelihood of the complete data
$R(\mathcal{S} \to \mathcal{S}')$	proposal transition density from state $\mathcal{S}$ to $\mathcal{S}'$
$\mathcal{B} = \mathcal{B}_1\mathcal{B}_2\mathcal{B}_3$	prior ratio × likelihood ratio × proposal probability ratio
$\mathcal{A}$	acceptance probability, $\mathcal{A} = \min(1, \mathcal{B})$
$\mathcal{J}$	Jacobian
λ_k	conditional intensity function of the point process at time k
$\boldsymbol{\theta}$	parameter vector that contains all unknown parameters
$p(x)$	probability density function
$F(x)$	cumulative distribution function, $F(x) = \int_{-\infty}^{x} p(z)\,dz$
$\Phi(x)$	gaussian cumulative distribution function
$\mathrm{erf}(x)$	error function
$\mathbb{I}(\cdot)$	indicator function
$\mathcal{U}(a, b)$	uniform distribution within the region (a, b)

- Given the discrete hidden state S_k, the observed numbers of total spikes across all tetrodes (i.e., MUA), $y_1, y_2, \ldots, y_K$ ($y_k \in \mathbb{N}$), follow probability distributions that depend on the time-varying rate λ_k,

$$\Pr(Y_k = y_k \mid S_k = i) = \frac{e^{-\lambda_k}\lambda_k^{y_k}}{y_k!}, \tag{2.2}$$

  where the parameter λ_k is determined by

$$\lambda_k = \exp(\mu + \alpha S_k + \beta(N_{k-1} - N_{k-J})), \tag{2.3}$$

  where $\exp(\mu)$ denotes the baseline firing rate and S_k denotes the hidden discrete-state variable at time k. The term $N_{k-1} - N_{k-J}$ represents the total number of spikes observed during the history period $(k-J, k-1]$, which accounts for the history dependence of neuronal firing. The choice of the length of history dependence is often determined empirically based on the preliminary data analysis, such as the histogram of the interspike interval (ISI). Equations 2.2 and 2.3 can be understood in terms of a generalized linear model (GLM) (e.g., McCullagh & Nelder, 1989; Truccolo, Eden, Fellow, Donoghue, & Brown, 2005), where the link function is a log function and the distribution is Poisson. Note that when $\beta = 0$ (i.e., history independence is assumed), we obtain an inhomogeneous Poisson process, and λ_k reduces to a Poisson rate parameter.

Taking the logarithm to both sides of equation 2.2, equation 2.3 can be rewritten as

$$\log \lambda_k = \mu + \alpha S_k + \beta \tilde{n}_k, \tag{2.4}$$

where $\tilde{n}_k = N_{k-1} - N_{k-J}$. More generally, we can split the time period $(k-J, k-1]$ into several windows (say, with equal duration δ), and equation 2.4 can be rewritten as

$$\begin{aligned}\log \lambda_k &= \mu + \alpha S_k + \sum_j \beta_j \tilde{n}_{k,j} \\ &= \mu + \alpha S_k + \boldsymbol{\beta}^T \tilde{\boldsymbol{n}}_k,\end{aligned} \tag{2.5}$$

where $\boldsymbol{\beta} = \{\beta_j\}$ and $\tilde{\boldsymbol{n}}_k = \{\tilde{n}_{k,j}\}$ are two vectors with proper dimensionality, and $\tilde{n}_{k,j} = N_{k-j\delta} - N_{k-(j+1)\delta}$ denotes the observed number of multiunit spike counts within the time interval $(k-(j+1)\delta, k-j\delta]$. If we further assume that the observations y_k at different time indices k are mutually independent, the observed data likelihood is given by

$$p(y_{1:K} \mid S_{1:K}, \boldsymbol{\theta}) = \prod_{k=1}^{K} \frac{\exp(-\lambda_k)\lambda_k^{y_k}}{y_k!}. \tag{2.6}$$

Note that $\lambda_k \equiv \lambda(S_k)$ is functionally dependent on the latent process S_k, although we have omitted it from the notation for brevity. In statistics, the hidden variables $\{S_k\}$ are treated as the missing data, $\{y_k\}$ as the observed (incomplete) data, and their combination $\{S_k, y_k\}$ as the complete data. Let $\boldsymbol{\theta}$ denote all of the unknown parameters; then the complete data likelihood is given by

$$p(S_{1:K}, y_{1:K} \mid \boldsymbol{\theta}) = p(y_{1:K} \mid S_{1:K}, \boldsymbol{\theta}) p(S_{1:K} \mid \boldsymbol{\theta}). \tag{2.7}$$

And the complete data log likelihood, denoted as $\mathcal{L}$, is derived as (by ignoring the constant)

$$\begin{aligned} \mathcal{L} = \log p(S_{0:K}, y_{1:K} \mid \boldsymbol{\theta}) = & \sum_{k=1}^{K} (y_k \log \lambda_k - \lambda_k) + \sum_{i=0}^{1} i \log \pi_i \\ & + \sum_{k=2}^{K} \sum_{i=0}^{1} \sum_{j=0}^{1} \xi_k(i, j) \log P_{ij}, \end{aligned} \tag{2.8}$$

where $\xi_k(i, j) = \Pr(S_{k-1} = i, S_k = j)$.

2.2 Forward-Backward and Viterbi Algorithms. The inference and learning procedure for the standard HMM is given by an efficient estimation procedure known as the EM algorithm, which is also known as the Baum-Welch algorithm (Baum et al., 1970; Baum, 1972). Rooted in maximum likelihood estimation, the EM algorithm iteratively and monotonically maximizes (or increases) the log-likelihood function given the incomplete data (Dempster et al., 1977). In the E-step, a forward-backward procedure is used to recursively estimate the hidden state posterior probability. In the M-step, based on the missing state statistics (estimated from the E-step), the reestimation procedure and Newton-Ralphson algorithm are used to estimate the unknown parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{P}, \mu, \alpha, \beta)$. In each full iteration, the EM algorithm iteratively maximizes the so-called Q-function,

$$\begin{aligned} Q(\boldsymbol{\theta}^{new} \mid \boldsymbol{\theta}^{old}) &= \mathbb{E}[\log p(\hat{S}_{1:K}, y_{1:K} \mid \boldsymbol{\theta}) \mid \boldsymbol{\theta}^{old}] \\ &= \mathbb{E}\Bigg[\sum_{k=1}^{K} (y_k \log \hat{\lambda}_k - \hat{\lambda}_k) + \sum_{i=0}^{1} i \log \hat{\pi}_i \\ &\quad + \sum_{k=2}^{K} \sum_{i=0}^{1} \sum_{j=0}^{1} \xi_k(i, j) \log \hat{P}_{ij} \mid \boldsymbol{\theta}^{old} \Bigg]; \end{aligned} \tag{2.9}$$

the new $\boldsymbol{\theta}^{new}$ is obtained by maximizing the incomplete data likelihood conditional on the old parameters $\boldsymbol{\theta}^{old}$; and the iterative optimization procedure

continues until the algorithm ultimately converges to a local maximum or a stationary point. For the self-containing purpose, we present a brief derivation of the EM algorithm (Rabiner, 1989) for the two-state HMM estimation problem.

2.2.1 E-Step: Forward-Backward Algorithm. In the E-step, the major task of the forward-backward procedure is to compute the conditional state probabilities for the two states:

$$\begin{aligned}\Pr(S_k = 1 \mid y_{1:K}, \boldsymbol{\theta}) &= \frac{\Pr(y_{1:K}, S_k = 1 \mid \boldsymbol{\theta})}{\Pr(y_{1:K} \mid \boldsymbol{\theta})} \\ &= \frac{\Pr(y_{1:K}, S_k = 1 \mid \boldsymbol{\theta})}{\sum_{l=0}^{1} \Pr(y_{1:K}, S_k = l \mid \boldsymbol{\theta})} \qquad (2.10)\\ \Pr(S_k = 0 \mid y_{1:K}, \boldsymbol{\theta}) &= 1 - \Pr(S_k = 1 \mid y_{1:K}, \boldsymbol{\theta}), \qquad (2.11)\end{aligned}$$

as well as the conditional state joint probability:

$$\begin{aligned}&\Pr(S_{k-1} = i, S_k = j \mid y_{1:K}, \boldsymbol{\theta}) \\ &= \frac{\Pr(y_{0:K}, S_{k-1} = i, S_k = j \mid \boldsymbol{\theta})}{\Pr(y_{1:K} \mid \boldsymbol{\theta})} \\ &= \frac{\Pr(y_{0:K}, S_{k-1} = i, S_k = j \mid \boldsymbol{\theta})}{\sum_{l=0}^{1} \sum_{m=0}^{1} \Pr(y_{1:K}, S_{k-1} = l, S_k = m \mid \boldsymbol{\theta})}. \qquad (2.12)\end{aligned}$$

To make the notation simple, in the derivation below, we let the conditional $\boldsymbol{\theta}$ be implicit in the equation.

To estimate equations 2.10 and 2.11, we first factorize the joint probability as

$$\begin{aligned}\Pr(y_{1:K}, S_k = l) &= \Pr(y_{1:k}, S_k = l) \Pr(y_{k+1:K} \mid y_{1:k}, S_k = l) \\ &= \Pr(y_{1:k}, S_k = l) \Pr(y_{k+1:K} \mid y_{1:k} S_k = l) \\ &\equiv a_k(l) b_k(l) \quad \text{for } l = 0, 1, \qquad (2.13)\end{aligned}$$

where

$$\begin{aligned}a_k(l) &= \Pr(y_{1:k}, S_k = l) \quad \text{for } k = 2, \ldots, K \\ a_1(l) &= \Pr(y_{1:k}, S_k = l) \Pr(y_1 \mid S_1 = l) \\ b_k(l) &= \Pr(y_{k+1:K} \mid S_k = l) \quad \text{for } k = 1, \ldots, K-1 \\ b_1(l) &= 1,\end{aligned}$$

and the forward and backward messages $a_k(l)$ and $b_k(l)$ can be computed recursively along the discrete-time index k (Rabiner, 1989):

$$\begin{aligned} a_k(l) &= \sum a_{k-1}(i) P_{il} \Pr(y_k \mid S_k = l) \\ &= \sum a_{k-1}(i) P_{il} \frac{\exp(-\lambda_k)\lambda_k^{y_k}}{y_k!} \\ b_k(l) &= \sum b_{k+1}(i) P_{li} \Pr(y_{k+1} \mid S_{k+1} = i) \\ &= \sum b_{k+1}(i) P_{li} \frac{\exp(-\lambda_{k+1})\lambda_{k+1}^{y_{k+1}}}{y_{k+1}!}, \end{aligned}$$

where P_{il} denotes the transition probability from state i to l.

Given $\{a_k, b_k\}$, we can estimate equation 2.12 by

$$\Pr(S_{k-1} = i, S_k = j \mid y_{1:K}, \boldsymbol{\theta}) = a_k(i) P_{ij} \Pr(y_{k+1} \mid S_{k+1} = j) b_{k+1}(j). \quad (2.14)$$

Furthermore, we can compute the observed likelihood (of the incomplete data) by

$$p(y_{1:K}) = \sum_{l=0}^{1} \Pr(y_{1:K}, S_K = l) = \sum_{l=0}^{1} a_K(l). \quad (2.15)$$

Given equations 2.13 and 2.15, the state posterior conditional probability is given by Bayes' rule,

$$\Pr(S_k = i \mid y_{1:K}) = \frac{\Pr(S_k = i, y_{1:K})}{p(y_{1:K})} = \frac{a_k(i) b_k(i)}{\sum_{l=0}^{1} a_K(l)}. \quad (2.16)$$

In the term of the computational overhead for the above-described two-state HMM, the forward-backward procedure requires a linear order of computational complexity $\mathcal{O}(4K)$ and memory storage $\mathcal{O}(2K)$.

2.2.2 M-Step: Reestimation and Newton-Ralphson Algorithm. In the M-step, we update the unknown parameters (based on their previous estimates) by setting the partial derivatives of the Q-function to zeros: $\frac{\partial Q(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$, from which we may derive either closed-form or iterative solutions.

Let $\xi_k(i, j) = \Pr(S_{k-1} = i, S_k = j \mid y_{1:K}, \boldsymbol{\theta})$ and $\gamma_k(i) = \Pr(S_k = i \mid y_{1:K}, \boldsymbol{\theta})$ denote, respectively, the conditional marginal and joint state probabilities (which are the sufficient statistics for the complete data log likelihood 2.9).

From the E-step, we may obtain

$$\gamma_k(i) = \frac{a_k(i)b_k(i)}{\sum_{i=0}^{1} a_k(l)b_k(l)} = \sum_j \xi_k(j,i) = \sum_j \xi_{k+1}(i,j). \tag{2.17}$$

The transition probabilities are given by Baum's reestimation procedure:

$$\hat{P}_{ij} = \frac{\sum_{k=2}^{K} \xi_k(i,j)}{\sum_{k=2}^{K} \sum_j \xi_k(i,j)} = \frac{\sum_{k=2}^{K} \xi_k(i,j)}{\sum_{k=2}^{K} \gamma_k(i)}. \tag{2.18}$$

Specifically, the transition probabilities P_{01} and P_{10} are estimated by closed-form expressions,

$$\hat{P}_{01} = \frac{\sum_{k=2}^{K} \xi_k(0,1)}{\sum_{k=2}^{K} \sum_{j=0}^{1} \xi_k(0,j)} = \frac{\sum_{k=2}^{K} \xi_k(0,1)}{\sum_{k=2}^{K} \gamma_k(0)}, \tag{2.19}$$

$$\hat{P}_{10} = \frac{\sum_{k=2}^{K} \xi_k(1,0)}{\sum_{k=2}^{K} \sum_{j=0}^{1} \xi_k(1,j)} = \frac{\sum_{k=2}^{K} \xi_k(1,0)}{\sum_{k=2}^{K} \gamma_k(1)}. \tag{2.20}$$

Next, we need to estimate the other unknown parameters (μ, α, β) that appear in the likelihood model. Since there is no closed-form solution for μ, α, and β, we may use the iterative optimization methods, such as the Newton-Ralphson algorithm or the iterative weighted least squares (IWLS) algorithm (e.g., Pawitan, 2001), to optimize the parameters in the M-step.

Let $\hat{S}_k = \sum_{i=0}^{1} i\,\gamma_k(i)$ denote the computed mean statistic of a hidden state at time k; by setting the derivatives of $\mathcal{L}$ with regard to the parameters α, μ, and β (and similarly for vector $\boldsymbol{\beta}$), to zeros, we obtain

$$\sum_{k=J}^{K} \hat{S}_k y_k = \sum_{k=J}^{K} \hat{S}_k \exp(\mu + \alpha \hat{S}_k + \beta \tilde{n}_k), \tag{2.21}$$

$$\sum_{k=J}^{K} \tilde{n}_k y_k = \sum_{k=J}^{K} \tilde{n}_k \exp(\mu + \alpha \hat{S}_k + \beta \tilde{n}_k), \tag{2.22}$$

$$\sum_{k=J}^{K} y_k = \sum_{k=J}^{K} \exp(\mu + \alpha \hat{S}_k + \beta \tilde{n}_k), \tag{2.23}$$

respectively. Typically, a fixed number of iterations is preset for the Newton-Ralphson algorithm in the internal loop within the M-step.

Finally, the convergence of the EM algorithm is monitored by the incremental changes of the log likelihood as well as the parameters. If the

absolute value of the change is smaller than 10^{-5}, the EM algorithm is terminated.

2.2.3 Viterbi Algorithm. Upon estimating parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{P}, \mu, \alpha, \beta)$, we can run the Viterbi algorithm (Viterbi, 1967; Forney, 1973) for decoding the most likely state sequences. The Viterbi algorithm is a dynamical programming method (Bellman, 1957) that uses the "Viterbi path" to discover the single most likely explanation for the observations. Specifically, the maximum a posteriori (MAP) state estimate $\hat{S}_k$ at time k is

$$\hat{S}_k^{\mathrm{MAP}} = \arg\max_{i \in \{0,1\}} \gamma_k(i) \quad 1 \le k \le K. \tag{2.24}$$

The computational overhead of the forward Viterbi algorithm has an overall time complexity $\mathcal{O}(4K)$ and space complexity $\mathcal{O}(2K)$.

3 Continuous-Time Markovian and Semi-Markovian State-Space Models

The discrete-time probabilistic model discussed in the previous section imposes strong assumptions on the transition between the UP and DOWN states. First, it is stationary in the sense that the transition probability is time invariant; second, the transition is strictly Markovian. In this section, we relax these assumptions and further develop continuous-time, data-driven (either Markovian or semi-Markovian) state-space models, which is more appropriate and realistic in characterizing the nature or statistics of the state transitions. In addition, the inference algorithm for the continuous-time models uses the estimation output from the discrete-time model (developed in section 2) as the initialization condition, which also helps to accelerate the algorithmic convergence process.

3.1 Continuous-Time Probabilistic Model. One important distinction between the discrete-time and continuous-time Markov chains is that the former allows state changes to occur only at regularly spaced intervals, whereas the latter is aperiodic, in that the time between state changes is exponentially distributed. Therefore, the notion of a "single-step transition probability" is no longer valid in continuous time since the "step" does not exist. In fact, the transition probability in continuous time is characterized by either the transition rate or the sojourn time probability density function (pdf) between the two state change events. Let us assume that the pdf of the sojourn time for state j characterized by a parameter vector θ_j. Hence, the transition probability between state 0 (DOWN) and 1 (UP) is characterized by the corresponding pdfs $p(\theta_0, z)$ and $p(\theta_1, z)$, respectively, where z is now the random variable in terms of holding time. For example, given the current

state status (say, state j) and current time t, the probability of escaping or changing the current state (to other different state) will be computed from the cumulative distribution function (cdf):

$$F(z) \equiv \Pr[0 \le z \le t] = \int_0^t p(\theta_j, z)\, dz, \tag{3.1}$$

and the probability of remaining in the present state will be computed from

$$\Pr[z > t] = 1 - F(z) = \int_t^\infty p(\theta_j, z)\, dz, \tag{3.2}$$

which is known as the survival function in reliability and survival analysis. As seen, the transition probability matrix in continuous time now depends on the elapsed time (starting from the state onset) as well as the present state status. In general, we write the transition probability matrix as a parameterized form $\boldsymbol{P}(\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\boldsymbol{\theta}_0, \boldsymbol{\theta}_1)$ characterizes the associated sojourn time pdf parameters for the DOWN (0) and UP (1) states. As we will see, choosing different probability density models for the sojourn time results in different formulations of the continuous-time Markov or semi-Markov chain.

In modeling the neural spike train point processes, the CIF characterizes the instantaneous firing probability of a discrete event (i.e., spike). Specifically, the product between the CIF $\lambda(t)$ and the time interval Δ tells approximately the probability of observing a spike within the interval $[t, t + \Delta)$ (e.g., Brown et al., 2003):

$$\lambda(t) = \lim_{\Delta \to 0} \frac{\Pr\{N(t + \Delta) - N(t) = 1 \mid \mathcal{H}_{0:t}\}}{\Delta}.$$

For each spike train, we model the CIF in a parametric form,[5]

$$\lambda^c(t) \equiv \lambda^c(t \mid \mathcal{H}_{0:t}) = \exp\left(\mu_c + \alpha_c S_t + \gamma_c \int_0^t e^{-\beta_c \tau} dN^c(t-\tau)\, d\tau\right), \tag{3.3}$$

where $\exp(\mu_c)$ denotes the baseline firing rate for the cth spike train and β_c denotes an exponential decaying parameter that takes into account the

[5]Here we assume that the individual CIF $\lambda^c(t)$ can be modeled as a GLM with $\log(\cdot)$ as a link function (Truccolo et al., 2005; Paninski, Pillow, & Lewi, 2007). One can also use other functions as the link function candidate, such as $\log(1 + \exp(\cdot))$ or the sigmoidal (bell-shaped) function, whichever better reflects the neuron's firing properties (e.g., Paninski, 2004). The choice of the functional form for the CIF does not affect the inference principle or procedure described below.

history dependence of firing from time 0 up to time t. The nonnegative term $\int_0^t e^{-\beta_c \tau} dN^c(t-\tau)\, d\tau$ defines a convolution between the exponential decaying window and the firing history of spike train c up to time t. Because of digitalized recording devices, all continuous-time signals are sampled and recorded in digital format with a very high sampling rate (32 kHz in our setup). Therefore, we still deal with the "discretized" version of a continuous-time signal. In this case, we sometimes use S_t and S_k interchangeably if no confusion occurs. However, as we see below, their technical treatments are rather different. In the context of continuous-time observations ($\Delta = 1$ ms), every time interval has at most one spike from each spike train. For computational ease, we approximate the integral in equation 3.3 with a finite discrete sum of firing history as follows:

$$\lambda_k^c \equiv \lambda^c(k \mid \mathcal{H}_{1:k}) = \exp\left(\mu_c + \alpha_c S_k + \boldsymbol{\beta}_c^T \tilde{\boldsymbol{n}}_k\right), \tag{3.4}$$

where $\tilde{\boldsymbol{n}}_k$ is a vector containing the number of spike counts within the past intervals, and the length of the vector defines a finite number of windows of spiking history. By assuming that the spike trains are mutually independent, the observed data likelihood is given by (Brillinger, 1988; Brown et al., 2003)

$$p\left(dN_{1:K}^{1:C} \mid S_{1:K}, \boldsymbol{\theta}\right) = \prod_{c=1}^{C} \prod_{k=1}^{K} \exp\left(dN_k^c \log\left[\lambda_k^c \Delta\right] - \lambda_k^c \Delta\right). \tag{3.5}$$

The complete statistics of the continuous-time latent process may be characterized by a triplet: $\mathcal{S} = (n, \boldsymbol{\tau}, \boldsymbol{\chi})$, where $\boldsymbol{\tau} = (\tau_0, \tau_1, \ldots, \tau_n)$ is a vector that contains the duration lengths of sojourn times of $\mathcal{S}$, and $\boldsymbol{\chi} = (\chi_0, \chi_1, \ldots, \chi_n)$ represents the states visited in these sojourns. Let $\nu_0 = 0$, $\nu_j = \sum_{i=0}^{j-1} \tau_i$ ($i = 1, 2, \ldots, n+1$) and $\nu_{n+1} = T$. Alternatively, the complete data likelihood, equation 3.5, can be rewritten in another form,

$$p\left(dN_{1:K}^{1:C} \mid \mathcal{S}, \boldsymbol{\theta}\right) = \prod_{c=1}^{C} \prod_{l=1}^{n} \Pr\left(dN_l^{1:C} \mid \chi_l, \boldsymbol{\theta}\right), \tag{3.6}$$

where $dN_l^{1:C}$ denotes all of spike train observations during the sojourn time $[\nu_{l-1}, \nu_l]$ for the continuous-time (semi-)Markov process $\{S(t)\}$.

