(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 September 2017 (21.09.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/161167 A1

(51) International Patent Classification:
A61B 5/11 (2006.01) *G06T 7/20* (2017.01)
G06T 7/00 (2017.01)

(21) International Application Number:
PCT/US2017/022781

(22) International Filing Date:
16 March 2017 (16.03.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/310,563 18 March 2016 (18.03.2016) US

(71) Applicant: **PRESIDENT AND FELLOWS OF HARVARD COLLEGE** [US/US]; 17 Quincy Street, Cambridge, Massachusetts 02138 (US).

(72) Inventors: **DATTA, Sandeep Robert**; 40 Homer Street, Newton, Massachusetts 02125 (US). **JOHNSON, Matthew J.**; 1204 Tournament Drive, Hillsborough, Massachusetts 94010 (US).

(74) Agents: **RESNICK, David S.** et al.; NIXON PEABODY LLP, 100 Summer St., Boston, Massachusetts 02110-2131 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

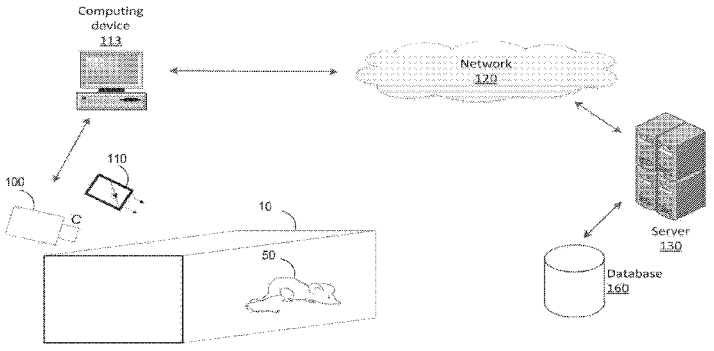
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(54) Title: AUTOMATICALLY CLASSIFYING ANIMAL BEHAVIOR



(57) Abstract: Systems and methods are disclosed to objectively identify sub-second behavioral modules in the three-dimensional (3D) video data that represents the motion of a subject. Defining behavioral modules based upon structure in the 3D video data itself - rather than using *a priori* definitions for what should constitute a measurable unit of action - identifies a previously-unexplored sub-second regularity that defines a timescale upon which behavior is organized, yields important information about the components and structure of behavior, offers insight into the nature of behavioral change in the subject, and enables objective discovery of subtle alterations in patterned action. The systems and methods of the invention can be applied to drug or gene therapy classification, drug or gene therapy screening, disease study including early detection of the onset of a disease, toxicology research, side-effect study, learning and memory process study, anxiety study, and analysis in consumer behavior.

WO 2017/161167 A1

AUTOMATICALLY CLASSIFYING ANIMAL BEHAVIOR

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH

[0001] This invention was made with government support under (1) National Institutes of Health (NIH) New Innovator Award No. DP20D007109 awarded by the NIH Office of the Director; and (2) NIH Research Project Grant Program No. RO1DC011558 awarded by the NIH National Institute on Deafness and Other Communication Disorders (NIDCD). The government has certain rights in the invention.

FIELD

[0002] The present invention is direct to system and methods for identifying and classifying animal behavior, human behavior or other behavioral metrics.

BACKGROUND

[0003] The following description includes information that may be useful in understanding the present invention. It is not an admission that any of the information provided herein is prior art or relevant to the presently claimed invention, or that any publication specifically or implicitly referenced is prior art.

[0004] The quantification of animal behavior is an essential first step in a range of biological studies, from drug discovery to understanding neurodegenerative disorders. It is usually performed by hand; a trained observer watches an animal behave, either live or on videotape, and records the timing of all interesting behaviors.

[0005] Behavioral data for a single experiment can include hundreds of mice, spanning hundreds of hours of video, necessitating a team of observers, which inevitably decreases the reliability and reproducibility of results. In addition, what constitutes an "interesting behavior" is essentially left to the human observer: while it is trivial for a human observer to assign an anthropomorphic designation to a particular behavior or series of behaviors (i.e., "rearing," "sniffing," "investigating," "walking," "freezing," "eating," and the like), there are almost certainly behavioral states generated by the mouse that are relevant to the mouse that defy simple human categorization.

[0006] In more advanced applications, video can be semi-automatically analyzed by a computer program. However, the brain generates behaviors that unfold smoothly over time and yet are composed of distinct patterns of motion. Individual sensory neurons that trigger action can perform behaviorally-relevant computations in as little as a millisecond, and neural populations that mediate behavior exhibit dynamics that evolve on timescales of 10s to 100s of milliseconds [1-8]. This fast neural activity interacts with slower neuromodulator systems to generate behaviors that are organized at multiple timescales simultaneously [9]. Ultimately understanding how neural circuits create complex behaviors — particularly spontaneous or innate behaviors expressed by freely-behaving animals — requires a clear framework for characterizing how behavior is organized at the timescales relevant to the nervous system.

SUMMARY

[0007] Although behaviors have been sculpted by evolution to enable animals to accomplish particular goals (such as finding food or a mate), it is not yet clear how these behaviors are organized over time, particularly at fast timescales. However, one powerful approach to characterizing the structure of behavior arises from ethology, which proposes that the brain builds coherent behaviors by expressing stereotyped modules of simpler action in specific sequences [10]. For example, both supervised and unsupervised classification approaches have identified potential behavioral modules expressed during exploration by C. elegans and by both larval and adult D. melanogaster [11-16]. These experiments have revealed an underlying structure to behavior in these organisms, which in turn has uncovered strategies used by invertebrate brains to adapt behavior to changes in the environment. In the case of C. elegans, navigation towards an olfactory cue is mediated at least in part by neural circuits that modulate the transition probabilities that connect behavioral modules into sequences over time; seemingly new sensory-driven behaviors (like positive chemotaxis) can therefore be generated by the worm nervous system through resequencing of a core set of behavioral modules [17-19]. Similar observations have been made for sensory-driven behaviors in fly larvae [11].

[0008] These insights into the underlying time-series structure of behavior arose from the ability to quantify morphological changes in worms and flies, and to use those data to identify behavioral modules [11-16]. However, it has been difficult to gain similar insight into the global organization of behavior in mammals. While innate exploratory, grooming, social approach, aggressive and reproductive behaviors in mice have all been divided by

investigators into potential modules, this approach to breaking up mammalian behaviors into parts depends upon human-specified definitions for what constitutes a meaningful behavioral module (e.g. running, mating, fighting) [20-25] and are therefore largely bounded by human perception and intuition. Particularly, human perception has difficulty identifying modules spanning a short timescale.

[0009] Systematically describing the structure of behavior in animals — and understanding how the brain alters that structure to enable adaptation — requires addressing three key issues. First, it is not clear which features of behavior are important to measure when attempting to modularize mouse behavior. Although most current methods track two-dimensional parameters such as the position, velocity or shape of the top-down or lateral outline of the mouse [20,22-24,26-28], mice exhibit complex three-dimensional pose dynamics that are difficult to capture but which may afford important insights into the organization of behavior. Second, given that behavior evolves on several timescales in parallel, it is not clear how to objectively identify the relevant spatiotemporal scales at which to modularize behavior. Finally, effectively characterizing behavior requires accommodating the fact that behavior is both stereotyped (a prerequisite for modularity) and variable (an inescapable feature of noisy nervous and motor systems) [29].

[0010] This variability raises significant challenges for algorithms tasked with identifying the number and content of the behavioral modules that are expressed during a given experiment, or with assigning any given instance of an observed action to a particular behavioral module. Furthermore, identifying the spatiotemporal scales at which naturalistic behaviors are organized has been a defining challenge in ethology, and thus to date most efforts to explore the underlying structure of behavior have relied on ad hoc definitions of what constitutes a behavioral module, and have focused on specific behaviors rather than systematically considering behavior as a whole. It is not clear whether spontaneous behaviors exhibited by animals have a definable underlying structure that can be used to characterize action as it evolves over time.

[0011] Furthermore, existing computerized systems for classification of animal behavior match parameters describing the observed behavior against hand-annotated and curated parametric databases. Therefore, in both the manual and existing semi-automated cases, subjective evaluation of the animal's behavioral state is built into the system — a human observer must decide ahead of time what constitutes a particular behavior. This biases

assessment of that behavior and limits the assessment to those particular behaviors the researcher can discriminate with human perception and is therefore limited, especially with respect to behaviors that occur on a short timescale. In addition, video acquisition systems deployed in these semi-supervised forms of behavioral analysis (nearly always acquiring data in two-dimensional) are only optimized for specific behaviors, thereby both limiting throughput and increasing wasted experimental effort through alignment errors.

Overview

[0012] Despite these challenges, the inventors have discovered systems and methods for automatically identifying and classifying behavior modules of animals by processing video recordings of the animals. In accordance with the principles of the invention, a monitoring method and system uses hardware and custom software that can classify animal behavior. Classification of an animal behavioral state is determined by quantitative measurement of animal posture in three-dimensions using a depth camera. In one embodiment, a 3D depth camera is used to obtain a stream of video images of the animal having both area and depth information. The background image (the empty experimental area) is then removed from each of the plurality of images to generate processed images having light and dark areas. The contours of the light areas in the plurality of processed images are found and parameters from both area and depth image information within the contours is extracted to form a plurality of multi-dimensional data points, each data point representing the posture of the animal at a specific time. The posture data points can then be clustered so that point clusters represent animal behaviors.

[0013] This data may then be fed into a model free algorithm, or fed into computation model to characterize the structure of naturalistic behavior. In some embodiments, the systems fit models for behavior using methods in Bayesian inference, which allows unsupervised identification of the optimal number and identity of behavioral modules from within a given dataset. Defining behavioral modules based upon structure in the 3D behavioral data itself — rather than using *a priori* definitions for what should constitute a measurable unit of action — identifies a previously-unexplored sub-second regularity that defines a timescale upon which behavior is organized, yields key information about the components and structure of behavior, offers insight into the nature of behavioral change, and enables objective discovery of subtle alterations in patterned action.

Example Application to Video of Mouse Exploring Open Field

[0014] In one example, the inventors measured how the shape of a mouse's body changes as it freely explores a circular open field. The inventors used depth sensors to capture three-dimensional (3D) pose dynamics of the mouse, and then quantified how the mouse's pose changed over time by centering and aligning the image of the mouse along the inferred axis of its spine.

[0015] Plotting this 3D data over time revealed that mouse behavior is characterized by periods during which pose dynamics evolve slowly, punctuated by fast transitions that separate these periods; this pattern appears to break up the behavioral imaging data into blocks consisting of a small number of frames typically lasting from 200-900 ms. This suggests that mouse behavior may be organized at two distinct timescales, the first defined by the rate at which a mouse's pose can change within a given block, and the second defined by the transition rate between blocks.

[0016] Characterizing mouse behavior within these blocks — and determining how behavior might differ between blocks — requires first estimating the timescales at which these blocks are organized. In some embodiments, to identify approximate boundaries between blocks, the behavioral imaging data was submitted to a changepoint algorithm designed to detect abrupt changes in the structure of data over time. In one example, this method automatically identified potential boundaries between blocks, and revealed that the mean block duration was about 350 ms.

[0017] Additionally, the inventors performed autocorrelation and spectral analysis, which provided complementary information about the timescale of behavior. Temporal autocorrelation in the mouse's pose largely dissipated within 400 ms (tau = 340 ± 58 ms,) and nearly all of the frequency content that differentiated behaving and dead mice concentrated between 1 and 6 Hz (measured by spectrum ratio, or Wiener filter, mean 3.75 ± .56 Hz); these results suggest that most of the dynamism in the mouse's behavior occurs within 200 - 900 ms timescale.

[0018] Additionally, visual inspection of the block-by-block pattern of behavior exhibited by a mouse reveals that each block appears to encode a brief motif of behavior (e.g. a turn to the right or left, a dart, a pause, the first half of a rear) separated from the subsequent behavioral motif by a fast transition. Taken together, these findings reveal a previously-

unappreciated sub-second organization to mouse behavior — during normal exploration mice express brief motifs of movement that appear to rapidly switch from one to another in series.

[0019] The finding that behavior is naturalistically broken up into brief motifs of motion indicates that each of these motifs is a behavioral module: a stereotyped and reused unit of behavior that the brain places into sequences to build more complex patterns of action. Next, systems and method are disclosed for identifying multiple examples of the same stereotyped sub-second motif of behavior.

Processing Algorithms and Methods for Identifying Modules in Video Data

[0020] To identify similar modules, mouse behavioral data may first be subject dimesionality reduction using, for example (1) principal component analysis, and (2) neural networks such as multi-layer perceptrons. For instance, using principal components analysis (PCA), the first two principal components may be plotted. Each block in the pose dynamics data corresponds to a continuous trajectory through PCA space; for example, an individual block associated with the mouse's spine being elevated corresponded to a specific sweep through PCA space. Scanning the behavioral data for matching motifs using a template matching method identified several additional examples of this sweep in different animals, suggesting that each of these PCA trajectories may represent individual instances in which a stereotyped behavioral module was reused.

[0021] Given this evidence for sub-second modularity, the inventors devised a series of computational models — each of which describes a different underlying structure for mouse behavior — trained these models on 3D behavioral imaging data, and determined which models predicted or identified the underlying structure of mouse behavior. Particularly, the inventors utilized computational inference methods (including Bayesian non-parametric approaches and Gibbs sampling) that are optimized to automatically identify structure within large datasets.

[0022] Each model differed in whether it considered behavior to be continuous or modular, in the possible contents of the modules, and in the transition structure that governed how modules were placed into sequence over time. To compare model performance, the models were tested to predict the contents and structure of real mouse behavioral data to which the models had not been exposed. Among the alternatives, the best quantitative predictions were made by a model that posits that mouse behavior is composed of modules

(each capturing a brief motif of 3D body motion) that switch from one to another at the sub-second timescales identified by our model-free analysis of the pose dynamics data.

AR-HMM Model

[0023] One model represented each behavioral module as a vector autoregressive (AR) process capturing a stereotyped trajectory through PCA space. Additionally in that model, the switching dynamics between different modules were represented using a Hidden Markov Model (HMM). Together, this model is referred to herein as "AR-HMM."

[0024] In some embodiments, AR-HMM makes predictions about mouse behavior based upon its ability to discover (within training data) the set of behavioral modules and transition patterns that provide the most parsimonious explanation for the overall structure of mouse behavior as it evolves over time. Accordingly, a trained AR-HMM can be used to reveal the identity of behavioral modules and their transition structure from within a behavioral dataset, and thereby expose the underlying organization of mouse behavior. After training, the AR-HMM can assign every frame of the training behavioral data to one of the modules it has discovered, revealing when any given module is expressed by mice during a given experiment.

[0025] Consistent with the AR-HMM recognizing the inherent block structure in the 3D behavioral data, the module boundaries identified by the AR-HMM respected the inherent block structure embedded within the pose dynamics data. Furthermore, the model-identified module duration distribution was similar to the changepoints-identified block duration distribution; however, the module boundaries identified by the AR-HMM refined the approximate boundaries suggested by the changepoints analysis (78 percent of module boundaries were within 5 frames of a changepoint). Importantly, the ability of the AR-HMM to identify behavioral modules depended upon the inherent sub-second organization of mouse pose data, as shuffling the frames that make up the behavioral data in small chunks (i.e. < 300 milliseconds) substantially degraded model performance, while shuffling the behavioral data in bigger chunks had little effect. These results demonstrate that the AR-HMM recognizes the inherent sub-second block structure of the behavioral data.

[0026] Additionally, specific behavioral modules identified by the AR-HMM encoded a set of distinct and reused motifs of motion. For instance, the PCA trajectories assigned by the model to one behavioral module traced similar paths through PCA space.

Consistent with each of these trajectories encoding a similar motif of action, collating and inspecting the 3D movies associated with multiple data instances of this specific module confirmed that it encodes a stereotyped motif of behavior, one human observers would refer to as rearing. In contrast, data instances drawn from different behavioral modules traced distinct paths through PCA space. Furthermore, visual inspection of the 3D movies assigned to each of these modules demonstrated that each encodes a repeatedly used and coherent pattern of three-dimensional motion that can be distinguished and labeled with descriptors (e.g., "walk," "pause," and "low rear" modules).

[0027] To quantitatively and comprehensively assess the distinctiveness of each behavioral module identified by the AR-HMM, we then performed a cross-likelihood analysis, which revealed that the data instances associated with a given module are best assigned to that module, and not to any of the other behavioral modules in the parse. In contrast, the AR-HMM failed to identify any well-separated modules in a synthetic mouse behavioral dataset that lacks modularity, demonstrating that the discovered modularity within the real behavioral data is a feature of the dataset itself rather than being an artifact of the model. Furthermore, restarting the model training process from random starting points returns the same or a highly similar set of behavioral modules, consistent with the AR-HMM homing in on and identifying an intrinsic modular structure to the behavioral data. Together these data suggest that mouse behavior — when viewed through the lens of the AR-HMM — is fundamentally organized into distinct sub-second modules.

[0028] Additionally, if the AR-HMM identifies behavioral modules and transitions that make up mouse behavior, then synthetic behavioral data generated by a trained AR-HMM can provide a reasonable facsimile of real pose dynamics data. The AR-HMM appeared to capture the richness of mouse behavior, as synthetic behavioral data (in the form of spine dynamics, or a 3D movie of a behaving mouse) was qualitatively difficult to distinguish from behavioral data generated by an actual animal. Mouse pose dynamics data therefore appear to have an intrinsic structure organized on sub-second timescales that is well-parsed by the AR-HMM into defined modules; furthermore, optimal identification of these modules and effective prediction of the structure of behavior requires overt modeling of modularity and switching dynamics.

SLDS SVAE Model

[0029] To both reduce redundant dimensions and make modeling computationally tractable, various techniques may be employed to dimensionally reduce each image. One method of reducing dimensionality, principal component analysis, will reduce the dimensionality to a linear space. However, the inventors have discovered that reduction of the dimensionality to only a linear space will not accommodate for various changes in the mice that are not related to behavior. This includes changes in mouse size, mouse breed, etc.

[0030] Accordingly, the inventors have discovered that using certain kinds of neural networks, such a multi-layer perceptrons, one can effectively reduce the dimensionality of images. Furthermore, these dimensionality reduced images provide an effective method to develop models that are agnostic to the size of the mouse, or other animals, and can account for other changes that are not behavior related. For instance, some neural networks that reduce the dimensionality to a three dimensional image manifold may be utilized.

[0031] Accordingly, based on these algorithms, the inventors developed an SVAE generative model and corresponding variational family algorithm. As an example, the inventors focus on a particular generative model for time series based on a switching linear dynamical system (SLDS) (Murphy, 2012; Fox et al, 2011), which illustrates how the SVAE can incorporate both discrete and continuous latent variables with rich probabilistic dependence.

[0032] The systems and methods of the present invention can be applied to a variety of animal species, such as animals in animal models, humans in clinical trials, humans in need of diagnosis and/or treatment for a particular disease or disorder. Without limitations, these animals include mice, dogs, cats, cows, pigs, goats, sheep, rats, horses, guinea pigs, rabbits, reptiles, zebrafish, birds, fruit flies, worms, amphibians (e.g., frogs), chickens, non-human primates, and humans.

[0033] The systems and methods of the present invention can be used in a variety of applications including, but not limited to, drug screening, drug classification, genetic classification, disease study including early detection of the onset of a disease, toxicology research, side-effect study, learning and memory process study, anxiety study, and analysis in consumer behavior.

[0034] The systems and methods of the present invention are particularly useful for diseases that affect the behavior of a subject. These diseases include neurodegenerative

diseases such as Parkinson's disease, Huntington's disease, Alzheimer's disease, and Amyotrophic lateral sclerosis, neurodevelopmental psychiatric disorders such as attention deficit hyperactivity disorder, autism, Down syndrome, Mendelsohnn's Syndrome, and Schizophrenia.

[0035] In some embodiments, the systems and methods of the present invention can be used to study how a known drug or test compound can alter the behavioral state of a subject. This can be done by comparing the behavioral representations obtained before and after the administration of the known drug or test compound to the subject. As used herein, the term "behavioral representation" refers to a set of sub-second behavioral modules and their transition statistics determined using the systems or methods of the invention. Without limitation, the behavioral representation can be in the form of a matrix, a table, or a heatmap.

[0036] In some embodiments, the systems and methods of the present invention can be used for drug classification. The systems and methods of the present invention can create a plurality of reference behavioral representations based on existing drugs and the diseases or disorders they treat, wherein each reference behavioral representation represents a class of drugs (e.g., antipsychotic drugs, antidepressants, stimulants, or depressants). A test behavioral representation can be compared to the plurality of reference behavioral representation, and if the test behavioral representation is similar to one of the plurality of reference behavioral representations, the test compound is determined to belong to the same class of drugs that is represented by said particular of reference behavioral representation. Without limitation, the test compound can be a small molecule, an antibody or an antigen-binding fragment thereof, a nucleic acid, a polypeptide, a peptide, a peptidomimetic, a polysaccharide, a monosaccharide, a lipid, a glycosaminoglycan, or combinations thereof.

[0037] In some embodiments, this may include a system for automatically classifying an animal's behavior as belonging to one class of drugs versus a list of alternatives. For instance, to develop the system, we may provide a training set of many mice under many different drug conditions, and build a linear or non-linear classifier to discover what combinations and ranges of features constitute membership in a particular drug class. This classifier may be then fixed as soon as training is completed, allowing us to apply it to previously unseen mice. Potential classifier algorithms may include logistic regression, support vector machine with linear basis kernel, support vector machine with radial basis

function kernel, multi-layer perceptron, random forest classifier, or k-Nearest Neighbors classifier.

[0038] Similar to drug classification, in some embodiments, the systems and methods of the present invention can be used in gene-function classification.

[0039] In someone embodiments of drug screening, an existing drug that is known to treat a particular disease or disorder can be administered to a first test subject. The systems and methods of the present invention can then be used on the first test subject to obtain a reference behavioral representation, which includes a set of behavioral modules that can characterize the therapeutic effects of the drug on the first test subject. Subsequently, a test compound can be administered to a second test subject of the same animal type as the first test subject. The systems and methods of the present invention can then be used on the second test subject to obtain a test behavioral representation. If the test behavioral representation is found to be similar to the reference behavioral representation, the test compound is determined to be effective in treating the particular disease or disorder. If the test behavioral representation is found to not be similar to the reference behavioral representation, the test compound is determined to be ineffective in treating the particular disease or disorder. It should be noted that the first and second test subject can each be a group of test subjects, and the behavioral representation obtained can be an average behavioral representation.