If we model each spike train as an inhomogeneous Poisson process with time-varying CIF $\lambda^c(t)$, the expected number of spike counts observed within the duration $[\nu_{l-1}, \nu_l]$ in the cth spike train is given by the integrated intensity (also referred to as cumulative hazard function):

$$\lambda_l^c = \int_{\nu_{l-1}}^{\nu_l} \lambda^c(t)\, dt. \tag{3.7}$$

Correspondingly, the observed data likelihood function, equation 3.5, is given by (Daley & Vere-Jones, 2002)

$$p\left(dN_{1:K}^{1:C} \mid \mathcal{S}, \boldsymbol{\theta}\right) = \prod_{c=1}^{C}\prod_{l=1}^{n}\left(\exp\left(-\lambda_l^c\right)\prod_{i=1}^{y_l^c}\lambda^c(t_i)\right)$$
$$= \prod_{c=1}^{C}\prod_{l=1}^{n}\left(\exp\left(-\lambda_l^c\right)\frac{\left(\lambda_l^c\right)^{y_l^c}}{y_l^c!}\right),$$

where y_l^c denotes the spike counts of the cth spike train during the interval $(\nu_{l-1}, \nu_l]$, and t_i denotes the continuous-time index of the ith spike of a specific spike train during the interval $(\nu_{l-1}, \nu_l]$.

Ultimately, we can write the complete-data log likelihood in a compact form:[6]

$$\begin{aligned}\mathcal{L} &= \log p(S_{0:T}, y_{1:n} \mid \boldsymbol{\theta})\\ &= \sum_{l=1}^{n}\sum_{j=0}^{1}\left(\xi_l(j, j)\log[F(\boldsymbol{\theta}_j, \tau_l)] + \sum_{i\neq j}\xi_l(i, j)\log[1 - F(\boldsymbol{\theta}_j, \tau_l)]\right)\\ &\quad + \sum_{c=1}^{C}\sum_{l=1}^{n}\left(\int_{\nu_{l-1}}^{\nu_l}\log\lambda^c(t)\,dN^c(t) - \lambda_l^c\right),\end{aligned} \tag{3.8}$$

where $\boldsymbol{\theta}_j$ denotes the parameter(s) of the probability model of the sojourn time associated with state j.

3.2 Continuous-Time Markov Chain. In a continuous-time Markov chain (i.e., Markov process), state transitions from one state to another can occur at any instant of time. Due to the Markov property, the time that the system spends in any given state is memoryless, and the distribution of the survival time depends on the state (but not on the time already spent in the state); in other words, the sojourn time is exponentially distributed,[7] which can be characterized by a single rate parameter. The rate parameter, also known as the continuous-time state transition rate, defines the probability per time unit that the system makes a transition from one state to the other

[6] In addition to the compact representation, another main reason for this reformulation is the efficiency and stability of numerical computation in calculating the observed data likelihood or likelihood ratio.

[7] The exponential distribution with mean $1/\lambda$ is the maximum entropy distribution among all continuous distributions with nonnegative support that have a mean $1/\lambda$.

during an infinitesimal time interval:

$$q_{ij} = \lim_{\Delta t \to 0} \frac{\Pr(S_{t+\Delta t} = j \mid S_t = i)}{\Delta t}, \quad i \neq j. \tag{3.9}$$

The total transition rate of state i satisfies the rate balance condition:

$$r_i \equiv q_{ii} = -\sum_{j \neq i} q_{ij}. \tag{3.10}$$

The holding time of the sojourn for state i follows an exponential distribution $\exp(-r_i\tau)$, or, equivalently, the transition times of state i are generated by a homogeneous Poisson process characterized by rate parameter r_i. For a two-state Markov chain, the transition rate matrix may be written as

$$Q = \begin{pmatrix} q_{00} & q_{01} \\ q_{10} & q_{11} \end{pmatrix} = \begin{pmatrix} r_0 & -r_0 \\ -r_1 & r_1 \end{pmatrix}. \tag{3.11}$$

For an exponential random variable z, its cdf is computed as $\Pr[z \leq t] = \int_{-\infty}^{t} re^{-rz}\,dz = 1 - e^{-rt}$, where re^{-rz} is the pdf of the exponential distribution with a rate parameter r. The reciprocal of the rate parameter, $1/r$, is sometimes called the survival parameter in the sense that the exponential random variable z that survives the duration of time has the mean $\mathbb{E}[z] = 1/r$. In light of equations 3.1 and 3.2, at a given specific time t, the probability of remaining within the current state sojourn is $\Pr[z > t] = 1 - \Pr[z \leq t] = \int_{t}^{\infty} re^{-rz}\,dz = e^{-rt}$. Let r_0 and r_1 denote the transition rate for states 0 and 1, respectively. Let $\tau = t - \nu$ denote the elapsed time from the state transition up to the current time instant t; then the parameterized transition probability $\boldsymbol{P} = \{P_{ij}\}$ is characterized by

$$P_{ij}(\tau \mid r_i) = \begin{cases} \exp(-r_i\tau), & i = j \\ 1 - \exp(-r_i\tau), & i \neq j \end{cases} \quad (i, j \in \{0, 1\}). \tag{3.12}$$

Now, the transition probability, instead of being constant, is a probabilistic function of the time after the Markov process makes a transition to or from a given state (the holding time from the last transition or the survival time to the next transition).

3.2.1 Imposing Physiological Constraints. Due to biophysical or physiological constraints, the sojourn time for a specific state might be subject

to a certain range constraint, reflected in terms of the pdf. Without loss of generality, let $p(\tau)$ denote the standard pdf for a random variable τ, and let $\tilde{p}(\tau)$ denote the "censored" version of $p(\tau)$,[8] which is defined by

$$\tilde{p}(\tau) = \frac{1}{c} p(\tau) \mathbb{I}_{[a,b]}(\tau) = \begin{cases} \frac{1}{c} p(\tau), & \tau \in [a, b] \\ 0, & \text{otherwise} \end{cases},$$

where $\mathbb{I}(\cdot)$ is an indicator function and $a > 0$ and $b \in (a, \infty)$ are the lower and upper bounds of the constrained random variable τ (which is always positive for the duration of the sojourn time), respectively. The scalar c is a normalized constant determined by $c = F(b) - F(a)$, where $F(\cdot)$ denotes the corresponding cdf of the standard pdf $p(\tau)$ and $F(\infty) = 1$. Likewise, the censored version of the cdf is computed by $\tilde{F}(\tau) = \int_{-\infty}^{\tau} \tilde{p}(\tau)\, d\tau = \frac{1}{c} \int_{a}^{\tau} p(\tau)\, d\tau$.

Suppose the sojourn time τ of a continuous-time Markov chain follows a censored version of the exponential distribution; then we can write its censored pdf as

$$\tilde{p}(\tau) = \begin{cases} \dfrac{r \exp(-r\tau)}{\exp(-ra)}, & \tau \geq a > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{3.13}$$

where the normalizing constant is given by $c = F(\infty) - F(a) = 1 - [1 - \exp(-ra)] = \exp(-ra)$.

3.3 Continuous-Time Semi-Markov Chain. In contrast to the Markov process, the semi-Markov process is a continuous-time stochastic process $\{S_t\}$ that draws the sojourn time $\{\nu_l\}$ spent in specific discrete states from a nonexponential distribution. In other words, the characterization of the sojourn time is no longer an exponential pdf. Table 2 lists a few examples of continuous-time probability models for characterizing the sojourn time duration for the interevent interval (Tuckwell, 1989). In general, the nonexponential censored pdf with a lower bound gives the flexibility to model the "refractory period" of the UP or DOWN state.

3.3.1 Two-Parameter Exponential Family of Distributions for the UP and DOWN State. To characterize the nonexponential survival time behavior of semi-Markov processes, here we restrict our attention to three probability distributions that belong to the two-parameter exponential family of

[8] *Censoring* is a term used in statistics that refers to the condition that the value of an observation is partially known or the condition that a value occurs outside the range of measuring tool.

Table 2: Summary of Continuous-Time Probability Models for the Transition Probability Density Function $p(\tau)$ (Where τ Is a Nonnegative or Positive Random Variable That Denotes the Holding Time), Cumulative Distribution Function $F(\tau)$, and Survival Function $1 - F(\tau)$.

	pdf $p(\tau)$	cdf $F(\tau)$	$1 - F(\tau)$	Domain
Exponential	$r \exp(-r\tau)$	$1 - \exp(-r\tau)$	$\exp(-r\tau)$	$\tau \in [0, \infty), r > 0$
Weibull	$r\alpha(r\tau)^{\alpha-1} \exp[-(r\tau)^{\alpha}]$	$1 - \exp[-(r\tau)^{\alpha}]$	$\exp[-(r\tau)^{\alpha}]$	$\tau \in [0, \infty), r > 0, \alpha > 0$
Gamma	$\tau^{s-1} \frac{\exp(-\tau/\theta)}{\Gamma(s)\theta^{s}}$	$\frac{\gamma(s,\tau/\theta)}{\Gamma(s)}$	$1 - \frac{\gamma(s,\tau/\theta)}{\Gamma(s)}$	$\tau \in [0, \infty), s > 0, \theta > 0$
Log normal	$\frac{1}{\tau\sqrt{2\pi\sigma^2}} \exp(-\frac{(\ln\tau-\mu)^2}{2\sigma^2})$	$\frac{1}{2} + \frac{1}{2}\mathrm{erf}[\frac{\ln\tau-\mu}{\sqrt{2}\sigma}]$	$\frac{1}{2} - \frac{1}{2}\mathrm{erf}[\frac{\ln\tau-\mu}{\sqrt{2}\sigma}]$	$\tau \in (0, \infty), \mu > 0, \sigma > 0$
Inverse gaussian	$\sqrt{\frac{s}{2\pi\tau^3}} \exp(-\frac{s(\tau-\mu)^2}{2\mu^2\tau})$	$\Phi(\sqrt{\frac{s}{\tau}}(\frac{t}{\mu} - 1)) + \exp(\frac{2s}{\tau})\Phi(-\sqrt{\frac{s}{\tau}}(\frac{t}{\mu} + 1))$	—	$\tau \in (0, \infty), \mu > 0, s > 0$

Note: $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-t^2)\, dt$ denotes the error function, $\Phi(z; \mu, \sigma) = \int_{-\infty}^{z} \exp(-\frac{(t-\mu)^2}{2\sigma^2})\, dt$ denotes the gaussian cumulative distribution function, and these two functions relate to each other by $\Phi(z) = \frac{1}{2}[1 + \mathrm{erf}(\frac{z}{\sqrt{2}})]$.

continuous probability distributions. We define the censored versions of three pdfs as follows:

- The censored gamma distribution $\tilde{p}(\tau; s, \kappa)$:

$$\tilde{p}(\tau; s, \kappa) = \frac{1}{c}\mathbb{I}_{[a,b]}(\tau)p(\tau; s, \kappa) = \frac{1}{c}\mathbb{I}_{[a,b]}(\tau)\tau^{s-1}\frac{\exp(-\tau/\kappa)}{\Gamma(s)\kappa^s},$$

  where κ and s represent the scale and shape parameters, respectively. If s is an integer, then the gamma distribution represents the sum of s exponentially distributed random variables, each with a mean κ. Similarly, c is a normalized constant for the censored pdf $\tilde{p}(\tau; s, \kappa)$: $c = F(b) - F(a)$, and $F(\cdot)$ is the cdf of the standard gamma distribution.
- The censored log-normal distribution $\tilde{p}(\tau; \mu, \sigma)$:

$$\begin{aligned}\tilde{p}(\tau; \mu, \sigma) &= \frac{1}{c}\mathbb{I}_{[a,b]}(\tau)p(\tau; \mu, \sigma)\\ &= \frac{1}{c}\mathbb{I}_{[a,b]}(\tau)\frac{1}{x\sigma\sqrt{2\pi}}\exp\left(-\frac{[\ln(\tau)-\mu]^2}{2\sigma^2}\right),\end{aligned}$$

  where the mean, median, and variance of the log-normal distribution are $\exp(\mu + \sigma^2/2)$, $\exp(\mu)$, and $\exp(2\mu + \sigma^2)[\exp(\sigma^2) - 1]$, respectively.
- The censored inverse gaussian distribution $\tilde{p}(\tau; \mu, s)$:

$$\begin{aligned}\tilde{p}(\tau; \mu, s) &= \frac{1}{c}\mathbb{I}_{[a,b]}(\tau)p(\tau; \mu, s)\\ &= \frac{1}{c}\mathbb{I}_{[a,b]}(\tau)\sqrt{\frac{s}{2\pi\tau^3}}\exp\left(-\frac{s(\tau-\mu)^2}{2\mu^2\tau}\right),\end{aligned}$$

  where μ and s represent the mean and shape parameters, respectively.

The choice of the last two probability distributions is mainly motivated by the empirical data analysis published earlier (Ji & Wilson, 2007). Typically, for a specific data set, a smoothed histogram analysis is conducted to visualize the shape of the distribution, and the Kolmogorov-Smirnov (KS) test can be used to empirically evaluate the fit of specific probability distributions. Mostly likely, none of parametric probability distribution candidate would fit perfectly (i.e., within 95% confidence interval) for the real-world data; we often choose the one that has the best fit.[9] In the simulation data shown in Figure 1, we have used the following constraints for the UP and DOWN states: [0.1, 3] (unit: second) for UP state and [0.05, 1] (unit: second) for DOWN state. The lower and upper bounds of these constraints are chosen in light of the results reported from Ji and Wilson (2007). Note that the shapes of the log-normal and inverse gaussian pdfs and cdfs are very similar, except that inverse gaussian distribution is slightly sharper when

[9] Alternatively, one can use a discrete nonparametric probability model for the sojourn time, which is discussed in section 5.

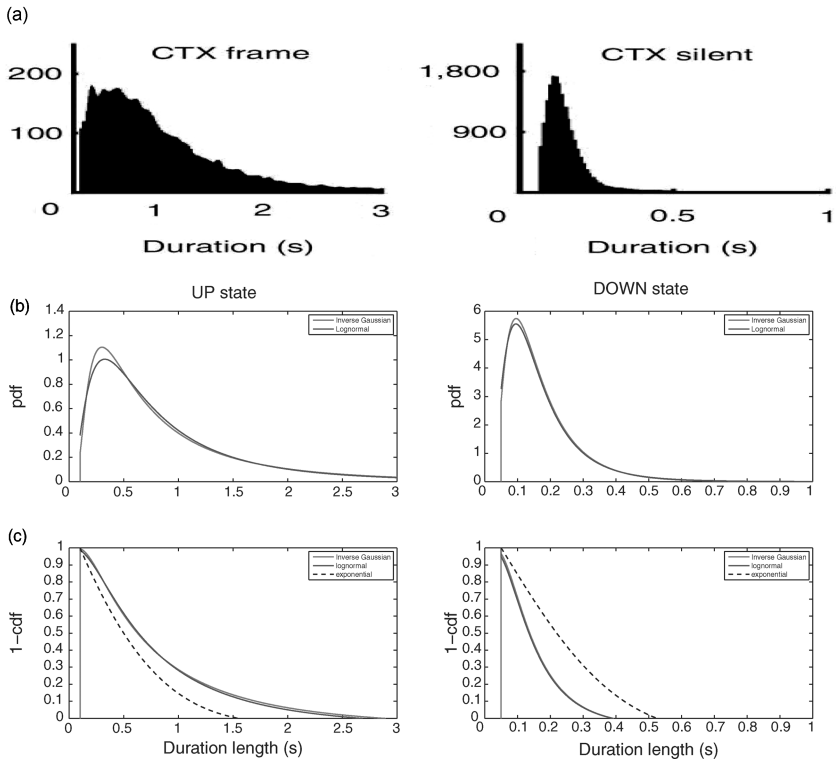


Figure 1: (a) The duration histograms of the UP (mean 0.96, median 0.67, support [0.1, 3], unit: second) and DOWN (mean 0.17, median 0.13, support [0.05, 1], unit: second) states of spiking data recorded from four behaving rats' visual cortex during SWS. Note that the statistics used in *b* are identical to those in (Ji & Wilson, 2007, Figure 2b). The *y*-axis in both plots shows the count statistics of all cortical UP and DOWN durations. (b) Censored versions of the log-normal and inverse gaussian pdfs for the UP (left panel) and DOWN (right panel) states. (c) Censored versions of the log-normal and inverse gaussian survival functions (1-cdf) for the UP (left panel) and DOWN (right panel) states. As comparison, the dashed lines in *b* and *c* show the holding time probability for an exponential distribution. Note that in the UP state, the holding time probability in both two-parameter distributions decays more slowly than that of the exponential distribution, whereas in the DOWN state, the holding time probability of exponential distribution decays more slowly than the others.

the value of the random variable is small (Takagi, Kumagai, Matsunaga, & Kusaka, 1997). In addition, the tail behavior of these two distributions differs; however, provided we consider only their censored versions (with finite duration range), the tail behavior is not a major concern here.

3.4 EM Algorithm. In the continuous-time model, we treat the individual spike trains separately and aim to estimate their own parameters. Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_{up}, \boldsymbol{\theta}_{down}, \{\mu_c\}_{c=1}^{C}, \{\alpha_c\}_{c=1}^{C}, \{\boldsymbol{\beta}_c\}_{c=1}^{C},)$ denote the unknown parameters of interest, where θ_{up} and θ_{down} represent the parameters associated with the parametric pdfs of the UP and DOWN states, respectively; the rest of the parameters are associated with the CIF model for respective spike trains. For an unknown continuous-time latent process $\{S(t); 0 \le t \le T\}$ (where $S(t) \in \{0, 1\}$), let n be the number of jumps between two distinct discrete states. Let $\mathcal{S} = (n, \boldsymbol{\tau}, \boldsymbol{\chi})$ be a triplet of the Markov or semi-Markov process, where $\boldsymbol{\tau} = (\tau_0, \tau_1, \ldots, \tau_n)$ is a vector that contains the duration of the sojourn time of $\mathcal{S}$ and $\boldsymbol{\chi} = (\chi_0, \chi_1, \ldots, \chi_n)$ represents the states visited in these sojourns.

Let $\mathcal{Y}$ denote all the spiking timing information of the observed spike trains. Similar to the discrete-time HMM, we aim to maximize the Q-function, defined as follows:

$$Q(\boldsymbol{\theta}) = \sum_{S} p(\mathcal{S} \mid \mathcal{Y}, \boldsymbol{\theta}) \log p(\mathcal{Y}, \mathcal{S} \mid \boldsymbol{\theta}). \tag{3.14}$$

The inference can be tackled in a similar fashion by the EM algorithm.

First, let us consider a simpler task where the transition time of the latent process is known and the number of state jumps is given. In other words, the parameters n and $\boldsymbol{\tau}$ are both available, so the estimation goal becomes less demanding. We need to estimate only $\boldsymbol{\chi}$ and $\boldsymbol{\theta}$.

Since the number of state transition, n, is known, $p(\mathcal{Y}, \mathcal{S} \mid \boldsymbol{\theta})$ is simplified to

$$p(\mathcal{S}, \mathcal{Y} \mid \boldsymbol{\theta}) = \sum_{l=1}^{n} P_{\mathcal{S}_{l-1}, \mathcal{S}_l}(\tau_l) P(\mathcal{Y}_l \mid \mathcal{S}(\chi_l)). \tag{3.15}$$

Let $\xi_l(i, j) = \Pr(\mathcal{S}(\chi_{l-1}) = i, \mathcal{S}(\chi_l) = j)$ and $\gamma_l(i) = \Pr(\mathcal{S}(\chi_l) = i)$. In the case of the continuous-time Markov chain, the complete data log likelihood is given by

$$\begin{aligned}
\mathcal{L}(\mathcal{S}, \boldsymbol{\theta}) &= \log p(S_{0:T}, \mathcal{Y} \mid \boldsymbol{\theta}) \\
&= \sum_{l=1}^{n} \sum_{j=0}^{1} \left(\xi_l(j, j)(-r_j \tau_l) + \sum_{i \neq j} \xi_l(j, i) \log[1 - \exp(-r_j \tau_l)] \right) \\
&\quad + \sum_{c=1}^{C} \sum_{l=1}^{n} \left(\int_{v_{l-1}}^{v_l} \log \lambda^c(t)\, dN^c(t) - \lambda_l^c \right),
\end{aligned} \tag{3.16}$$

where $\boldsymbol{\theta} = (r_0, r_1, \mu_1 \alpha_1, \beta_1, \ldots, \mu_C, \alpha_C, \beta_C)$ denotes the augmented parameter vector that contains all of the unknown parameters. In the case of the

continuous-time semi-Markov chain where the sojourn time is modeled by a nonexponential probability distribution, we can write the log-likelihood function as

$$\begin{aligned}\mathcal{L} &= \log p(S_{0:T}, \mathcal{Y} \mid \boldsymbol{\theta}) \\ &= \sum_{l=1}^{n} \left(\sum_{j=0}^{1} \xi_l(j, j) \log[1 - F(\tau_l; \boldsymbol{\theta}_j)] + \sum_{i \neq j} \xi_l(j, i) \log[F(\tau_l; \boldsymbol{\theta}_j)] \right) \\ &\quad + \sum_{c=1}^{C} \sum_{l=1}^{n} \left(\int_{v_{l-1}}^{v_l} \log \lambda^c(t)\, dN^c(t) - \lambda_l^c \right), \end{aligned} \tag{3.17}$$

where $F(\cdot)$ denotes the cdf of the nonexponential probability distribution.

Conditioned on the parameter $\boldsymbol{\theta}$, the posterior probabilities ξ_l and γ_l (for $l = 1, \ldots, n$) can be similarly estimated by the forward-backward algorithm as in the E-step for the HMM, whereas in the M-step, the new parameter $\boldsymbol{\theta}^{new}$ is obtained by

$$\boldsymbol{\theta}^{new} = \arg\max_{\boldsymbol{\theta}} \sum_{\mathcal{S}} \Pr\{\mathcal{S} \mid \mathcal{Y}, \boldsymbol{\theta}^{old}\} \log(\Pr\{\mathcal{S}, \mathcal{Y} \mid \boldsymbol{\theta}\}). \tag{3.18}$$

More specifically, for the parameters associated with the transition probability density model, we might, for example, assume that the UP and DOWN state durations are both log normal distributed with parameters $\boldsymbol{\theta}_j = \{\mu_j, \sigma_j\} (j = 0, 1)$, and they can be estimated by

$$\begin{aligned}\{\mu_j, \sigma_j\} = \arg\max_{\mu,\sigma} \Bigg\{ &\sum_{l=1}^{n} \xi_l(j, j) \log[1 - F(\tau_l; \mu, \sigma)] \\ &+ \sum_{l=1, i \neq j}^{n} \xi_l(j, i) \log[F(\tau_l; \mu, \sigma)] \Bigg\}. \end{aligned} \tag{3.19}$$

In light of Table 2, setting the derivatives of μ_j and σ_j to zeros yields

$$\begin{aligned} 0 &= \sum_{l=1}^{n} \frac{-1}{\sigma} \exp\left(-\frac{(\ln \tau_l - \mu)^2}{2\sigma^2} \right) \left(\frac{\xi_l(j, j)}{1 - F(\tau_l; \mu, \sigma)} + \frac{\sum_{i \neq j} \xi_l(j, i)}{F(\tau_l; \mu, \sigma)} \right), \\ 0 &= \sum_{l=1}^{n} \frac{\mu - \ln \tau_l}{\sigma^2} \exp\left(-\frac{(\ln \tau_l - \mu)^2}{2\sigma^2} \right) \left(\frac{\xi_l(j, j)}{1 - F(\tau_l; \mu, \sigma)} + \frac{\sum_{i \neq j} \xi_l(j, i)}{F(\tau_l; \mu, \sigma)} \right), \end{aligned}$$

where we have used $\frac{d\,\mathrm{erf}(z)}{dz} = \frac{2}{\sqrt{\pi}} \exp(-z^2)$ in light of Table 2. The above two equations can be solved iteratively with the Newton-Ralphson algorithm.

Once the state estimate $\hat{S}(t)$ is available from the E-step,[10] the parameters associated with the likelihood model can also be estimated by the Newton-Ralphson or the IWLS algorithm,

$$\{\mu_c, \alpha_c, \beta_c\} = \arg\max_{\mu,\alpha,\boldsymbol{\beta}} \left(\sum_{l=1}^{n} \int_{v_{l-1}}^{v_l} \log \lambda^c(t)\, dN^c(t) - \lambda_l^c \right), \tag{3.20}$$

where $\lambda^c(t)$ and λ_l^c are defined by equations 3.3 (or 3.4) and 3.7, respectively.

Notably, the EM algorithm described above has a few obvious drawbacks. Basically, it assumes that the information as to when and how many state transitions occur during the latent process is given; once the number of state jumps (say, n) is determined, it does not allow the number n to change, so it is incapable of online model selection. Furthermore, it is very likely that the EM algorithm suffers from the local maximum problem, especially if the initial conditions of the parameters are far from the true estimates. In the following, we present an alternative inference method to overcome these two drawbacks, and the method can be regarded as a generalization of the EM algorithm, except that the E-step state estimation is replaced by a Monte Carlo sampling procedure. This method is often called the Monte Carlo EM (MCEM) algorithm (Chan & Ledolter, 1995; McLachlan & Krishnan, 1996; Tanner, 1996; Stjernqvist, Rydén, Sköld, & Staaf, 2007). The essence of MCEM is the theory of Markov chain Monte Carlo (MCMC), which will be detailed below.

3.5 Monte Carlo EM Algorithm. The basic idea of MCMC sampler is to draw a large number of samples randomly from the posterior distribution and then obtain a sample-based numerical approximation of the posterior distribution. Unlike other Monte Carlo samplers (such as importance sampling and rejection sampling), the MCMC method is well suited for sampling from a complex and high-dimensional probability distribution. Instead of drawing independent samples from the posterior distribution directly, MCMC constructs a Markov chain such that its equilibrium will eventually approach the posterior distribution. The Markov chain theory states that given an arbitrary initial value, the chain will ultimately converge to the equilibrium point provided the chain is run sufficiently long. In practice, determining the convergence as well as the "burn-in time" for MCMC samplers requires diagnostic tools (see Gilks, Richardson, & Spiegelhalter, 1995, for general discussions of the MCMC methods). Depending on the specific problem, the MCMC method is typically computationally intensive, and the convergence process can be very slow. Nevertheless, here we focus

[10]Note that $\hat{S}(t)$ is not the same as χ_l ($l = 1, 2, \ldots, n$) in that the former is stochastic and the latter is deterministic.

on methodology development, and therefore the computational demand is not the main concern. Specifically, constructing problem-specific MCMC proposal distributions (densities) is the key to obtain an efficient MCMC sampler that has a fast convergence speed and a good mixing property (Brooks, Guidici, & Roberts, 2003). For the variable-dimension RJMCMC method, we present some detailed mathematical treatments in appendix A.

The Monte Carlo EM (MCEM) algorithm works just like an ordinary EM algorithm, except that in the E-step, the expectation operations (i.e., computation of sufficient statistics) are replaced by Monte Carlo simulations of the missing data. Specifically, M realizations of the latent process $S(t)$ $(0 \le t \le T)$ are simulated, and in this case the Q-function can be written as

$$Q(\mathcal{S}, \boldsymbol{\theta}) \approx \hat{Q}(\mathcal{S}, \boldsymbol{\theta}) = \frac{1}{M} \sum_{m=1}^{M} Q\big(\mathcal{S}^{(m)}, \boldsymbol{\theta}\big), \tag{3.21}$$

where $\mathcal{S}^{(m)}$ denotes the mth simulated latent process for the unknown state (missing data).

The M-step of MCEM is the same as the conventional M-step in the EM algorithm. Specifically, the parameters of the CIF appearing in the likelihood model are estimated using equation 3.20. However, the estimation of the parameters for the UP or DOWN state sojourn depends on the type of probability distribution. Here we distinguish three different possible scenarios.

First, when the latent process is a continuous-time Markov chain and the sojourn time durations for the UP and DOWN states are both exponentially distributed, then $\boldsymbol{\theta}_{up}$ and $\boldsymbol{\theta}_{down}$ correspond to the rate parameters r_1 and r_0, respectively. The Q-function for a single Monte Carlo realization of $\mathcal{S}$ can be written as

$$\begin{aligned} Q(\mathcal{S}, \boldsymbol{\theta}) = & \sum_{i=0, j\neq i}^{1} (\hat{n}_{ij} \log q_{ij} + r_i T_i) \\ & + \sum_{c=1}^{C} \sum_{l=1}^{n} \left(\int_{v_{l-1}}^{v_l} \log \lambda^c(t)\, dN^c(t) - \lambda_l^c \right), \end{aligned} \tag{3.22}$$

where n_{ij} denotes the number of jumps from state i to state j during $[0, T]$, and $T_i = \int_0^T \mathbb{I}(S(t) = i)\, dt$ denote the total time or the sojourn length of $\{S(t)\}$ spent in state i during $[0, T]$. Let $n_i = \sum_{l=0}^{n} \mathbb{I}(\chi_l = i)$ denote the number of events that occur while $\{S(t)\}$ is in state i; then it is known that the transition rate matrix can be estimated by $\hat{q}_{ij} = n_{ij}/T_i$ and $r_i = q_{ii} = n_i/T_i$, where n_{ij}, n_i, and T_i are the sufficient statistics (Rydén, 1996). In this case, r_i corresponds to the MLE. With M Monte Carlo realizations, the rate

parameter will be estimated by

$$r = \frac{1}{M}\sum_{m=1}^{M}\frac{\sum_{l=0}^{n}\mathbb{I}\big(\chi_l^{(m)}=0\big)}{\int_0^T \mathbb{I}\big(S^{(m)}(t)=0\big)\,dt}.$$

In the second scenario, when the latent process is a continuous-time semi-Markov chain where the sojourn time durations for both UP and DOWN states are nonexponentially distributed, the Q-function can be written as

$$Q(\mathcal{S},\boldsymbol{\theta}) = \frac{1}{M}\sum_{m=1}^{M}\sum_{l=0}^{n-1}\left(\sum_{j=0}^{1}\sum_{j\to i}^{i\neq j}\log\left[1-F\left(\tau_l^{(m)};\boldsymbol{\theta}_j\right)\right]\right) + \sum_{c=1}^{C}\sum_{l=1}^{n}\left(\int_{v_{l-1}}^{v_l}\log\lambda^c(\overline{S}(t),\boldsymbol{\theta})\,dN^c(t)-\lambda_l^c\right), \tag{3.23}$$

where $\overline{S}(t)=\frac{1}{M}\sum_{m=1}^{M}\hat{S}^{(m)}(t)$ is obtained from the Monte Carlo mean statistic of M simulated latent processes. Similarly, the parameters of the nonexponential probability distributions can be estimated by their MLE based on their Monte Carlo realizations. For instance, in the case of inverse gaussian distribution, the mean parameter is given by $\mu_j=\frac{1}{n_jM}\sum_{m=1}^{M}\sum_{l=0}^{n}\tau_l^{(m)}\mathbb{I}(\chi_l^{(m)}=j)$, and the shape parameter is given by $\sigma_j=\big[\frac{1}{n_jM}\sum_{m=1}^{M}\sum_{l=0}^{n}((\tau_l^{(m)})^{-1}-\mu_j^{-1})\mathbb{I}(\chi_l^{(m)}=j)\big]^{-1}$. In the case of log-normal distribution, the mean parameter is given by $\mu_j=\frac{1}{n_jM}\sum_{m=1}^{M}\sum_{l=0}^{n}\mathbb{I}(\chi_l^{(m)=}j)\ln\tau_l^{(m)}$, and the SD parameter is given by $\sigma_j=\frac{1}{n_jM}\sum_{m=1}^{M}\sum_{l=0}^{n}\mathbb{I}(\chi_l^{(m)}=j)\mid\ln\tau_l^{(m)}-\mu_j\mid$.

If, in the third scenario, one of the state sojourn time durations is exponentially distributed and the other is nonexponential, then the resulting latent process is still a semi-Markov chain, and the estimation procedure remains similar to that in the above two cases.

3.5.1 Initialization of the MCMC Sampler. It is important to choose sensible initial values for the simulated (semi-) Markov chain since a poor choice of the initial $\mathcal{S}^{(0)}$ can lead to a sampler that takes a very long time to converge or result in a poor mixing of the (semi-) Markov chain. In our experiment, we typically use a discrete-time HMM model (with a 10 ms bin size) to estimate the hidden state sequence and then interpolate it to obtain an initial estimate of the continuous-time state process (with 1 ms bin size), from which we obtain the initial values of $\{n,\boldsymbol{\tau},\boldsymbol{\chi}\}$.

3.5.2 Algorithmic Procedure. In summary, the MCEM algorithm is run as follows:

- Initialize the MCMC sampler state for $\mathcal{S}=\{n,\boldsymbol{\tau},\boldsymbol{v}\}$.

- Iterate the MCEM algorithm's E and M steps until the log likelihood reaches a local maximum or saddle point.
  1. Monte Carlo E-step: Given an initial state $\mathcal{S}^{(0)}$, run the RJMCMC sampling procedure to draw M Monte Carlo samples $\{\mathcal{S}^{(m)}\}_{m=1}^{M}$, based on which to compute the necessary Monte Carlo sufficient statistics.
  2. M-step: estimate the parameters $\{\boldsymbol{\theta}_{up}, \boldsymbol{\theta}_{down}\}$ with their MLE.
  3. M-step: optimize the parameters $\{\mu_c, \alpha_c, \boldsymbol{\beta}_c\}_{c=1}^{C}$ according to equation 3.20.
- Upon convergence, reconstruct the hidden state $\hat{S}(t)$ in the continuous-time domain.
- Compute $\lambda^c(t)$ for each spike train c, and conduct goodness-of-fit tests (see below) for all spike trains being modeled.

3.5.3 Reconstruction of Hidden State. There are two ways to reconstruct the hidden state of the latent process. The first is to apply the Viterbi algorithm once the MCEM inference is completed (when n and $\boldsymbol{\tau}$ have been determined). In the second, and simpler, approach, we can determine the state by the following rule (Ball et al., 1999). For $m = 1, 2, \ldots, M$, let

$$\hat{S}^{(m)}(t) = \begin{cases} 1, & \text{if } \chi_t^{(m)} \in \text{UP}, \\ 0, & \text{if } \chi_t^{(m)} \in \text{DOWN}, \end{cases} \tag{3.24}$$

and let $\overline{S}(t) = \frac{1}{M}\sum_{m=1}^{M} \hat{S}^{(m)}(t)$, and the point estimate of the hidden state is

$$\hat{S}(t) = \begin{cases} 1 & \text{if } \overline{S}(t) \geq 0.5, \\ 0 & \text{if } \overline{S}(t) < 0.5. \end{cases} \tag{3.25}$$

Furthermore, the marginal prior probability of the hidden state is given by

$$\hat{\pi}_i = \frac{1}{M}\sum_{m=1}^{M} \mathbb{I}\big(S^{(m)}(0) = i\big). \tag{3.26}$$

3.5.4 Goodness-of-Fit Tests. Upon estimating the CIF model $\lambda^c(t)$ for each spike train (see equation 3.3), the goodness of fit of the estimated model is tested in light of the time-rescaling theorem (Brown, Barbieri, Ventura, Kass, & Frank, 2002). Specifically, given a point process specified by J discrete events: $0 < u_1 < \cdots < u_J < T$, define the random variables $z_j = \int_{u_{j-1}}^{u_j} \lambda(\tau)\, d\tau$ for $j = 1, 2, \ldots, J-1$. Then the random variables z_js are independent, unit-mean exponentially distributed. By introducing the variable of transformation $v_j = 1 - \exp(-z_j)$, v_js are independent and uniformly distributed within the region $[0, 1]$. Let $g_j = \Phi^{-1}(v_j)$ (where $\Phi(\cdot)$ denotes the cdf of the standard gaussian distribution); then g_js will be

independent standard gaussian random variables. Furthermore, the standard Kolmogorov-Smirnov (KS) test is used to compare the cdf of v_j against that of the random variables uniformly distributed within [0, 1]. The KS statistic is the maximum deviation of the empirical cdf from the uniform cdf. To compute it, the v_js are sorted from the smallest to the largest value; then we plot values of the cdf of the uniform density defined as $\frac{j-0.5}{J}$ against the ordered v_js. The points should lie on the 45 degree line. In a Cartesian plot of the empirical cdf as the y-coordinate versus the uniform cdf as the x-coordinate, the 95% confidence interval lines are $y = x \pm \frac{1.36}{(J-1)^{1/2}}$. The KS distance, defined as the maximum distance between the KS plot and the 45 degree line, is used to measure the lack of fit between the model and the data.

Furthermore, we measure the independence of the time-rescaled time series by computing the autocorrelation function of g_js: ACF(m) = $\frac{1}{J-m}\sum_{j=1}^{J-m} g_j g_{j+m}$. Since g_js are normally distributed, if they are independent, then they are also uncorrelated; hence, ACF(m) shall be small for all values of m, and the associated 95% confidence interval is $0 \pm \frac{1.96}{(J-1)^{1/2}}$.

3.5.5 Implementation and Convergence. Let the triple $\mathcal{S} = (n, \boldsymbol{\tau}, \boldsymbol{\chi})$ denote all the information of the continuous-time latent process, which contains n state jumps and $n+1$ durations of corresponding sojourn times, and the discrete states that are visited in the sojourns.

To simulate the Markov chain, we first obtain the initial conditions for both the state and parameters $\{\mathcal{S}^{(0)}, \boldsymbol{\theta}^{(0)}\}$. Next, we run the MCMC sampler (see appendix A for details) to generate a sequence of Monte Carlo samplers $\{\mathcal{S}^{(k)}\}$ for $k = 1, 2, \ldots, M$, and the realizations $\{\mathcal{S}^{(k)}\}$ can be viewed as the samples drawn from the conditional posterior $p(\mathcal{S} \mid \mathcal{Y}, \boldsymbol{\theta})$. At each MCEM step, the parameter vector $\boldsymbol{\theta}$ is updated in the Monte Carlo M-step using the sufficient statistics obtained from $p(\mathcal{S} \mid \mathcal{Y}, \boldsymbol{\theta})$. Typically, to reduce the correlation of the simulated samples, a "burn-in" period is discarded at the beginning of the simulated (semi-) Markov chain. Even after the burn-in period, the successive realizations of $\{\mathcal{S}^{(k)}\}$ might still be highly correlated. This problem can be alleviated by using the "thinning" or subsampling technique: every N_p simulated samples in the chain is used. Although the thinning technique can increase the Monte Carlo variance of the estimate (Geyer, 1992), it is widely used in practice to reduce the correlation among the samples. In our experiment, we typically chose $N_p = 10$.

At the end of each Monte Carlo sampling step, the sufficient statistics for computing the acceptance probability and updating the parameter $\boldsymbol{\theta}$ (in the M-step) need to be stored, and all new information $(n, \boldsymbol{\tau}, \boldsymbol{\chi})$ needs to be updated each time $\mathcal{S}$ changes.

In terms of convergence, Markov chain theory tells us that when $M \to \infty$, the samples are asymptotically drawn from the desired target (equilibrium) distribution. However, choosing the proper value of M is often problem dependent, and the convergence diagnosis of the MCMC sampler remains

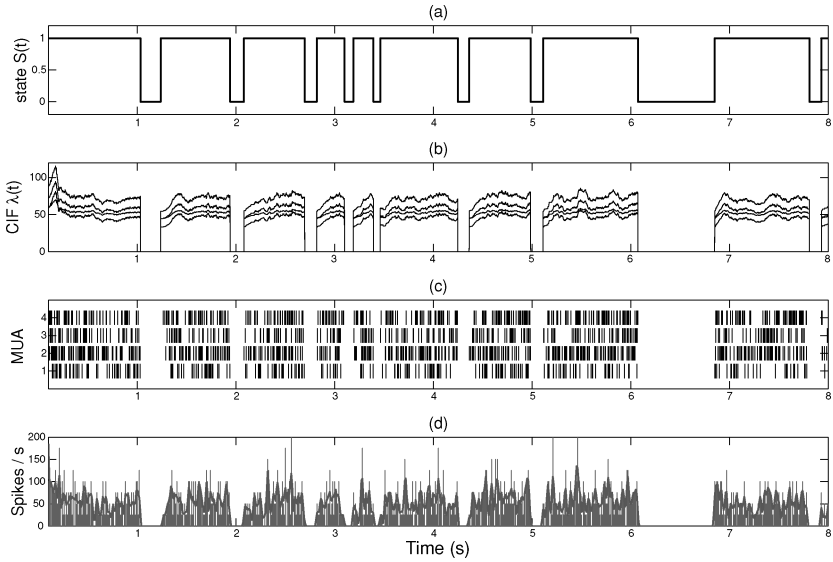


Figure 2: Synthetic data. (a) The simulated UP and DOWN hidden state process. (b) The simulated time-varying traces of conditional intensity function (CIF) $\lambda^c(t)$ ($c = 1, \ldots, 4$). (c) The four simulated spike trains. (d) The averaged firing rate across four spike trains (the solid gray curve corresponds to the temporally smoothed firing rate using a 30 ms width gaussian kernel).

an active research topic in the literature (e.g., Gelman & Rubin, 1992; Cowles & Carlin, 1996).

4 Experimental Results

4.1 Synthetic Data. First, we simulate four spike trains with the time-rescaling theorem (see Figure 2 for a snapshot of one realization). The latent state variable is assumed to be drawn from a two-state discrete space: $S(t) \in \{0, 1\}$. The simulation is conducted with a 1 ms time bin size with the following model:

$$\lambda^c(t) = \exp(\mu_c + \alpha_c S(t) + \beta_c \tilde{n}(t)), \quad (c = 1, \ldots, 4),$$

where $\tilde{n}(t)$ denotes the number of spike counts across all spike trains during the previous 100 ms prior to the current time index t, and the parameters of individual spike trains are set as follows:

$$\mu_1 = -3.5, \quad \mu_2 = -4.0, \quad \mu_3 = -3.8, \quad \mu_4 = -3.8,$$

Table 3: Comparison of Experimental Results on the Simulation Data.

	State Estimation Error Rate		
Method	Mean ± SD	Best	Worst
Threshold based	2.91 ± 0.31%	2.41%	3.48%
Discrete HMM-EM	1.52 ± 0.34	1.07	2.07
Continuous MCEM	1.26 ± 0.42	0.74	1.95

Note: Mean performance is averaged over 10 independent random trials.

$$\alpha_1 = 7.0, \quad \alpha_2 = 8.0, \quad \alpha_3 = 7.6, \quad \alpha_4 = 7.6,$$
$$\beta_1 = 0.06, \quad \beta_2 = 0.05, \quad \beta_3 = 0.03, \quad \beta_4 = 0.05.$$

For the simulated hidden latent process, the total duration is $T = 30$ s, and the number of jumps varies from 35 to 45, yielding an average occurrence rate of state transitions of about 80 min^{-1}. Furthermore, we assume that the sojourn time durations for both UP and DOWN states follow a log-normal distribution. For the UP state, the survival time length is randomly drawn from $\text{logn}(-0.4005, 0.8481)$ (such that the mean and median value of the duration are 0.67 s and 0.96 s, respectively), with a lower bound of 0.15 s; and for the DOWN state, the survival time length is randomly drawn from $\text{logn}(-1.9661, 0.6231)$ (such that the mean and median value of the duration are 0.14 s and 0.17 s, respectively), with a lower bound of 0.05 s.

To test the discrete-time HMM model, the spike trains are binned in 10 ms and collected by spike counts. We employed the EM algorithm with the following initialization parameters: $\pi_0 = \pi_1 = 0.5$, $P_{00} = P_{11} = 0.9$, $P_{01} = P_{10} = 0.1$. For the synthetic data, the EM algorithm typically converges within 200 iterations. The forward-backward algorithm computes all necessary sufficient statistics. Upon convergence, the Viterbi algorithm produced the ultimate state sequence output, yielding an average decoding error rate of 1.52% (averaged over 10 independent runs). In this case, since the CIF model is given (no model mismatch issue is involved), the decoding error rate is reasonably low even with the discrete-time HMM. As a comparison, we also employed the threshold-based method (Ji & Wilson, 2007; see appendix B for brief descriptions) to classify the UP and DOWN states using the simulated spike trains. It was found (see Table 3) that the discrete-time HMM method yields better performance than the threshold-based method. Figure 3 plots a snapshot of hidden state estimation obtained from the discrete-time HMM in our experiments.