[0040] Similar to drug screening, in some embodiments, the systems and methods of the present invention can be used in gene-therapy screening. Gene therapies can include delivery of a nucleic acid and gene knockout.

[0041] In some embodiments, the systems and methods of the present invention can be used in the study of disease or disorder. For example, the systems and methods of the invention can be used to discover new behavioral modules in subjects having a particular disease or disorder. For example, the systems and methods of the present invention can permit early diagnosis of a disease or disorder by identifying a reference behavioral representation in subjects having the disease or disorder or subjects that are in the process of developing the disease or disorder. If the reference behavioral representation or a significant portion thereof is also observed in a subject suspected of having the disease or disorder, the

subject is diagnosed as having the disease or disorder. Thus early clinical interventions can be administered to the subject.

[0042] Additionally, in some embodiments, the systems and methods of the present invention can be used in the in the study of consumer behavior, for example, how a consumer responds to a scent (e.g., perfume). The systems and methods of the present invention can be used to identify a reference behavioral representation that represents positive reactions to the scent. In the presence of the scent, a person exhibiting the reference behavioral representation or a significant portion thereof is determined to be reacting positively to the scent. Reference behavioral representation that represents negative reactions to the scent can also be identified and used to gauge a person's reaction.

BRIEF DESCRIPTION OF THE DRAWINGS

[0043] The accompanying drawings, which are incorporated in and constitute a part of this specification, exemplify the embodiments of the present invention and, together with the description, serve to explain and illustrate principles of the invention. The drawings are intended to illustrate major features of the exemplary embodiments in a diagrammatic manner. The drawings are not intended to depict every feature of actual embodiments nor relative dimensions of the depicted elements, and are not drawn to scale.

[0044] FIG. 1 depicts, in accordance with various embodiments of the present invention, a diagram of a system designed to capture video data of an animal;

[0045] FIG. 2A depicts, in accordance with various embodiments of the present invention, a flow chart showing processing steps performed on video data;

[0046] FIG. 2B depicts, in accordance with various embodiments of the present invention, a flow chart showing processing steps performed on video data;

[0047] FIG. 3 depicts, in accordance with various embodiments of the present invention, a flow chart showing analysis performed on the video data output from the processing steps;

[0048] FIG. 4 depicts, in accordance with various embodiments of the present invention, a flow chart showing the implementation of an AR-HMM algorithm;

[0049] FIG. 5A depicts, in accordance with various embodiments of the present invention, a graph showing the proportion of frames explained by each module (Y axis), plotted against the set of modules, sorted by usage (X axis);

[0050] FIG. 5B depicts, in accordance with various embodiments of the present invention, a graph showing modules (X axis) sorted by usage (Y axis) with Bayesian credible intervals indicated;

[0051] FIGs. 6A - 6E depict, in accordance with various embodiments of the present invention, the influences of the physical environment on module usage and spatial pattern of expression. FIG. 6A. Modules identified by the AR-HMM sorted by usage (n = 25 mice, 500 total minutes, data from circular open field). FIG. 6B. Hinton diagram of the observed bigram probabilities, depicting the probability that any pair of modules are observed as ordered pairs. FIG. 6C. Module usage, sorted by context. Mean usages across animals depicted with dark lines, with bootstrap estimates depicted in fainter lines (n=100). Marked modules discussed in main text and shown in FIG. 6D: square = circular thigmotaxis, circle = rosette, diamond = square thigmotaxis, cross = square dart. FIG. 6D. Occupancy graph of mice in circular open field (left, n=25, 500 minutes total) indicating average spatial positions across all experiments. Occupancy graph depicting deployment of circular thigmotaxis module (middle, average orientation across the experiment indicated as arrow field) and circle-enriched rosette module (right, orientation of individual animals indivated with arrows). FIG. 6E. Occupancy graph of mice in square box (left, n=15, 300 minutes total) indicating cumulative spatial positions across all experiments. Occupancy graph depicting a square-enriched thigmophilic module (middle, average orientation across the experiment indicated as arrow field), and square-specific darting module (right, orientation of individual animals indivated with arrows).

[0052] FIG. 7 depicts, in accordance with various embodiments of the present invention, a histogram depicting the average velocity of the modules that were differentially upregulated and interconnected after TMT exposure “freezing” compared to all other modules in the dataset.

[0053] FIGs. 8A – 8E depict, in accordance with various embodiments of the present invention, how odor avoidance alters transition probabilities. FIG. 8A. Occupancy plot of mice under control conditions (n=24, 480 total minutes) and exposed to the monomolecular

fox-derived odorant trimethylthiazoline (TMT, 5% dilution in carrier DPG, n=15, 300 total minutes) in the lower left quadrant (arrow). FIG. 8B. Module usage plot sorted by "TMT-ness. Dark lines depict mean usages, bootstrap estimates depicted in fainter lines. Marked modules discussed in this specification and FIG. 8E: square = sniff in TMT quadrant, circle = freeze away from TMT. FIG. 8C, left and middle. Behavioral state maps for mice exploring a square box under control conditions (blank) and after TMT exposure, with modules depicted as nodes (usage proportional to the diameter of each node), and bigram transition probabilities depicted as directional edges. The two-dimensional layout is meant to minimize the overall distance between all connected nodes and is seeded by spectral clustering to emphasize neighborhood structure. FIG. 8C. Statemap depiction of the difference between blank and TMT. Usage differences are indicated by the newly sized colored circles (upregulation indicated in blue, downregulation indicated in red, blank usages indicated in black). Altered bigram probabilities are indicated in the same color code. FIG. 8D. Mountain plot depicting the joint probability of module expression and spatial position, plotted with respect to the TMT corner (X axis); note that the "bump" two-thirds of the way across the graph occurs due to the two corners equidistant from the odor source. FIG. 8E. Occupancy plot indicating spatial position in which mice after TMT exposure emit an investigatory sniffing module (left) or a pausing module.

[0054] FIGs. 9A-9C depict, in accordance with various embodiments of the present invention, how the AR-HMM disambiguates wild-type, heterozygous and homozygous mice. FIG. 9A. Usage plot of modules exhibited by mice (n = 6 +/+, n = 4 +/-, n = 5 -/-, open field assay, 20 minute trials), sorted by "mutant-ness". Mean usages across animals depicted with dark lines, with bootstrap estimates depicted in fainter lines. FIG. 9B. State map depiction of baseline OFA behavior for +/+ animals as in FIG. 4C (left); difference state maps as in FIG. 4C between the +/+ and +/- genotype (middle), and +/+ and -/- genotype (right). FIG. 9C. Illustration of the "waddle" module in which the hind limbs of the animal are elevated above the shoulder girdle, and the animal locomotes forward with a wobbly gait.

[0055] FIGs. 10A – 10B depict, in accordance with various embodiments of the present invention, how optogenetic perturbation of the motor cortex yields both neomorphic and physiological modules. FIG. 10A. Mountain plot depicting the probability of expression of each behavioral module (each assigned a unique color on the Y axis) as a function of time (X axis), with two seconds of light stimulation initiated at time zero (each plot is the average

of 50 trials). Note that because of the trial structure (in which mice were sequentially exposed to increasing light levels) modest variations in the baseline pattern of behavior are captured before light onset across conditions. Stars indicate two modules that are expressed during baseline conditions that are also upregulated at intermediate powers (11 mW) but not high powers (32 mW); cross indicates pausing module upregulated at light offset. FIG. 10B. Average position of example mice (with arrows indicating orientation over time) of the two modules induced under the highest stimulation conditions. Note that these plots are taken from one animal and representative of the complete dataset (n=4); because of variability in viral expression the threshold power required to elicit behavioral changes varied from animal to animal, but all expressed the spinning behaviors identified in FIG. 10A.

[0056] FIGs. 11A – 11C depict, in accordance with various embodiments of the present invention, how depth imaging reveals block structure in mouse pose dynamics data. FIG. 11A depicts Imaging a mouse in the circular open field with a standard RGB camera (left) and a 3D depth camera (right, mouse height is color mapped, mm = mm above floor) captures the three-dimensional pose of the mouse. FIG. 11B depicts an arrow that indicates the inferred axis of the animal's spine; all mouse images are centered and aligned along this axis to enable quantitative measurements of pose dynamics over time during free behavior. Visualization of pose data reveals inherent block structure within 3D pose dynamics. Compression of pre-processed and spine-aligned data through the random projections technique reveals sporadic sharp transitions in the pose data as it evolves over time. Similar data structure was observed in the raw data and in the height of the spine of the animal as it behaves (upper panel, spine height at any given position is colormapped, mm = mm above floor). When the animal is rearing (as it is here at the beginning of the datastream), its cross-sectional profile with respect to the camera becomes smaller; when the animal is on all fours its profile becomes larger. FIG. 11C shows a changepoints analysis which identifies potential boundaries between these blocks (normalized probability of a changepoint indicated in the trace at the bottom of the behavioral data). Plotting the duration of each block as identified by the changepoints analysis reveals a block duration distribution (n=25, 500 total minutes imaging, mean = 358 ms, SD 495 ms). Mean block duration values are plotted in black, with the duration distribution associated with each individual mouse plotted in gray. FIG. 11C, middle and right. Autocorrelation analysis reveals that the rate of decorrelation in the mouse's pose slows after about 400 milliseconds (left, mean plotted in dark blue, individual mouse autocorrelations plotted in light blue, tau = 340 ± 58 ms). Plotting the ratio in spectral

power between a behaving and dead mouse (right, mean plotted in black, individual mice plotted in grey) reveals most behavioral frequency content is represented between 1 and 6 Hz (mean = 3.75 ± .56 hz);

[0057] FIGs. 12A-12D depict, in accordance with various embodiments of the present invention, how mouse pose dynamics data contains reused behavioral modules. FIG. 12A depicts how a projection of mouse pose data into Principal Components (PC) space (bottom) reveals that the individual blocks identified in the pose data encode reused trajectories. After subjecting mouse pose data to principal components analysis, the values of the first two PCs at each point in time were plotted in a two-dimensional graph (point density is colormapped). Tracing out the path associated with a block highlighted by changepoints analysis (top) identifies a trajectory through PC space (white). By searching through pose data using a template matching procedure, additional examples of this block were identified that encoded similar trajectories through PC space (time indicated as progression from blue to red), suggesting that the template block represented a reused motif of motion. FIG. 12B depicts modeling mouse pose data with the AR-HMM identifies individual behavioral modules. The AR-HMM parses the behavioral data into a limited set of identifiable modules (top – marked "labels", each module is uniquely color coded). Multiple data instances associated with a single behavioral module each take a stereotyped trajectory through PCA space (bottom left, trajectories in green); multiple trajectories define behavioral sequences (bottom center). Depicting the side-on view of the mouse (inferred from depth data, bottom right) reveals that each trajectory within a behavioral sequence encodes a different elemental action (time within the module is indicated as increasingly darker lines, from module start to end). FIG. 12C depicts isometric-view illustrations of the three-dimensional imaging data associated with walk, pause and low rear modules. FIG. 12D depicts cross-likelihood analysis depicting the probability that a data instance assigned to a particular module will be effectively modeled by another module. Cross-likelihoods were computed for the open field dataset, and the likelihood that any given data instance assigned to a particular module would be accurately modeled by a different module is heatmapped (units are nats, where enats is the likelihood ratio); note the high-likelihood diagonal, and the low likelihoods associated for all off-diagonal comparisons. Plotting the same metric on a model trained on synthetic data whose autocorrelation structure matches actual mouse data but which lacks any modularity reveals that the AR-HMM fails to identify modules in the absence of underlying modularity in the training data.

[0058] FIGs. 13A-13B depict, in accordance with various embodiments of the present invention, block and autocorrelation structure in Mouse Depth Imaging Data. FIG. 13A depicts that a block structure is present in random projections data, spine data and raw pixel data derived from aligned mouse pose dynamics. FIG. 13B illustrates that live mice exhibit significant block structure in imaging data (left panels), while dead mice do not (right panels). Compression does not significantly affect autocorrelation structure mouse pose dynamics data. Raw pixels, PCA data and random projections representing the same depth dataset (left panel) all decorrelate at approximately the same rate, demonstrating that data compression does not influence fine-timescale correlation structure in the imaging data. This correlation structure is not observed if mice poses evolve as if taking a Levy flight (middle panel) or random walk (right panel), suggesting that live mice express a specific sub-second autocorrelation structure potentially associated with switching dynamics.

[0059] FIG. 14 depicts, in accordance with various embodiments of the present invention, the variance explained after dimensional rejection using Principal Components Analysis. A Plot comparing variance explained (Y axis) with the number of included PCA dimensions (X axis) reveals that 88 percent of the variance is captured by the first 10 principal components; this number of dimensions was used for data analysis by the AR-HMM.

[0060] FIG. 15 depicts, in accordance with various embodiments of the present invention, comparative modeling of mouse behavior. A series of computational models of behavior were composed, each instantiating a distinct hypothesis about the underlying structure of behavior, and each of these models was trained on mouse behavioral data (in the form of the top 10 principal components extracted from aligned depth data). These models included a Gaussian model (which proposes that mouse behavior is a single Gaussian in pose space), a GMM (a Gaussian Mixture Model, which proposes that mouse behavior is a mixture of Gaussians in pose space), a Gaussian HMM (a Gaussian Hidden Markov Model, which proposes that behavior created from modules, each a Gaussian in pose space, that are interconnected in time with definable transition statistics), a GMM HMM (a Gaussian Mixture Model Hidden Markov Model, which proposes that behavior created from modules, each a mixture of Gaussians in pose space, that are interconnected in time with definable transition statistics), an AR model (which proposes that mouse behavior is a single, continuous autoregressive trajectory through pose space), an AR MM (which proposes that

mouse behavior is built from modules, each of which encodes a autoregressive trajectory through pose space, and which transition from one to another randomly), and a AR sHMM (which proposes that mouse behavior is built from modules, each of which encodes a autoregressive trajectory through pose space, and which transition from one to another with definable transition statistics). The performance of these models at predicting the structure of mouse behavioral data these models to which these models had not been exposed is shown on the Y axis (measured in likelihood units, and normalized to the performance of the Gaussian model), and the ability of each model to predict behavior on a frame-by-frame basis is shown on the X axis (upper). Three slices are taken through this plot at different points in time, demonstrating that the optimal AR HMM outperforms alternative models at timescales at which the switching dynamics inherent in the data come into play (e.g. after more than 10 frames, error bars are SEM).

[0061] FIG. 16 depicts, in accordance with various embodiments of the present invention, duration distributions for blocks and modules that are qualitatively similar. Percentage of blocks/modules of a given duration (Y axis) plotted against block duration (X axis) reveals roughly similar duration distributions for the changepoints algorithm identified blocks, and the model-identified behavioral modules. These distributions are expected to be similar although not identical, as the changepoints algorithm identifies local changes in data structure, while the model identifies modules based upon their contents and their transition statistics; note that the model has no direct access to the "local fracture" metrics used by the changepoints algorithm.

[0062] FIG. 17 depicts, in accordance with various embodiments of the present invention, how shuffling behavioral data at fast timescales that lowers AR-HMM performance.

[0063] FIG. 18 depicts, in accordance with various embodiments of the present invention, a visualization of model-generated mouse behavior, each of the models was trained on behavioral data (left) and then allowed to generate its "dream" version of mouse behavior (right); here that output is visualized at the shape of the spine of the animal over time. The individual modules identified by each model are indicated as a color code underneath each model (marked "labels").

[0064] FIG. 19 depicts, in accordance with various embodiments of the present invention, how module interconnectivity is sparse. Without thresholding the average module is interconnected with 16.85 ±.95 other modules; this modest interconnectivity falls sharply with even modest thresholding (X axis, thresholding applied to bigram probabilities), consistent with sparse temporal interconnectivity between individual behavioral modules.

[0065] FIG. 20 depicts, in accordance with various embodiments of the present invention, the identification filtering parameters. To filter data from the Kinect we used iterative an iterative median filtering approach in which we applied a median filter iteratively both in space and in time; this approach has been shown to effectively maintain data structure while smoothing away noise. To identify optimal filter settings, we imaged dead mice that were differentially posed in rigor mortis; ideal filter settings would distinguish mice that were posed differently, but be unable to distinguish data from the same mouse. Filter setting are indicated as ((pixels), (frames)) with the numbers within each parenthesis referring to the iterative settings for each round of filtering. To assess filter performance, we computed a within/between pose correlation ratio (Y axis), in which the mean spatial correlation for all frames of the same pose was divided by the mean spatial correlation for all frames of different poses. This revealed that light filtering (with settings ((3), (3,5))) optimized discriminability in the data.

[0066] FIG. 21 depicts, in accordance with various embodiments of the present invention, identifying changepoint algorithm parameters. By optimizing against the changepoints ratio (number of changepoints identified in live mice versus dead mice, Y axis), clear optimal values were identified via grid scanning for sigma and H (left two panels). This changepoint ratio was not highly sensitive to K; a setting of 48 (at the observed maximum) was therefore chosen.

[0067] FIG. 22 depicts, in accordance with various embodiments of the present invention, graphical model for the AR-HMM. The shaded nodes labeled y_t for time indices t=1,2,…,T represent the preprocessed 3D data sequence. Each such data node y_t has a corresponding state node x_t which assigns that data frame to a behavioral mode. The other nodes represent the parameters which govern the transitions between modes (i.e. the transition matrix π) and the autoregressive dynamical parameters for each mode (i.e. the set of parameters θ).

[0068] FIG. 23 depicts, in accordance with various embodiments of the present invention, a graphical depiction of dimensionality reduction using a neural network to form an image manifold.

[0069] FIG. 24 depicts, in accordance with various embodiments of the present invention, graphical representations of structured variable autoencoders;

[0070] FIG. 25 depicts, in accordance with various embodiments of the present invention, the application of structured variation autoencoders. Figure 4 is an example of generative completion using video data of a mouse. Figure 5 is a graphical example of filtering and production for 1D bouncing data. Figure 6 is comparison on the dot problem of natural gradient updates (bottom trend line) and standard gradient updates (above). Figure 7 is a 2D grid in mouse image manifold coordinates.

[0071] In the drawings, the same reference numbers and any acronyms identify elements or acts with the same or similar structure or functionality for ease of understanding and convenience. To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the Figure number in which that element is first introduced.

DETAILED DESCRIPTION

[0072] In some embodiments, properties such as dimensions, shapes, relative positions, and so forth, used to describe and claim certain embodiments of the invention are to be understood as being modified by the term "about."

[0073] Various examples of the invention will now be described. The following description provides specific details for a thorough understanding and enabling description of these examples. One skilled in the relevant art will understand, however, that the invention may be practiced without many of these details. Likewise, one skilled in the relevant art will also understand that the invention can include many other obvious features not described in detail herein. Additionally, some well-known structures or functions may not be shown or described in detail below, so as to avoid unnecessarily obscuring the relevant description.

[0074] The terminology used below is to be interpreted in its broadest reasonable manner, even though it is being used in conjunction with a detailed description of certain

specific examples of the invention. Indeed, certain terms may even be emphasized below; however, any terminology intended to be interpreted in any restricted manner will be overtly and specifically defined as such in this Detailed Description section.

[0075] While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any inventions or of what may be claimed, but rather as descriptions of features specific to particular implementations of particular inventions. Certain features that are described in this specification in the context of separate implementations can also be implemented in combination in a single implementation. Conversely, various features that are described in the context of a single implementation can also be implemented in multiple implementations separately or in any suitable sub-combination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a sub-combination or variation of a sub-combination.

[0076] Similarly while operations may be depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the implementations described above should not be understood as requiring such separation in all implementations, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

Overview

[0077] The inventors have discovered systems and methods for automatically and objectively identifying and classifying behavior modules of animals by processing video data of the animals. These systems may classify animal behavioral state by quantitative measurement, processing, and analysis of an animal posture or posture trajectory in three-dimensions using a depth camera. These system and methods obviate the need for *a priori* definition for what should constitute a measurable unit of action, thus making the classification of behavioral states objective and unsupervised.

[0078] In one aspect, the invention relates to a method for analyzing the motion of a subject to separate it into sub-second modules, the method comprising: (i) processing three dimensional video data that represent the motion of the subject using a computational model to partition the video data into at least one set of sub-second modules and at least one set of transition periods between the sub-second modules; and (ii) assigning the at least one set of sub-second modules to a category that represents a type of animal behavior.

[0079] FIG. 1 illustrates an embodiment of the process a system may utilize to automatically classify video frames or sets of frames into behavior modules. For instance, the system may include a video recorder 100 and tracking system 110. In some embodiments, video recorder 100 may be a 3D depth camera and the tracking system 110 may project structured infrared light into the experimental field 10. Infrared receivers on the tracking system may be able to determine the location of an object based on parallax. In some embodiments, the video recorder 100 may be connected to the tracking system 110 or in some embodiments they may be separate components.

[0080] The video recorder 100 may output data related to video images and or tracking data from the tracking system 110 to a computing device 113. In some embodiments, the computing device 113 will perform pre-processing of the data locally before sending over a network 120 to be analyzed by a server 130 and to be saved in a database 160. In other embodiments, the data may be processed, and fit locally on a computing device 113.

[0081] In one embodiment, a 3D depth camera 100 is used to obtain a stream of video images of the animal 50 having both area and depth information. The background image (the empty experimental area) is then removed from each of the plurality of images to generate processed images having light and dark areas. The contours of the light areas in the plurality of processed images can be found and parameters from both area and depth image information within the contours can then be extracted to form a plurality of multi-dimensional data points, each data point representing the posture of the animal at a specific time. The posture data points can then be clustered so that point clusters represent animal behaviors.

[0082] Then, the preprocessed depth-camera video data may be input into the various models in order to classify the video data into sub-second “modules” and transition periods that describe repeated units of behavior that are assembled together to form coherent

behaviors observable by the human eye. The output of the models that classify the video data into modules may output several key parameters including: (1) the number of behavioral modules observed within a given set of experimental data (i.e. the number of states), (2) the parameters that describe the pattern of motion expressed by the mouse associated with any given module (i.e. state-specific autoregressive dynamical parameters), (3) the parameters that describe how often any particular module transitions to any other module (i.e. the state transition matrix), and (4) for each video frame an assignment of that frame to a behavioral module (i.e. a state sequence associated with each data sequence). In some embodiments, these latent variables were defined by a generative probabilistic process and were simultaneously estimated using Bayesian inference algorithms.