Next, we applied the MCEM algorithm to refine the latent state estimation in the continuous-time domain. Naturally, with a smaller bin size, the continuous-time model allows precisely segmenting the UP and DOWN states for identifying the location of state transition. With the initial

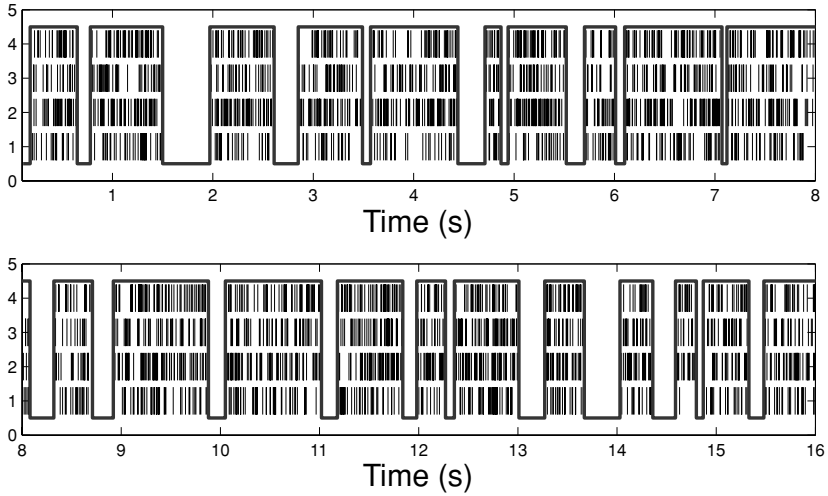


Figure 3: A snapshot of UP and DOWN state estimation obtained from the discrete-time HMM for the simulated spike train data.

conditions obtained from the discrete-time EM algorithm, we simulated the Markov chains for 20,000 iterations and discarded the first 1000 iterations ("burn-in" period). For the synthetic data, the MCEM algorithm converged after 20 to 30 iterations, and we were able to further improve the decoding accuracy by reducing the average error rate to 1.26%. As seen in Table 3, the continuous-time model outperformed the discrete-time HMM model in terms of the lower estimation error. However, in terms of estimating the correct number of state transitions, the HMM obtained nearly 100% accuracy in all 10 Monte Carlo trials (except for two trials that miscount two more jumps); in this sense, the HMM estimation result can be treated as a very good initial state as the input to the continuous-time semi-Markov chain model. In addition, the continuous-time model yields a 10 times greater information transmission rate (1 bit/ms) than the discrete-time model (1 bit/10 ms). We also computed the probability distribution statistics of the UP and DOWN states from both estimation methods. In the discrete-time HMM, we used the sample statistics of the UP and DOWN state durations as the estimated results. These results were also used as the initial values for the continuous-time semi-Markov process, where the MCEM algorithm was run to obtain the final estimate. The results are summarized in Table 4.

Once the estimates of $\{S(t)\}$ and $\{\mu_c, \alpha_c, \beta_c\}$ become available, we compute $\lambda^c(t)$ for the simulated spike trains in continuous time (with $\Delta = 1$ ms). Furthermore, the goodness-of-fit tests are employed to the rescaled time series, and the KS plots and the autocorrelation plots for the simulated spike trains are shown in Figure 4. As seen from the figure, these plots fall almost

Table 4: Duration Length Statistics of the UP and DOWN States from the Simulation Data.

	True	Sample Statistics (from HMM)	Estimated (from MCEM)
Mean (UP)	−0.4005	−0.4468	−0.4212
SD (UP)	0.8481	0.6827	0.7735
Mean (DOWN)	−1.9661	−2.1708	−2.0256
SD (DOWN)	0.6231	0.6335	0.6301

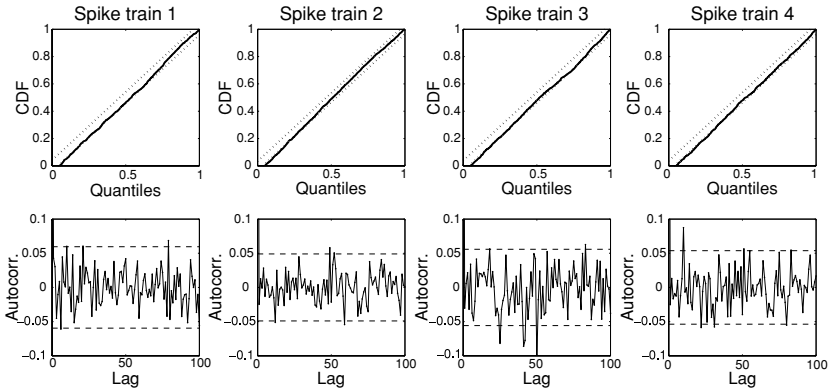


Figure 4: The fitted KS plots (top row) and autocorrelation plots (bottom row) for the four simulated spike trains from one Monte Carlo experiment (dotted and dashed lines in the plots indicate the 95% confidence bounds).

within the 95% confidence bounds, indicating the model fit is sufficiently satisfactory.

Finally, we also do extra simulation studies by examining the sensitivity of different methods regarding the change of two factors: the modulation gains of the hidden state and the number of observed spike trains. The first issue examines the impact of the global network activity during the UP state, that is, the α_c component appearing in $\lambda^c(t)$. Specifically, we modify the gain parameters of individual spike trains (while keeping remaining parameters unchanged) as follows:

$$\alpha_1 = 6.7, \quad \alpha_2 = 7.8, \quad \alpha_3 = 7.3, \quad \alpha_4 = 7.3,$$

such that each λ_c is reduced about 20% during the UP state period. It appears that the average performance of the threshold-based method degraded from the original 2.91% to 3.62% (with a trial-and-error selected threshold), while the performance of the probabilistic models remained almost unchanged.

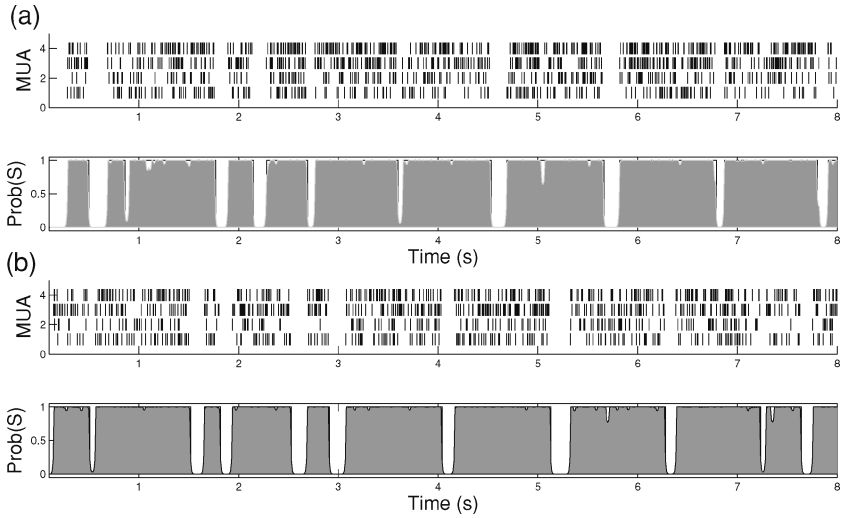


Figure 5: Snapshot illustrations of simulated synthetic spike trains and the estimated state posterior probability from the (a) HMM and (b) continuous-time semi-Markov model (b). The shaded area denotes the posterior probability of the hidden state being in an UP state. The estimation error rates (compared with the ground truth) in these two cases are 1.9% and 1.4%, respectively.

This is partly because of the fact that a correct generative model is used and the uncertainties of the hidden state were taken into account during the final estimation (see Figure 5). In the meantime, if α_c is decreased more and more, the mean MUA firing rate will be significantly decreased, the rate difference between UP and DOWN periods is reduced, and therefore the ambiguity between them increases. At some point, it can be imagined that all methods will break down unless the bin size is increased accordingly (at the cost of loss of accuracy in the classification boundary). Due to space limitations, we do not explore this issue further here.

The second issue examines the estimation accuracy of the missing variable against the number of observed variables. It is expected that as more and more observations are added, the resulting discrepancy between the threshold-based method and the probabilistic models will also become smaller, since the uncertainty of the hidden state is less (or the posterior of the hidden variable is larger with more spike train observations). In our simulations, we did extra experiments by either reducing or increasing the number of simulated spike trains, followed by rechecking the results across different setups for different methods. The estimation error results are summarized in Figure 6. Specifically, for the threshold-based method, as more and more spike trains are added, its estimation error gradually improves.

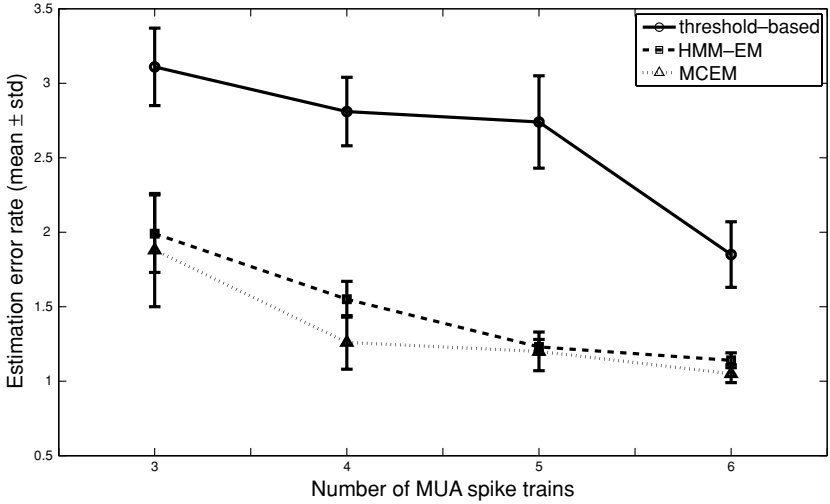


Figure 6: The estimation error comparison of different methods by varying the number of spike train observations (the statistics are computed based on five independent simulated trials). In all conditions, the spike trains are generated using the same conditions: $\mu_c = -3.6$, $\alpha_c = 7.2$, and $\beta_c = 0.05$.

This is expected since the threshold selection criterion (see appendix B) heavily depends on the number of the spike train observations, and adding more spike train observations help to disambiguate the boundary between the UP and DOWN states. Meanwhile, for the probabilistic models, the estimation performance either slightly improves (in the discrete-time HMM) or remains roughly the same (in the continuous-time model). This is partly because adding more observations will also increase the number of parameters to be estimated in the continuous-time model, so the difficulty of inference also increases; whereas the HMM performance is likely to saturate quickly due to either the insufficiency of the model or the local minimum problem inherent in the EM algorithm. This observation implies that the probabilistic models are particularly valuable when the number of spike train observations is relatively small and that the simple threshold-based method becomes more and more reliable in terms of estimation accuracy—yet its performance is still worse than that of two probabilistic models. This is probably because it is difficult to find an optimal kernel smoothing parameter or the two threshold values (see appendix B). Moreover, the threshold-based method cannot produce the statistics of interest (e.g., posterior probability, transition probability).

4.2 Real-World Spike Train Data. Next, we apply our models to validate some real-world simultaneously recorded spike trains collected from

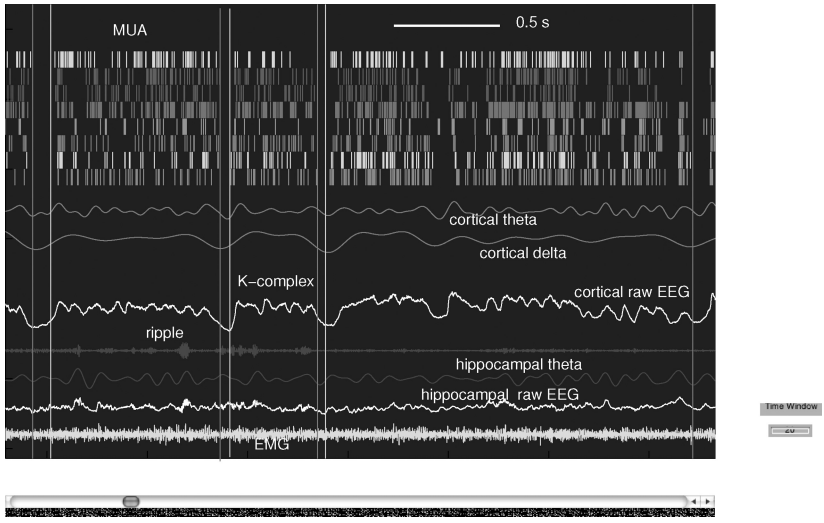


Figure 7: A snapshot of recordings of cortical MUA, raw cortical EEG, cortical theta wave (4–8 Hz), cortical delta wave (2–4 Hz), raw hippocampal EEG, hippocampal ripple power (more than 100 Hz), hippocampal theta wave, and EMG.

one behaving rat (Vijayan, 2007), where the MUA, cortical and hippocampal EEGs, and EMG have been simultaneously recorded (see Figure 7 for a snapshot). We presented the spike train data of a single animal (in one day), recorded from the primary somatosensory cortex (SI) during SWS. Neurophysiological studies of neural spike trains and EEGs across different rats and different recording days, as well as the comparison between the cortex and hippocampus, will be presented elsewhere. In this study, 20 clearly identified cortical cells from eight tetrodes were recorded and sorted. All spiking activity from these eight tetrodes was used to determine the UP and DOWN states.

For this study, we selected about 15.7 minutes of recordings from a total of 11 (interrupted) SWS periods of one rat,[11] where the mean $\pm$ SD length of SWS periods is 85.7 ± 35.8 s (maximum 156.1 s, minimum 30.7 s). We pulled out the multiunit spikes from eight tetrodes (no spike sorting is necessary here). For each spike train (from one tetrode), we empirically chose the

[11]The sleep stage classification was based on the recordings of electromyography (EMG) and hippocampal and cortical EEGs (ripple power, theta and delta power). SWS is characterized as having low EMG, high hippocampal ripple, low hippocampal theta (4–8 Hz), and high cortical delta (2–4 Hz). In practice, we varied the cut-off thresholds of those parameters (via grid search) to obtain a suboptimal SWS classification for a specific rat.

following CIF model, as defined in the continuous-time domain:[12]

$$\begin{aligned}\lambda^c(t) &= \exp\left(\mu_c + \alpha_c S_t + \gamma_c \int_0^t e^{-\beta_c \tau} dN(t-\tau)\, d\tau\right) \\ &\approx \exp\left(\mu_c + \alpha_c S_t + \gamma_c \int_0^{\overline{\tau}} e^{-\beta_c \tau} dN(t-\tau)\, d\tau\right) \\ &\approx \exp\left(\mu_c + \alpha_c S_t + \gamma_c \boldsymbol{\beta}_c^T \boldsymbol{N}_{t-\overline{\tau}:t}\right),\end{aligned}$$

where the exponential decaying parameter β_c is initially set to a value that lets $e^{-\beta_c \tau} \approx 0$ for $\tau > \overline{\tau} = 100$ ms, which leads to the second line of approximation; the third line of approximation appears when we replace the continuous convolution with a discrete vector product, in which $\boldsymbol{\beta}$ denotes the vector containing 100 coefficients sampled from $e^{-\beta_c \tau}$ with a 1 ms temporal resolution, and $\boldsymbol{N}_{t-\overline{\tau}:t}$ denotes the vector containing 100 0 or 1 elements that indicate, respectively the absence or presence of spikes. For the initial values, we set $\gamma_c = 0.01$ for all spike trains; μ_c and α_c were hand-tuned based on a small data set.[13]

Since $\boldsymbol{\theta}$ will be largely dependent on $\mathcal{S}$ in the MCEM algorithm, a sensible choice of initial state $\mathcal{S}^{(0)}$ is important for the convergence of the MCMC sampler. We initialized the state with the results obtained from the discrete-time HMM (10 ms bin size) and interpolated the intermediate missing values in the continuous-time domain (1 ms bin size). The rate parameter defined for the HMM (see equation 2.5) was assumed as follows:[14]

$$\lambda_k = \exp(\mu + \alpha S_k + \beta_1 n_{k-2:k-1} + \beta_2 n_{k-4:k-2} + \beta_3 n_{k-6:k-4}),$$

and $n_{k-2:k-1}$ defines the number of spike counts (across all spike trains) within the previous 10 ms time interval prior to time index k or t_k (with bin size 10 ms). Therefore, the spiking history dependence is described by the number of spike counts in the past three history windows: 10–20 ms, 20–40 ms, and 40–60 ms. The initial parameters were set as $\mu = -0.5, \alpha = 1, \boldsymbol{\beta} \equiv [\beta_1, \beta_2, \beta_3]^T = \mathbf{0}$. The discrete-time HMM converged after about 100 iterations when the log likelihood stops to increase. After

[12]The model was empirically verified by model selection based on the GLM fit of a small data set using the glmfit function in Matlab. The model with the lowest deviance (defined by twofold negative log likelihood) or the smallest Akaike's information criterion (AIC) value will be chosen.

[13]A sensible initial parameter value will be an important factor for obtaining a fast convergence of the simulated Markov chain. In practice, one can fit a small spike train data set (with preidentified hidden state values) with a GLM.

[14]We computed the mean and variance of spike counts given all 10 ms time bins and obtained a mean 2.42 and a variance 4.45. The deviance between the mean and variance statistics suggested that the inhomogeneous Poisson probability model is inaccurate, and this fact motivated us to include history-dependent covariates in the rate parameter.

Table 5: State Estimation Discrepancy Between the Proposed Algorithms and the Threshold-Based Method for the Real-World Spike Trains Data.

Algorithm	Discrepancy Percentage	Number of Jumps, n	Bin Size
Threshold-based	—	2986	10 ms
Discrete HMM-EM	4.42%	3223	10 ms
Continuous MCEM	3.04%	2576	1 ms

Table 6: Estimated Statistics of the UP and DOWN State Durations (in msec) for the Real-World Spike Train Data.

	UP State			DOWN State		
Sojourn Duration Length	Threshold Based	HMM-EM	MCEM	Threshold Based	HMM-EM	MCEM
Minimum	40	40	58	40	20	53
Maximum	8510	4450	5059	270	290	302
Median	510	330	446	60	60	77
Mean $\pm$ SD	739 ± 727	507 ± 493	644 ± 638	67 ± 31	74 ± 36	83 ± 37

that, we ran the Viterbi algorithm to obtain an initial guess of $\{n, \boldsymbol{\tau}, \boldsymbol{\chi}\}$ for the continuous-time model. It is assumed that if $S_k^{(0)} = S_{k+1}^{(0)}$, the same state spans the region $[k\Delta, (k+1)\Delta]$ ($\Delta = 10$ ms), while if $S_k^{(0)} \neq S_{k+1}^{(0)}$, then a single jump occurs at time $(k + 0.5)\Delta$. Furthermore, we initialized the parameters of individual spike trains that were obtained from a GLM fit (based on about 500 ms of empirical data analysis; see note 12). The HMM estimation results are summarized in Tables 5 and 6. Since there is no ground truth about the latent process for the real-world data, we compared the HMM's state estimate with that obtained from the threshold-based method. It appears that the HMM tends to discover more state transitions than the threshold-based method (see Table 5), some of which might be false alarms. Figure 8 presents an illustrated example.[15] In order to determine which estimation result (from both methods) is correct, we might require other available information (such as the cortical EEG or hippocampal EEG) to help determine the "true" state.[16] Direct comparison of different methods is difficult for real data since there is no single ground truth. Typically it was found that the HMM method yields more frequent state jumps than the

[15] A video demo file is provided online (https://neurostat.mit.edu/zhechen/UpDownDemo.avi) for readers interested in a detailed result comparison for a selected 5 minute recording.

[16] The cortical EEG averages have special waveforms triggered by the start and the end times of the UP state; furthermore, ripple events (150–300 Hz) occur much more frequently during the UP state (Ji & Wilson, 2007).

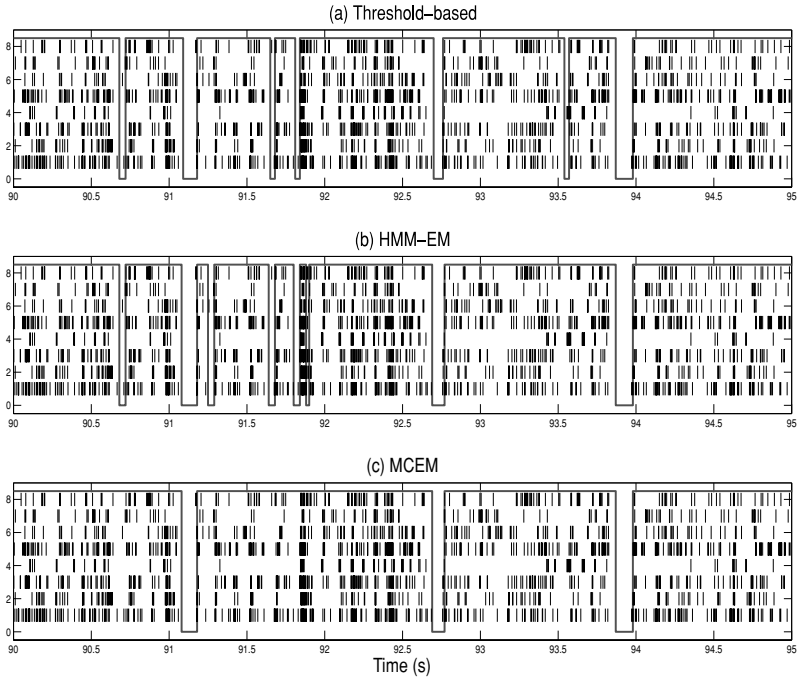


Figure 8: Real-world MUA spike trains of eight tetrodes recorded from the primary somatosensory cortex of one rat (note that each tetrode might contain varying number of single cells). (a) A selected 5 s segment of the MUA spike trains during SWS and its UP and DOWN state classification via the threshold-based method (segmented by the thick solid line). (b) The hidden state estimation result obtained from the discrete-time HMM (used as the initial state for the continuous-time RJMCMC sampler). (c) The hidden state estimation obtained from the MCEM algorithm. In this example, the MCEM algorithm merged several neighboring sojourns that were decoded differently from the HMM.

threshold method (see Table 5); this is simply due to the fact that while estimating the hidden state, the algorithm does not consider the neighboring state information and evaluates only the individual likelihood within each single time interval (of 10 ms); this tends to yield many single-state jumps with short durations. In contrast, the threshold-based method is designed to merge those short silent intervals with their neighboring sojourns (see step 3 in appendix B). However, the selection of the threshold is rather ad hoc, and the classification results require many hand-tuned parameter setups (such as kernel smoothing, bin size, and minimum SWS cut-off length), which does not offer a robust and consistent criterion across different data sets.

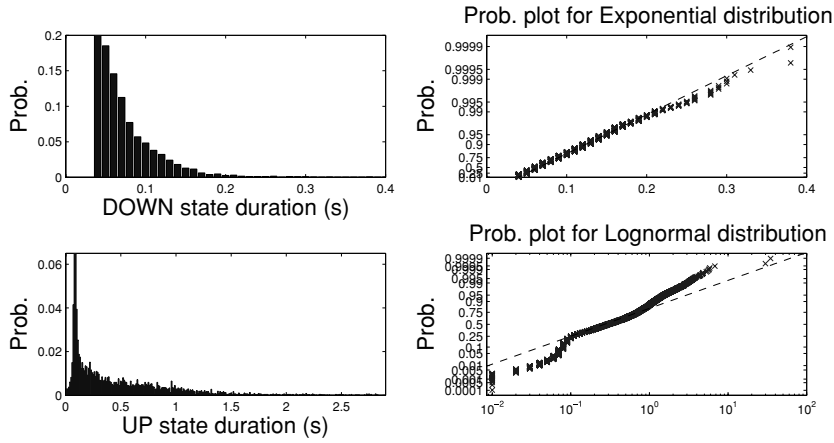


Figure 9: Fitting the real-world sojourn-time duration length for the DOWN and UP states, where the UP or DOWN state classification is obtained from the discrete-time HMM estimation result. (Left panels) Histograms. (Right panels) Fitting the sojourn durations with exponential (for the DOWN state) and log-normal (for the UP state) distributions. If the sample data fit the tested probability density, the data points will approximately match the straight line in the plot.

In order to choose a proper parametric model for the sojourn time duration for the UP and DOWN states, we used the state classification result from the discrete-time HMM (see Figure 9). Based on the histogram data analysis, we chose the exponential pdf as the probability model for the sojourn duration of the DOWN state and the log-normal pdf as the probability model for the sojourn duration of the UP state. We also computed their sample statistics (mean, SD) and used them as the initial parameters for the continuous-time probabilistic model. The lower bounds for the UP and DOWN state duration lengths are both set as 40 ms.

Since the recording time of the real-world spike trains data is rather long (about 60 times longer than the synthetic data), the computational overhead is much greater for the MCEM algorithm. In RJMCMC sampling, 300,000 iterations were run to simulate the Markov chain,[17] and the first 3000 iterations were discarded as the burn-in period. After that, we fed the obtained parameter estimates using the complete data set. After an additional 100 MCEM iterations, the algorithm converged (when the iterative log-likelihood increase is sufficiently small), and we obtained the final parameter estimates. With these estimates, we simulated another 1000 realizations of the semi-Markov chain $\mathcal{S}$ and used them for the final hidden

[17] In implementation by Matlab version 7.0, that roughly amounts to about 15 hours of CPU time in a personal computer equipped with an Intel Core 2 Du processor.

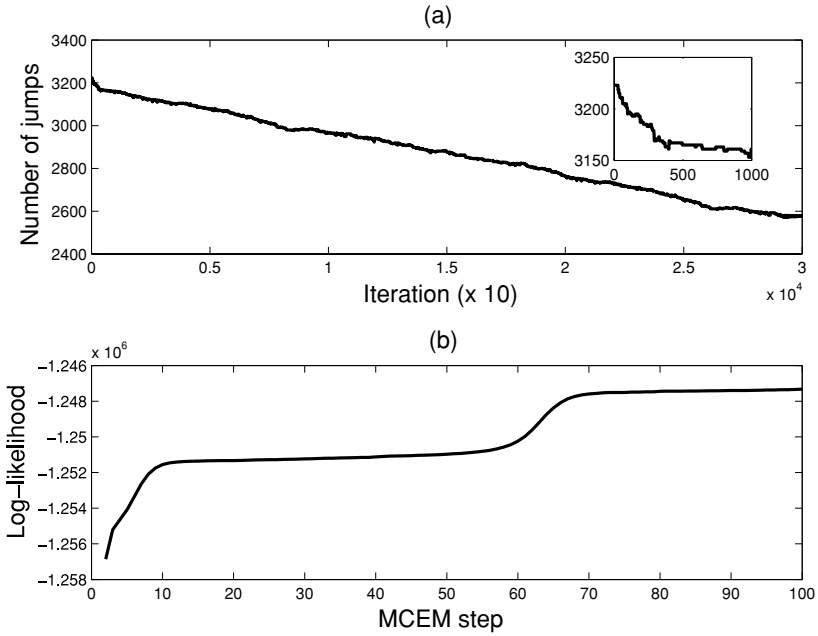


Figure 10: Convergence plot of the simulated Markov chain. (a) Trajectory of the number of state jumps (inset: the trajectory within the first 1000 iterations). (b) Trajectory of the log likelihood in running the MCEM algorithm.

state reconstruction (see equations 3.22 and 3.23). The convergence plots of the semi-Markov chain and the MCEM algorithm are shown in Figure 10.

Several noteworthy comments are in order:

- As a comparison, we also used the estimated hidden state $\{S(t)\}$ to fit a GLM model (using glmfit function in Matlab, $\Delta = 1$ ms) by modeling the history dependence with eight discrete windows (1–5, 5–10, 10–15, 15–20, 20–30, 30–40, 40–50 ms). Upon fitting the GLM model, we obtained the estimated spiking history dependence coefficients for the individual spike trains (see Figure 11); as seen from the results, their curves all have an approximately exponential-decaying shape. Finally, the KS plots and autocorrelation plots are shown in Figures 12 and 13, respectively. Overall, the goodness of fit is quite satisfactory.
- In comparison with the discrete-time HMM-EM method (see Tables 5 and 6), the continuous-time MCEM method yields less frequent state jumps. As a consequence, the MCEM result is accompanied with less short sojourn durations since it allows a potential merge of neighboring sojourns during the RJMCMC procedure (see move type 3 in appendix A) that considers the joint likelihoods of the neighboring

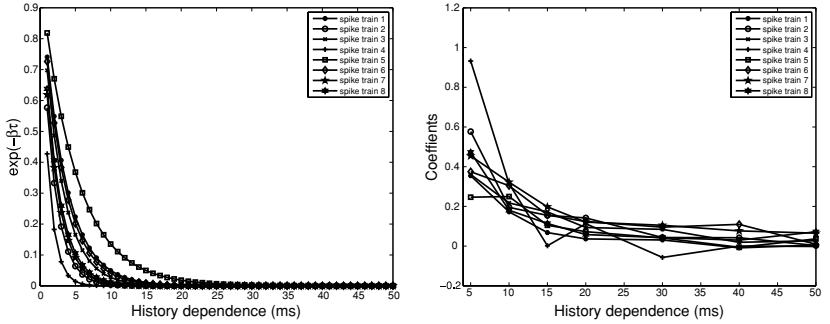


Figure 11: (Left panel) Estimated exponential decaying filters $e^{-\beta_c \tau}$ for the recorded spike trains shown in Figure 8. (Right panel) Estimated history dependence coefficients estimated for the eight spike trains (based on GLM fit using seven discrete windows of history spike counts: 1–5, 5–10, 10–15, 15–20, 20–30, 30–40, 40–50 ms). The estimated history-dependent firing coefficients exhibit an exponential-like decaying curve (for all eight spike trains).

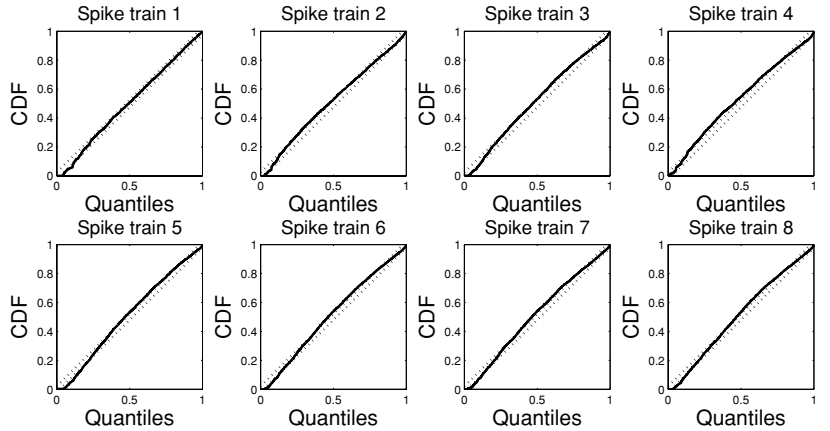


Figure 12: Fitted KS plots of the real-world MUA spike trains (dotted lines along the diagonal indicate the 95% confidence bounds).

sojourns. Furthermore, in comparison with the threshold-based method, the continuous-time semi-Markov model is more powerful in representing the uncertainty as well as inferring the underlying neural dynamics. Its estimated model parameters (the shape of the transition and duration probability density) might reveal the some neural mechanism or physiology behind the transitory dynamics (e.g., the inhibitory period after the last transition event). In our experiments, the MCEM method obtained the lowest estimate of the number of state transitions (see Table 5), yielding a transition occurrence

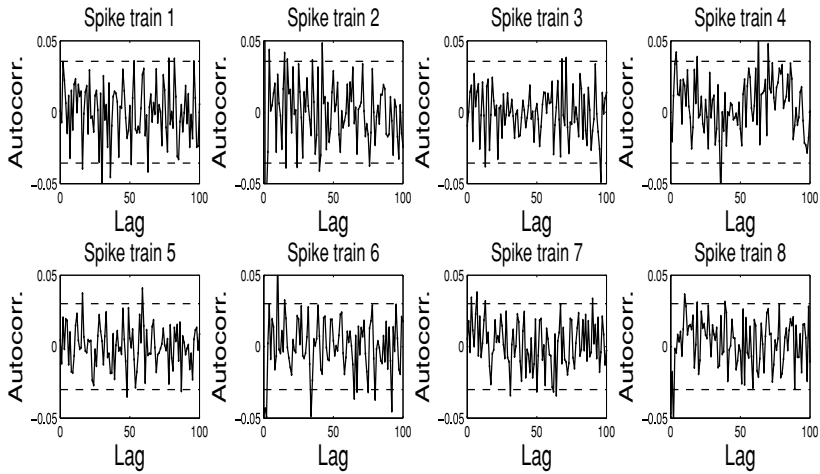


Figure 13: Autocorrelation plots for the real-world MUA spike trains (dashed lines indicate the 95% confidence bounds).

rate about 82 min^{-1} (slightly greater than the rate reported in visual cortex; Ji & Wilson, 2007). Despite its flexibility, the MCEM method is much more computationally intensive than the HMM-EM method. The implementation of HMM is simpler and has a faster convergence speed (the EM algorithm typically converged within 100–200 steps, although the local maximum problem remains). In contrast, the MCEM method relies on simulation of state sequences at every iteration and is required to evaluate the point-process joint likelihood (see equation 3.5) for each possible move. The calculation of every single spike's likelihood contribution is time-consuming and is a bottleneck in the computation. In addition, the convergence speed of the MCEM algorithm becomes slower in the end. This is because when the Markov chain gradually approaches the equilibrium, many moves are rejected and a small modification of the hidden state $\mathcal{S}$ or the parameter $\boldsymbol{\theta}$ would not change very much in terms of the joint log likelihood of the data. This can be seen in the flat plateau of the log-likelihood curve near the end of the convergence in Figure 10b. For the real-world spike train data, the simulation of Markov chain needs to be very long in order to pass through all of move possibilities, especially if the number of potential state transitions is large (here, on the order of thousands). Even so, no optimum stop criterion can be provided with a guarantee; hence, the trade-off between the computation cost and the estimation accuracy remains in any Monte Carlo optimization problem.

- To compare these three classification methods, we also computed the cortical EEG averages (mean $\pm$ standard error of mean) triggered

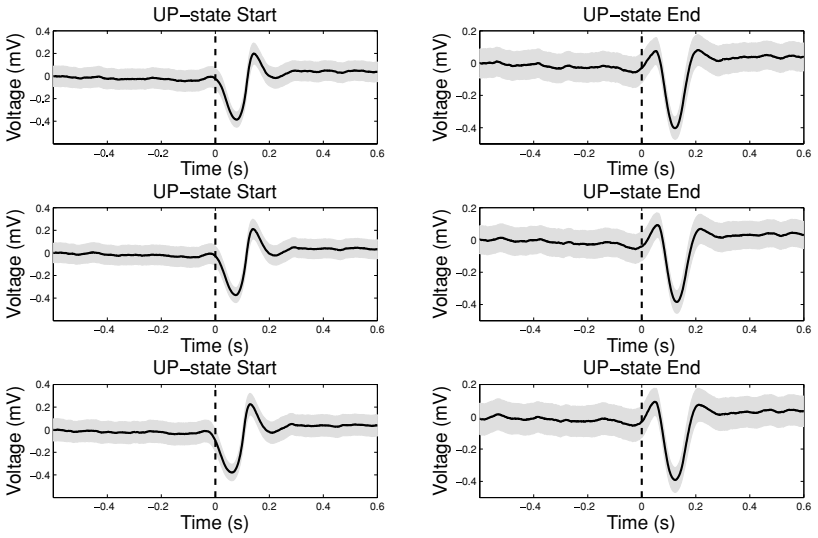


Figure 14: Cortical EEG averages (mean ± standard error of the mean, shown by trace width) triggered by the classified UP state start and end time stamps (for visualization purposes, the standard error of the mean in all plots is amplified by 10 times its original value). From top to bottom: Results from the threshold-based method, the HMM method, and the MCEM method. The start of the UP state is aligned with the K-complex signal that has a biphasic wave switching from a negative dip to a positive peak, which lasts about 200 ms.

by the their UP state starting and ending time stamps, respectively (recall note 15). The results are compared in Figure 14. Although the figures all look similar (due to large timescale), on close examination of the plots, it appears that the EEG averages from the MCEM method result in a more accurate detection of the onset of the UP state.

- Given the MUA spike train data analyzed for the behaving rat, the latent process S_t stays longer during the UP state than the DOWN state, indicating that the population neurons remained dominantly active during SWS.[18] Whether these neuronal firing patterns contain any "memory replay" compared with the earlier firing pattern during the RUN behavior will be the subject of future investigation.

4.3 Firing Pattern Analysis Within the UP States. As observed from the analysis of the recorded multiple spike trains, the somatosensory cortical

[18]This is in contrast to the anesthetized animals, in which the DOWN states occupy a larger fraction of time than the UP states.

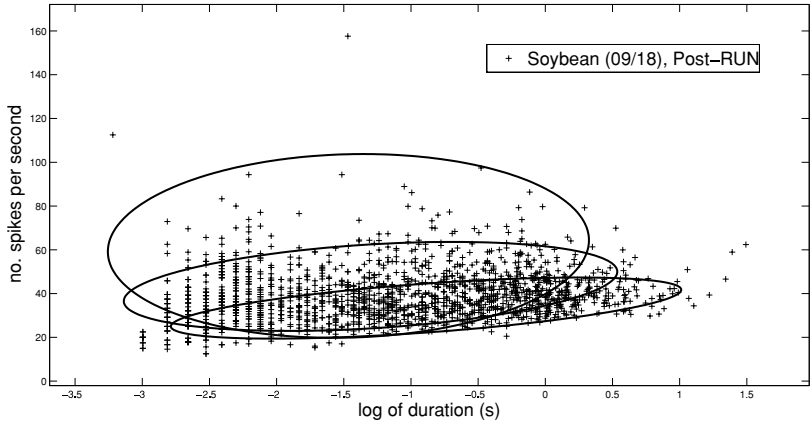


Figure 15: Gaussian mixture clustering for the two firing features (log of duration and number of spikes per second). Here, the optimal number of mixtures is 3; the ellipses represent the two-dimensional gaussian shapes with different covariance structures.

neurons undergo near-synchronous transitions between the UP and DOWN states, from every tens of milliseconds to a few seconds or so. The neuronal firing activities inside the UP state are mainly characterized by duration length and spiking rate. It would be interesting to see if there are any firing "patterns" embedded in these UP-state periods, in either multiunit or single-unit activity (e.g., Luczak et al., 2007; Ji & Wilson, 2007).

On estimating the latent state process, we obtain two features: one is the log (natural basis) of duration, the other the number of spikes per second per tetrode. After collecting these two features from the experimental data shown earlier, we resort to the clustering tool for feature visualization. The soft-clustering algorithm we use here is a greedy clustering algorithm (Verbeek, Vlassis, & Kröse, 2003) based on fitting a gaussian mixture model. In the greedy learning algorithm, the optimal number of mixtures is automatically determined during the learning process. The algorithm was run 20 times, and the best result (with the highest log likelihood) was chosen (see Figure 15). Hence, the neuronal firing pattern can be characterized by a finite number of parameters (mean and covariance), from which we can compare the different neuronal firing across different animals, different days, different brain regions (cortex versus hippocampus), different sleep stages (Pre-RUN versus Post-RUN), and so on. Since this article mainly focuses on the statistical modeling methodology, further quantitative data analysis and its link to neurophysiology will be presented and discussed elsewhere.

5 Discussion

5.1 Model Mismatch or Misspecification for the Spike Train. When investigating the real-world recording spike trains data, an important task of computational modeling is to identify the functional form of CIF (see equations 2.3 and 3.3 or 3.4). Unlike the simulated spike train data, the CIF of the real-world spike trains is not known in advance and needs to be identified before the probabilistic inference is carried on. In this article, for simplicity, we have presumed that the CIF can be approximated by a GLM (McCullagh & Nelder, 1989; Truccolo et al., 2005) which includes the hidden state and firing history as variables. We have also assumed that the spike trains across tetrodes are mutually independent. Most likely, the neuronal spiking is influenced not only by its own firing history but also by the other spike trains. Despite these simplifications, we think the models presented here still serve as a valuable first step to represent the temporal dynamics of the observed MUA spike trains. Above all, to quote George Box, "All models are wrong, but some are useful." In addition, theoretically, given sufficient data and under some regular conditions (Pawitan, 2001), the MLE for a GLM is consistent even when the model (e.g., the link function) is chosen incorrectly (Paninski, 2004). From a practical point of view, we have ignored the possible estimation bias here.

For the real-world spike train data, there is no ground truth available for $\mathcal{S}$. A common practice is to select a small data set, and the UP and DOWN states are first identified by the threshold-based or the HMM method and reconfirmed by human inspection (with extra help of EEG measurements). Based on that information and the assumption that the CIF might be identified by a GLM, we can use the GLM fit for model selection. The model fit would be shown by the deviance and validated by the KS test. If the KS plot falls inside the 95% confidence intervals, it indicates that the CIF model fits well with the given spike train data. Unfortunately, in practice, this is not always the case given only a limited amount of the observed data and an economical size of parameter space for the GLM, indicating a lack of discrepancy between the model and the data.

5.2 Discrete Probability Model for the Sojourn Time. In this article, the sojourn time survival function for the UP and DOWN states is assumed and modeled as being continuous and parametric. More generally, if the histogram analysis of the data indicates that the true distribution is far away from any parametric (exponential or nonexponential) probability density, we might also employ a discretized probability model for the survival probability of the sojourn time. Specifically, let $[a, b]$ denote the range for the sojourn time; we may split the range evenly into L bins and model the discrete probability at each piece as $P_i(x) = \Pr\{a + (i-1)(b-a)/L \le x < a + i(b-a)/L\}$ $(i = 1, 2, \ldots, L)$. Then the probabilistic model of the sojourn

time will be fully characterized by two sets of the parameters, $\{P_i^{up}\}$ and $\{P_j^{down}\}$, where $\sum_{i=1}^{L} P_i^{up} = 1$ and $\sum_{j=1}^{L} P_j^{down} = 1$. In this case, the inference algorithm will be slightly different in that the M-step of the MCEM algorithm will be modified with a reestimation procedure (see equation 3.18), but the E-step remains unchanged.

5.3 Adding Intermediate States. Although in this article, we have exclusively discussed a two-state (0 and 1) Markov model, it is easy to extend the framework to a general N-state Markovian model. Indeed, it is quite possible to add an intermediate state between DOWN and UP as the transitory state. The reason for this argument arises from the observation from the real-world MUA spike trains: in many circumstances, there is no clear evidence that the MUA are either up-modulated or completely silent. Nor does the EEG in any obvious fashion help to differentiate these ambiguous periods. Unfortunately, how to define a transitory state presumably remains a nontrivial problem, and no attempt was made to explore this direction here.

5.4 Fully Bayesian Inference. In the MCEM algorithm discussed above, we consider Monte Carlo sampling only in the E-step, whereas the M-step uses a standard deterministic optimization procedure. Potentially we can use the MCMC procedure for both state and parameter estimation $\{\mathcal{S}^{(k)}, \boldsymbol{\theta}^{(k)}\}$ for $k = 1, 2, \ldots, M$, from which we can obtain the full posterior distribution $p(\mathcal{S}, \boldsymbol{\theta} \mid \mathcal{Y})$ instead of the marginal $p(\mathcal{S} \mid \boldsymbol{\theta}, \mathcal{Y})$. Take, for example, the parameters associated with the sojourn time pdf; we can define the gamma prior for the exponential distribution or a conjugate prior for the inverse gaussian (Banerjee & Bhattacharyya, 1979); for the parameters associated with the CIF model, we may define a gaussian prior. In this case, the M-step would be replaced by iterative Gibbs sampling. The detailed exploration of such a fully Bayesian inference approach, however, is beyond the scope of this article.

5.5 Limitation of Our Approach. There are several obvious assumptions used in our statistical modeling approach. First, the statistical mutual independence is assumed across neural spike trains, without explicit modeling of the recurrent network activity.[19] Second, the observed data are assumed to be stationary in the sense that the state transition and the CIF parameters are estimated from a long period of spike train recordings when those parameters are assumed to remain constant. Finally, an identical CIF model is also assumed across all neural spike trains. Nevertheless, these limitations by no means diminish the value of the models and methods

[19] Recently, complementary work has been reported in modeling self-organized recurrent network model of excitatory and inhibitory neurons for spontaneous UP and DOWN state transitions (Kang, Kitano, & Fukai, 2008).

proposed here, since this article can be treated as a pilot study toward the ultimate modeling goal.