Camera Setup and Initialization

[0083] Various methods may be utilized to record and track video images of animals 50 (e.g., mice). In some embodiments, the video recorded may be recorded in three dimensions. Various apparatuses are available for this function, for instance the experiments disclosed herein utilized Microsoft's Kinect for Windows. In other embodiments, the following additional apparatuses may be utilized: (1) stereo-vision cameras (which may include groups of two or more two-dimensional cameras calibrated to produce a depth image, (2) time-of-flight depth cameras (e.g. CamCube, PrimeSense, Microsoft Kinect 2, structured illumination depth cameras (e.g. Microsoft Kinect 1), and x-ray video.

[0084] The video recorder 100 and tracking system 110 may project structured infrared light onto the imaging field 10, and compute the three-dimensional position of objects in the imaging field 10 upon parallax. The Microsoft Kinect for Windows has a minimum working distance (in Near Mode) of 0.5 meters; by quantitating the number of missing depth pixels within an imaged field, the optimal sensor positioned may be determined. For example, the inventors have discovered that the optimal sensor position for a Kinect is between 0.6 and 0.75 meters away from the experimental field depending on ambient light conditions and assay material.

Data Acquisition

[0085] Data output from the video recorder 100 and tracking system 110 may be received by and processed by a computing device 113 that processes the depth frames and saves them in a suitable format (e.g., binary or other format). In some embodiments, the data from the video recorder 100 and tracking system 110 may be directly output over a network 120 to a server 130, or may be temporarily buffered and/or sent over a USB or other connection to an associated computing device 113 that temporarily stores the data before sending over a network 120 to a centralized server 130 for further processing. In other embodiments, the data may be processed by an associated computer 113 without sending over a network 120.

[0086] For instance, in some embodiments, data output from a Kinect may be sent to a computer over a USB port utilizing custom Matlab or other software to interface the Kinect via the official Microsoft .NET API that retrieves depth frames at a rate of 30 frames per second and saves them in raw binary format (16-bit signed integers) to an external hard-drive or other storage device. Because USB3.0 has sufficient bandwidth to allow streaming of the data to an external hard-drive or computing device with storage in real-time. However, in some embodiments, a network may not have sufficient bandwidth to remotely stream the data in real time.

Data Pre-Processing

[0087] In some embodiments, after the raw images of the video data are saved, various pre-processing may take place to isolate the animal in the video data and orient the images of the animal along a common axis for further processing. In some embodiments, the orientation of the head may be utilized to orient the images in a common direction. In other embodiments, an inferred direction of the spine may be incorporated.

[0088] For instance, tracking the evolution of an imaged mouse's pose over time requires identifying the mouse within a given video sequence, segmenting the mouse from the background (in this case the apparatus the mouse is exploring), orienting the isolated image of the mouse along the axis of its spine, correcting the image for perspective distortions, and then compressing the image for processing by the model.

Isolating Video Data of the Animal

[0089] FIG. 2A illustrates a process the system may perform for isolating a region of interest and subtracting background images to isolate the video data of the animal 50. First, to isolate the experimental arena in which the mouse is behaving, the system may first identify a region-of-interest (ROI) 210 for further analysis. In other embodiments, the region-of-interest 210 may include the entire field of view 10 of recorded video data. To isolate the region, one may manually trace along the outside edge of any imaged arena; pixels outside the ROI 210 may be set to zero to prevent spurious object detection. In other embodiments, the system may automatically define a ROI 210 using various methods. In some embodiments, the system may filter the raw imaging data with an iterative median filter, which is well suited to removing correlated noise from the sensor, for example, in a Kinect.

[0090] After selecting the region of interest 210, the raw images may be cropped to the region of interest 215. Then missing pixel values can be input 225, after which an X, Y, and Z position can be calculated 230 for each pixel, and the pixel position can be resampled. Accordingly, the images can be resampled onto real-world coordinates. Then, the system calculates the median real-world coordinate background image 240, and those can be subtracted form the real-world coordinate images 245.

[0091] To subtract the background image of the arena from the video data, various techniques may be performed, including for example, subtracting the median value of a portion of the video data for a set time period (e.g. 30 seconds). For instance, in some embodiments, the first 30 seconds of data from any imaging stream may be subtracted from all video frames and any spurious values less than zero may be reset to zero.

[0092] To further ensure the analysis focuses on the animal, the system may binarize the image (or perform similar processes using thresholds) and eliminate any objects that did not survive a certain number of iterations of morphological opening. Accordingly, once this is finished, the system may perform the additional processing illustrated in FIG. 2B. Accordingly, the background subtracted images (mouse video data) 250 may be filtered and the artifacts may be removed 255. In some embodiments, this may involve iterative median filtering.

[0093] The animal in the image data may then be identified by defining it as the largest object within the arena that survived the subtraction and masking procedures, or by blob detection 260. Then, the image of the mouse may be extracted 265.

Identifying the Orientation of the Animal

[0094] The centroid of the animal (e.g. mouse) may then be identified 270 as the center-of-mass of the preprocessed image or by other suitable methods; an ellipse may then be fit to its contour 285 to detect its overall orientation. In order to properly orient the mouse 280, various machine learning algorithms may be trained (e.g. a random forest classifier) on a set of manually-oriented extracted mouse images. Given an image, the orientation algorithm then returns an output indicating whether the mouse's head is oriented correctly or not.

[0095] Once the position is identified, additional information may be extracted 275 from the video data including the centroid, head and tail positions of the animal, orientation, length, width, height, and each of their first derivatives with respect to time. Characterization of the animal's pose dynamics required correction of perspective distortion in the X and Y axes. This distortion may be corrected by first generating a tuple of (x,y,z) coordinates for each pixel in real-world coordinates, and then resampling those coordinates to fall on an even grid in the (x,y) plane using Delaunay triangulation.

Output to a Model Based or Model-Free Algorithm

[0096] As illustrated in FIG. 3, the output of the orientation corrected images in some embodiments will be to a principle component analysis time series 310 or other statistical methods for reducing data points. In some embodiments, the data will be run through a model fitting algorithm 315 such as the AR-HMM algorithm or SLDS SVAE algorithm disclosed herein, or may be run through a model free algorithm 320 as disclosed in order to identify behavior modules 300 contained within the video data. Additionally, in some embodiments, the PCA time series will not be performed. In some embodiments, a multi-layer perceptron will be utilized to reduce the dimensionality.

[0097] In embodiments with model-free algorithms 320, various combinations of algorithms can be utilized with the goal of isolating sub-second modules of behavior that have similar orientation profile and trajectories. Disclosed herein are some examples of these

algorithms, however, additional algorithms could be envisioned that segment the data into behavior modules.

Reducing dimensionality of Image

[0098] In some embodiments, both that include model-free algorithms 320 or the model fitting 315 algorithm, the information captured in each pixel often is either highly correlated (neighboring pixels) or uninformative (pixels on the border of the image that never represent the mouse's body). To both reduce redundant dimensions and make modeling computationally tractable, various techniques may be employed to dimensionally reduce each image. For example, a five-level wavelet decomposition may be performed, thereby transforming the image into a representation in which each dimension captured and pooled information at a single spatial scale; in this transformation, some dimensions may code explicitly for fine edges on the scale of a few millimeters, while others encoded broad changes over spatial scales of centimeters.

[0099] This wavelet decomposition however will expand the dimensionality of the image. In order to reduce this dimensionality, various techniques may be applied.

[00100] In some embodiments, random projections technique may be utilized to reduce the dimensionality of the data. Random projections is an approach that produces new dimensions derived from an original signal, with dimensionality D_orig, by randomly weighting each original dimension, and then summing each dimension according to that weighting, producing a single number per data point. This procedure can be repeated several times, with new random weightings, to produce a set of "randomly projected" dimensions. The Johnson–Lindenstrauss lemma shows that distances between points in the original dataset with dimensionality D_orig is preserved in the randomly projected dimensions, D_proj, where D_proj < D_orig.

[00101] In other embodiments, principal components analysis may then be applied to these vectors, in order to project the wavelet coefficients into ten dimensions, which the inventors have found still captures > 95% of total variance. For instance, principle components may be built using a canonical dataset of 25 C57 BL/6 mice, aged 6 weeks, recorded for 20 minutes each, and all datasets were projected into this common pose space. Accordingly, the output of the PCA may then be input into the modeling algorithm for module identification.

[00102] However, PCA will reduce the dimensionality to a linear space. The inventors have discovered that reduction of dimensionality to a linear space will not accommodate for various changes in the mice that are not related to behavior. This includes changes in mouse size, mouse breed, etc.

[00103] Accordingly, the inventors have discovered that using certain kinds of neural networks, such a multi-layer perceptrons, one can effectively reduce the dimensionality of the images while maintaining a more robust represnetation. For instance, the inventors proposed a structured variational autoencoder for dimensionality reduction as disclosed herein. Furthermore, these dimensionality reduced images provide an effective method to develop models that are agnostic to the size of the mouse, or other animals, and can account for other changes that are not behavior related. For instance, some neural networks that reduce the dimensionality to a ten dimensional image manifold may be utilized.

Model-Free Algorithms: Identifying Behavior Module Length

[00104] In some embodiments that have a model-free algorithm 320, in order to evaluate time-scale over which an animal's behavior is self-similar — which reflects the rate at which an animal transitions from one pattern of motion to another — an autocorrelation analysis may be performed. Because some data smoothing is required to remove sensor-specific noise, computing the auto-correlogram as the statistical correlation between time-lagged versions of a signal will result in a declining auto-correlogram, even for an animal (e.g. mouse) that is posed in rigor mortis. Therefore, correlation distance between all 10 dimensions of the mouse's pose data may be utilized as the comparator between time-lagged versions of the time-series signal in question, resulting in a flat autocorrelation function of value ~1.0 for a dead animal, and a declining autocorrelation function for a behaving animal (e.g., mouse). The rate at which this auto-correlogram declines in a behavior mouse is a measure of a fundamental timescale of behavior, which may be characterized as a time-constant, tau, of an exponentially-decaying curve. Tau can be fitted using the Levenberg-Marquardt algorithm (non-linear least squares) using the SciPy optimization package.

[00105] In some embodiments, a power-spectral density (PSD) analysis may be performed on the mouse behavioral data to further analyze its time domain structure. For instance, a Wiener filter may be utilized to identify the time frequencies that must be boosted in the signal derived from a dead mouse in order to best match a behaving mouse. This can be

implemented simply by taking the ratio of the PSD of a behaving mouse over the PSD of a dead mouse. In some embodiments, the PSD may be computed using the Welch periodogram method, which takes the average PSD over a sliding window across the entire signal.

Model-Free Algorithms: Locating Change Points for Transition Periods

[00106] In some embodiments where a model is not used to identify modules 320, various methods may be utilized to identify the changepoints for the transition periods. Plotting the random projections of the mouse depth image over time yields obvious striations, each a potential changepoint over time. To automate the identification of these changepoints, which represent potential boundaries between the block structure apparent in the random projections data, a simple changepoint identification technique called the filtered derivative algorithm may be utilized. For example, an algorithm can be employed that calculates the derivative of the per-frame unit-normalized random projections with a lag of k=4 frames. For each time point, for each dimension, an algorithm may determine whether the signal has crossed some threshold h=0.15 mm. Then, the binary changepoint indicator signal may be summed across each of D=300 random projection dimensions, and then the resulting 1D signal may be smoothed with a Gaussian filter with a kernel standard deviation of sigma=0.43 frames. Change points may then be identified as the local maxima of this smoothed 1D time-series. This procedure depends in part upon the specific values of the parameters k, h and sigma; for example, those values that maximize the number of changepoints in the behaving mouse while yielding no change points in a dead mouse may be utilized.

Model-Free Algorithms: Identifying Similar or Repeating Modules

[00107] In some embodiments, where data is being analyzed without a model 320, certain algorithms may be utilized to identify similar or repeated modules. Accordingly, a set of repeating modules may be identified as the vocabulary or syllables of the animal behavior. Therefore, to determine whether any reasonably long snippet of behavior (greater than just a few frames) was ever "repeated" (without reliance on a underlying model for behavior), the system may utilize a template matching procedure to identify similar trajectories through PCA or MLP manifold space. To identify similar trajectories, for example, the systems and methods may calculate the Euclidean distance between some target snippet, the "template", and every possible snippet of equal length (often defined by the approximate block

boundaries identified by changepoints analysis). Other similar methods could be employed for identifying modules, including other statistical based methods.

[00108] In some embodiments, the collection of modules that are similar would be selected as the most similar snippets, ignoring snippets discovered that were shifted less than 1 second from each other (to ensure we select behavioral snippets that occur distanced in time from each other, and also in separate mice).

Data Modeling

[00109] In other embodiments, systems and methods may be employed that identify behavior modules in video data utilizing data models 315. For instance, a data model may implement the well-established paradigm of generative probabilistic modeling, which is often used to model complex dynamical processes. This class of models is generative in the sense that it describes a process by which observed data can be synthetically generated by the model itself, and they are probabilistic because that process is defined mathematically in terms of sampling from probability distributions. In addition, by fitting an interpretable model to data, the data were 'parsed' in a manner that revealed the latent variable structure that the model posits gave rise to the data (including parameters describing the number and identities of the states as well as parameters describing the transitions between the states).

[00110] In some embodiments, the model 315 may be expressed utilizing a Bayesian framework. The Bayesian framework provides a natural way to express hierarchical models for the organization of behavior, priors or regularizers that reflect known or observed constraints on the patterns of motion within the 3D dataset, and a coherent representation of uncertainty. This framework also provides significant and well-established computational machinery for inferring key parameters of any model. Within the Bayesian framework, for a particular model structure (e.g. the spatiotemporal nature of the states and their possible transitions) and prior distributions on the latent variables, the data fixes a posterior distribution over the latent variables.

[00111] Below, the model-based methods used to characterize behavior are defined in two steps: first, a mathematical definition of the generative model and priors used, and second, a description of the inference algorithms.

Example Model for Identifying Behavior Modules – AR-HMM

[00112] In some embodiments, systems may utilize a discrete-time hidden Markov model 315 (HMM) to identify behavior modules. HMMs encompass a range of stochastic processes for modeling sequential and time series data. The HMM model posits that at each point in time (e.g. for every frame of imaging data), the mouse is within a discrete state (Markov state) that can be given a label. Each Markov state represents a brief three-dimensional motif of motion the animal undertakes while within that state. Because observed three-dimensional behavior of mice appears to depend upon the specific pattern of motion the animal expressed in the immediate past, ideally each Markov state would predict the mouse's future behavior based upon its immediate past pose dynamics. Each Markov state is therefore composed of both a latent discrete component, which identifies the behavioral mode of the animal, and a number of lags of the observation sequence, which are used to predict the short-timescale behavior of the animal based on the behavioral mode. This model structure is often called a switching vector-autoregressive (SVAR) model or autoregressive HMM (AR-HMM).

[00113] FIG. 4 provides an example of how an AR-HMM algorithm can convert input data (spine aligned depth imaging data 305 that has been dimensionally reduced 405 using PCA 310) into a fit model that describes the number of behavioral modules and the trajectories they encode through PCA space, the module-specific duration distributions that govern how long any trajectory within a given module lasts, and the transition matrix that describes how these individual modules interconnect over time.

[00114] In addition, the AR-HMM can be configured to assign a label to every frame of the training data associating it with a given behavioral module. After pre-processing and dimensional reduction 405, imaging data is broken into training 415 and test sets 410. The training set 415 is then submitted to the AR-HMM 315. After randomly initializing the parameters of the model 315 (which here refers to the autoregressive parameters that describe each module's trajectory through PCA space, the transition matrix describing the probabilities that governs temporal interconnections between modules, the duration distribution parameters that describe how long any instance of a given module is likely to last, and the labels assigned to each frame of imaging data associating that frame with a particular module) the AR-HMM attempts to fit the model 315 by varying one parameter while holding the others constant. The AR-HMM alternates between two main updates: the

algorithm 315 first attempts to segment the imaging data into modules given a fixed set of transition statistics and a fixed description of the AR parameters that describe any given module, and then the algorithm switches to fixing the segmentation and updating the transition matrix and the AR parameters 455. The AR-HMM 315 uses a similar approach to assigning any given frame of imaging data to a given module. It first computes the probability of that a given module is the "correct" module, which is proportional to a measure of how well the state's corresponding autoregressive parameters 455 describe the data at that time index and how well the resulting state transitions agree with the transition matrix 450.

[00115] In the second step, the AR-HMM 315 varies the autoregressive parameters 455 and transition parameters 450 to better fit the assigned data, thus updating the each of the behavioral modules and the model of the transitions among modes. The product of this process are the parameters described 455, the quality of these parameters in terms of describing behavior are then evaluated using likelihood measurements of the data that was held-out from training 475.

[00116] By identifying the discrete latent states 445 associated with 3D pose sequence data, an HMM model 315 can identify segments of data that exhibit similar short-timescale motion dynamics and explain such segments in terms of reused autoregressive parameters. For each observation sequence there is an unobserved state sequence: if the discrete state at time index t is x_t=i, then the probability that the discrete state x_(t+1) takes on value j is a deterministic function of i and j and is independent of all previous states. Symbolically,

[00117] $$p(x_{t+1}|x_t, x_{t-1}, x_{t-2}, \ldots, x_1) = p(x_{t+1}|\, x_t)$$

[00118] $$p(x_{t+1} = j\,|x_t = i) = \pi_{ij}$$

[00119] where π is a transition matrix 450 in which the (i,j) element is the probability of transitioning from state i to state j. In some embodiments, the discrete state's dynamics may be fully parameterized by the transition matrix, which is considered here not to change with time. One of the tasks of the inference algorithm (described below) was to infer probable values for the discrete state sequences and the transition matrix governing their deployment, thus inferring a sequence of reused behavioral modules and transition patterns that govern how these modules are connected over time.

[00120] Given a discrete state sequence, a corresponding 3D pose data sequence can be modeled as a conditionally vector autoregressive (VAR) process. Each state-specific vector autoregression can capture short-timescale motion dynamics particular to the corresponding discrete state; in other words, each behavioral module can be modeled as its own autoregressive process. More precisely, given the discrete state x_t of the system at any time index t, the value of the observed data vector at that time index y_t is distributed according to a state-specific noisy regression on K previous values of the observation sequence, y_(t-1),…,y_(t-K). The inference algorithm may also be tasked with inferring the most probable values for each state's autoregressive dynamical parameters as well as the number of lags used in the dynamics.

[00121] In some embodiments, these switching autoregressive dynamics defined the core of the AR-HMM. However, because different animal populations or experimental conditions are expected to give rise to differences in behavior, when considering two or more such experimental conditions models may be built hierarchically: different experimental conditions may be allowed to share the same library of state-specific VAR dynamics but learned their own transition patterns as well as any unique VAR dynamical modes. This simple extension allows a model to reveal changes in the parameters due to changes in the experiment. Furthermore, the compositional Bayesian inference algorithms employed immediately extends such hierarchical models.

[00122] To employ Bayesian inference methods, unknown quantities, including the transition matrix 450 and the autoregressive parameters 455 that describe each state 445, can be treated with a uniform representation as latent random variables. In particular, weak prior distributions 465 can be placed on these quantities and their posterior distributions 465 after conditioning on observed 3D imaging data were investigated. For the autoregressive parameters, a prior that included a Lasso-like penalty can be used to encourage uninformative lag indices to have their corresponding regression matrix coefficients tend to zero.

[00123] For the transition matrix 450, a hierarchical Dirichlet process 435 prior can used, to regularize the number of discrete latent states 445. In addition, the transition matrix 450 prior also included a sticky bias, which is a single nonnegative number that controlled the tendency of the discrete states to self-transition. Because this parameter controls the timescale of the inferred switching dynamics, this parameter can be set such that the output of the model inference algorithms matches (as closely as possible) the model-free duration

distribution determined by a changepoint analysis as disclosed herein (or other method of identifying the module length) and the autocorrelogram generated from the preprocessed and unmodeled 3D pose data. In some embodiments, this parameter can be tuned — for example to define the prior over the timescale of behavior.

[00124] In some embodiments, simpler models can be used than the AR-HMM model by removing certain portions of the model structure. For instance, removing the discrete switching dynamics captured in the transition matrix and replacing them with a mixture model may generate an alternative model in which the distribution over each discrete state does not depend on its previous state. This would be the case if animals had a set of behavioral modules from which to choose, and the likelihoods of expressing any given one of them did not depend on the order in which they appear. This simplification resulted in the autoregressive mixture model (AR-MM).

[00125] Alternatively, replacing the conditionally autoregressive dynamics with simple state-specific Gaussian emissions results in a Gaussian-emission HMM (G-HMM); this model explores the hypothesis that each behavioral module is best described by a simple pose, rather than being a dynamical trajectory. Applying both simplifications yields a Gaussian mixture model (G-MM), in which behavior is simply a sequence of poses over time in which the probability of expressing any given pose does not depend on the prior pose. Removing the switching dynamics yields a pure autoregressive (AR) or linear dynamical system (LDS) model, in which behavior is described as a trajectory through pose space without any reused discrete behavioral modules at all.

Analysis of Behavior Modules

[00126] In some embodiments, systems may provide the ability to provide an indication of the relationship between behavior modules, describe the most frequently used behavior modules, or perform other useful analysis of behavior modules.

[00127] For example, in order to represent the grammatical relationship between behavioral syllables, the probability (e.g. bigram) that two syllables were found occurring one after the other (a "bigram" of modules) can be calculated as a fraction of all observed bigrams. In some embodiments, to calculate this value for each pair (i,j) of modules, for example, a square n x n matrix, A, may be utilized where n is the number of total modules in the label sequence. Then, the systems and methods may scan through the label sequences that

were saved at the last iteration of Gibbs sampling, incrementing the entry A[i,j] for every time the system identifies a syllable i directly preceding a syllable j. At the end of the label sequence, the system may divide by the number of total bigrams observed.

[00128] In order to visually organize those modules that were specifically up-regulated or selectively expressed as a result of a manipulation, the system may assign a selectivity index to each module. For example, where p(condition) indicates the percent usage of a module in a condition, the system may sort modules in the circular open field versus square box comparison by (p(circle)-p(square)/(p(circle) + p(square)). In the comparison between blank odor and fox odor (TMT), the system may sort modules by (p(tmt) - p(blank)) / (p(tmt) + p(blank)).