5.6 Future Work. We have considered several future investigation efforts in the line with the work reported here. From a computational modeling point of view, we can extend the model by including continuous-valued observations (e.g., Srinivasan, Eden, Mitter, & Brown, 2007). For instance, the LFP or EEG measurements have been simultaneously recorded from both cortical and hippocampal regions. The detection of K-complexes from the cortical EEG and detection of the sharp wave-ripple complexes (SPW-Rs) from the hippocampal EEG would be beneficial to the identification of UP and DOWN states (Siriota et al., 2003; Battaglia et al., 2004; Ji & Wilson, 2007). Furthermore, it is possible to build a more complex SSM by allowing both continuous- and discrete-valued hidden variables—for instance, a switching SSM where the two latent processes interact with each other (e.g., Ghahramani & Hinton, 2000; Srinivasan et al., 2007).

From a neurophysiological point of view, we are also interested in studying the population neuronal firing causality and latency between the cortex and hippocampus, as well as their spike patterns relevant to the rat's RUN behavior. It is well known that sleep is a key factor that may promote the transfer of memory from the hippocampus to the cortex, and during sleep, replays in these two regions occur synchronously (Mehta, 2007; Ji & Wilson, 2007). Based on the extracelluar recordings (MUA and LFP), it would be interesting to investigate the UP and DOWN activities during multiple processing stages and sites in the cortico-hippocampal circuits, and the UP and DOWN state transitions can be used to quantify the functional connectivity of the neural circuits. An in-depth exploration of the LFP (e.g., K-complexes, SPW-Rs), with single- and multiunit firing activities in both cortical and hippocampal regions, would be key to understanding memory consolidation during sleep.

6 Conclusion

We have developed both discrete- and continuous-time probabilistic models and inference algorithms for inferring population neurons' UP and DOWN states, using the MUA spike trains. Compared to the deterministic threshold-based method (see appendix B) used in the literature, our probabilistic paradigms offer a stochastic approach to analyze the spike trains as well as provide a generative model to simulate the spike trains. Furthermore, the hidden state estimate is treated as a random variable with certain uncertainty (encoded by its posterior probability), whereas the threshold-based method cannot represent such uncertainties.

The discrete-time HMM provides a reasonable state estimate with a rather fast computing speed. However, the model is restricted to locate the UP and DOWN state transition with a relatively large time bin size (here,

10 ms). Another drawback of the HMM is that it is prone to get stuck in the local solution; in other words, the number of state transitions typically remains unchanged after a few EM iterations. In contrast, one advantage of the continuous-time probabilistic model is that it allows estimating the exact locations of state jumps. By using the RJMCMC sampling technique, the number of jumps as well as the locations of the jumps are allowed to be modified during the inference procedure, which offers a way to escape from the local minimum and tackle the model selection problem. The only shortcoming of the RJMCMC method is its greater computational complexity and the tremendous demand of computational power. In practice, the number of steps required to reach equilibrium often demands sensible initial conditions and diagnostic monitoring during the convergence process. We found that the inference solution obtained from the discrete-time HMM yields a reasonable initial state sequence to feed into the MCMC sampler. Once the number and the locations of the state jumps are determined, we can use the Monte Carlo statistics to infer the latent process. For practitioners who are more concerned about the processing speed than the accuracy of hidden state estimation, the discrete-time HMM might offer a reasonable guess (depending on the data characteristics). Nevertheless, no claim is made here that our proposed models and algorithms could always produce a correct UP or DOWN state classification result. The final justification might still rely on careful human inspection, but our estimation results certainly provide a good starting point with high confidence for follow-up.

In analyzing the simultaneously recorded spike trains, identifying an accurate CIF model is crucial to the probabilistic inference. However, there is no free-lunch-recipe to obtain the ultimate answer. In practice, it often requires some empirical data analysis (Brown, Kass, & Mitra, 2004; Kass, Ventura, & Brown, 2005), such as the interspike interval histogram, firing-rate trend dependence analysis, or fitting a GLM (Truccolo et al., 2005) or a non-Poisson model (Barbieri, Quirk, Frank, Wilson, & Brown, 2001; Brown et al., 2003).

Finally, we hope that our proposed statistical models can shed some light on developing physiologically plausible mechanistic models. A better understanding of the transition mechanism between the UP and DOWN states would also help to improve the statistical description of the data.

Appendix A: Reversible-Jump MCMC

A.1 Background. Reversible-jump MCMC (RJMCMC) is a Metropolis-Hastings-type sampling algorithm with a transdimensional proposal. The term *reversible jump* refers to the ability of the Markov chain to "jump" between two parameter spaces that have different dimensions. The sampler explores the state spaces of variable dimensionality by various modifications through the Metropolis-Hastings proposals. Each Metropolis-Hastings proposal has a respective reverse proposal. For every

proposal, the acceptance probability is computed according to a certain rule. The goal of the RJMCMC algorithm is to design efficient moves that allow the simulated Markov chain to reach the desired equilibrium (posterior) distribution within a reasonable amount of time. Unlike the fixed-dimensional MCMC algorithms, RJMCMC allows the state transitions to occur between spaces with different dimensions, say, $\mathcal{S} \to \mathcal{S}'$, where $dim(\mathcal{S}) \neq dim(\mathcal{S}')$.

In this appendix, we present a detailed elaboration of the RJMCMC algorithm in the context of simulating a continuous-time (semi-) Markov chain for the problem. In the following, we use similar notations and formulations of Ball et al. (1999).

A.2 Derivation. Let n denote the number of jumps between two distinct discrete states in the latent process $\{S(t); 0 \le t \le T\}$, where $S(t) \in \{0, 1\}$. Let $\mathcal{S} = (n, \boldsymbol{\tau}, \boldsymbol{\chi})$ be a triplet of the (semi-) Markov process, where $\boldsymbol{\tau} = (\tau_0, \tau_1, \ldots, \tau_n)$ is a vector that contains the duration of the sojourn time of $\mathcal{S}$, and $\boldsymbol{\chi} = (\chi_0, \chi_1, \ldots, \chi_n)$ represents the states visited in these sojourns. Let $\nu_0 = 0$, $\nu_j = \sum_{i=0}^{j-1} \tau_i$ $(j = 1, 2, \ldots, n+1)$, and $\nu_{n+1} = T$. Furthermore, we assume that for both UP and DOWN states, the sojourn time duration τ has a lower bound a $(\tau \ge a > 0)$ but no upper bound; consequently, the associated pdf has to be rectified accordingly.

The following seven types of moves are considered in the Metropolis-type proposal:

1. Move a boundary between two successive sojourns of $\mathcal{S}$. First, decide which boundary to move by sampling j uniformly from $\{0, 1, \ldots, n-1\}$. Let $\chi_j = i_1$, $\chi_{j+1} = i_2$, and then we have two alternative sampling options:[20]
   - Sample u from a uniform distribution $\mathcal{U}(a_{i_1}, \tau_j + \tau_{j+1} - a_{i_2})$, where a_{i_1}, a_{i_2} denote the lower-bound constraints of the sojourn time of states χ_j and χ_{j+1}, respectively. The proposal $\mathcal{S}'$ is obtained by moving the boundary between the jth and $(j+1)$st sojourns from ν_{j+1} to $\nu_j + u$. In this case, $\mathcal{S}' = (n', \boldsymbol{\tau}', \boldsymbol{\chi}')$, where $\tau'_j = u$, $\tau'_{j+1} = \tau_j + \tau_{j+1} - u$.
   - Sample u from a gaussian distribution $\mathcal{N}(0, \sigma^2)$, where σ^2 denotes the (user-specified) variance parameter. The proposal $\mathcal{S}'$ is obtained by moving the boundary between the jth and $(j+1)$st sojourns from ν_{j+1} to $\nu_{j+1} + u$. In this case, $\mathcal{S}' = (n', \boldsymbol{\tau}', \boldsymbol{\chi}')$, $\tau'_j = \tau_j + u$, $\tau'_{j+1} = \tau_{j+1} - u$,

   For both sampling options, $n' = n$, $\tau'_l = \tau_l$ $(l \neq j, j+1)$, and $\chi'_l = \chi_l$ $(l = 1, 2, \ldots, n')$.

[20] Given a reasonably accurate initial state, the second sampling option would be more efficient since its rejection rate would be lower. For the current continuous-time estimation problem, the SD parameter σ in the second sampling option is chosen to be 2 ms, twice that of the bin size.

2. Insert a sojourn in one of the existing sojourns of $\mathcal{S}$. First, sample l^* uniformly from $\{0, 1, \ldots, n\}$, and let $i = \chi_{l^*}$. Determine the state for the inserted sojourn by sampling j from the probability $\tilde{p}_1(j \mid i, \boldsymbol{\theta})$ $(j \neq i)$. In the two-state space, $\tilde{p}_1(j \mid i) = 1$ (i.e., deterministic). Sample u from a uniform distribution $\mathcal{U}(a_i, \tau_{l^*} - a_i)$ (where a_i denotes the lower bound of the sojourn time for state i), and then sample v from the censored version of the exponential (or log-normal or inverse gaussian) distribution with parameters $\boldsymbol{\theta}_j \equiv r_j$ (or $\boldsymbol{\theta}_j \equiv \{\mu_j, \sigma_j\}$ for log normal, or $\boldsymbol{\theta}_j \equiv \{\mu_j, s_j\}$ for the inverse gaussian) truncated at $\tau_{l^*} - u - a_i$, that is, from the distribution that has the following conditional pdf $\tilde{p}(v \mid u)$

$$\tilde{p}(v \mid u) = \frac{p_v(\boldsymbol{\theta}_j; v)}{F_v(\boldsymbol{\theta}_j; \tau_{l^*} - u - a_i) - F_v(\boldsymbol{\theta}_j; a_j)} \times (a_j \leq v \leq \tau_{l^*} - u - a_i), \tag{A.1}$$

where a_j denotes the lower bound of the sojourn time for state j. Then a sojourn in state j $(j \neq i)$ of length v is inserted in the l^*th sojourn of $\mathcal{S}$. And the new proposal $\mathcal{S}' = (n', \boldsymbol{\tau}', \boldsymbol{\chi}')$ is given by $n' = n + 2$, $(\tau'_l, \chi'_l) = (\tau_l, \chi_l)$ $(l = 0, 1, l^* - 1)$, $\tau'_{l^*} = u$, $\tau'_{l^*+1} = v$, $\tau'_{l^*+2} = \tau_{l^*} - u - v$, $(\tau'_l, \chi'_l) = (\tau_{l-2}, \chi_{l-2})$ $(l = l^* + 3, l^* + 4, \ldots, n + 2)$, $\chi'_{l^*} = \chi'_{l^*+2} = \chi_{l^*} (= i)$, $\chi'_{l^*+1} = j$. Note that if $\tau_{l^*} - u - 2a_i < a_j$, move 2 will not be executed.
3. Delete an intermediate sojourn of $\mathcal{S}$ whose two adjacent sojourns are in the same state (namely, merge one sojourn with its neighboring sojourns). Sample l^* uniformly from $\{0, 1, \ldots, n - 1\}$, and delete the l^*th sojourn of $\mathcal{S}$. The new $\mathcal{S}'$ is given by $n' = n - 2$, $(\tau'_l, \chi'_l) = (\tau_l, \chi_l)$ $(l = 0, 1, \ldots, l^* - 2)$, $\tau'_{l^*-1} = \tau_{l^*-1} + \tau_{l^*} + \tau_{l^*+1}$, $\chi'_{l^*-1} = \chi_{l^*-1}$, $(\tau'_l, \chi'_l) = (\tau_{l+2}, \chi_{l+2})$ $(l = l^*, l^* + 1, \ldots, n - 2)$.
4. Split the first sojourn of $\mathcal{S}$. First, let $i = \chi_0$, and sample u from the uniform distribution $\mathcal{U}(a_j, \tau_0 - a_i)$, where a_i and a_j denote the lower-bound constraints of the sojourn time for states i and j, respectively $(j \neq i$; in the two-state case, the choice of j is deterministic). The new proposal $\mathcal{S}'$ is given by $n' = n + 1$, $\tau'_0 = u$, $\chi'_0 = j$, $\tau'_1 = \tau_0 - u$, $\chi'_1 = i$, $(\tau'_l, \chi'_l) = (\tau_{l-1}, \chi_{l-1})$ $(l = 1, 2, \ldots, n - 1)$. Note that if $\tau_0 < a_i + a_j$, move 4 will not be executed.
5. Delete the first sojourn of $\mathcal{S}$. This move is deterministic. The new proposal $\mathcal{S}'$ is given by $n' = n - 1$, $\tau'_0 = \tau_0 + \tau_1$, $\chi'_0 = \chi_1$, $(\tau'_l, \chi'_l) = (\tau_{l+1}, \chi_{l+1})$ $(l = 1, 2, \ldots, n - 1)$.
6. Split the last sojourn of $\mathcal{S}$. First, let $i = \chi_n$, and sample u from a uniform distribution $\mathcal{U}(a_i, \tau_n)$, where a denotes the lower-bound constraint of the sojourn time for state i. Next, sample j from the distribution $\tilde{p}_3(j \mid i)$ $(j \neq i$; in the two-state state space, the choice of j is deterministic). The new proposal $\mathcal{S}'$ is given by $n' = n + 1$, $(\tau'_l, \chi'_l) =$

(τ_l, χ_l) $(l = 1, 2, \ldots, n-1)$, $\tau'_n = u$, $\chi'_n = \chi_n$, $\tau'_{n+1} = \tau_n - u$, $\chi'_{n+1} = j$. Note that if $\tau_n < a_i$, move 6 will not be executed.

7. Delete the last sojourn of $\mathcal{S}$. This move type is deterministic, and the new proposal $\mathcal{S}'$ is given by $n' = n-1$, $(\tau'_l, \chi'_l) = (\tau_l, \chi_l)$ $(l = 1, 2, \ldots, n-2)$, $\tau'_{n-1} = \tau_{n-1} + \tau_n$, $\chi'_{n-1} = \chi_{n-1}$.

Of the above moves, moves 2 to 7 are discrete and transdimensional, and move 1 is continuous. For the convenience of technical treatment, we classify the seven moves into three classes: $A = \{1\}$, $B = \{2, 4, 6\}$, $C = \{3, 5, 7\}$, which correspond to *boundary move, insertion*, and *deletion*, respectively. Specifically, the individual moves in class B are the respective inverses of those in class C.

The move types chosen for the $\mathcal{S}$ update are obtained by sampling independently from the distribution of q_i $(i = 1, 2, \ldots, 7)$ as follows. First, the class of move type is determined by sampling from the distribution (q_A, q_B, q_C), where, for example, q_B represents the probability that a class B move is chosen. Second, if a class B or class C move is chosen, the specific individual move is found by sampling $\tilde{j}$ from a distribution $(\tilde{q}_1, \tilde{q}_2, \tilde{q}_3)$. By this setup, the probability of q_i will be determined by $q_1 = q_A, q_2 = q_B\tilde{q}_1, q_4 = q_B\tilde{q}_2, q_6 = q_B\tilde{q}_3, q_3 = q_C\tilde{q}_1, q_5 = q_C\tilde{q}_2, q_7 = q_C\tilde{q}_3$. For the problem here, we may use the following setup: $(q_A, q_B, q_C) = (\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$, and $\tilde{q}_1 \propto \frac{n-1}{n+1}$, $\tilde{q}_2 \propto \frac{1}{n+1}$, $\tilde{q}_3 \propto \frac{1}{n+1}$, such that the normalization constraint $\sum_{i=1}^{7} q_i = 1$ is satisfied. The $\tilde{q}_2$ and $\tilde{q}_3$ correspond to probabilities for selecting the first and last sojourns, respectively, and $\tilde{q}_1$ for the other sojourns.

To compute the acceptance probabilities: for $i = 1, 2, \ldots, 7$, let $R^{(i)}(\mathcal{S} \to \mathcal{S}'; \boldsymbol{\theta})$ be the density of the transition kernel associated with the proposal $\mathcal{S}'$ for the move type (i) and let q_i be the probability of choosing the move type (i). Since moves 2 to 7 are discrete (unlike move type 1), their $R^{(i)}(\mathcal{S} \to \mathcal{S}'; \boldsymbol{\theta})$ are probabilities instead of probability densities. Hence, the density of the transition kernel for a new proposal $\mathcal{S}'$ is given by

$$R(\mathcal{S} \to \mathcal{S}'; \boldsymbol{\theta}) = \sum_{i=1}^{7} q_i R^{(i)}(\mathcal{S} \to \mathcal{S}'; \boldsymbol{\theta}).$$

Among the seven move proposals, move type 1 does not change the dimension of $\mathcal{S}$, and its acceptance probability is $\mathcal{A} = \min(1, \mathcal{B})$, where

$$\mathcal{B} = \frac{p(\mathcal{S}', \boldsymbol{\theta}, y) R(\mathcal{S}' \to \mathcal{S}; \boldsymbol{\theta})}{p(\mathcal{S}, \boldsymbol{\theta}, y) R(\mathcal{S} \to \mathcal{S}'; \boldsymbol{\theta})}. \tag{A.2}$$

On the other hand, move types 2 to 7 change the dimension of $\mathcal{S}$, and their acceptance probabilities are given by $\mathcal{A} = \min(1, \mathcal{B})$, where

$$\mathcal{B} = \frac{p(\mathcal{S}', \boldsymbol{\theta}, y) R(\mathcal{S}' \to \mathcal{S}; \boldsymbol{\theta})}{p(\mathcal{S}, \boldsymbol{\theta}, y) R(\mathcal{S} \to \mathcal{S}'; \boldsymbol{\theta})} |\mathcal{J}|, \tag{A.3}$$

where $|\mathcal{J}|$ denotes the determinant of the Jacobian. The Jacobian measures the ratio of the volume of two state spaces.

For presentation convenience, we often factorize the acceptance probability as $\mathcal{B} = \mathcal{B}_1\mathcal{B}_2\mathcal{B}_3$ (prior ratio × likelihood ratio × proposal probability ratio),[21] where

$$\mathcal{B}_1 = \frac{p(\mathcal{S}' \mid \boldsymbol{\theta})}{p(\mathcal{S} \mid \boldsymbol{\theta})}, \quad \mathcal{B}_2 = \frac{p(y \mid \mathcal{S}', \boldsymbol{\theta})}{p(y \mid \mathcal{S}, \boldsymbol{\theta})}, \quad \mathcal{B}_3 = \frac{R(\mathcal{S}' \to \mathcal{S}; \boldsymbol{\theta})}{R(\mathcal{S} \to \mathcal{S}'; \boldsymbol{\theta})}. \tag{A.4}$$

In the context of two-state continuous-time (semi-) Markov chain that is used in this article, we compute these three probability ratios for these seven moves in the following.

A.3 Move Types

A.3.1 Move Type 1. For the first sampling option, let $i_1 = \chi_j$ and $i_2 = \chi_{j+1}$, and let $\lambda^c_{i_1} = \int_{v_j}^{v_{j+1}} \lambda^c(t; S_{v_j:v_{j+1}} = i_1)\, dt$, $\lambda^c_{i_2} = \int_{v_{j+1}}^{v_{j+2}} \lambda^c(t; S_{v_{j+1}:v_{j+2}} = i_2)\, dt$, $\tilde{\lambda}^c_{i_1} = \int_{v_j}^{v_j+u} \lambda^c(t; S_{v_j:v_j+u} = i_1)\, dt$, and $\tilde{\lambda}^c_{i_2} = \int_{v_j+u}^{v_{j+2}} \lambda^c(t; S_{v_j+u:v_{j+2}} = i_2)\, dt$, and let $y^c_{i_1}$, $y^c_{i_2}$, $\tilde{y}^c_{i_1}$, and $\tilde{y}^c_{i_2}$ denote the number of spike counts for spike train c during the time intervals $[v_j, v_{j+1}]$, $[v_{j+1}, v_{j+2}]$, $[v_j, v_j + u]$, and $[v_j + u, v_{j+2}]$, respectively. The probability ratios for move type 1 are calculated as follows

$$\mathcal{B}_1^{(1)} = \frac{\tilde{p}(\boldsymbol{\theta}_{i_1}; u)}{\tilde{p}(\boldsymbol{\theta}_{i_1}; \tau_j)} \frac{\tilde{p}(\boldsymbol{\theta}_{i_2}; \tau_{j+1} + \tau_j - u)}{\tilde{p}(\boldsymbol{\theta}_{i_2}; \tau_{j+1})} \quad \text{(option 1)}$$

$$\mathcal{B}_2^{(1)} = \prod_{c=1}^{C} \left\{ \exp(\lambda^c_{i_1} + \lambda^c_{i_2} - \tilde{\lambda}^c_{i_1} - \tilde{\lambda}^c_{i_2}) \times \frac{\prod_{k=1}^{\tilde{y}^c_{i_1}} \lambda^c(t_k, \mathcal{S}'_{v_j:v_j+u}) \prod_{k=1}^{\tilde{y}^c_{i_2}} \lambda^c(t_k, \mathcal{S}'_{v_j+u:v_{j+2}})}{\prod_{k=1}^{y^c_{i_1}} \lambda^c(t_k, \mathcal{S}_{v_j:v_{j+1}}) \prod_{k=1}^{y^c_{i_2}} \lambda^c(t_k, \mathcal{S}_{v_{j+1}:v_{j+2}})} \right\} \quad \text{(option 1)}$$

$$\mathcal{B}_3^{(1)} = \frac{R(\mathcal{S}' \to \mathcal{S}; \boldsymbol{\theta})}{R(\mathcal{S} \to \mathcal{S}'; \boldsymbol{\theta})} = 1,$$

where $\tilde{p}(\boldsymbol{\theta}; \cdot)$ is defined by the censored version of the parametric pdf of the sojourn time (for either UP or DOWN state). The ratios for the second sampling option are conceptually similar, and we show only $\mathcal{B}_1^{(1)}$ here:

$$\mathcal{B}_1^{(1)} = \frac{\tilde{p}(\boldsymbol{\theta}_{i_1}; \tau_j + u)}{\tilde{p}(\boldsymbol{\theta}_{i_1}; \tau_j)} \frac{\tilde{p}(\boldsymbol{\theta}_{i_2}; \tau_{j+1} - u)}{\tilde{p}(\boldsymbol{\theta}_{i_2}; \tau_{j+1})} \quad \text{(option 2)}.$$

[21] To avoid numerical problems, it is more convenient to calculate the ratios in the logarithm domain.

A.3.2 Move Type 2. Let $i_1 = i, i_2 = j$, and let $y^c_{i_1}$, $\tilde{y}^c_{i_1}$, $\tilde{y}^c_{i_2}$, $\hat{y}^c_{i_1}$ denote the number of spike counts for spike train c during the time intervals $[\nu_{l^*}, \nu_{l^*+1}]$, $[\nu_{l^*}, \nu_{l^*} + u]$, $[\nu_{l^*} + u, \nu_{l^*} + u + v]$, and $[\nu_{l^*} + u + v, \nu_{l^*+1}]$, respectively. Let $\lambda^c_{i_1} = \int_{\nu_{l^*}}^{\nu_{l^*+1}} \lambda^c(t; S_{\nu_{l^*}:\nu_{l^*+1}} = i_1)\, dt$, $\tilde{\lambda}^c_{i_1} = \int_{\nu_{l^*}}^{\nu_{l^*}+u} \lambda^c(t; S_{\nu_{l^*}:\nu_{l^*}+u} = i_1)\, dt$, $\tilde{\lambda}^c_{i_2} = \int_{\nu_{l^*}+u}^{\nu_{l^*}+u+v} \lambda^c(t; S_{\nu_{l^*}+u:\nu_{l^*}+u+v} = i_2)\, dt$, and $\hat{\lambda}^c_{i_1} = \int_{\nu_{l^*}+u+v}^{\nu_{l^*+1}} \lambda^c(t; S_{\nu_{l^*}+u+v:\nu_{l^*+1}} = i_1)\, dt$. Then

$$\mathcal{B}^{(2)}_1 = \frac{\tilde{p}(\boldsymbol{\theta}_{i_1}; u)\tilde{p}(\boldsymbol{\theta}_{i_2}; v)\tilde{p}(\boldsymbol{\theta}_{i_1}; \tau_{l^*} - u - v)}{\tilde{p}(\boldsymbol{\theta}_{i_1}; \tau_{l^*})}$$

$$\mathcal{B}^{(2)}_2 = \prod_{c=1}^{C} \left\{ \exp(\lambda^c_{i_1} - \tilde{\lambda}^c_{i_1} - \tilde{\lambda}^c_{i_2} - \hat{\lambda}^c_{i_1}) \times \frac{\prod_{k=1}^{\tilde{y}^c_{i_1}} \lambda^c(t_k, \mathcal{S}'_{\nu_{l^*}:\nu_{l^*}+u}) \prod_{k=1}^{\tilde{y}^c_{i_2}} \lambda^c(t_k, \mathcal{S}'_{\nu_{l^*}+u:\nu_{l^*}+u+v}) \prod_{k=1}^{\hat{y}^c_{i_1}} \lambda^c(t_k, \mathcal{S}'_{\nu_{l^*}+u+v:\nu_{l^*+1}})}{\prod_{k=1}^{y^c_{i_1}} \lambda^c(t_k, \mathcal{S}_{\nu_{l^*}:\nu_{l^*+1}})} \right\}$$

$$\mathcal{B}^{(2)}_3 = \frac{R(\mathcal{S}' \to \mathcal{S}; \boldsymbol{\theta})}{R(\mathcal{S} \to \mathcal{S}'; \boldsymbol{\theta})} = \frac{\frac{q_3}{n+1}}{\frac{q_2}{n+1}\frac{1}{\tau_{l^*} - 2a_i}\tilde{p}(v \mid u)} = \frac{\tau_{l^*} - 2a_i}{\tilde{p}(v \mid u)},$$

where in computing $\mathcal{B}^{(2)}_3$, we have used $q_2 = q_3$ and $p(u) = \frac{1}{\tau_{l^*} - 2a_i}$, and $\tilde{p}(v \mid u)$ is given by equation A.1. Since move type 2 changes the dimension of the $\mathcal{S}$, we need to compute the associated Jacobian's determinant. Note that $\boldsymbol{\tau}'$ is obtained from $\boldsymbol{\tau}$ from an invertible deterministic function $\boldsymbol{\tau}' \leftarrow \mathcal{T}(\boldsymbol{\tau}, u, v) = (\tau_0, \tau_1, \ldots, \tau_{l^*-1}, u, v, \tau_{l^*} - a_i - u - v, \tau_{l^*+1}, \tau_{l^*+2}, \ldots, \tau_n)$, whose Jacobian is then given by

$$|\mathcal{J}| = \left| \frac{\partial \mathcal{S}'(u, v, \tau^*_l - a_i - u - v))}{\partial \mathcal{S}(u, v, \tau^*_l)} \right| = \begin{vmatrix} \frac{\partial u}{\partial \tau_{l^*}} & \frac{\partial u}{\partial u} & \frac{\partial u}{\partial v} \\ \frac{\partial v}{\partial \tau_{l^*}} & \frac{\partial v}{\partial u} & \frac{\partial v}{\partial v} \\ \frac{\partial(\tau_{l^*} - a_i - u - v)}{\partial \tau_{l^*}} & \frac{\partial(\tau_{l^*} - a_i - u - v)}{\partial u} & \frac{\partial(\tau_{l^*} - a_i - u - v)}{\partial v} \end{vmatrix} = \begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{vmatrix} = 1.$$

A.3.3 Move Type 3. Moves 3 and 2 are inverses of each other. Let $i_1 = \chi_{l^*-1}$ and $i_2 = \chi_{l^*}$, and let $\lambda^c_{i_1} = \int_{\nu_{l^*-1}}^{\nu_{l^*}} \lambda^c(t; S_{\nu_{l^*-1}:\nu_{l^*}} = i_1)\, dt$, $\lambda^c_{i_2} = \int_{\nu_{l^*}}^{\nu_{l^*+1}} \lambda^c(t; S_{\nu_{l^*}:\nu_{l^*+1}} = i_2)\, dt$, $\hat{\lambda}^c_{i_1} = \int_{\nu_{l^*+1}}^{\nu_{l^*+2}} \lambda^c(t; S_{\nu_{l^*+1}:\nu_{l^*+2}} = i_1)\, dt$, $\tilde{\lambda}^c_{i_1} = \int_{\nu_{l^*-1}}^{\nu_{l^*+2}} \lambda^c(t; S_{\nu_{l^*-1}:\nu_{l^*+2}} = i_1)\, dt$, and let $y^c_{i_1}$, $y^c_{i_2}$, $\hat{y}_{i_1}$, and $\tilde{y}^c_{i_1}$ denote the numbers of spike counts for spike train c within the intervals $[\nu_{l^*-1}, \nu_{l^*}]$, $[\nu_{l^*}, \nu_{l^*+1}]$, $[\nu_{l^*+1}, \nu_{l^*+2}]$, and $[\nu_{l^*-1}, \nu_{l^*+2}]$, respectively. It is noted that $\boldsymbol{\tau}'$ can be obtained from an invertible deterministic function $\mathcal{T}(\boldsymbol{\tau}) = (\tau_0, \tau_1, \ldots, \tau_{l^*-1}, \tau_{l^*+1}, \tau_{l^*+2}, \ldots, \tau_n)$, and $|\mathcal{J}| = 1$. The

probability ratios are given as

$$\mathcal{B}_1^{(3)} = \frac{\tilde{p}(\boldsymbol{\theta}_{i_1};\, \tau_{l^*-1} + \tau_{l^*} + \tau_{l^*+1})}{\tilde{p}(\boldsymbol{\theta}_{i_1},\, \tau_{l^*-1})\tilde{p}(\boldsymbol{\theta}_{i_2},\, \tau_{l^*})\tilde{p}(\boldsymbol{\theta}_{i_1},\, \tau_{l^*+1})}$$

$$\mathcal{B}_2^{(3)} = \prod_{c=1}^{C} \left\{ \exp(\lambda_{i_1}^c + \lambda_{i_2}^c + \hat{\lambda}_{i_1}^c - \tilde{\lambda}_{i_1}^c) \right.$$

$$\left. \times \frac{\prod_{k=1}^{\tilde{y}_{i_1}^c} \lambda^c(t_k, \mathcal{S}'_{v_{l^*-1}:v_{l^*+2}})}{\prod_{k=1}^{y_{i_1}^c} \lambda^c(t_k, \mathcal{S}_{v_{l^*-1}:v_{l^*}}) \prod_{k=1}^{y_{i_2}^c} \lambda^c(t_k, \mathcal{S}_{v_{l^*}:v_{l^*+1}}) \prod_{k=1}^{\hat{y}_{i_1}^c} \lambda^c(t_k, \mathcal{S}_{v_{l^*+1}:v_{l^*+2}})} \right\}$$

$$\mathcal{B}_3^{(3)} = \frac{\frac{q_2}{n+1} \frac{1}{(\tau_{l^*-1}+\tau_{l^*}+\tau_{l^*+1})} \tilde{p}_v(\tau_{l^*} \mid \tau_{l^*+1})}{\frac{q_3}{n+1}} = \frac{\tilde{p}_v(\tau_{l^*} \mid \tau_{l^*+1})}{\tau_{l^*-1} + \tau_{l^*} + \tau_{l^*+1}}$$

where $q_2 = q_3$, and $\tilde{p}_v(\tau_{l^*} \mid \tau_{l^*+1})$ is determined from

$$\tilde{p}_v(\tau_{l^*} \mid \tau_{l^*+1}) = \frac{p(\boldsymbol{\theta}_{i_2};\, \tau_{l^*})}{\int_{a_{i_2}}^{\tau_{l^*}+\tau_{l^*+1}} p(\boldsymbol{\theta}_{i_2};\, z)\, dz}.$$

A.3.4 Move Type 4. Let $i_1 = j$, $i_2 = i$, and let $\lambda_{i_2}^c = \int_0^{v_1} \lambda^c(t;\, S_{0:v_1} = i_2)\, dt$, $\tilde{\lambda}_{i_2}^c = \int_0^u \lambda^c(t;\, S_{0:u} = i_2)\, dt$, $\tilde{\lambda}_{i_1}^c = \int_u^{v_1} \lambda^c(t;\, S_{u:v_1} = i_1)\, dt$, and let $y_{i_2}^c$, $\tilde{y}_{i_2}^c$, and $\tilde{y}_{i_1}^c$ denote the number of spike counts for spike train c observed within the intervals $[0, v_1]$, $[0, u]$, and $[u, v_1]$, respectively. Let π_{i_1} and π_{i_2} denote the prior probabilities of the initial sojourn in state i_1 and i_2, respectively. Then we have

$$\mathcal{B}_1^{(4)} = \frac{\pi_{i_1}}{\pi_{i_2}} \frac{\tilde{p}(\boldsymbol{\theta}_{i_1};\, u)\tilde{p}(\boldsymbol{\theta}_{i_2};\, \tau_0 - u)}{\tilde{p}(\boldsymbol{\theta}_{i_2};\, \tau_0)}$$

$$\mathcal{B}_2^{(4)} = \prod_{c=1}^{C} \left\{ \exp(\lambda_{i_2}^c - \tilde{\lambda}_{i_2}^c - \tilde{\lambda}_{i_1}^c) \frac{\prod_{k=1}^{\tilde{y}_{i_2}^c} \lambda^c(t_k, \mathcal{S}'_{0:u}) \prod_{k=1}^{\tilde{y}_{i_1}^c} \lambda^c(t_k, \mathcal{S}'_{u:v_1})}{\prod_{k=1}^{y_{i_2}^c} \lambda^c(t_k, \mathcal{S}_{0:v_1})} \right\}$$

$$\mathcal{B}_3^{(4)} = \frac{q_5}{q_4 \frac{1}{\tau_0 - a_{i_2} - a_{i_1}}} = \tau_0 - a_{i_2} - a_{i_1},$$

where $q_4 \equiv q_B \tilde{q}_2 = q_C \tilde{q}_2 \equiv q_5$. Note that $\boldsymbol{\tau}'$ is obtained from $\boldsymbol{\tau}$ from an invertible deterministic function $\boldsymbol{\tau}' \leftarrow \mathcal{T}(\boldsymbol{\tau}, u) = (u, \tau_0 - u, \tau_1, \tau_2, \ldots, \tau_n)$. Then it follows that

$$|\mathcal{J}| = \begin{vmatrix} \frac{\partial u}{\partial \tau_0} & \frac{\partial u}{\partial u} \\ \frac{\partial(\tau_0 - u)}{\partial \tau_0} & \frac{\partial(\tau_0 - u)}{\partial u} \end{vmatrix} = \begin{vmatrix} 0 & 1 \\ 1 & -1 \end{vmatrix} = 1.$$

A.3.5 Move Type 5. Moves 5 and 4 are inverses of each other. Let $i_1 = \chi_0, i_2 = \chi_1, \lambda_{i_1}^c = \int_0^{\nu_1} \lambda^c(t; S_{0:\nu_1} = i_1)\,dt, \lambda_{i_2}^c = \int_{\nu_1}^{\nu_2} \lambda^c(t; S_{\nu_1:\nu_2} = i_1)\,dt, \tilde{\lambda}_{i_2}^c = \int_0^{\nu_1} \lambda^c(t; S_{0:\nu_2} = i_2)\,dt$, and let $y_{i_1}^c$, $y_{i_2}^c$, and $\tilde{y}_{i_2}^c$ denote the observed number of spike counts for spike train c within the intervals $[0, \nu_1]$, $[\nu_1, \nu_2]$, and $[0, \nu_2]$, respectively. Then we have

$$\mathcal{B}_1^{(5)} = \frac{\pi_{i_2}}{\pi_{i_1}} \frac{\tilde{p}(\boldsymbol{\theta}_{i_2}; \tau_0 + \tau_1)}{\tilde{p}(\boldsymbol{\theta}_{i_1}; \tau_0)\tilde{p}(\boldsymbol{\theta}_{i_2}; \tau_1)}$$

$$\mathcal{B}_2^{(5)} = \prod_{c=1}^{C} \left\{ \exp(\lambda_{i_1}^c + \lambda_{i_2}^c - \tilde{\lambda}_{i_2}^c) \frac{\prod_{k=1}^{\tilde{y}_{i_2}^c} \lambda^c(t_k, \mathcal{S}'_{0:\nu_2})}{\prod_{k=1}^{y_{i_1}^c} \lambda^c(t_k, \mathcal{S}_{0:\nu_1}) \prod_{k=1}^{y_{i_2}^c} \lambda^c(t_k, \mathcal{S}_{\nu_1:\nu_2})} \right\}$$

$$\mathcal{B}_3^{(5)} = \frac{q_4 \frac{1}{\tau_0 + \tau_1}}{q_5} = \frac{1}{\tau_0 + \tau_1},$$

where $q_4 = q_5$. Similarly, $\boldsymbol{\tau}' \leftarrow \mathcal{T}(\boldsymbol{\tau}) = (\tau_0 + \tau_1, \tau_2, \ldots, \tau_n)$, and $|\mathcal{J}| = 1$.

A.3.6 Move Type 6. Let $i_1 = i, i_2 = j, \lambda_{i_1}^c = \int_{\nu_n}^{\nu_{n+1}} \lambda^c(t; S_{\nu_n:\nu_{n+1}} = i_1)\,dt, \tilde{\lambda}_{i_1}^c = \int_{\nu_n}^{\nu_n+u} \lambda^c(t; S_{\nu_n:\nu_n+u} = i_1)\,dt, \tilde{\lambda}_{i_2}^c = \int_{\nu_n+u}^{\nu_{n+1}} \lambda^c(t; S_{\nu_n+u:\nu_{n+1}} = i_2)\,dt$, and let $y_{i_1}^c$, $\tilde{y}_{i_1}^c$, and $\tilde{y}_{i_2}^c$ denote the number of spike counts for spike train c observed within the intervals $[\nu_n, \nu_{n+1}]$, $[\nu_n, \nu_n + u]$, and $[\nu_n + u, \nu_{n+1}]$, respectively. Then we have

$$\mathcal{B}_1^{(6)} = \frac{\tilde{p}(\boldsymbol{\theta}_{i_1}; u)\tilde{p}(\boldsymbol{\theta}_{i_2}; \tau_n - u)}{\tilde{p}(\boldsymbol{\theta}_{i_1}; \tau_n)}$$

$$\mathcal{B}_2^{(6)} = \prod_{c=1}^{C} \left\{ \exp(\lambda_{i_1}^c - \tilde{\lambda}_{i_2}^c - \tilde{\lambda}_{i_2}^c) \times \frac{\prod_{k=1}^{\tilde{y}_{i_1}^c} \lambda^c(t_k, \mathcal{S}'_{\nu_n:\nu_n+u}) \prod_{k=1}^{\tilde{y}_{i_2}^c} \lambda^c(t_k, \mathcal{S}'_{\nu_n+u:\nu_{n+1}})}{\prod_{k=1}^{y_{i_1}^c} \lambda^c(t_k, \mathcal{S}_{\nu_n:\nu_{n+1}})} \right\}$$

$$\mathcal{B}_3^{(6)} = \frac{q_7}{q_6 \frac{1}{\tau_n - a_{i_1}}} = \tau_n - a_{i_1},$$

where $q_6 = q_7$. Similarly, $\boldsymbol{\tau}' \leftarrow \mathcal{T}(\boldsymbol{\tau}, u) = (\tau_0, \tau_1, \ldots, \tau_{n-1}, u, \tau_n - u)$, and

$$|\mathcal{J}| = \begin{vmatrix} \frac{\partial u}{\partial u} & \frac{\partial u}{\partial \tau_n} \\ \frac{\partial(\tau_n - u)}{\partial u} & \frac{\partial(\tau_n - u)}{\partial \tau_n} \end{vmatrix} = \begin{vmatrix} 1 & 0 \\ -1 & 1 \end{vmatrix} = 1.$$

A.3.7 Move Type 7. Moves 7 and 6 are inverses of each other. Let $i_1 = \chi_{n-1}, i_2 = \chi_n, \lambda_{i_1}^c = \int_{\nu_{n-1}}^{\nu_n} \lambda^c(t; S_{\nu_{n-1}:\nu_n} = i_1)\,dt, \lambda_{i_2}^c = \int_{\nu_n}^{\nu_{n+1}} \lambda^c(t; S_{\nu_n:\nu_{n+1}} = i_2)\,dt,$

$\tilde{\lambda}^c_{i_1} = \int_{\nu_{n-1}}^{\nu_{n+1}} \lambda^c(t; S'_{\nu_{n-1}:\nu_{n+1}} = i_1)\, dt$, and let $y^c_{i_1}$, $y^c_{i_2}$ and $\tilde{y}^c_{i_1}$ denote the observed number of spike counts for spike train c within the intervals $[\nu_{n-1}, \nu_n]$, $[\nu_n, \nu_{n+1}]$, and $[\nu_{n-1}, \nu_{n+1}]$, respectively. Then we have

$$\mathcal{B}_1^{(7)} = \frac{\tilde{p}(\boldsymbol{\theta}_{i_1}; \tau_{n-1} + \tau_n)}{\tilde{p}(\boldsymbol{\theta}_{i_1}; \tau_{n-1})\tilde{p}(\boldsymbol{\theta}_{i_2}; \tau_n)}$$

$$\mathcal{B}_2^{(7)} = \prod_{c=1}^{C} \left\{ \exp(\lambda^c_{i_1} + \lambda^c_{i_2} - \tilde{\lambda}^c_{i_1}) \frac{\prod_{k=1}^{\tilde{y}^c_{i_1}} \lambda^c(t_k, \mathcal{S}'_{\nu_{n-1}:\nu_{n+1}})}{\prod_{k=1}^{y^c_{i_1}} \lambda^c(t_k, \mathcal{S}_{\nu_{n-1}:\nu_n}) \prod_{k=1}^{y^c_{i_2}} \lambda^c(t_k, \mathcal{S}_{\nu_n:\nu_{n+1}})} \right\}$$

$$\mathcal{B}_3^{(7)} = \frac{q_6 \frac{1}{\tau_{n-1}+\tau_n}}{q_7} = \frac{1}{\tau_{n-1} + \tau_n},$$

where $q_6 = q_7$. In addition, we have $\boldsymbol{\tau}' \leftarrow \mathcal{T}(\boldsymbol{\tau}) = (\tau_0, \ldots, \tau_{n-2}, \tau_{n-1} + \tau_n)$ and $|\mathcal{J}| = 1$.

A.4 Heuristics for Efficient RJMCMC Sampling. The experimental recordings are relatively long (varying 15–30 minutes for different rats or dates), and the MCMC sampling for the continuous-time model (with 1 ms bin size) is quite time-consuming. We need to design an efficient (problem-specific) sampling procedure for tackling this problem. One important issue is the initialization of state. As discussed earlier, this will be obtained from the estimation result of the discrete-time HMM. Another issue is to design data-driven MCMC proposals (e.g., Tu & Zhu, 2002) that "intelligently" select moves that also satisfy the detailed balance condition. Specifically, we use a few heuristics in carrying out RJMCMC sampling:

- Move type 1: Given a reasonably initialized state, use option 2 instead of option 1.
- Move type 2: Implement it favorably for those long sojourn time durations, and execute it only for those sojourn time durations with at least four times the minimum length.
- Move type 3: Implement it favorably for those short sojourn time durations.
- Move type 4: Execute it only when the initial sojourn time has at least four times the minimum length.
- Move type 6: Execute it only when the final sojourn time has at least four times the minimum length.

As far as the current UP and DOWN estimation problem is concerned, move types 1 and 3 are the most important ones. When implementing move type 3, we employ a heuristic importance sampling trick. Specifically, the probability of choosing a sojourn time to merge with its neighboring sojourns is inversely proportional to the sojourn length: the shorter the duration, the more likely to be picked out to be merged. Similarly, this

trick can be utilized in move type 2 to determine where to split a DOWN state sojourn. The probability of the selected position will be inversely proportional to the observed number of instantaneous MUA spike counts.