Statemap visualizations

[00129] The system may also output the syllable bigram probabilities and syllable usages on n syllables on a graph $G = (V, E)$ in which each node $i \in V = \{1,2, \ldots, n\}$ corresponds to syllable i and each directed edge $(i, j) \in E = \{1,2, \ldots, n\}^2 \setminus \{(i, i): i \in V\}$ corresponds to a bigram. The graph may be output as a set of circular nodes and directed arcs so that the size of each node is proportional to the corresponding syllable's usage and the width and opacity of each arc is proportional to the corresponding bigram's probability within a minimum and maximum range depicted in the figure legends. To lay out each graph in a reproducible non-(pseudo-)random way (up to global rotation of the figure), the system may initialize the position of the nodes using the spectral layout algorithm and fine-tuned node positions using the Fructherman-Reingold iterative force-directed layout algorithm; we used both algorithms can be used as implemented in the NetworkX software package.

Overview of Main Inference Algorithms

[00130] In some embodiments, inference algorithms may be applied to the models 315 to estimate the parameters. For example, an approximate Bayesian inference can be performed using Gibbs sampling, a Markov Chain Monte Carlo (MCMC) inference algorithm. In the MCMC paradigm, the inference algorithm constructs approximate samples from the posterior distribution of interest, and these samples are used to compute averages or as a proxy for posterior modes. The sequence of samples produced by the algorithm dwells in regions of high posterior probability while escaping regions of low posterior probability or bad local optima. In the main AR-HMM model, the latent variables of interest include the

vector autoregressive parameters, the hidden discrete state sequence, and the transition matrix (e.g. the autoregressive parameters that define the pose dynamics within any given behavioral module, the sequence of the modules, and the transition probabilities between any given module and any other module). Applying the MCMC inference algorithm to 3D imaging data generate a set of samples of these latent variables for the AR-HMM.

[00131] The Gibbs sampling algorithm has a natural alternating structure, directly analogous to the alternating structure of expectation-maximization (EM) and variational mean field algorithms. Applied to the AR-HMM, after initialization to a random sample from the prior, the algorithm can be alternated between two main updates: first, the algorithm can resample the hidden discrete state sequences given the transition matrix and autoregressive parameters, and second, the algorithm can resample the parameters given the hidden states.

[00132] In other words, the algorithm 315 first attempts to segment the imaging data into modules 300 given a fixed set of transition statistics and a fixed description of the AR parameters that describe any given module, and then the algorithm switches to fixing the segmentation and updating the transition matrix 450 and the AR parameters 455. To assign each of the 3D pose video frames to one of the behavioral modes 300 in the first step of this process, the state label 445 for a particular time index can be sampled randomly from the set of possible discrete states, where the probability of sampling a given state can be proportional to a measure of how well the state's corresponding autoregressive parameters described the data at that time index and how well the resulting state transitions agree with the transition matrix 450. In the second step, given the assignment of data subsequences to states, the autoregressive parameters and transition parameters can be resampled to fit the assigned data, thus updating the dynamical model of each of the behavioral modes and the model of the transitions among modes. The procedure implemented by the Gibbs sampling algorithm can be noisy, enabling the algorithm to escape local maxima that may prevent the algorithm from effectively exploring parameter space.

EXAMPLES

[00133] Below are disclosed examples of the specific implementations of the models described herein for performing the disclosed examples. Variations of these models may be implemented to identify behavior models.

Prior on the transition matrix

[00134] A sticky HDP prior was placed on the transition matrix π with concentration parameters $\alpha, \gamma > 0$ and sticky parameter $\kappa > 0$

[00135]
$$\begin{aligned} \rho_j &\overset{\text{iid}}{\sim} \text{Beta}(1,\gamma) & \beta_i &= (1-\rho_i)\prod_{j<i}\rho_j \\ \pi_i &\overset{\text{iid}}{\sim} \text{DP}(\alpha\beta + \kappa\delta_i) & i &= 1,2,\ldots \end{aligned}$$

[00136] where δ_{ij} is 1 when $i = j$ and is 0 otherwise and π_i denotes the ith row of π. Gamma priors are placed on α and γ, setting $\alpha \sim \text{Gamma}(1,1/100)$ and $\gamma \sim \text{Gamma}(1,1/100)$.

Generation of the discrete state sequence

[00137] Given the transition matrix, the prior on a discrete state sequence x was

[00138] $$x_t \sim \pi_{x_{t-1}} \quad t = 2,3,\ldots,T$$

[00139] where x_1 is generated by the stationary distribution under π.

Prior on the autoregressive parameters

[00140] The autoregressive parameters $\theta = \{\theta^{(i)}\}_{i=1}^{\infty} = \{A^{(i)}, b^{(i)}, \Sigma\ (i)\}$ for each state $i = 1,2,\ldots$ were sampled from a Matrix Normal Inverse-Wishart prior :

[00141] $$(A,b), \Sigma \sim \text{MNIW}(\nu_0, S_0, M_0, K_0)$$

[00142] or equivalently

[00143]
$$\begin{aligned} \Sigma &\sim \text{InvWishart}(\nu_0, S_0) \\ \text{vec}((A,b)) &\sim \text{Normal}(\text{vec}(M_0), \Sigma \otimes K_0) \end{aligned}$$

[00144] where $\otimes$ denotes a Kronecker product and (A,b) denotes the matrix formed by appending b to A as a column. In addition, a block ARD prior on K_0 is used to encourage uninformative lags to be shrunk to zero:

[00145] $$K_0 = \text{diag}(k_1,\ldots,k_{KD}) \qquad k_i \overset{\text{iid}}{\sim} \text{InvGamma}(1/25,1/25).$$

Generation of the 3D pose sequence principle components

[00146] Given the autoregressive parameters and discrete state sequence, the data sequence y was generated according to an affine autoregressiion :

[00147] $$y_t \sim \text{Normal}(A^{(x_t)}\tilde{y}_{t-1} + b^{(x_t)}, \Sigma^{(x_t)}) \quad t = K+1, K+2, \ldots, T$$

[00148] where $\tilde{y}$ denotes a vector of K lags:

[00149] $$\tilde{y}_t \overset{\text{def}}{=} [y_{t-K}^T \quad y_{t-K+1}^T \quad \cdots \quad y_{t-1}^T]^T$$

[00150] The alternative models are special cases of the AR-HMM and were constructed by adding constraints. In particular, the Gaussian-emission HMM (G-HMM) corresponds to constraining $A^{(i)} = 0$ for each state index i. Similarly, the autoregressive mixture (AR-MM) and Gaussian mixture (GMM) correspond to constraining the transition matrix to be constant across rows, $\pi_{ij} = \pi_{i'j} = \pi_j$ for each i and i', in the AR-HMM and G-HMM, respectively.

Specific implementation of inference algorithms to examples

[00151] As discussed above, the Gibbs sampling inference algorithm alternated between two principal stages: updating the segmentation of the data into modules given a fixed transition matrix and autoregressive parameters, and updating the transition matrix and autoregressive parameters given a fixed segmentation. Mathematically, updating the segmentation sampled the label sequence x conditioned on the values of the data y, the autoregressive parameters θ, and the transition matrix π; that is, sampling the conditional random variable x | θ,π,y. Similarly, updating the transition matrix and autoregressive parameters given the segmentation sampled π|x and θ|x,y, respectively.

[00152] For inference in the AR-HMM the weak limit approximation to the Dirichlet process was used, in which the infinite model was approximated by a finite one. That is, choosing some finite approximation parameter L, β and π were modeled using finite Dirichlet distributions of size L

[00153] $$\beta \sim \text{Dir}(\gamma/\text{L}, \cdots, \gamma/\text{L})$$

[00154] $$\pi_k \sim \text{Dir}(\alpha\beta_1, \cdots, \alpha\beta_j + \kappa\delta_{kj}, \cdots, \alpha\beta_L)$$

[00155] where π_k denotes the ith row of the transition matrix. This finite representation of the transition matrix allowed the state sequence x to be resampled as a block and for large L provides an arbitrarily good approximation to the infinite Dirichlet process.

[00156] Using a weak limit approximation, the Gibbs sampler for the AR-HMM iterated resampling the conditional random variables

[00157] $x \mid \pi, \theta, y \quad \theta | x, y \quad$ and $\quad \beta, \pi | x$

[00158] For simplicity, throughout this section notation for conditioning on hyperparameters and the superscript notation for multiple observation sequences is suppressed.

Sampling $x \mid \pi, \theta, y$

[00159] Sampling the state labels x given the dynamical parameters, π and θ, and the data y corresponds to segmenting the 3D video sequence and assigning each segment to a behavioral mode that describes its statistics.

[00160] Given the observation parameters θ and the transition parameters π, the hidden state sequence x is Markov with respect to a chain graph. The standard HMM backward message passing recursions are

[00161]
$$B_t(k) = p(y_{t+1:T} | \theta, \pi, x_t = k)$$
$$= \sum_{j=1}^{K} p(x_{t+1} = j \,| x_t = k, \pi)\, p(y_{t+1} | x_{t+1} = j, \theta)\, B_{t+1}(j)$$

[00162] for $t = 1,2, \ldots, T-1$ and $k = 1,2, \ldots, K$, where $B_T(k) = 1$ and where $y_{t+1:T} = (y_{t+1}, y_{t+2}, \ldots, y_T)$. Using these messages, the conditional distribution of the first state x_1, marginalizing over all the future states $x_{2:T}$ is

[00163] $$p(x_1 = k \,| \pi, \theta, y) \propto p(x_1 = k | \pi) p(y_1 | x_1 = k, \theta) B_1(k)$$

[00164] which can be sampled efficiently. Given a sampled value $\bar{z}_1$, the conditional distribution of the second state x_2 is

[00165] $p(x_2 = k \,|\pi, \theta, y, x_1 = \bar{z}) \propto p(x_2 = k \,|x_1 = \bar{z_1}, \pi)\, p(y_2|\, x_2 = k, \theta) B_2(k).$

[00166] Therefore after passing HMM messages backward the state sequence can be recursively sampled forwards.

Sampling $\theta|\, x, y$

[00167] Sampling the autoregressive parameters θ given the state sequence x and the data sequence y corresponds to updating each mode's dynamical parameters to describe the 3D video data segments assigned to it.

[00168] To resample the observation parameters θ conditioned on a fixed sample of the state sequence x and the observations y one can exploit conjugacy between the autoregressive likelihood and the MNIW prior. That is, the conditional also follows the MNIW distribution:

[00169] $p\big(A^{(k)}, \Sigma^{(k)}\big|x, y, S_0, \nu_0, M_0, K_0\big) = p\big(A^{(k)}, \Sigma^{(k)}\big|S_n, \nu_n, M_n, K_n\big)$

[00170] where (S_n, ν_n, M_n, K_n) are posterior hyperparameters that are functions of the elements of y assigned to state k as well as the preceding lagged observations:

[00171]
$$S_n = S_0 + S_{yy^\top} + (M_0 K_0^{-1} M_0^\top - M_n K_n^{-1} M_N^\top)$$
$$M_n = \big(M_0 K_0^{-1} + S_{\tilde{y}\tilde{y}^\top}\big) K_n$$
$$K_n = \big(K_0^{-1} + S_{\tilde{y}\tilde{y}^\top}\big)^{-1}$$
$$\nu_n = \nu_0 + n$$

[00172] where

[00173]
$$S_{yy^\top} = \sum_{t:x_t=k} y_t\, y_t^\top \quad S_{\tilde{y}\tilde{y}^\top} = \sum_{t:x_t=k} \tilde{y}_t\, \tilde{y}_t^\top$$
$$S_{y\tilde{y}^\top} = \sum_{t:x_t=k} y_t\, \tilde{y}_t^\top \quad n = \#\{t: x_t = k\}.$$

[00174] Therefore resampling $\theta|\, x, y$ includes three steps: collecting statistics from the data assigned to each state, forming each state's posterior hyperparameters, and updating each state's observation parameter by simulating a draw from the appropriate MNIW. Simulating $(A, \Sigma) \sim \mathrm{MNIW}(S_n, \nu_n, M_n, K_n)$ proceeds as

[00175]
$$\Sigma \sim \text{InvWishart}(S_n, \nu_n)$$
$$A = M_n + \Sigma^{\frac{1}{2}} G \, \mathrm{K_n}^{-\frac{1}{2}} \quad \text{where} \quad G_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0,1).$$

Sampling $\beta, \pi | \, x$

[00176] Sampling the transition parameters π and β given the state sequence x corresponds to updating the probabilities of transitions among behavioral modules to reflect the transition patterns observed in the state sequence. Updating β encouraged redundant behavioral modes to be pruned from the model, while updating each π_{ij} fit the transitions observed from state i to state j.

[00177] Resampling the transition parameters β and π, which are draws from the weak limit approximation to the (sticky) HDP, was performed using an auxiliary variable sampling scheme. That is, $\beta,\backslash pi \mid x$ was generated by first sampling auxiliary variables $m \mid \beta, x$. Then $\beta,\backslash pi \mid x, m$ was generated by first sampling from the marginal $\beta | \, m$ and then the conditional $\pi | \, \beta, x$.

[00178] The matrix of transition counts in the sampled state sequence x is

[00179] $$n_{kj} = \#\{t : x_t = k, \; x_{t+1} = j, \; t = 1,2, \ldots, T-1\}.$$

[00180] Suppressing conditioning notation for simplicity, the auxiliary variables $m = \{m_{kj} : k, j = 1,2, \ldots, K\}$ are sampled via

[00181] $$m_{kj} = \sum_{l=1}^{n_{kj}} b_{kjl} \quad \text{where} \quad b_{kjl} \overset{\text{iid}}{\sim} \text{Bernoulli}\left(\frac{\alpha\beta_j}{\alpha\beta_j + \kappa} \frac{\alpha\beta_j + \kappa\delta_{kj}}{\alpha\beta_j + l + \kappa\delta_{kj}}\right)$$

[00182] where Bernoulli(p) denotes a Bernoulli random variable that takes value 1 with probability p and takes value 0 otherwise. Note that the update for the HDP-HMM without a sticky bias corresponds to setting $\kappa = 0$ in these updates.

[00183] Given the auxiliary variables, the update to β is a Dirichlet-multinomial conjugate one, where

[00184] $$\beta | \, m \sim \text{Dir}(\tfrac{\gamma}{K} + m_{\cdot 1}, \tfrac{\gamma}{K} + m_{\cdot 2}, \cdots, \gamma/K \; + m_{\cdot K})$$

[00185] where $m_{\cdot j} = \sum_{k=1}^{K} m_{kj}$ for $j = 1,2, \ldots, K$. The update to $\pi | \, \beta, x$ is similar, with

[00186] $\pi_k | \beta, x \sim \mathrm{Dir}(\alpha\beta_1 + n_{k1}, \cdots, \alpha\beta_j + n_{kj} + \kappa\delta_{kj}, \cdots, \backslash alpha\ \beta_K + n_{kK})$.

Application of the Models to the Examples

[00187] Datasets from the open-field, odor, and genetic manipulation experiments were modeled jointly to increase statistical power. Because the neural implants associated with the optogenetics experiment modestly altered the profile of the animal, these data were modeled separately. In all experiments, the first 10 principal components for each frame of each imaged mouse were gathered. Data were then subdivided and assigned either a "train" or a "test" label, in a 3:1 train:test ratio. The mice labeled "test" were held-out from the training process, and used to test generalization performance via measurement held-out likelihood. This approach allowed us to directly compare algorithms whose composition reflected different underlying structures for behavior.

[00188] We trained models on data using the procedures described herein; modeling was robust to both initialization settings and to parameter and hyperparameter settings (with the exception of kappa, see below). Specifically, the number of lags used in our AR observation distributions and the number of used states in our transition matrix with an HDP prior was found to be robust to the particular hyperparameter settings on both priors. We varied the hyperparameters of our sparsifying ARD prior by several orders of magnitude, and held-out likelihood, the number of used lags, and the number of used states varied negligibly. We also varied the hyperparameters of our HDP prior by several orders of magnitude and again observed no change to the number of used states or held-out likelihood. All jointly-trained data shared observation distributions, but each treatment class was allowed its own transition matrix. Each model was updated through 1000 iterations of Gibbs sampling; upon the last iteration of Gibbs sampling the model output was saved; all further analysis was performed on this final update.

[00189] The "stickiness" of the duration distribution of our behavioral modules — defined by the kappa setting of the model — influenced the average duration of behavioral modules discovered by the AR-HMM; this allowed us to control the temporal scale at which behavior was modeled. As discussed in the main text, autocorrelation, power spectral density, and the changepoint algorithm identified switching dynamics at a specific sub-second temporal scale (as encapsulated by the changepoints duration distribution and reflected by the spectrogram and autocorrelogram). We therefore empirically set the kappa stickiness

parameter of the time-series model to best match the duration distribution discovered by changepoint detection. To find the kappa setting at which these distributions were best matched, we minimized the Kolmogorov-Smirnov distance between the inter-changepoint interval distribution and the posterior behavioral module duration distribution through a dense grid search.

Mouse Strains, Housing and Habituation

[00190] Unless otherwise noted, all experiments were performed on 6-8 week old C57/BL6 males (Jackson Laboratories). Mice from the rorβ and rbp4 strains were habituated and tested identically to the reference C57/BL6 mice. Mice were brought into our colony at 4 weeks of age, where they were group-housed for two weeks in a reverse 12 hours light/12 hours dark cycle. On the day of testing, mice were brought into the laboratory in a light-tight container, where they were habituated in darkness for 30 minutes before testing.

Example 1. Behavioral assays: Innate Exploration

[00191] To address these possibilities, we first used the AR-HMM to define the baseline architecture of mouse exploratory behavior in the open field, and then asked how this template for behavior was modified through distinct manipulations of the external world.

[00192] For the open field assay (OFA), mice were habituated as noted above, and then placed in the middle of a circular 18" diameter enclosure with 15"-high walls (US Plastics), immediately after which 3D video recording was begun. The animal was allowed to freely explore the enclosure for the 30 minute experimental period. Mice whose behavior was assessed in a square box were handled and measured identically to the OFA, except in the odor box described below.

[00193] The AR-HMM identified ~60 reliably-used behavioral modules (51 modules explained 95 percent of imaging frames, and 65 modules explained 99 percent of imaging frames, Figs. 5A, 5B) from the circular open field dataset, which is representative of normal mouse exploratory behavior in the laboratory (Fig. 6A, n=25 animals, 20 minute trials). FIG. 5A shows the proportion of frames explained by each module (Y axis), plotted against the set of modules, sorted by usage (X axis). Ninety-five percent of frames were explained by 51 behavioral modules; ninety-nine percent of frames were explained by 62 behavioral modules in the open field dataset.

[00194] FIG. 5B shows modules (X axis) sorted by usage (Y axis) with Bayesian credible intervals indicated. Note that all the credible intervals are smaller than the SEs computed based upon the bootstrap estimates (Fig. 5B). As noted above, many of these modules encode human-describable components of behavior (e.g. rears, walking, pauses, turns).

[00195] The AR-HMM also measures the probability that any given module precedes or follows any other module; in other words, after model training each module is assigned a pairwise transition probability with every other module in the set; these probabilities summarize the sequences of modules that were expressed by the mouse during behavior. Plotting these transition probabilities as a matrix revealed that they were highly non-uniform, with each module preferentially connected in time to some modules and not others (Fig. 6B; average node degree without thresholding 16.82 ± .95, after thresholding bigram probabilities lower than 5 percent, 4.08 ± .10). This specific connectivity between pairs of modules restricted the module sequences that were observed in the dataset (8900/~125,000 possible trigrams) demonstrating that certain module sequences were favored; this observation suggests that mouse behavior is predictable, as knowing what the mouse is doing at any given moment in time informs an observer about what the mouse is likely to do next. Information theoretic analysis of the transition matrix confirmed that mouse behavior is significantly predictable, as the average per-frame entropy rate was low relative to a uniform transition matrix (without self-transitions 3.78 ± .03 bits, with self-transitions .72 ± .01 bits, entropy rate in uniform matrix 6.022 bits), and the average mutual information between interconnected modules was significantly above zero (without self-transitions 1.92 ± .02 bits, with self-transitions 4.84 bits ± .03 bits). This deterministic quality to behavior likely serves to ensure that the mouse emits coherent patterns of motion; consistent with this possibility, upon inspection frequently-observed module sequences were found to encode different aspects of exploratory behavior.

[00196] The behavior expressed by mice in the circular open field reflects a context-specific pattern of locomotor exploration. We hypothesized that mice would adapt to changes in apparatus shape by focally altering the structure of behavior to generate new pose dynamics to interact with specific physical features of the environment; to test this hypothesis, we imaged mice within a smaller square box and then co-trained our model with both the circular open field data and square data, thereby enabling direct comparisons of

modules and transition under both conditions (n = 25 mice in each condition). Although mice tended to explore the corners of the square box and the walls of the circular open field, the overall usage of most modules was similar between these apparatuses, consistent with exploratory behavior sharing many common features across arenas (Fig. 6C). The AR-HMM also identified a small number of behavioral modules that were deployed extensively in one context, but negligibly or not at all in the other, consistent with the idea that different physical environments drive expression of new behavioral modules (Fig. 6C, for all usage differences discussed below $p<10^{-3}$ based upon bootstrap estimation).

[00197] Interestingly, these "new" modules are not only deployed during physical interactions with specific features of the apparatus — which would be predicted to elicit new pose dynamics — but also during unconstrained periods of exploration. For example, one circular arena-specific module encoded a thigmotactic behavior in which the mouse locomotes near the arena wall with a body posture that matches the curvature of the wall. This module was also expressed when the mouse is closer to the center of the circular arena and not in physical contact with the wall, demonstrating that expression of this module is not simply the direct consequence of physical interactions with the wall but rather reflects the behavioral state of the mouse in a curved arena; while thigmotaxis also occurred in the square box, the associated behavioral module encodes locomotion with a straight body and was used during straight trajectories in both square and circular apparatuses (Figs. 6D-E, middle panels). Similarly, within the square box mice expressed a context-specific module that encodes a dart from the center of the square to one of the adjacent corners; this pattern of motion likely was a consequence of the square having a small central open field, and was not the specific product of a physical constraint placed upon the mouse.

[00198] A number of additional modules were found to be preferentially expressed in one context or the other; these upregulated modules appeared to encode behaviors that were deployed in allocentric patterns specified by the shape of the arena. In the circular arena, for example the mouse preferentially expressed a rear in which the mouse's body points outwards while it pauses near the center of the open field, while in the smaller square box mice preferentially executed a high rear in the corners of the box (Fig. 6E, data not shown). These results suggest that what the mouse does (i.e. its egocentric behavior) is modulated based upon where in space the mouse is (i.e. its allocentric position). Taken together, these data demonstrate that mice adapt to new physical environments, at least in part, through

recruitment of a limited set of context-specific behavioral modules (that encode context-appropriate pose dynamics) into baseline patterns of action; these new modules — along with other modules whose expression is enriched in one context or the other — are differentially deployed in space to respond to changes in the environment.