A.5 Special Example. In what follows, we derive a special example, in which the sojourn time durations of both UP and DOWN states are modeled by a censored exponential distribution as given in equation 3.12. This example can be viewed as a special case of the result from Ball et al. (1999) in which no constraint was imposed for the pdf. Let r_0 and r_1 denote the rate parameters associated with the exponential distribution for the DOWN and UP states, respectively. And let $a_0 > 0$ and $a_1 > 0$ denote the lower bounds of the sojourn durations for the DOWN and UP states, respectively. The probability ratios $\mathcal{B}_1$ and $\mathcal{B}_3$ for the seven move types are as follows:

- Move type 1:

$$\mathcal{B}_1^{(1)} = \frac{c_1 r_{i_1} \exp(-r_{i_1} u) c_2 r_{i_2} \exp[-r_{i_2}(\tau_{j+1} + \tau_j - u)]}{c_3 r_{i_1} \exp(-r_{i_1} \tau_j) c_4 r_{i_2} \exp(-r_{i_2} \tau_{j+1})}$$

$$= \frac{c_1 c_2 \exp[(r_{i_1} - r_{i_2})(\tau_i - u)]}{c_3 c_4} \qquad \text{(option 1)}$$

$$\mathcal{B}_1^{(1)} = \frac{c_5 r_{i_1} \exp[-r_{i_1}(\tau_j + u)] c_5 r_{i_2} \exp[-r_{i_2}(\tau_{j+1} - u)]}{c_3 r_{i_1} \exp(-r_{i_1} \tau_j) c_4 r_{i_2} \exp(-r_{i_2} \tau_{j+1})}$$

$$= \frac{c_5 c_6 \exp[(r_{i_1} - r_{i_2}) u]}{c_3 c_4} \qquad \text{(option 2)}$$

$$\mathcal{B}_3^{(1)} = 1 \qquad \text{(options 1 and 2)},$$

  where $c_1, c_2, c_3, c_4, c_5, c_6$ are the normalized coefficients (details are ignored here; see the description after equation 3.13)
- Move type 2:

$$\mathcal{B}_1^{(2)} = \frac{c_1 r_{i_1} \exp(-r_{i_1} u) c_2 r_{i_2} \exp(-r_{i_2} v) c_3 r_{i_1} \exp[-r_{i_1}(\tau_{l^*} - u - v)]}{c_4 r_{i_1} \exp(-r_{i_1} \tau_{l^*})}$$

$$= \frac{c_1 c_2 c_3 r_{i_1} r_{i_2} \exp[(r_{i_1} - r_{i_2}) v]}{c_4}$$

$$\mathcal{B}_3^{(2)} = \frac{(\tau_{l^*} - 2a_{i_1})\big(\exp(-r_{i_2} a_{i_2}) - \exp[-r_{i_2}(\tau_{l^*} - u - a_{i_1})]\big)}{r_{i_2} \exp(-r_{i_2} v)},$$

  where c_1, c_2, c_3, c_4 are the normalized coefficients
- Move type 3:

$$\mathcal{B}_1^{(3)} = \frac{c_1 r_{i_1} \exp[-r_{i_1}(\tau_{l^*-1} + \tau_{l^*} + \tau_{l^*+1})]}{c_2 r_{i_1} \exp(-r_{i_1} \tau_{l^*-1}) c_3 r_{i_2} \exp(-r_{i_2} \tau_{l^*}) c_4 r_{i_1} \exp(-r_{i_1} \tau_{l^*+1})}$$

$$= \frac{c_1 \exp[(r_{i_2} - r_{i_1})\tau_{l^*}]}{c_2 c_3 c_4 r_{i_1} r_{i_2}}$$

$$\mathcal{B}_3^{(3)} = \frac{r_{i_2} \exp(-r_{i_2}\tau_{l^*})}{(\tau_{l^*-1} + \tau_{l^*} + \tau_{l^*+1})\big(\exp(-r_{i_2}a_{i_2}) - \exp[-r_{i_2}(\tau_{l^*} + \tau_{l^*+1})]\big)},$$

where c_1, c_2, c_3, c_4 are the normalized coefficients

- Move type 4:

$$\mathcal{B}_1^{(4)} = \frac{\pi_{i_1}}{\pi_{i_2}} \frac{c_1 r_{i_1} \exp(-r_{i_1}u) c_2 r_{i_2} \exp[-r_{i_2}(\tau_0 - u)]}{c_3 r_{i_2} \exp(-r_{i_2}\tau_0)}$$

$$= \frac{\pi_{i_1}}{\pi_{i_2}} \frac{c_1 c_2 r_1 \exp[(r_{i_2} - r_{i_1})u]}{c_3}$$

$$\mathcal{B}_3^{(4)} = \tau_0 - a_{i_1} - a_{i_2},$$

where c_1, c_2, c_3 are the normalized coefficients

- Move type 5:

$$\mathcal{B}_1^{(5)} = \frac{\pi_{i_2}}{\pi_{i_1}} \frac{c_1 r_{i_2} \exp[-r_{i_2}(\tau_0 + \tau_1)]}{c_2 r_{i_1} \exp(-r_{i_1}\tau_0) c_3 r_{i_2} \exp(-r_{i_2}\tau_1)}$$

$$= \frac{\pi_{i_2}}{\pi_{i_1}} \frac{c_1 \exp[(r_{i_1} - r_{i_2})\tau_0]}{c_2 c_3 r_1}$$

$$\mathcal{B}_3^{(5)} = \frac{1}{\tau_0 + \tau_1},$$

where c_1, c_2, c_3 are the normalized coefficients

- Move type 6:

$$\mathcal{B}_1^{(6)} = \frac{c_1 r_{i_1} \exp(-r_{i_1}u) c_2 r_{i_2} \exp[-r_{i_2}(\tau_n - u)]}{c_3 r_{i_1} \exp(-r_{i_1}\tau_n)}$$

$$= \frac{c_1 c_2 r_2 \exp[(r_{i_1} - r_{i_2})(\tau_n - u)]}{c_3}$$

$$\mathcal{B}_3^{(6)} = \tau_n - a_{i_1},$$

where c_1, c_2, c_3 are the normalized coefficients

- Move type 7:

$$\mathcal{B}_1^{(7)} = \frac{c_1 r_{i_1} \exp[-r_{i_1}(\tau_{n-1} + \tau_n)]}{c_2 r_{i_1} \exp(-r_{i_1}\tau_{n-1}) c_3 r_{i_2} \exp(-r_{i_2}\tau_n)} = \frac{c_1 \exp[(r_{i_2} - r_{i_1})\tau_n]}{c_2 c_3 r_2}$$

$$\mathcal{B}_3^{(7)} = \frac{1}{\tau_{n-1} + \tau_n},$$

where c_1, c_2, c_3 are the normalized coefficients.

Appendix B: Threshold-Based Method for Classifying UP and DOWN States

The standard threshold-based method for determining the UP and DOWN states based on MUA spike trains (Ji & Wilson, 2007) consists of three major steps.

First, we bin the spike trains into 10 ms windows and calculate the raw spike counts for all time intervals. The raw count signal is smoothed by a gaussian window with an SD of 30 ms to obtain the smoothed count signal over time. We then calculate the first minimum (count threshold value) of the smoothed spike count histogram during SWS. As the spike count has been smoothed, the count threshold value may be a noninteger value.

Second, based on the count threshold value, we determine the active and silent periods for all 10 ms bins. The active periods are set to 1, and silent periods are set to 0. Next, the duration lengths of all silent periods are computed. We then calculate the first local minimum (gap threshold) of the histogram of the silent period durations.

Third, based on the gap threshold value, we merge those active periods separated by silent periods in duration less than the gap threshold. The resultant active and silent periods are classified as the UP and DOWN states, respectively. Finally, we recalculate the duration lengths of all UP and DOWN state periods and compute their respective histograms and sample statistics (min, max, median, mean, SD).

In summary, the choices of the spike count threshold and the gap threshold will directly influence the UP and DOWN state classification and their statistics (in terms of duration length and occurrence frequency). However, the optimal choices of these two hand-tuned parameters are rather ad hoc and dependent on several issues (e.g., kernel smoothing, bin size; see Figure 16 for an illustration). In some cases, no minimum can be found in the smoothed histogram, and then the choice of the threshold is problematic. Note that the procedure will need to be repeated for different data sets such that the UP and DOWN states statistics can be compared.

Acknowledgments

The research reported here was supported by NIH/NIDA grant R01-DA015644 to E.N.B. and M.A.W., an NIH Director Pioneer Award DP1-OD003646 to E.N.B., and an NIH/NHLBI grant R01-HL084502 to Z.C. and R.B.S.V. was supported by NIH institutional NRSA grant T32 HL07901. We thank R. Haslinger, D. Ji, and D. Nguyen for some helpful discussions. We also thank two anonymous reviewers for their valuable comments that helped to improve the presentation of this article.

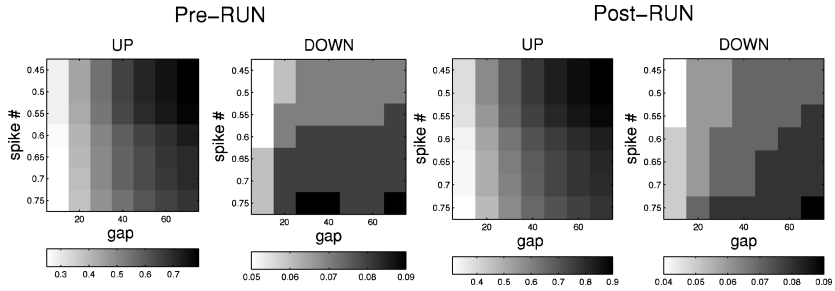


Figure 16: Threshold-based method for estimating the median duration length of the UP and DOWN states (data from the same rat on a different day) in which two thresholds are chosen by grid search. The abscissa represents the gap threshold (in millisecond), and the ordinate represents the smoothed spike count threshold. The map's units are shown in seconds.

References

Abeles, M., Bergman, H., Gat, I., Meilijson, I., Seidemann, E., Tishby, N., et al. (1995). Cortical activity flips among quasi-stationary states. *Proc. Natl. Acad. Sci. USA, 92*, 8616–8620.

Achtman, N., Afshar, A., Santhanam, G., Yu, B. M., Ryu, S. I., & Shenoy, K. V. (2007). Free paced high-performance brain-computer interfaces. *Journal of Neural Engineering, 4*, 336–347.

Albert, P. S. (1991). A two-state Markov mixture model for a time series of epileptic seizure counts. *Biometrics, 47*, 1371–1381.

Ball, F. G., Cai, Y., Kadane, J. B., & O'Hagan, A. (1999). Bayesian inference for ion-channel gating mechanisms directly from single-channel recordings, using Markov chain Monte Carlo. *Proceedings of the Royal Society of London, A455*, 2879–2932.

Bannerjee, A. K., & Bhattacharyya, G. K. (1979). Bayesian results for the inverse gaussian distribution with an application. *Technometrics, 21*(2), 247–251.

Barbieri, R., Quirk, M. C., Frank, L. M., Wilson M. A., & Brown E. N. (2001). Construction and analysis of non-Poisson stimulus response models of neural spike train activity. *Journal of Neuroscience Methods, 105*, 25–37.

Battaglia, F. P., Sutherland, G. R., & McNaughton, B. L. (2004). Hippocampal sharp wave bursts coincide with neocortical "up-state" transitions. *Learning and Memory, 11*, 697–704.

Baum, L. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic function of Markov processes. *Inequalities, 3*, 1–8.

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics, 41*(1), 164–171.

Bellman, R. (1957). *Dynamic programming*. Princeton, NJ: Princeton University Press.

Brillinger, D. R. (1988). Maximum likelihood analysis of spike trains of interacting nerve cells. *Biological Cybernetics, 59*, 189–200.

Brooks, S., Giudici, P., & Roberts, G. (2003). Efficient construction of reversible jump MCMC proposal distributions (with discussion). *Journal of the Royal Society of London, Series B65*(1), 3–56.

Brown, E. N. (2005). Theory of point processes for neural systems. In C. C. Chow, B. Gutkin, D. Hansel, C. Meunier, & J. Dalibard (Eds.), *Methods and models in neurophysics* (pp. 691–727). Amsterdam: Elsevier.

Brown, E. N., Barbieri, R., Eden, U. T., & Frank, L. M. (2003). Likelihood methods for neural data analysis. In J. Feng (Ed.), *Computational neuroscience: A comprehensive approach* (pp. 253–286). London: CRC Press.

Brown, E. N., Barbieri, R., Ventura, V., Kass R. E., & Frank L. M. (2002). The time-rescaling theorem and its application to neural spike data analysis. *Neural Comput., 14*(2), 325–346.

Brown, E. N., Kass R. E., & Mitra, P. P. (2004). Multiple neural spike train data analysis: State-of-the-art and future challenges. *Nature Neuroscience, 7*, 456–461.

Buzsáki, G. (2006). *Rhythms of the brain*. New York: Oxford University Press.

Chan, K. S., & Ledolter, J. (1995). Monte Carlo EM estimation for time series models involving counts. *Journal of the American Statistical Association, 90*, 242–252.

Chung, S. H., Krishnamurthy, V., & Moore, J. B. (1991). Adaptive processing techniques based on hidden Markov models for characterizing very small channel currents buried in noise and deterministic interferences. *Philosophical Transactions of the Royal Society, London B*, 357–384.

Coleman, T. P., & Brown, E. N. (2006). A recursive filter algorithm for state estimation from simultaneously recorded continuous-valued, point process and binary observations. In *Proc. Asilomar Conference on Signals, Systems, and Computers* (pp. 1949–1953). Piscataway, NJ: IEEE Press.

Cowles, M. K., & Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association, 91*, 883–904.

Cox, D. R., & Isham, V. (1980). *Point processes*. London: Chapman & Hall.

Daley, D., & Vere-Jones, D. (2002). *An introduction to the theory of point processes, Vol. 1: Elementary theory and methods*. New York: Springer-Verlag.

Danóczy, M. G., & Hahnloser, R. H. R. (2006). Efficient estimation of hidden state dynamics from spike trains. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems, 18* (pp. 227–234), Cambridge, MA: MIT Press.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39*(1), 1–38.

Deng, L., & Mark, J. W. (1993). Parameter estimation for Markov modulated Poisson processes via the EM algorithm with time discretization. *Telecommunication Systems, 1*(1), 321–338.

Durbin, R., Eddy, S., Krough, A., & Mitchison, G. (1998). *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge: Cambridge University Press.

Eden, U. T., & Brown, E. N. (2008). Mixed observation filtering for neural data. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'08)* (pp. 5201–5203). Piscataway, NJ: IEEE Press.

Ephraim, Y., & Merhav, N. (2002). Hidden Markov processes. *IEEE Transactions on Information Theory, 48*, 1518–1569.

Escola, S., & Paninski, L. (2008). *Hidden Markov models for the complex stimulus-response relationship of multi-state neurons*. Conference abstract, Frontiers in Computational Neuroscience, Bernstein Symposium.

Forney, G. D. (1973). The Viterbi algorithm. *Proceedings of the IEEE, 61*(3), 268–278.

Fredkin, D. R., & Rice, J. A. (1992). Maximum likelihood estimation and identification directly from single channel recordings. *Proceedings of the Royal Society of London, B249*, 125–132.

Fujisawa, S., Matsuki, N., & Ikegaya, Y. (2005). Single neurons can induce phase transitions of cortical recurrent networks with multiple internal states. *Cerebral Cortex, 16*(5), 639–654.

Gat, I., Tishby, N., & Abeles, M. (1997). Hidden Markov modeling of simultaneously recorded cells in the associated cortex of behaving monkeys. *Networks: Computation in Neural Systems, 8*(3), 297–322.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Sciences, 7*, 457–472.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science, 7*, 473–511.

Ghahramani, Z., & Hinton, G. E. (2000). Variational learning for switching state-space models. *Neural Comput., 12*(4), 831–864.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (Eds.). (1995). *Markov chain Monte Carlo in practice*. London: Chapman & Hall/CRC.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika, 82*, 711–732.

Guon, Y. (2003). Estimating hidden semi-Markov chains from discrete sequences. *Journal of Computational Graphical Statistics, 12*, 604–639.

Haider, B., Duque, A., Hasenstaub, A. R., & McCormick, D. A. (2006). Neocortical network activity in vivo is generated through a dynamic balance of excitation and inhibition. *Journal of Neuroscience, 26*, 4535–4545.

Haider, B., Duque, A., Hasentaub, A. R., Yu, Y., & McCormick, D. A. (2007). Enhancement of visual responsiveness by spontaneous local network activity in vivo. *Journal of Neurophysiology, 97*, 4186–4202.

Haslinger, R., Ulbert, I., Moore, C. I. Brown, E. N., & Devor, A. (2006). Analysis of LFP phase predicts sensory response of barrel cortex. *Journal of Neurophysiology, 96*, 1658–1663.

Ji, D., & Wilson, M. A. (2007). Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature Neuroscience, 10*(1), 100–107.

Jones, L. M., Fontanini, A., Sadacca, B. F., & Katz, D. B. (2007). Natural stimuli evoke analysis dynamic sequences of states in sensory cortical ensembles. *Proc. Natl. Acad. Sci. USA, 104*, 18772–18777.

Kang, S., Kitano, K., & Fukai, T. (2008). Structure of spontaneous UP and DOWN transitions self-organizing in a cortical network model. *PLoS Computational Biology, 4*(3): e1000022 doi:10.1371/journal.pcbi.1000022.

Kass, R. E., Ventura, V., & Brown, E. N. (2005). Statistical issues in the analysis of neuronal data. *Journal of Neurophysiology, 94*, 8–25.

Kemere, C., Santhanam, G., Yu, B. M., Afshar, A., Ryu, S. I., Meng, T. H., et al. (2008). Detecting neural-state transitions using hidden Markov models for motor cortical prostheses. *Journal of Neurophysiology, 100*, 2441–2452.

Le, N. D., Leroux, B. G., & Puterman, M. L. (1992). Exact likelihood evaluation in a Markov mixture model for time series of seizure counts. *Biometrics, 48*, 317–323.

Luczak, A., Barthó, P., Marguet, S. L., Buzsáki, G., & Harris, K. D. (2007). Sequential structure of neocortical spontaneous activity in vivo. *Proc. Natl. Acad. Sci., 104*, 347–352.

McCullagh, P., & Nelder, J. (1989). *Generalized linear models*. London: Chapman & Hall.

McLachlan, G. J., & Krishnan, T. (1996). *The EM algorithm and extensions*. Hoboken, NJ: Wiley.

Mehta, M. R. (2007). Cortico-hippocampal interaction during up-down states and memory consolidation. *Nature Neuroscience, 10*(1), 13–15.

Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems, 15*, 243–262.

Paninski, L., Pillow, J., & Lewi, J. (2007). Statistical models for neural encoding, decoding, and optimal stimulus design. In P. Cisek, T. Drew, & J. Kalaska (Eds.), *Computational neuroscience: Theoretical insights into brain function*. Amsterdam: Elsevier.

Pawitan, Y. (2001). *In all likelihood: Statistical modelling and inference using likelihood*. New York: Oxford University Press.

Prerau, M. J., Smith, A. C., Eden, U. T., Yanike, M., Suzuki, W., & Brown, E. N. (2008). A mixed filter algorithm for cognitive state estimation from simultaneously recorded continuous-valued and binary measures of performance. *Biological Cybernetics, 99*(1), 1–14.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257–286.

Radons, G., Becker, J. D., Dülfer, B., & Krüger, J. (1994). Analysis, classification, and coding of multielectrode spike trains with hidden Markov models. *Biological Cybernetics, 71*(4), 359–373.

Robert, C. P., Rydén, T., & Titterington, D. M. (2000). Bayesian inference in hidden Markov models through reversible jump Markov chain Monte Carlo method. *Journal of the Royal Statistical Society, Series B62*, 57–75.

Roberts, W. J. J., Ephraim, Y., & Dieguez, E. (2006). On Rydén's EM algorithm for estimating MMPPs. *IEEE Signal Processing Letters, 13*(6), 373–376.

Rydén, T. (1996). An EM algorithm for estimation in Markov-modulated Poisson processes. *Computational Statistics and Data Analysis, 21*, 431–447.

Sanchez-Vives, M. V., & McCormick, D. A. (2000). Cellular and network mechanisms of rhythmic recurrent activity in neocortex. *Nature Neuroscience, 3*, 1027–1034.

Sirota, A., Csicsvari, J., Buhl, D., & Buzsáki, G. (2003). Communication between neocortex and hippocampus during sleep in rodents. *Proc. Nat. Acad. Sci. USA, 100*, 2065–2069.

Smith A. C., & Brown E. N. (2003). Estimating a state-space model from point process observations. *Neural Comput., 15*, 965–991.

Srinivasan, L., Eden, U. T., Mitter, S. K., & Brown, E. N. (2007). General-purpose filter design for neural prosthetic devices. *Journal of Neurophysiology, 98*, 2456–2475.

Stjernqvist, S., Rydén, T., Sköld, M., & Staaf, J. (2007). Continuous-index hidden Markov modelling of array CGH copy number data. *Bioinformatics, 23*(8), 1006–1014.

Takagi, K., Kumagai, S., Matsunaga, I., & Kusaka, Y. (1997). Application of inverse gaussian distribution to occupational exposure data. *Annals of Occupational Hygiene, 41*(5), 505–514.

Tanner, M. A. (1996). *Tools for statistical inference* (3rd ed.). New York: Springer-Verlag.

Truccolo, W., Eden U.T., Fellow, M., Donoghue, J. D., & Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble and covariate effects. *Journal of Neurophysiology, 93*, 1074–1089.

Tu, Z., & Zhu, S. (2002). Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(5), 657–673.

Tuckwell, H. C. (1989). *Stochastic processes in the neurosciences*. Philadelphia: SIAM.

Verbeek, J. J., Vlassis, N., & Kröse, B. (2003). Efficient greedy learning of gaussian mixture models. *Neural Comput., 15*(2), 469–485.

Vijayan, S. (2007). *Characterization of activity in the primary somatosensory cortex during active behavior and sleep*. Unpublished doctoral dissertation, Harvard University.

Viterbi, J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory, 13*, 260–269.

Volgushev, M., Chauvette, S., Mukovski, M., & Timofeev, I. (2006). Precise long-range synchronization of activity and silence in neocortical neurons during slow-wave sleep. *Journal of Neuroscience, 26*, 5665–5672.

Wolansky, T., Clement, E. A., Peters, S. R., Palczak, M. A., & Dickson, C. T. (2006). Hippocampal slow oscillation: A novel EEG state and its coordination with ongoing neocortical activity. *Journal of Neuroscience, 26*, 6213–6229.

Xi, J., & Kass, R. E. (2008). *Detection of bursts in neuronal spike trains using hidden semi-Markov point process models*. Unpublished manuscript.

Received June 11, 2008; accepted December 17, 2008.