Example 2. Behavioral assays: Stimulus-Driven Innate Behaviors – Response to Odorants

[00199] Because mice express the same underlying behavioral state — locomotor exploration — in both the circle and the square one might predict that the observed changes to behavioral modules in this case would be focal and limited in extent. We therefore asked how the underlying structure of behavior is altered when mice are exposed to a sensory cue — within an otherwise-constant physical environment — that drives a global change in behavioral state that includes the expression of new and motivated actions.

[00200] To assess innate behavioral responses to volatile odorants, we developed an odor delivery system that spatially isolates odors in specific quadrants of a square box. Each 12" x 12" box was constructed of ¼" black matte acrylic (Altech Plastics), with 3/4" holes patterning the bottom of the box in a cross formation, and a 1/16" thick glass cover (Tru Vue). These holes were tapped and connected via PTFE tubing to a vacuum manifold (Sigma Aldrich) that provides negative pressure to isolate odors within quadrants. Odor was injected into the box through ½" NPT-⅜" pipe-fittings (Cole-Parmer). Filtered air (1.0L/min) was blown over odorant-soaked blotting paper (VWR) placed at the bottom of Vacutainer syringe vials (Covidien). The odorized airstream was then passed through corrugated PTFE tubing (Zeus) into one of the four pipe-fittings in a corner of the odor box.

[00201] We verified the ability of the odor box to isolate odors within specified quadrants by visualizing vaporized odor or smoke through sheet illumination of the box with a low-power handheld HeNe laser. This approach allowed us to tune the vacuum flow and odor flow rates to achieve odor isolation, which was verified using a photoionization device (Aurora Scientific). To eliminate the possibility of cross contamination between experiments, the odor boxes were soaked in a 1% Alconox solution overnight, then thoroughly cleaned with a 70% ethanol solution. Mice were habituated to the experimental room for 30 minutes before the initiation of the experiment. Under control conditions, dipropylene glycol with air (1.0 L/min) was delivered to each of the four corners of the

apparatus before a single mouse was placed in the center of the box and allowed to freely explore while 3D video records were acquired for 20 minutes. The same cohort of animals was tested for odor responses by subsequently repeating the experiment with odorized air delivered to one of the four quadrants. All 3D video recordings are performed in total darkness. TMT was obtained from Pherotech, and used at 5% concentration.

[00202] Mice exploring the square box were therefore exposed to the aversive fox odor trimethylthiazoline (TMT), which was delivered to one quadrant of the box via olfactometer. This odorant initiates a complex and profound behavioral state change including odor investigation, and escape and freezing behaviors that are accompanied by increases in corticosteroid and endogenous opioid levels. Consistent with these known effects, mice sniffed the odor-containing quadrant, and then avoided the quadrant containing the predator cue and displayed prolonged periods of immobility traditionally described as freezing behavior (FIG. 7). FIG. 7 shows a histogram depicting the average velocity of the modules that were differentially upregulated and interconnected after TMT exposure "freezing" compared to all other modules in the dataset.

[00203] Surprisingly, this suite of new behaviors was encoded by the same set of behavioral modules that were expressed during normal exploration; several modules were up- or down-regulated after TMT exposure, but no new modules were introduced or eliminated relative to control (n=25 animals in control conditions, n = 15 in TMT, model was co-trained on both datasets simultaneously). Instead, TMT altered the usage of and transition probabilities between specific modules, leading to newly-favored behavioral sequences that encode TMT-regulated behaviors (for all usage and transition differences discussed below $p<10^{-3}$ based upon bootstrap estimation).

[00204] Plotting the module transitions altered after exposure to TMT defined two neighborhoods within the behavioral statemap; the first included an expansive set of modules and interconnections that were modestly downregulated by TMT, and the second included a focused set of modules and transitions that were upregulated by TMT). During normal behavior these newly-interconnected modules were temporally dispersed and individually appear to encode for different morphological forms of pausing or balling up. In contrast, under the influence of TMT these modules were concatenated into new sequences that, upon inspection and quantification, were found to encode freezing behavior (average during-sequence velocity -.14 ± .54 mm/sec, for other modules 34.7 ± 53 mm/sec). For example, the

most commonly-expressed freezing trigram was expressed 716 times after TMT exposure (in 300 minutes of imaging), as opposed to just 17 times under control conditions (in 480 minutes of imaging). The TMT-induced neighborhood structure imposed upon these pausing modules to create freezing demonstrates that behavior can be altered through focal changes in transition probabilities. This local rewriting of transition probabilities was accompanied by an increase in the overall determinism of mouse behavior — its global pattern of action became more predictable as a consequence of TMT exposure (per frame entropy rate fell from 3.92 ± .02 bits to 3.66 ± .08 bits without self-transitions, and from .82 ± .01 bits to .64 ± .02 bits with self-transitions) — consistent with the mouse enacting an deterministic avoidance strategy.

[00205] Proximity to the odor source also governed the pattern of expression of specific behavioral modules (Figs. 8D-8E). For example, a set of freezing-related modules tended to be expressed in the quadrant most distal from the odor source, while the expression of an investigatory rearing module (whose overall usage was not altered by TMT) was specifically enriched within the odor quadrant (Figs. 8D-8E). Together, these findings suggest two additional mechanisms through which the mouse nervous system can generate new and adaptive behaviors. First, the transition structure between individual modules that are otherwise normally associated with a different behavioral state, such as locomotor exploration, can be altered to generate new behaviors such as freezing. Second, the spatial patterns of deployment of pre-existing modules and sequences can be regulated to support motivated behaviors such as odor investigation and avoidance. Behavioral modules are not, therefore, simply reused over time, but instead act as flexibly interlinked components of behavioral sequences whose expression is dynamically regulated both in time and space.

Example 3. The effect of genes and neural circuits on modules

[00206] As described above, the fine-timescale structure of behavior is selectively vulnerable to changes in the physical or sensory environment that influence action over timescales of minutes. Furthermore, the AR-HMM appears to comprehensively encapsulate the pattern of behavior expressed by the mouse (within the limits of our imaging). These observations suggest that the AR-HMM — which affords a systematic window into mouse behavior at the sub-second timescale— may be able to both quantify obvious behavioral phenotypes and to reveal new or subtle phenotypes induced after experimental manipulations that influence behavior across a range of spatiotemporal scales.

[00207] To explore how changes in individual genes — which act on timescales of the lifetime of the mouse — might impact fast behavioral modules and transitions, we characterized the phenotype of mice mutant for the retinoid-related orphan receptor 1β (Ror1β) gene, which is expressed in neurons in the brain and spinal cord; we selected this mouse for analysis because homozygous mutant animals exhibit abnormal gait[37-40], which we would expect to be detected by the AR-HMM. After imaging and modeling, littermate control mice were found to be nearly indistinguishable from fully inbred C57/Bl6 mice, whereas mutant mice expressed a unique behavioral module that encoded a waddling gait (Fig. 9A, 9C). This alteration in behavior was accompanied by its converse: the expression of five behavioral modules encoding normal forward locomotion at different speeds in wild-type and C57 mice was downregulated in the Ror1β mutant (Fig. 9A, average during-module velocity = 114.6 ± 76.3 mm/sec). In addition, expression of a set of four modules that encoded brief pauses and headbobs were also upregulated (Fig. 9A, average during-module velocity = 8.8 ± 25.3 mm/sec); this pausing phenotype had not previously been reported in the literature. Interestingly, heterozygous mice – which have no reported phenotype[37-40], appear normal by eye, and exhibit wild-type running wheel behavior[40] — also were found to express a fully-penetrant mutant phenotype: they overexpressed the same set of pausing modules that were upregulated in the full Ror1β mutants, while failing to express the more dramatic waddling phenotype (Fig. 9A).

[00208] The AR-HMM therefore describes the pathological behavior of Ror1β mice as the combination of a single neomorphic waddling module and the increased expression of a small group of physiological modules encoding pausing behaviors; heterozygote mice express a defined subset of these behavioral abnormalities, whose penetrance is not intermediate but equals that observed in the mutant. These results suggest that the sensitivity of the AR-HMM enables fractionation of severe and subtle behavioral abnormalities within the same litter of animals, enables discovery of new phenotypes, and facilitates comparisons amongst genotypes. These experiments also demonstrate that genotype-dependent variations in behavior, the consequence of the indelible and lifetime alteration of a specific gene in the genome, can influence module expression and transition statistics that operate on timescales of milliseconds.

Example 4. Behavioral Assays: Optogenetics – Effect of Neural Activity on Modules

[00209] Finally, we wished to ask whether the behavioral structure captured by the AR-HMM would offer insight into fleeting or unreliable changes in behavior. We therefore briefly triggered neural activity in motor circuits, and asked how stimulation at different levels of intensity influenced the moment-to-moment organization of behavior. We unilaterally expressed the light-gated ion channel Channelrhodopsin-2 in corticostriatal neurons[41,42] and assessed behavioral responses before, during and after two seconds of light-mediated activation of motor cortex (n= 4 mice, model was trained separately from previous experiments).

[00210] Four adult male *Rbp4*-Cre (The Jackson Laboratory) mice were anesthetized with 1.5% isoflurane and placed in a stereotaxic frame (Leica). Microinjection pipettes (O.D. 10-15 μm) were inserted into the left motor cortex (coordinates from Bregma: 0.5 AP, -1 ML, 0.60 DV). 0.5 μl of AAV5.EF1a.DIO.hChR2(H134R)-eYFP.WPRE.hGH (~10^{12} infectious units/mL, Penn Vector Core) were injected in each mouse over 10 minutes followed by an additional 10 minutes to allow diffusion of viral particles away from the injection site. After the injection, a bare optic fiber with a zirconia ferrule (O.D. 200 μm, 0.37 numerical aperture) was inserted 100 μm above the injection site and secured to the skull with acrylic cement (Lang). Twenty-eight days following the viral injection, mice were placed in a circular arena and the optical implant was coupled to a laser pump (488 nm, CrystaLaser) via a patch-chord and a rotary joint (Doric Lenses). The laser was directly controlled from a PC. After 20 minutes of familiarization to the arena, the optostimulation was started. The laser power, the pulse width, the inter-pulse interval and the inter-train interval were controlled by custom-made software (NI Labview). Each train of laser pulses consisted of 30 pulses (pulse width: 50 ms) at 15 Hz. The interval between successive trains was set to 18 seconds. 50 trains were delivered for each laser intensity. The animal was progressively exposed to higher laser intensities over the course of the experiment.

[00211] At the lowest power levels no light-induced changes in behavior were observed, while at the highest power levels the AR-HMM identified two behavioral modules whose expression was reliably induced by the light (Fig. 10A). Neither of these modules were expressed during normal mouse locomotion; inspection revealed them to encode two forms of spinning behavior (differing in their length and the angle of turn), in which the mouse traces out semi-circles or donuts in space (Fig. 10B). The induction of neomorphic behaviors after

strong unilateral motor cortex stimulation is not surprising, although it is important to note that the AR-HMM both recognized these behaviors as new and encapsulated them as two unique behavioral modules. However, we noted that approximately 40 percent of the time, the overall pattern of behavior did not return to baseline for several seconds after light offset. This deviation from baseline was not due to continued expression of the modules triggered at light onset; instead, mice often expressed a pausing module (average during-module velocity = .8 ± 7 mm/sec) at light offset as if "resetting" after a non-volitional movement.

[00212] The behavioral changes induced by high intensity optogenetic stimulation were reliable, as on essentially every trial the animal emitted one of the two spinning modules. We then asked whether the sensitivity of the AR-HMM would enable quantitative analysis of more subtle changes in behavior, as occurs in intermediate regimes of motor cortex stimulation that elicit unreliable emission of specific behavioral modules. We therefore titrated the levels of light stimulation down until one of the two neomorphic behavioral modules was no longer detected, and the other was expressed on only 25 percent of trials. Surprisingly, we were then could detect the upregulation of a second set of behavioral modules, each of which was expressed about 25 percent of the time (Fig. 10A). These modules were not neomorphic, but rather were normally expressed during physiological exploration, and encoded a turn and a head-bobbing behavior (data not shown). While each of these individual light-regulated modules was emitted unreliably, taken in aggregate the behavioral changes across all modules suggested that lower-level neural activation reliably influenced behavior, but largely through inducing physiological rather than neomorphic actions (Fig. 10A). Taken together, the detection of both stimulus-locked induction of behavioral modules and the lingering effects of stimulation of module usage demonstrates that neurally-induced changes in behavior can influence the sub-second structure of behavior. Furthermore, the identification of a physiologically-expressed set of light-regulated behavioral modules — whose induction would not have been apparent under strong stimulation conditions — also suggests that the AR-HMM can reveal subtle relationships between neural circuits and the time-series structure of behavior.

Example 5: Reducing Dimensionality – Probabilistic Graphical Models and Variational Autoencoders

[00213] As illustrated in FIG. 3, after correction of the orientation of the images, methods may be utilized to reduce the dimensionality of the data. For instance, each image

may be a 900 dimensional vector, and therefore reducing dimensionality is quite important for model analysis. In some embodiments, both that include model-free algorithms 320 or the model fitting 315 algorithm, the information captured in each pixel often is either highly correlated (neighboring pixels) or uninformative (pixels on the border of the image that never represent the mouse's body). To both reduce redundant dimensions and make modeling computationally tractable, various techniques may be employed to dimensionally reduce each image.

[00214] In some examples, the output of the orientation corrected images in some embodiments will be to a principle component analysis time series 310 or other statistical methods for reducing data points. However, PCA reduces the dimensionality to a linear space. The inventors have discovered that reduction of the dimensionality to a linear space may not accommodate for various changes in the mice that are not related to behavior. This includes changes in mouse size, mouse breed, etc.

[00215] Accordingly, the inventors have discovered that using certain kinds of neural networks, such a multi-layer perceptrons, one can effectively reduce the dimensionality of the images. Furthermore, these dimensionality reduced images provide an effective method to develop models that are agnostic to the size of the mouse, or other animals, and can account for other changes that are not behavior related. For instance, some neural networks that reduce the dimensionality to a ten dimensional image manifold may be utilized.

[00216] The inventors developed a new framework for unsupervised learning that composes probabilistic graphical models with deep learning methods and combines their respective strengths to reduce dimensionality. Their method uses graphical models to express structured probability distributions and recent advances from deep learning to learn flexible feature models and bottom-up recognition networks. All components of these models are learned simultaneously using a single objective, and accordingly the inventors developed scalable fitting algorithms that can leverage natural gradient stochastic variational inference, graphical model message passing, and backpropagation with the reparameterization trick.

[00217] Unsupervised probabilistic modeling often has two goals: first, to learn a model that is flexible enough to represent complex high-dimensional data, such as images or speech recordings, and second, to learn model structure that is interpretable, admits meaningful priors, and generalizes to new tasks. That is, it is often not enough just to learn

the probability density of the data: one also wants to learn a meaningful representation. Probabilistic graphical models (Koller & Friedman, 2009; Murphy, 2012) provide many tools to build such structured representations, but can be limited in their capacity and may require significant feature engineering before being applied to data. Alternatively, advances in deep learning have yielded not only flexible, scalable generative models for complex data like images but also new techniques for automatic feature learning and bottom-up inference (Kingma & Welling, 2014; Rezende etal., 2014).

[00218] Consider the problem of learning an unsupervised generative model for a depth video of a tracked freely-behaving mouse, illustrated in FIG. 23. Learning interpretable representations for such data, and studying how those representations change as the animal's genetics are edited or its brain chemistry altered, can create powerful behavioral phenotyping tools for neuroscience and for high-throughput drug discovery (Wiltschko et al., 2015.) Each frame of the video is a depth image of a mouse in a particular pose, and so even though each image is encoded as 30 x 30 = 900 pixels, the data lie near a low-dimensional nonlinear manifold. A good generative model must not only learn this manifold but also represent many other salient aspects of the data.

[00219] For example, from one frame to the next, the corresponding manifold points should be close to one another, and in fact the trajectory along the manifold may follow very structured dynamics. To inform the structure of these dynamics, a natural class of hypotheses used in ethology and neurobiology (Wiltschko et al., 2015) is that the mouse's behavior is composed of short, reused actions, such as darts, rears, and grooming bouts. Therefore a natural representation would include discrete states with each state capturing the simple dynamics of a particular primitive action, a representation that would be difficult to encode in an unsupervised recurrent neural network model.

[00220] These two tasks, of learning the image manifold and learning a structured dynamics model, are complementary: we want to learn the image manifold not just as a set but in terms of manifold coordinates in which the structured dynamics model fits the data well. A similar modeling challenge arises in speech (Hinton et al., 2012), where high-dimensional data lie near a low-dimensional manifold because they are generated by a physical system with relatively few degrees of freedom (Deng, 1999), but also include the discrete latent dynamical structure of phonemes, words, and grammar (Deng, 2004).

[00221] To address these challenges, the inventors have developed graphical models for representing structured probability distributions and used ideas from variational autoencoders (Kingma & Welling, 2014) for learning not only the nonlinear feature manifold but also bottom-up recognition networks to improve inference. Thus the method enables the combination of flexible deep learning feature models with structured Bayesian priors, including nonparametric models.

[00222] This approach yields a single variational inference objective in which all components of the model are learned simultaneously. Furthermore, we develop a scalable fitting algorithm that combines several advances in efficient inference, including stochastic variational inference (Hoffman et ah, 2013), graphical model message passing (Roller & Friedman, 2009), and backpropagation with the reparameterization trick (Kingma & Welling, 2014). Thus our algorithm can leverage conjugate exponential family structure where it exists to efficiently compute natural gradients with respect to some variational parameters, enabling effective second-order optimization (Martens, 2015), while using backpropagation to compute gradients with respect to all other parameters. The general approach may be referred to as the structured variational autoencoder (SVAE). The SVAE is illustrated herein using graphical models based on switching linear dynamical systems (SLDS) (Murphy, 2012; Fox et al., 2011).

Natural gradient stochastic variational inference

[00223] Stochastic variational inference (SVI) (Hoffman et al., 2013) applies stochastic gradient ascent to a mean field variational inference objective in a way that exploits exponential family conjugacy to efficiently compute natural gradients (Amari, 1998; Martens, 2015). Consider a model composed of global latent variables , local latent variables where θ, local latent variables

$$x = \{x_n\}_{n=1}^N, \text{ and observed data } y = \{y_n\}_{n=1}^N,$$

$$p(\theta, x, y) = p(\theta) \prod_{n=1}^{N} p(x_n|\theta) p(y_n|x_n, \theta), \qquad (1)$$

where ρ(θ) is the natural exponential family conjugate prior to the exponential family $p(x_n, y_n|\theta)$,

$$\ln p(\theta) = \langle \eta_\theta, t_\theta(\theta) \rangle - \ln Z_\theta(\eta_\theta) \quad (2)$$

$$\ln p(x_n, y_n|\theta) = \langle \eta_{xy}(\theta), t_{xy}(x_n, y_n) \rangle - \ln Z_{xy}(\eta_{xy}(\theta)) = \langle t_\theta(\theta), (t_{xy}(x_n, y_n), 1) \rangle. \quad (3)$$

Consider the mean field family $q(\theta)q(x) = q(\theta)\Pi_n q(x_n)$. Because of the conjugacy structure, the optimal global mean field factor $q(\theta)$ is in the same family as the prior $\rho(\theta)$,

$$\ln q(\theta) = \langle \tilde{\eta}_\theta, t_\theta(\theta) \rangle - \ln Z_\theta(\tilde{\eta}_\theta). \quad (4)$$

The mean field objective on the global variational parameters, optimizing out the local variational factors $q(x)$, can then be written

$$\mathcal{L}(\tilde{\eta}_\theta) \triangleq \max_{q(x)} \mathbb{E}_{q(\theta)q(x)} \left[\ln \frac{p(\theta, x, y)}{q(\theta)q(x)} \right] \leq \ln p(y) \quad (5)$$

and the natural gradient of the objective (5) decomposes into a sum of local expected sufficient statistics (IHoffman et al., 2013):

$$\tilde{\nabla}_{\tilde{\eta}_\theta} \mathcal{L}(\tilde{\eta}_\theta) = \eta_\theta + \sum_{n=1}^{N} \mathbb{E}_{q^*(x_n)}(t_{xy}(x_n, y_n), 1) - \tilde{\eta}_\theta, \quad (6)$$

where $q^*(x_n)$ is a locally optimal local mean field factor. Thus we can compute a stochastic natural gradient update for our global mean field objective by sampling a data point y_n, optimizing the local mean field factor $q(x_n)$, and computing scaled expected sufficient statistics.

2.2. Variational autoencoders

[00224] The variational autoencoder (VAE) (Kingma & Welling, 2014; Rezende et al., 2014) is a recently proposed model and inference method that links neural network autoencoders (Vincent et al., 2008) with mean field variational Bayes. Given a high-dimensional dataset such as a collection of images, the VAE models each observation yn in terms of a low-dimensional latent variable y_n and a nonlinear observation model with parameters:

$$x_n \overset{\text{iid}}{\sim} \mathcal{N}(0, I), \quad n = 1, 2, \ldots, N \tag{7}$$

$$y_n \,|\, x_n, \vartheta \sim \mathcal{N}(\mu(x_n; \vartheta), \Sigma(x_n; \vartheta)) \tag{8}$$

where

$$h_\ell(x_n) = f(W_\ell h_{\ell-1}(x_n) + b_\ell), \; \ell = 1, 2, \ldots, L, \tag{9}$$

$$\mu(x_n; \vartheta) = W_\mu h_L(x_n) + b_\mu, \tag{10}$$

$$\Sigma(x_n; \vartheta) = \operatorname{diag}(\exp(W_{\sigma^2} h_L(x_n) + b_{\sigma^2})), \tag{11}$$

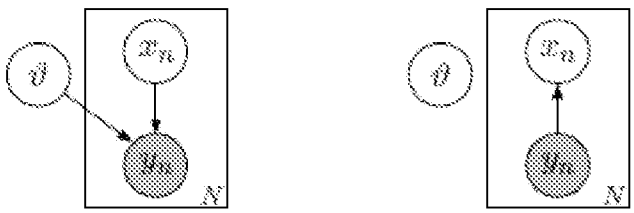
$$\vartheta = (\{(W_\ell, b_\ell)\}_{\ell=1}^{L}, (W_\mu, b_\mu), (W_{\sigma^2}, b_{\sigma^2})), \tag{12}$$

Because we will reuse this particular MLP construction, we introduce the notation

$$(\mu(x_n; \vartheta), \Sigma(x_n; \vartheta)) = \mathrm{MLP}(x_n; \vartheta). \tag{13}$$

To approximate the posterior, the variational autoencoder uses the mean field family:

$$q(\vartheta) q(x \,|\, y) = q(\vartheta) \prod_{n=1}^{N} q(x_n \,|\, y_n). \tag{14}$$



(a) VAE generative model. (b) VAE variational family.

Figure 2. Graphical models for the variational autoencoder.

[00225] A key insight of the variational autoencoder is to use a conditional variational density *q(xn | yn),* where the parameters of the variational distributiondepend on the corresponding data point. In particular, we can take the mean and covariance parameters of *q(xn \ yn)* to be *n(yn\(p)* and E(yrl; </>), respectively, where

$$(\mu(y_n; \phi), \Sigma(y_n; \phi)) = \mathrm{MLP}(y_n; \phi) \tag{15}$$

denotes a set of MLP parameters. Thus the variational distribution *q(xn | ijn)* acts like a stochastic *encoder* from an observation to a distribution over latent variables, while the forward model *p(yn | xn)* acts as a stochastic *decoder* from a latent variable value to a distribution over observations.

[00226] The resulting mean field objective expresses a variational Bayesian version of an autoencoder. The variational parameters are then the encoder parameters and the decoder parameters, and the objective is

$$\mathcal{L}(\tilde{\eta}_\theta, \phi) \triangleq \mathbb{E}_{q(\vartheta)q(x \mid y)}\left[\ln \frac{p(\vartheta)}{q(\vartheta)} + \sum_{n=1}^{N} \ln \frac{p(x_n, y_n \mid \vartheta)}{q(x_n \mid y_n)}\right].$$

[00227] To optimize this objective efficiently, Kingma & Welling (2014) applies a reparameterization trick. To simplify notation and computation, First, we rewrite the objective as

$$\mathcal{L}(\vartheta^*, \phi) = \mathbb{E}_{q(x \mid y)} \ln p(y \mid x, \vartheta^*) - \mathrm{KL}(q(x \mid y) \,\|\, p(x)).$$

The term *KL(q(x \ y) \\p(x))* is the KL divergence between two Gaussians and its gradient with respect to f can be computed in closed form. To compute stochastic gradients of the expectation term, since a random variable can be parameterized as

$$x_n = g(\phi, \epsilon) \triangleq \mu_q(y_n; \phi) + \Sigma_q(y_n; \phi)^{\frac{1}{2}}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I),$$

The expectation term can be rewritten in terms of its gradient approximated via Monte Carlo over,

$$\nabla_{\vartheta^*,\phi}\mathbb{E}_q \ln p(y \mid x, \vartheta^*) \approx \sum_{n=1}^{N} \nabla_{\vartheta^*,\phi} \ln p(y_n \mid g(\phi, \hat{\epsilon}_n), \vartheta^*)$$

These gradient terms can be computed using standard backpropagation. For scalability, the sum over data points is also approximated via Monte Carlo.

Generative model and variational family

[00228] Accordingly, based on these algorithms, the inventors developed an SVAE generative model and corresponding variational family. To be concrete we focus on a particular generative model for time series based on a switching linear dynamical system (SLDS) (Murphy, 2012; Fox et al, 2011), which illustrates how the SVAE can incorporate both discrete and continuous latent variables with rich probabilistic dependence.

[00229] the approach described here applies to a broad set of probabilistic graphical models and is not limited only to time series. First, Section 3.1 describes the generative model, which illustrates the combination of a graphical model expressing latent structure with a flexible neural net to generate observations. Next, Section 3.2 describes the structured variational family, which leverages both graph-structured mean field approximations and flexible recognition networks.

3.1. A switching linear dynamical system with nonlinear observations

[00230] A switching linear dynamical system (SLDS) represents data in terms of continuous latent states that evolve according to a discrete set of linear dynamics. At each time, there is a discrete-valued latent state, which indexes the dynamical mode, and a continuous-valued latent state that evolves according to that mode's linear Gaussian dynamics:

$$x_{t+1} = A^{(z_t)} x_t + B^{(z_t)} u_t, \quad u_t \overset{\text{iid}}{\sim} \mathcal{N}(0, I), \qquad (16)$$

FIG. 24 illustrates graphical models for the SLDS generative model and corresponding structure CRF variational family.

The discrete latent state evolves according to Markov dynamics,

$$z_{t+1} \mid z_t, \pi \sim \pi^{(z_t)} \qquad (17)$$

The initial states are generated separately:

$$z_1 \mid \pi_{\text{init}} \sim \pi_{\text{init}}, \tag{18}$$

$$x_1 \mid z_1, \mu_{\text{init}}, \Sigma_{\text{init}} \sim \mathcal{N}(\mu_{\text{init}}^{(z_1)}, \Sigma_{\text{init}}^{(z_1)}). \tag{19}$$

[00231] Thus inferring the latent variables and parameters of an SLDS identifies a set of reused dynamical modes, each described as a linear dynamical system on latent states, in addition to Markov switching between different linear dynamics. The dynamical parameters may be denoted:

$$\theta = (\pi, \pi_{\text{init}}, \{(A^{(k)}, B^{(k)}, \mu_{\text{init}}^{(k)}, \Sigma_{\text{init}}^{(k)})\}_{k=1}^{K}).$$

At each time, the continuous latent state gives rise to an observation that is conditionally Gaussian

$$y_t \mid x_t, \vartheta \sim \mathcal{N}(\mu(x_t; \vartheta), \Sigma(x_t; \vartheta)). \tag{20}$$

In a typical SLDS (Fox et al., 2011), one can write

$$y_t = Cx_t + Dv_t, \quad v_t \overset{\text{iid}}{\sim} \mathcal{N}(0, I). \tag{21}$$

[00232] However, to enable flexible modeling of images and other complex features, the algorithm can allow the dependence to be a more general nonlinear model. In particular, we consider the following equations for MLPs:

$$(\mu(x_t; \vartheta), \Sigma(x_t; \vartheta)) = \text{MLP}(x_t; \vartheta). \tag{22}$$

[00233] Note that by construction the density is in the exponential family. We can choose the prior *p(0)* to be a natural exponential family conjugate prior, writing

$$\ln p(\theta) = \langle \eta_\theta, t_\theta(\theta) \rangle - \ln Z_\theta(\eta_\theta) \tag{23}$$

$$\begin{aligned} \ln p(z, x|\theta) &= \langle \eta_{zx}(\theta), t_{zx}(z, x) \rangle - \ln Z_{zx}(\eta_{zx}(\theta)) \\ &= \langle t_\theta(\theta), (t_{zx}(z, x), 1) \rangle. \end{aligned} \tag{24}$$

[00234] We can also use a Bayesian nonparametric prior, and generating a discrete state sequence according to a hierarchical Dirichlet process (HDP) HMM (Fox et al., 2011). Though the Bayesian nonparametric case is not discussed further, the algorithms developed here immediately extend to the HDP-HMM using the methods in Johnson & Willsky (2014).

[00235] This construction contains the generative model of the VAE, described above, as a special case. Specifically, the VAE uses the same class of MLP observation models, but each latent value xt is modeled as an independent and identically distributed Gaussian, while the SVAE model proposed here allows a rich joint probabilistic structure. The SLDS generative model also includes as special cases the Gaussian mixture model (GMM), Gaussian-emission discrete-state HMM (G-HMM), and Gaussian linear dynamical system (LDS), and thus the algorithms developed here for the SLDS directly specialize to these models.

[00236] While using conditionally linear dynamics within each state may seem limited, the flexible nonlinear observation distribution greatly extends the capacity of such models. Indeed, recent work on neural word embeddings (Mikolov et al., 2013) as well as neural image models (Radford et al., 2015) has demonstrated learned latent spaces in which linear structure corresponds to meaningful semantics.

[00237] For example, addition and subtraction of word vectors can correspond to semantic relationships between words, and translation in an image model's latent space can correspond to an object's rotation. Therefore, linear models in a learned latent space can yield significant expressiveness while enabling fast probabilistic inference, interpretable priors and parameters, and a host of other tools. In particular, linear dynamics allow one to learn or encode information about timescales and frequencies: the eigenvalue spectrum of each transition matrix $A^{(k)}$ directly represents its characteristic timescales, and so we can control and interpret the structure of linear dynamics in ways that nonlinear dynamics models do not allow.

3.2. Variational family and CRF recognition networks

[00238] Described here is a structured mean field family with which can perform variational inference in the posterior distribution of the generative model from Section 3.1. This mean field family illustrates how an SVAE can leverage not only graphical model and exponential family structure but also learn bottom-up inference networks. As shown below,

these structures allow us to compose several efficient inference algorithms including SVI, message passing, backpropagation, and the reparameterization trick.

[00239] In mean field variational inference, one constructs a tractable variational family by breaking dependencies in the posterior (Wainwright & Jordan, 2008). To construct a structured mean field family for the generative model developed in Section 3.1, one ca break the posterior dependencies between the dynamics parameters θ, the observation parameters, the discrete state sequence and the continuous state sequence, writing the corresponding a factorized density as

$$q(\theta, \vartheta, z_{1:T}, x_{1:T}) = q(\theta)q(\vartheta)q(z_{1:T})q(x_{1:T}). \quad (25)$$

[00240] Note that this structured mean field family does not break the dependencies among the discrete states or among the continuous states as in a naive mean field model because these random variables are highly correlated in the posterior. By preserving joint dependencies across time, these structured factors provide a much more accurate representation of the posterior while still allowing tractable inference via graphical model message passing (Wainwright & Jordan, 2008).

[00241] To leverage bottom-up inference networks, one can parameterize the factor as a conditional random field (CRF) (Murphy, 2012). That is, using the fact that the optimal factor is Markov according to a chain graph, we write it terms of pairwise potentials and node potentials as

$$q(x_{1:T}) \propto \left(\prod_{t=1}^{T-1} \psi(x_t, x_{t+1})\right)\left(\prod_{t=1}^{T} \psi(x_t; y_t, \phi)\right) \quad (26)$$

where the node potential is a function of the observation. Specifically, we choose each node potential to be a Gaussian factor in which the precision matrix and potential vector depend on the corresponding observation through an MLP,

$$\psi(x_t; y_t, \phi) \propto \exp\left\{-\frac{1}{2}x_t^T J(y_t; \phi)x_t + h(y_t; \phi)^T x_t\right\},$$
$$(h(y_t; \phi), J(y_t; \phi)) = \mathrm{MLP}(y_t; \phi), \quad (27)$$

using the notation from Section 2.2. These *local recognition networks* allows one to fit a regression from each observation to a probabilistic guess at the corresponding latent state. Using graphical model inference, these local guesses can be synthesized with the dynamics model into a coherent joint factor over the entire state sequence.

[00242] This structured mean field family can be directly compared to the fully factorized family used in the variational autoencoder described above. That is, there is no graph structure among the latent variables of the VAE. The SVAE generalizes the VAE by allowing the output of the recognition network to be arbitrary potentials in a graphical model, such as the node potentials considered here. Furthermore, in the SVAE some of the graphical model potentials are induced by the probabilistic model rather than being the output of a recognition network; for example, the optimal pairwise potentials are induced by the variational factors on the dynamical parameters and latent discrete states, and the forward generative model (see Section 4.2.1). Thus the SVAE provides a way to combine bottom-up information from flexible inference networks with top-down information from other latent variables in a structured probabilistic model.

[00243] When $p(\theta)$ is chosen to be a conjugate prior, as in Eq. (23), the optimal factor $q(\theta)$ is in the same exponential family:

$$q(\theta) = \exp\left\{\langle \tilde{\eta}_\theta, t_\theta(\theta)\rangle - \log Z_\theta(\tilde{\eta}_\theta)\right\}, \qquad (28)$$

To simplify notation, as in Section 2.2 we take the variational factor on the observation parameters to be a singular distribution the mean field objective in terms of the global variational parameters is then

$$\mathcal{L}(\tilde{\eta}_\theta, \vartheta^*, \phi) \triangleq \max_{q(z)q(x)} \mathbb{E}_{q(\theta)q(z)q(x)}\left[\ln \frac{p(\theta, z, x)p(y \mid x, \vartheta^*)}{q(\theta)q(z)q(x)}\right] \qquad (29)$$

where, as in Eq. (5), the maximization is over the free parameters of the local variational factors In Section 4 it is shown how to optimize this variational objective.

4. Learning and inference

[00244] Disclosed in this section is an efficient algorithm for computing stochastic gradients of the SVAE objective in Eq. (29). These stochastic gradients can be used in a generic optimization routine such as stochastic gradient ascent or Adam (Kingma & Ba, 2015).

[00245] As disclosed, the SVAE algorithm is essentially a combination of SVI (Hoffman et ah, 2013) and AEVB (Kingma & Welling, 2014), described in Sections 2.1 and 2.2, respectively. By drawing on SVI, the SVAE algorithm is able to exploit exponential family conjugacy structure, when it is available, to efficiently compute natural gradients with respect to some variational parameters. Because natural gradients are adapted to the geometry of the variational family and are invariant to model reparameterizations (Amari & Nagaoka, 2007) natural gradient ascent provides an effective second-order optimization method (Martens & Grosse, 2015; Martens, 2015). By drawing on AEVB, these algorithms can fit both general nonlinear observation models and flexible bottom-up recognition networks.

[00246] The algorithm is split into two parts. First, in Section 4.1 disclosed is the general algorithm for computing gradients of the objective in terms of the results from a model specific inference subroutine. Next, in Section 4.2 disclosed is this model applied to an inference subroutine for the SLDS.

4.1. SVAE algorithm

[00247] Here the compute stochastic gradients are computed of the SVAE mean field objective (29) using the results of a model inference subroutine. The algorithm is summarized in Algorithm 1.

[00248] For scalability, the stochastic gradients used here are computed on minibatches of data. To simplify notation, assume the dataset is a collection of N sequences, each of length T. One can sample one sequence uniformly at random and compute a stochastic gradient with it. It is also possible to sample subsequences and compute controllably-biased stochastic gradients (Foti et al., 2014).

[00249] The SVAE algorithm computes a natural and standard gradient. To computer these gradients, as in Section 2.2 we split the objective as

$$\mathbb{E}_{q(x)} \ln p(y \mid x, \vartheta^*) - \mathrm{KL}(q(\theta, z, x) \,\|\, p(\theta, z, x)). \quad (30)$$

Algorithm 1 Computing gradients of the SVAE objective

Input: Variational dynamics parameter $\tilde{\eta}_\theta$ of $q(\theta)$, observation model parameter ϑ^*, recognition network parameters ϕ, sampled sequence $y^{(n)}$

function SVAEGRADIENTS($\tilde{\eta}_\theta$, ϑ^*, ϕ, $y^{(n)}$)

$\psi \leftarrow \textsc{Recognize}(y^{(n)}, \phi)$

$(\hat{x}^{(n)}(\phi), t_{zx}^{(n)}, \mathrm{KL}(\phi)) \leftarrow \textsc{Inference}(\tilde{\eta}_\theta, \psi)$

$\tilde{\nabla}_{\tilde{\eta}_\theta} \mathcal{L} \leftarrow \eta_\theta + N(t_{zx}^{(n)}, 1) - \tilde{\eta}_\theta$

$\nabla_{\vartheta^*, \phi} \mathcal{L} \leftarrow N \nabla_{\vartheta^*, \phi} \ln p(y^{(n)} \mid \hat{x}^{(n)}(\phi), \vartheta^*) - \mathrm{KL}(\phi)$

return natural gradient $\tilde{\nabla}_{\tilde{\eta}_\theta} \mathcal{L}$, gradient $\nabla_{\vartheta^*, \phi} \mathcal{L}$

end function

[00250] Note that only the second term depends on the variational dynamics parameter. Furthermore, it is a KL divergence between two members of the same exponential family (Eqs. (23) and (28)), and so as in Hoffman et al. (2013) and Section 2.1 we can write the natural gradient of (30) as:

$$\tilde{\nabla}_{\tilde{\eta}_\theta} \mathcal{L} = \eta_\theta + \sum_{n=1}^{N} \mathbb{E}_{q(z)q(x)}(t_{zx}(z^{(n)}, x^{(n)}), 1) - \tilde{\eta}_\theta \quad (31)$$

where *q(z)* and *q(x)* are taken to be locally optimal local mean field factors as in Eq. (6). Therefore by sampling the sequence index n uniformly at random, an unbiased estimate of the natural gradient is given by:

$$\tilde{\nabla}_{\tilde{\eta}_\theta} \mathcal{L} \approx \eta_\theta + N \mathbb{E}_{q(z)q(x)}(t_{zx}(z^{(n)}, x^{(n)}), 1) - \tilde{\eta}_\theta. \quad (32)$$

We abbreviate:

$$t_{zx}^{(n)} \triangleq \mathbb{E}_{q(z)q(x)} t_{zx}(z^{(n)}, x^{(n)}).$$

These expected sufficient statistics are computed efficiently using the model inference subroutine described in Section 4.2.

Algorithm 2 Model inference subroutine for the SLDS

Input: Variational dynamics parameter $\widetilde{\eta}_\theta$ of $q(\theta)$, node potentials $\{\psi(x_t; y_t)\}_{t=1}^T$ from recognition network

function INFERENCE($\widetilde{\eta}_\theta$, $\{\psi(x_t; y_t)\}_{t=1}^T$)
  Initialize factor $q(x)$
  repeat
    $q(z) \propto \exp\{\mathbb{E}_{q(\theta)q(x)} \ln p(z, x \mid \theta)\}$
    $q(x) \propto \exp\{\mathbb{E}_{q(\theta)q(z)} \ln p(x \mid z, \theta)\} \prod_t \psi(x_t; y_t)$
  until $q(z)$ and $q(x)$ converge
  draw sample $\hat{x} \sim q(x)$
  $\bar{t}_{zx} \leftarrow \mathbb{E}_{q(z)q(x)} t_{zx}(z, x)$
  $\mathrm{KL} \leftarrow \mathrm{KL}(q(\theta) \,\|\, p(\theta)) + N\,\mathrm{KL}(q(z)q(x) \,\|\, p(z, x))$
  return sample $\hat{x}$, expected stats. $\bar{t}_{zx}$, divergence KL
end function

\

[00251] Thus we must differentiate through the procedures that the model inference subroutine uses to compute these quantities. Performing this differentiation efficiently for the SLDS corresponds to backpropagation through message passing.

4.2. Model inference subroutine

[00252] Because the VAE corresponds to a particular SVAE with limited latent probabilistic structure, this inference subroutine can be viewed as a generalization of two steps in the AEVB algorithm (Kingma & Welling, 2014). However, the inference subroutine of an SVAE may in general perform other computations: first, because the SVAE can include other latent random variables and graph structure, the inference subroutine may optimize local mean field factors or run message passing. Second, because the SVAE can perform stochastic natural gradient updates on the global factor, the inference subroutine may also compute expected sufficient statistics.

[00253] To simplify notation, one can drop the sequence index *n*, writing *y* in place of $y^{(n)}$. The algorithm is summarized in Algorithm 2.

4.2.1. OPTIMIZING LOCAL MEAN FIELD FACTORS

[00254] As in the SVI algorithm of Section 2.1, for a given data sequence *y* one can optimize the local mean field factors. That is, for a fixed global parameter factorwith natural parameter and fixed node potentials output by the recognition network, we optimize the variational objective with respect to both the local variational factor on discrete latent states and the local variational factor on continuous latent states. This optimization can be

performed efficiently by exploiting the SLDS exponential family form and the structured variational family.

4.2.2. SAMPLES, EXPECTED STATISTICS, AND KL

[00255] After optimizing the local variational factors, the model inference subroutine uses the optimized factors to draw samples, compute expected sufficient statistics, and compute a KL divergence. The results of these inference computations, are then used to compute gradients of the SVAE objective.

6. Experiments

6.1. A bouncing dot in ID

[00256] As a representative toy problem, consider a sequence of one-dimensional images in which a dot bounces from one edge of the image to the other at a fixed speed. FIG. 25 shows the results of inference in an LDS SVAE fit to this problem. The top panel shows the noisy image observations over time. The second panel shows the model's inference about both past and future images: the model is conditioned on observations to the left of the vertical red line and hence is performing filtering, while to the right of the vertical red line the model is predicting. The figure shows that the model, having learned the appropriate low-dimensional representation and dynamcis, is able to predict coherently far into the future.

[00257] One can also use the dot problem to illustrate the significant optimization advantage provided by the natural gradients with respect to the variational dynamics parameters. In Figure 6 natural gradient updates are compared with standard gradient updates at three different learning rates. The natural gradient algorithm not only learns much faster but also is more stable: while the natural gradient update used a step size of 0.1, standard gradient dynamics were unstable at step sizes of both 0.1 and 0.05 and were terminated early. While a step size of 0.01 yielded a stable standard gradient update, training is orders of magnitude slower than with the natural gradient algorithm.

6.2. MOUSE BEHAVIORAL PHENOTYPING

[00258] The goal of behavioral phenotyping is to identify patterns of behavior and study how those patterns change when the animal's environment, genetics, or brain function are altered. Here the inventors use the 3D depth camera dataset from Wiltschko et al. (2015)

to show how the SLDS SVAE can learn a flexible yet structured generative model for such video data.

[00259] The VAE's nonlinear observation model is key to learning a manifold for the depth images of the mouse. Figure 7 (referenced in FIG. 25) shows images corresponding to points on a random 2D grid in the latent space, illustrating how the nonlinear observation model can generate accurate images. An SVAE learns this feature manifold simultaneously while fitting the structured latent probabilistic model.

[00260] Figure 4(referenced in FIG. 25) illustrates some of the learned dynamical structure, which shows a generative video completion task. The figure contains both model-generated data and corresponding real data in alternating rows. Within the model-generated data, the data between the two red lines were generated without conditioning on any corresponding observations, while the data outside the two red lines were generated conditionally.

CONCLUSION

[00261] Disclosed herein is a new model class and corresponding inference algorithms that draw on both probabilistic graphical models and flexible feature representations from deep learning. In the context of time series, this approach provides several new nonlinear models which can be used for inference, estimation, and even control. For example, by maintaining latent linear structure in the SVAE, some dynamic programming control problems may remain tractable.

[00262] While this disclosure focused on time series models, particularly the SLDS and related models, the construction presented here is more general: the basic strategy of learning flexible bottom-up inference networks for CRF potentials, and then combining that bottom-up information with coherent probabilistic inference in a structured model, may be relevant wherever graphical models have proven useful. The SVAE also enables many other tools from probabilistic modeling to be combined with more recent deep learning approaches, including hierarchical modeling, structured regularization and automatic relevance determination, and easy handling of missing data.

References

[00263] 1 Fettiplace, R. & Fuchs, P. A. Mechanisms of hair cell tuning. *Annual review of physiology* **61**, 809-834, (1999).

[00264] 2 Fettiplace, R. & Kim, K. X. The Physiology of Mechanoelectrical Transduction Channels in Hearing. *Physiological reviews* **94**, 951-986, (2014).

[00265] 3 Gollisch, T. & Herz, A. M. V. Disentangling Sub-Millisecond Processes within an Auditory Transduction Chain. *PLoS Biology* **3**, e8, (2005).

[00266] 4 Kawasaki, M., Rose, G. & Heiligenberg, W. Temporal hyperacuity in single neurons of electric fish. *Nature* **336**, 173-176, (1988).

[00267] 5 Nemenman, I., Lewen, G. D., Bialek, W. & de Ruyter van Steveninck, R. R. Neural Coding of Natural Stimuli: Information at Sub-Millisecond Resolution. *PLoS computational biology* **4**, e1000025, (2008).

[00268] 6 Peters, A. J., Chen, S. X. & Komiyama, T. Emergence of reproducible spatiotemporal activity during motor learning. *Nature* **510**, 263-267, (2014).

[00269] 7 Ritzau-Jost, A., Delvendahl, I., Rings, A., Byczkowicz, N., Harada, H., Shigemoto, R., Hirrlinger, J., Eilers, J. & Hallermann, S. Ultrafast Action Potentials Mediate Kilohertz Signaling at a Central Synapse. *Neuron* **84**, 152-163, (2014).

[00270] 8 Shenoy, K. V., Sahani, M. & Churchland, M. M. Cortical Control of Arm Movements: A Dynamical Systems Perspective. *Annual review of neuroscience* **36**, 337-359, (2013).

[00271] 9 Bargmann, C. I. Beyond the connectome: How neuromodulators shape neural circuits. *BioEssays* **34**, 458-465, (2012).

[00272] 10 Tinbergen, N. *The study of instinct*. (Clarendon Press, 1951).

[00273] 11 Garrity, P. A., Goodman, M. B., Samuel, A. D. & Sengupta, P. Running hot and cold: behavioral strategies, neural circuits, and the molecular machinery for thermotaxis in C. elegans and Drosophila. *Genes & Development* **24**, 2365-2382, (2010).

[00274] 12 Stephens, G. J., Johnson-Kerner, B., Bialek, W. & Ryu, W. S. Dimensionality and Dynamics in the Behavior of C. elegans. *PLoS computational biology* **4**, e1000028, (2008).

[00275] 13 Stephens, G. J., Johnson-Kerner, B., Bialek, W. & Ryu, W. S. From Modes to Movement in the Behavior of Caenorhabditis elegans. *PLoS ONE* **5**, e13914, (2010).

[00276] 14 Vogelstein, J. T., Vogelstein, J. T., Park, Y., Park, Y., Ohyama, T., Kerr, R. A., Kerr, R. A., Truman, J. W., Truman, J. W., Priebe, C. E., Priebe, C. E., Zlatic, M. & Zlatic, M. Discovery of brainwide neural-behavioral maps via multiscale unsupervised structure learning. *Science (New York, NY)* **344**, 386-392, (2014).

[00277] 15 Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the structure of drosophilid behavior. (2013).

[00278] 16 Croll, N. A. Components and patterns in the behaviour of the nematode Caenorhabditis elegans. *Journal of zoology* **176**, 159-176, (1975).

[00279] 17 Pierce-Shimomura, J. T., Morse, T. M. & Lockery, S. R. The fundamental role of pirouettes in Caenorhabditis elegans chemotaxis. *Journal of Neuroscience* **19**, 9557-9569, (1999).

[00280] 18 Gray, J. M., Hill, J. J. & Bargmann, C. I. A circuit for navigation in Caenorhabditis elegans. *Proceedings of the National Academy of Sciences of the United States of America* **102**, 3184-3191, (2005).

[00281] 19 Miller, A. C., Thiele, T. R., Faumont, S., Moravec, M. L. & Lockery, S. R. Step-response analysis of chemotaxis in Caenorhabditis elegans. *Journal of Neuroscience* **25**, 3369-3378, (2005).

[00282] 20 Jhuang, H., Garrote, E., Yu, X., Khilnani, V., Poggio, T., Steele, A. D. & Serre, T. Automated home-cage behavioural phenotyping of mice. *Nature Communications* **1**, 68, (2010).

[00283] 21 Stewart, A., Liang, Y., Kobla, V. & Kalueff, A. V. Towards high-throughput phenotyping of complex patterned behaviors in rodents: Focus on mouse self-grooming and its sequencing. *Behavioural brain ...*, (2011).

[00284] 22 Ohayon, S., Avni, O., Taylor, A. L., Perona, P. & Egnor, S. E. R. Automated multi-day tracking of marked mice for the analysis of social behavior. *Journal of neuroscience methods*, 1-25, (2013).

[00285] 23 de Chaumont, F., Coura, R. D.-S., Serreau, P., Cressant, A., Chabout, J., Granon, S. & Olivo-Marin, J.-C. Computerized video analysis of social interactions in mice. *Nature Methods* **9**, 410-417, (2012).

[00286] 24 Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S. & Branson, K. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nature Methods* **10**, 64-67, (2013).

[00287] 25 Weissbrod, A., Shapiro, A., Vasserman, G., Edry, L., Dayan, M., Yitzhaky, A., Hertzberg, L., Feinerman, O. & Kimchi, T. Automated long-term tracking and social behavioural phenotyping of animal colonies within a semi-natural environment. *Nature Communications* **4**, 2018, (2013).

[00288] 26 Spink, A. J., Tegelenbosch, R. A., Buma, M. O. & Noldus, L. P. The EthoVision video tracking system--a tool for behavioral phenotyping of transgenic mice. *Physiology & behavior* **73**, 731-744, (2001).

[00289] 27 Tort, A. B. L., Neto, W. P., Amaral, O. B., Kazlauckas, V., Souza, D. O. & Lara, D. R. A simple webcam-based approach for the measurement of rodent locomotion and other behavioural parameters. *Journal of neuroscience methods* **157**, 91-97, (2006).

[00290] 28 Gomez-Marin, A., Partoune, N., Stephens, G. J., Louis, M. & Brembs, B. Automated tracking of animal posture and movement during exploration and sensory orientation behaviors. *PLoS ONE* 7, e41642, (2012).

[00291] 29 Colgan, P. W. *Quantitative ethology*. (John Wiley & Sons, 1978).

[00292] 30 Fox, E. B., Sudderth, E. B., Jordan, M. I. & Willsky, A. S. in *Proc. International Conference on Machine Learning* (2008).

[00293] 31 Fox, E. B., Sudderth, E. B., Jordan, M. I. & Willsky, A. S. Bayesian Nonparametric Inference of Switching Dynamic Linear Models. *IEEE Transactions on Signal Processing* **59**, (2011).

[00294] 32 Johnson, M. J. & Willsky, A. S. The Hierarchical Dirichlet Process Hidden Semi-Markov Model. *Arxiv* **abs/1203.3485**, (2012).

[00295] 33 Teh, Y. W., Jordan, M. I. & Beal, M. J. Hierarchical dirichlet processes. *Journal of the american ...*, (2006).

[00296] 34 Geman, S. & Geman, D. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721-741, (1984).

[00297] 35 Wallace, K. J. & Rosen, J. B. Predator odor as an unconditioned fear stimulus in rats: elicitation of freezing by trimethylthiazoline, a component of fox feces. *Behav Neurosci* **114**, 912-922, (2000).

[00298] 36 Fendt, M., Endres, T., Lowry, C. A., Apfelbach, R. & McGregor, I. S. TMT-induced autonomic and behavioral changes and the neural basis of its processing. *Neurosci Biobehav Rev* **29**, 1145-1156, (2005).

[00299] 37 André, E., Conquet, F., Steinmayr, M., Stratton, S. C., Porciatti, V. & Becker-André, M. Disruption of retinoid-related orphan receptor beta changes circadian behavior, causes retinal degeneration and leads to vacillans phenotype in mice. *The EMBO journal* **17**, 3867-3877, (1998).

[00300] 38 Liu, H., Kim, S.-Y., Fu, Y., Wu, X., Ng, L., Swaroop, A. & Forrest, D. An isoform of retinoid-related orphan receptor β directs differentiation of retinal amacrine and horizontal interneurons. *Nature Communications* **4**, 1813, (2013).

[00301] 39 Eppig, J. T., Blake, J. A., Bult, C. J., Kadin, J. A., Richardson, J. E. & Group, M. G. D. The Mouse Genome Database (MGD): facilitating mouse as a model for human biology and disease. *Nucleic Acids Research* **43**, D726-736, (2015).

[00302] 40 Masana, M. I., Sumaya, I. C., Becker-Andre, M. & Dubocovich, M. L. Behavioral characterization and modulation of circadian rhythms by light and melatonin in C3H/HeN mice homozygous for the RORbeta knockout. *American journal of physiology. Regulatory, integrative and comparative physiology* **292**, R2357-2367, (2007).

[00303] 41 Glickfeld, L. L., Andermann, M. L., Bonin, V. & Reid, R. C. Cortico-cortical projections in mouse visual cortex are functionally target specific. *Nature Neuroscience* **16**, 219-226, (2013).

[00304] 42 Mei, Y. & Zhang, F. Molecular tools and approaches for optogenetics. *Biological psychiatry* **71**, 1033-1038, (2012).

[00305] 43 Lashley, K. S. (ed Lloyd A Jeffress) (Psycholinguistics: A book of readings, 1967).

[00306] 44 Sherrington, C. The Integrative Action of the Nervous System. *The Journal of Nervous and Mental Disease*, (1907).

[00307] 45 Bizzi, E., Tresch, M. C., Saltiel, P. & d'Avella, A. New perspectives on spinal motor systems. *Nature Reviews Neuroscience* **1**, 101-108, (2000).

[00308] 46 Drai, D., Benjamini, Y. & Golani, I. Statistical discrimination of natural modes of motion in rat exploratory behavior. *Journal of neuroscience methods* **96**, 119-131, (2000).

[00309] 47 Brown, T. G. in Proceedings of the Royal Society of London Series B (1911).

[00310] 48 Crawley, J. N. Behavioral phenotyping of rodents. *Comparative medicine* **53**, 140-146, (2003).

[00311] 49 Anderson, D. J. & Perona, P. Toward a science of computational ethology. *Neuron* **84**, 18-31, (2014).

[00312] 50 Berg, H. C. & Brown, D. A. Chemotaxis in Escherichia coli analysed by three-dimensional tracking. *Nature* **239**, 500-504, (1972).

[00313] 51 Berg, H. C. Chemotaxis in bacteria. *Annual review of biophysics and bioengineering* **4**, 119-136, (1975).

[00314] 52 Berg, H. C. Bacterial behaviour. *Nature* **254**, 389-392, (1975).

[00315] 53 Hong, W., Kim, D.-W. & Anderson, D. J. Antagonistic Control of Social versus Repetitive Self-Grooming Behaviors by Separable Amygdala Neuronal Subsets. *Cell* **158**, 1348-1361, (2014).

[00316] 54 Lin, D., Boyle, M. P., Dollar, P., Lee, H., Lein, E. S., Perona, P. & Anderson, D. J. Functional identification of an aggression locus in the mouse hypothalamus. *Nature* **470**, 221-226, (2011).

[00317] 55 Swanson, L. W. Cerebral hemisphere regulation of motivated behavior. *Brain research* **886**, 113-164, (2000).

[00318] 56 Aldridge, J. W. & Berridge, K. C. Coding of serial order by neostriatal neurons: a "natural action" approach to movement sequence. *The Journal of neuroscience : the official journal of the Society for Neuroscience* **18**, 2777-2787, (1998).

[00319] 57 Aldridge, J. W., Berridge, K. C. & Rosen, A. R. Basal ganglia neural mechanisms of natural movement sequences. *Canadian Journal of Physiology and Pharmacology* **82**, 732-739, (2004).

[00320] 58 Jin, X., Tecuapetla, F. & Costa, R. M. Basal ganglia subcircuits distinctively encode the parsing and concatenation of action sequences. *Nature Publishing Group* **17**, 423-430, (2014).

[00321] 59 Tresch, M. C. & Jarc, A. The case for and against muscle synergies. *Current opinion in neurobiology* **19**, 601-607, (2009).

[00322] 60 Flash, T. & Hochner, B. Motor primitives in vertebrates and invertebrates. *Current opinion in neurobiology* **15**, 660-666, (2005).

[00323] 61 Bizzi, E., Cheung, V. C. K., d'Avella, A., Saltiel, P. & Tresch, M. Combining modules for movement. *Brain Research Reviews* **57**, 125-133, (2008).

[00324] 62 Tresch, M. C., Saltiel, P. & Bizzi, E. The construction of movement by the spinal cord. *Nature Neuroscience* **2**, 162-167, (1999).

[00325] 63 Berwick, R. C., Okanoya, K., Beckers, G. J. L. & Bolhuis, J. J. Songs to syntax: the linguistics of birdsong. *Trends in cognitive sciences* **15**, 113-121, (2011).

[00326] 64 Wohlgemuth, M. J., Sober, S. J. & Brainard, M. S. Linked control of syllable sequence and phonology in birdsong. *Journal of Neuroscience* **30**, 12936-12949, (2010).

[00327] 65 Markowitz, J. E., Ivie, E., Kligler, L. & Gardner, T. J. Long-range Order in Canary Song. *PLoS computational biology* **9**, e1003052, (2013).

[00328] 66 Fentress, J. C. & Stilwell, F. P. Letter: Grammar of a movement sequence in inbred mice. *Nature* **244**, 52-53, (1973).

SELECTED EMBODIMENTS

[00329] Although the above description and the attached claims disclose a number of embodiments of the present invention, other alternative aspects of the invention are disclosed in the following further embodiments.

1. A method for analyzing the motion of a subject to separate it into modules, the method comprising:

processing three dimensional video data that represents the motion of the subject using a computational model to partition the video data into at least one set of modules and at least one set of transition statistics between the modules; and

assigning the at least one set of modules to a category that represents a type of animal behavior.

2. The method of embodiment 1, said processing comprises a step of isolating the subject from the background in the video data.

3. The method of embodiment 2, said processing further comprises a step of identifying an orientation of a feature of the subject on a set of frames of the video data with respect to a coordinate system common to each frame.

4. The method of embodiment 3, said processing further comprises a step of modifying the orientation of the subject in at least a subset of the set of frames so that the feature is oriented in the same direction with respect to the coordinate system to output a set of aligned frames.

5. The method of embodiment 4, said processing further comprises a step of processing the aligned frames using a principal component analysis (PCA) to output pose dynamics data, wherein the pose dynamics data represents a pose of the subject for each aligned frame through principal component space.

6. The method of embodiment 4, said processing further comprises a step of processing the aligned frames using a multi-layer perceptron (MLP) to output pose dynamics data, wherein the pose dynamics data represents a pose of the subject for each aligned frame through a manifold space.

7. The method embodiment 5, said processing further comprises a step of processing the aligned frames with a computational model to temporally segment the pose dynamics data into separate sets of modules wherein each of the sub-second module in a set of modules exhibits similar pose dynamics.

8. The method embodiment 7, wherein said model is a switching linear dynamic system (SLDS) model.

9. The method embodiment 7, wherein said multi-layer perceptron is a structured variational autoencoder.

10. The method embodiment 6, wherein the model is trained using gradient descent and backpropagation.

11. The method embodiment 7, wherein said processing said aligned frames with the MLP happens concurrently with said processing of said frames with the computational model.

12. The method of embodiment 5, further comprising a step of displaying a representation of each of the sets of modules that occur with a frequency above a threshold in the three dimensional video data.

13. The method of embodiment 1, wherein the computational model comprises modeling the sub-second modules as a vector autoregressive process representing a stereotyped trajectory through PCA space.

14. The method of embodiment 1, wherein the computational model comprises modeling transition periods between sub-second modules using a Hidden Markov Model.

15. The method of embodiment 1, wherein the three dimensional video data is first processed to output a series of points in a multidimensional vector space, wherein each point represents the 3D pose dynamics of the subject.

16. The method of any one of embodiments 1-10, wherein the subject is an animal in an animal study.

17. The method of any one of embodiments 1-10, wherein the subject is a human.

18. A method for analyzing the motion of a subject to separate it into modules, the method comprising:

pre-processing three dimensional video data that represents the motion of the subject to isolate the subject from the background;

identifying an orientation of a feature of the subject on a set of frames of the video data with respect to a coordinate system common to each frame;

modifying the orientation of the subject in at least a subset of the set of frames so that the feature is oriented in the same direction with respect to the coordinate system to output a set of aligned frames;

processing the aligned frames using a multi-layer perceptron (MLP) to output pose dynamics data, wherein the pose dynamics data represents a pose of the subject for each aligned frame through a three dimensional graphical space;

processing the aligned frames to temporally segment the pose dynamics data into separate sets of sub-second modules wherein each of the sub-second module in a set of modules exhibits similar pose dynamics; and

displaying a representation of each of the sets of modules that occur with a frequency above a threshold in the three dimensional video data.

19. The method of embodiment 18, wherein the processing the aligned frames step is performed using a model free algorithm.

20. The method of embodiment 19, wherein the model free algorithm comprises computing an auto-correlogram.

21. The method of embodiment 18, wherein the processing the aligned frames step is performed using a model based algorithm.

22. The method of embodiment 21, wherein the model based algorithm is an AR-HMM algorithm.

23. The method of embodiment 21, wherein the model based algorithm is an SLDS algorithm.

24. The method of embodiment 18, wherein said multi-layer perceptron is an SVAE.

25. The method of embodiment 24 wherein the SVAE and MLP are trained using a variational inference objective and performing gradient ascent.

26. The method of embodiment 25, wherein the SVAE and MLP are trained simultaneously.

27. The method of any one of embodiments 18-22, wherein the subject is an animal in an animal study.

28. The method of any one of embodiments 18-22, wherein the subject is a human.

29. The method of any one of embodiments 18-22, wherein the subject is analyzed over a period of time long enough for the subject to change in size.

30. The method of embodiment 25, wherein the SVAE and MLP are trained using data based on different strains of mice or rats.

31. A method of classifying a test compound, the method comprising:

identifying a test behavioral representation that includes a set of modules in a test subject after the test compound is administered to the test subject;

comparing the test behavioral representation to a plurality of reference behavioral representations, wherein each reference behavioral representation represents each class of drugs; and

determining that the test compound belongs to a class of drugs if the test behavioral representation is identified by a classifier as matching the reference behavioral representation representing said class of drugs.

32. The method of embodiment 31, wherein the test behavioral representation is identified by

receiving three dimensional video data representing the motion of the test subject;

processing the three dimensional data using a computational model to partition the data into at least one set of modules and at least one set of transition periods between the modules; and

assigning the at least one set of modules to a category that represents a type of animal behavior.

33. The method of embodiment 32, wherein the computational model comprises modeling the sub-second modules as a vector autoregressive process representing a stereotyped trajectory through principal component analysis (PCA) space.

34. The method of embodiment 32, wherein the computational model comprises modeling the sub-second modules as a SLDS that is fitted simultaneously while an MLP learns a feature manifold for the sub-second modules.

35. The method of embodiment 34, wherein the MLP is an SVAE.

36. The method of embodiment 33, wherein the computational model comprises modeling the transition periods using a Hidden Markov Model.

37. The method of any one of embodiments 31 - 36, wherein the three dimensional video data is first processed to output a series of points in a multidimensional vector space, wherein each point represents the 3D pose dynamics of the test subject.

38. The method of any one of embodiments 31 - 37, wherein the test compound is selected from the group consisting of a small molecule, an antibody or an antigen-binding fragment thereof, a nucleic acid, a polypeptide, a peptide, a peptidomimetic, a polysaccharide, a monosaccharide, a lipid, a glycosaminoglycan, and a combination thereof.

39. The method of any one of embodiments 31 – 38 wherein the test subject is an animal in an animal study.

40. A method for analyzing the motion of a subject to separate it into modules, the method comprising:

receiving three dimensional video data representing the motion of the subject before and after administration of an agent to the subject;

pre-processing the three dimensional video data to isolate the subject from the background;

identifying an orientation of a feature of the subject on a set of frames of the video data with respect to a coordinate system common to each frame;

modifying the orientation of the subject in at least a subset of the set of frames so that the feature is oriented in the same direction with respect to the coordinate system to output a set of aligned frames;

processing the aligned frames using a multi-layer perceptron (MLP) to output pose dynamics data, wherein the pose dynamics data represents a pose of the subject for each aligned frame through a three dimensional feature manifold;

processing the aligned frames with a computational model to temporally segment the pose dynamics data into separate sets of modules wherein each of the sub-second module in a set of sub-second modules exhibits similar pose dynamics;

determining the quantity of modules in each set of sub-second modules before administration of the agent to the subject;

determining the quantity of modules in each set of sub-second modules after administration of the agent to the subject;

comparing the quantity of modules in each set of sub-second modules before and after administration of the agent to the subject; and

outputting an indication of the change in frequency of expression of the quantity of modules in each set of modules before and after administration of the agent to the subject.

41. The method of embodiment 40, wherein each set of sub-second modules is classified into a predetermined behavior module based on comparison to reference data representing behavior modules.

42. The method of embodiment 40 or 41, wherein the change in frequency of expression of the quantity of modules in each set of modules before and after administration of the agent to the subject is compared to the reference data representing a change in frequency of expression of modules after exposure to known categories of agents.

43. The method of embodiment 42, comprising the further step of classifying the agent as one of the plurality of known categories of agents based on the comparison to reference data representing the change in frequency after exposure to known categories of agents.

44. The method of any one of embodiments 40-42, wherein the agent is a pharmaceutically active compound.

45. The method of any one of embodiments 40 - 42wherein the agent is visual or auditory stimulus.

46. The method of any one of embodiments 40-42, wherein the agent is an odorant.

47. The method of any one of embodiments 40-46, wherein the subject is an animal in an animal study.

48. The method of any one of embodiments 40-46 wherein the subject is a human.

COMPUTER & HARDWARE IMPLEMENTATION OF DISCLOSURE

[00330] It should initially be understood that the disclosure herein may be implemented with any type of hardware and/or software, and may be a pre-programmed general purpose computing device. For example, the system may be implemented using a server, a personal computer, a portable computer, a thin client, or any suitable device or devices. The disclosure and/or components thereof may be a single device at a single location, or multiple devices at a single, or multiple, locations that are connected together using any appropriate communication protocols over any communication medium such as electric cable, fiber optic cable, or in a wireless manner.

[00331] It should also be noted that the disclosure is illustrated and discussed herein as having a plurality of modules which perform particular functions. It should be understood that these modules are merely schematically illustrated based on their function for clarity purposes only, and do not necessary represent specific hardware or software. In this regard, these modules may be hardware and/or software implemented to substantially perform the particular functions discussed. Moreover, the modules may be combined together within the disclosure, or divided into additional modules based on the particular function desired. Thus, the disclosure should not be construed to limit the present invention, but merely be understood to illustrate one example implementation thereof.

[00332] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other. In some implementations, a server transmits data (e.g., an HTML page) to a client device (e.g., for purposes of displaying data to and receiving user input from a user interacting with the client device). Data generated at the client device (e.g., a result of the user interaction) can be received from the client device at the server.

[00333] Implementations of the subject matter described in this specification can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the subject matter

described in this specification, or any combination of one or more such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), an inter-network (e.g., the Internet), and peer-to-peer networks (e.g., ad hoc peer-to-peer networks).

[00334] Implementations of the subject matter and the operations described in this specification can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Implementations of the subject matter described in this specification can be implemented as one or more computer programs, i.e., one or more modules of computer program instructions, encoded on computer storage medium for execution by, or to control the operation of, data processing apparatus. Alternatively or in addition, the program instructions can be encoded on an artificially-generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus. A computer storage medium can be, or be included in, a computer-readable storage device, a computer-readable storage substrate, a random or serial access memory array or device, or a combination of one or more of them. Moreover, while a computer storage medium is not a propagated signal, a computer storage medium can be a source or destination of computer program instructions encoded in an artificially-generated propagated signal. The computer storage medium can also be, or be included in, one or more separate physical components or media (e.g., multiple CDs, disks, or other storage devices).

[00335] The operations described in this specification can be implemented as operations performed by a "data processing apparatus" on data stored on one or more computer-readable storage devices or received from other sources.

[00336] The term "data processing apparatus" encompasses all kinds of apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, a system on a chip, or multiple ones, or combinations, of the foregoing The apparatus can include special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit). The apparatus can also

include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, a cross-platform runtime environment, a virtual machine, or a combination of one or more of them. The apparatus and execution environment can realize various different computing model infrastructures, such as web services, distributed computing and grid computing infrastructures.

[00337] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, declarative or procedural languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, object, or other unit suitable for use in a computing environment. A computer program may, but need not, correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub-programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[00338] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform actions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

[00339] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for performing actions in accordance with instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical

disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio or video player, a game console, a Global Positioning System (GPS) receiver, or a portable storage device (e.g., a universal serial bus (USB) flash drive), to name just a few. Devices suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

CONCLUSION

[00340] The various methods and techniques described above provide a number of ways to carry out the invention. Of course, it is to be understood that not necessarily all objectives or advantages described can be achieved in accordance with any particular embodiment described herein. Thus, for example, those skilled in the art will recognize that the methods can be performed in a manner that achieves or optimizes one advantage or group of advantages as taught herein without necessarily achieving other objectives or advantages as taught or suggested herein. A variety of alternatives are mentioned herein. It is to be understood that some embodiments specifically include one, another, or several features, while others specifically exclude one, another, or several features, while still others mitigate a particular feature by inclusion of one, another, or several advantageous features.

[00341] Furthermore, the skilled artisan will recognize the applicability of various features from different embodiments. Similarly, the various elements, features and steps discussed above, as well as other known equivalents for each such element, feature or step, can be employed in various combinations by one of ordinary skill in this art to perform methods in accordance with the principles described herein. Among the various elements, features, and steps some will be specifically included and others specifically excluded in diverse embodiments.

[00342] Although the application has been disclosed in the context of certain embodiments and examples, it will be understood by those skilled in the art that the

embodiments of the application extend beyond the specifically disclosed embodiments to other alternative embodiments and/or uses and modifications and equivalents thereof.

[00343] In some embodiments, the terms "a" and "an" and "the" and similar references used in the context of describing a particular embodiment of the application (especially in the context of certain of the following claims) can be construed to cover both the singular and the plural. The recitation of ranges of values herein is merely intended to serve as a shorthand method of referring individually to each separate value falling within the range. Unless otherwise indicated herein, each individual value is incorporated into the specification as if it were individually recited herein. All methods described herein can be performed in any suitable order unless otherwise indicated herein or otherwise clearly contradicted by context. The use of any and all examples, or exemplary language (for example, "such as") provided with respect to certain embodiments herein is intended merely to better illuminate the application and does not pose a limitation on the scope of the application otherwise claimed. No language in the specification should be construed as indicating any non-claimed element essential to the practice of the application.

[00344] Certain embodiments of this application are described herein. Variations on those embodiments will become apparent to those of ordinary skill in the art upon reading the foregoing description. It is contemplated that skilled artisans can employ such variations as appropriate, and the application can be practiced otherwise than specifically described herein. Accordingly, many embodiments of this application include all modifications and equivalents of the subject matter recited in the claims appended hereto as permitted by applicable law. Moreover, any combination of the above-described elements in all possible variations thereof is encompassed by the application unless otherwise indicated herein or otherwise clearly contradicted by context.

[00345] Particular implementations of the subject matter have been described. Other implementations are within the scope of the following claims. In some cases, the actions recited in the claims can be performed in a different order and still achieve desirable results. In addition, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results.

[00346] All patents, patent applications, publications of patent applications, and other material, such as articles, books, specifications, publications, documents, things, and/or the

like, referenced herein are hereby incorporated herein by this reference in their entirety for all purposes, excepting any prosecution file history associated with same, any of same that is inconsistent with or in conflict with the present document, or any of same that may have a limiting affect as to the broadest scope of the claims now or later associated with the present document. By way of example, should there be any inconsistency or conflict between the description, definition, and/or the use of a term associated with any of the incorporated material and that associated with the present document, the description, definition, and/or the use of the term in the present document shall prevail.

[00347] In closing, it is to be understood that the embodiments of the application disclosed herein are illustrative of the principles of the embodiments of the application. Other modifications that can be employed can be within the scope of the application. Thus, by way of example, but not of limitation, alternative configurations of the embodiments of the application can be utilized in accordance with the teachings herein. Accordingly, embodiments of the present application are not limited to that precisely as shown and described.

CLAIMS

1. A method for analyzing the motion of a subject to separate it into modules, the method comprising:

processing three dimensional video data that represents the motion of the subject using a computational model to partition the video data into at least one set of modules and at least one set of transition statistics between the modules; and

assigning the at least one set of modules to a category that represents a type of animal behavior.

2. The method of claim 1, said processing comprises a step of isolating the subject from the background in the video data.

3. The method of claim 2, said processing further comprises a step of identifying an orientation of a feature of the subject on a set of frames of the video data with respect to a coordinate system common to each frame.

4. The method of claim 3, said processing further comprises a step of modifying the orientation of the subject in at least a subset of the set of frames so that the feature is oriented in the same direction with respect to the coordinate system to output a set of aligned frames.

5. The method of claim 4, said processing further comprises a step of processing the aligned frames using a multi-layer perceptron (MLP) to output pose dynamics data, wherein the pose dynamics data represents a pose of the subject for each aligned frame through a manifold space.

6. The method claim 5, wherein said multi-layer perceptron is a structured variational autoencoder.

7. The method of claim 1, wherein the three dimensional video data is first processed to output a series of points in a multidimensional vector space, wherein each point represents the 3D pose dynamics of the subject.

8. The method of any one of claims 1-10, wherein the subject is an animal in an animal study.

9. The method of any one of claims 1-10, wherein the subject is a human.

10. A method for analyzing the motion of a subject to separate it into modules, the method comprising:

pre-processing three dimensional video data that represents the motion of the subject to isolate the subject from the background;

identifying an orientation of a feature of the subject on a set of frames of the video data with respect to a coordinate system common to each frame;

modifying the orientation of the subject in at least a subset of the set of frames so that the feature is oriented in the same direction with respect to the coordinate system to output a set of aligned frames;

processing the aligned frames using a multi-layer perceptron (MLP) to output pose dynamics data, wherein the pose dynamics data represents a pose of the subject for each aligned frame through a three dimensional graphical space;

processing the aligned frames to temporally segment the pose dynamics data into separate sets of sub-second modules wherein each of the sub-second module in a set of modules exhibits similar pose dynamics; and

displaying a representation of each of the sets of modules that occur with a frequency above a threshold in the three dimensional video data.

11. The method of claim 10, wherein the processing the aligned frames step is performed using a model free algorithm.

12. The method of claim 11, wherein the model free algorithm comprises computing an auto-correlogram.

13. The method of claim 10, wherein the processing the aligned frames step is performed using a model based algorithm.

14. The method of claim 13, wherein the model based algorithm is an AR-HMM algorithm.

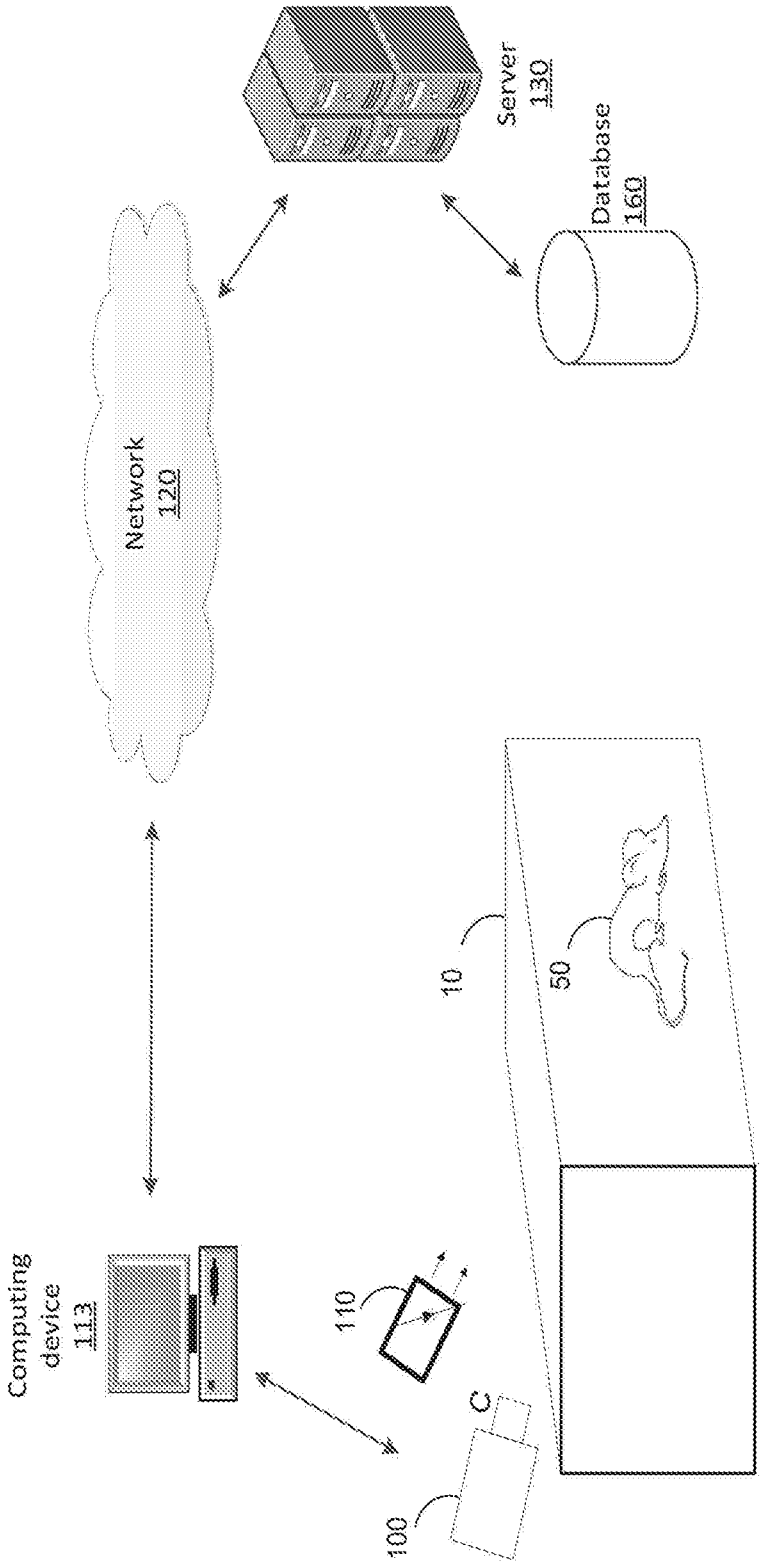

Server
130
Database
160
Network
120
10
50
Computing
device
113
110
C
100

FIG. 1

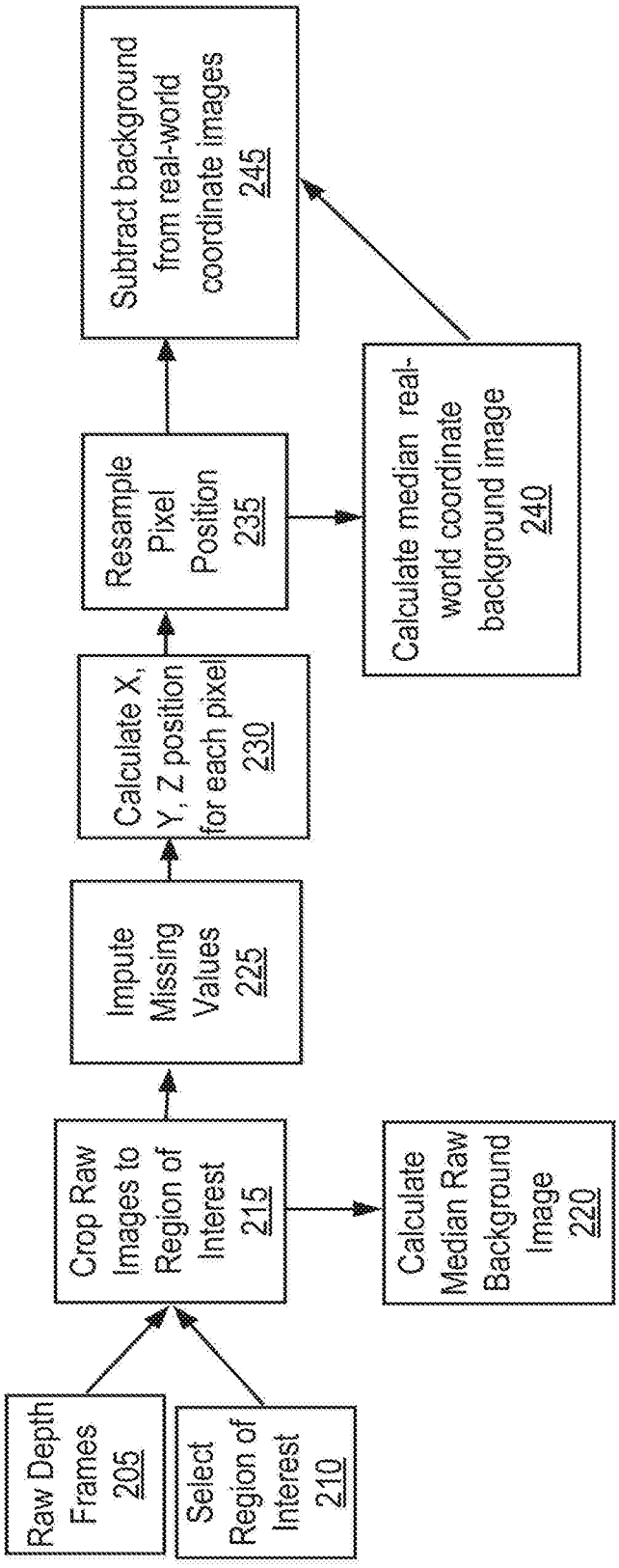

Raw Depth Frames 205
Select Region of Interest 210
Crop Raw Images to Region of Interest 215
Calculate Median Raw Background Image 220
Impute Missing Values 225
Calculate X, Y, Z position for each pixel 230
Resample Pixel Position 235
Calculate median real-world coordinate background image 240
Subtract background from real-world coordinate images 245

FIG. 2A

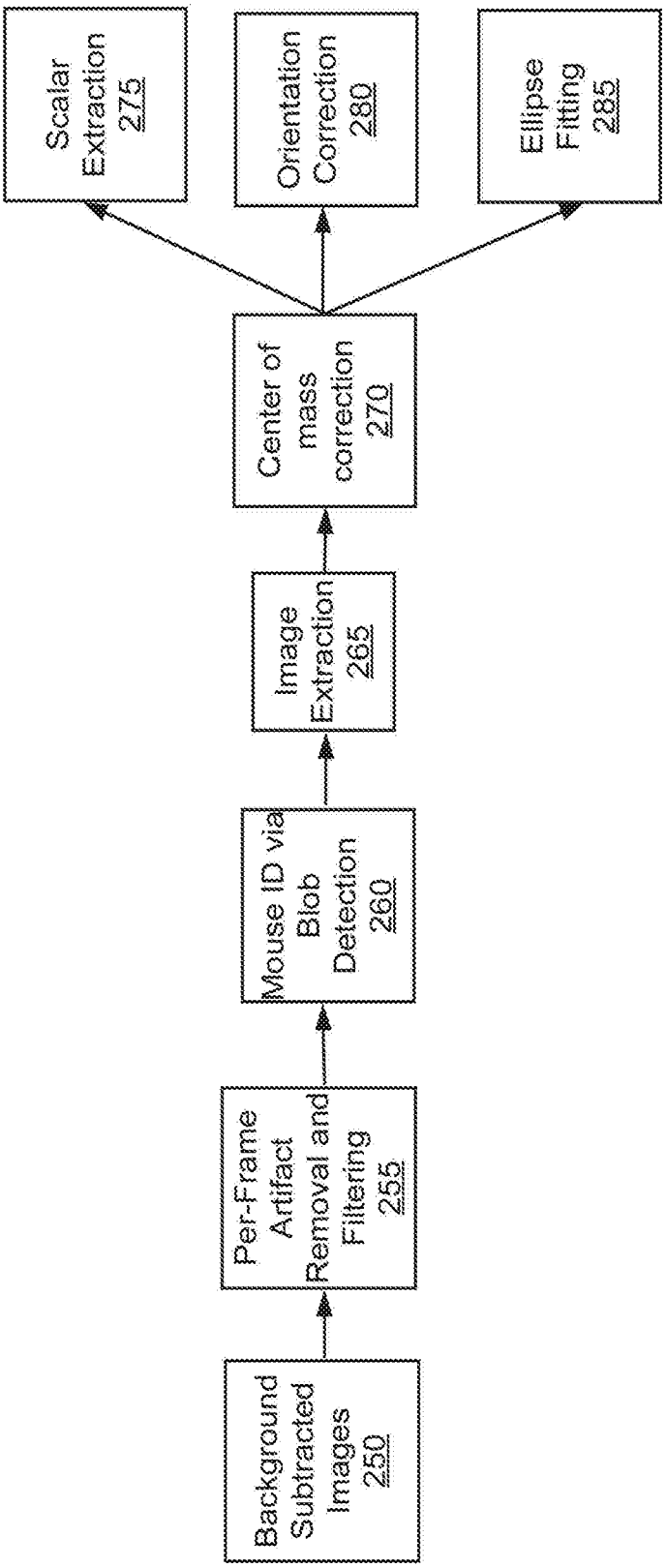

Background Subtracted Images 250
Per-Frame Artifact Removal and Filtering 255
Mouse ID via Blob Detection 260
Image Extraction 265
Center of mass correction 270
Scalar Extraction 275
Orientation Correction 280
Ellipse Fitting 285

FIG. 2B

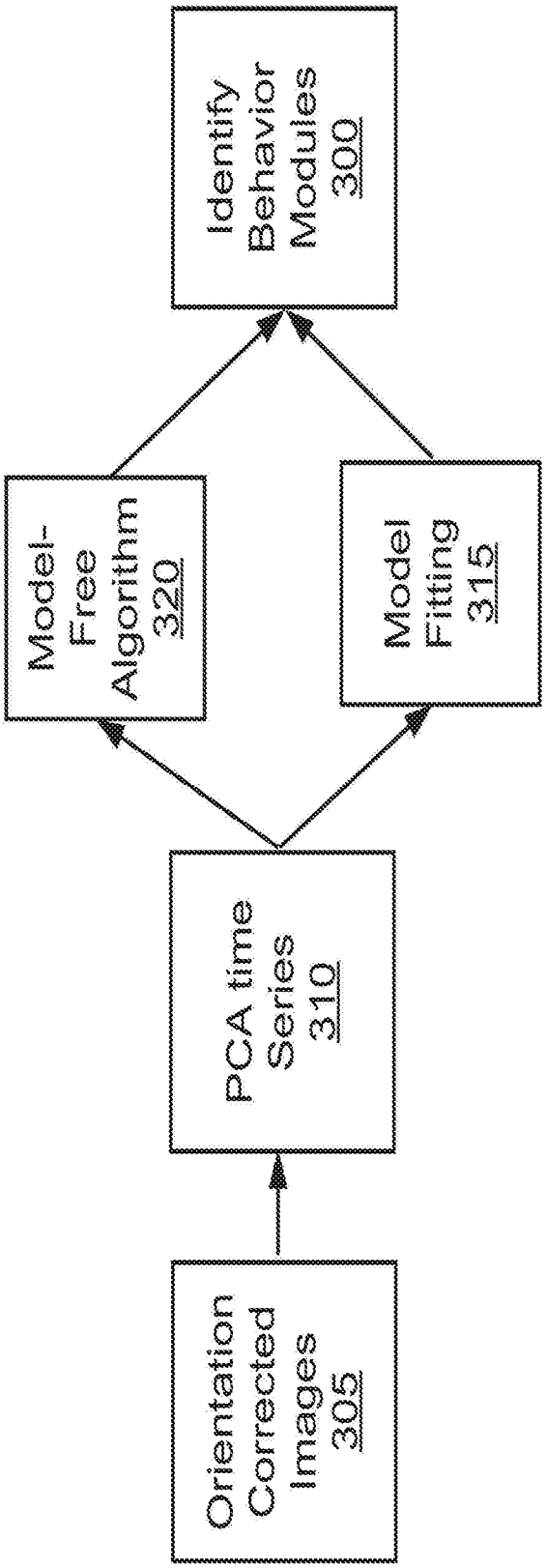

Identify Behavior Modules 300
Model-Free Algorithm 320
Model Fitting 315
PCA time Series 310
Orientation Corrected Images 305

FIG. 3

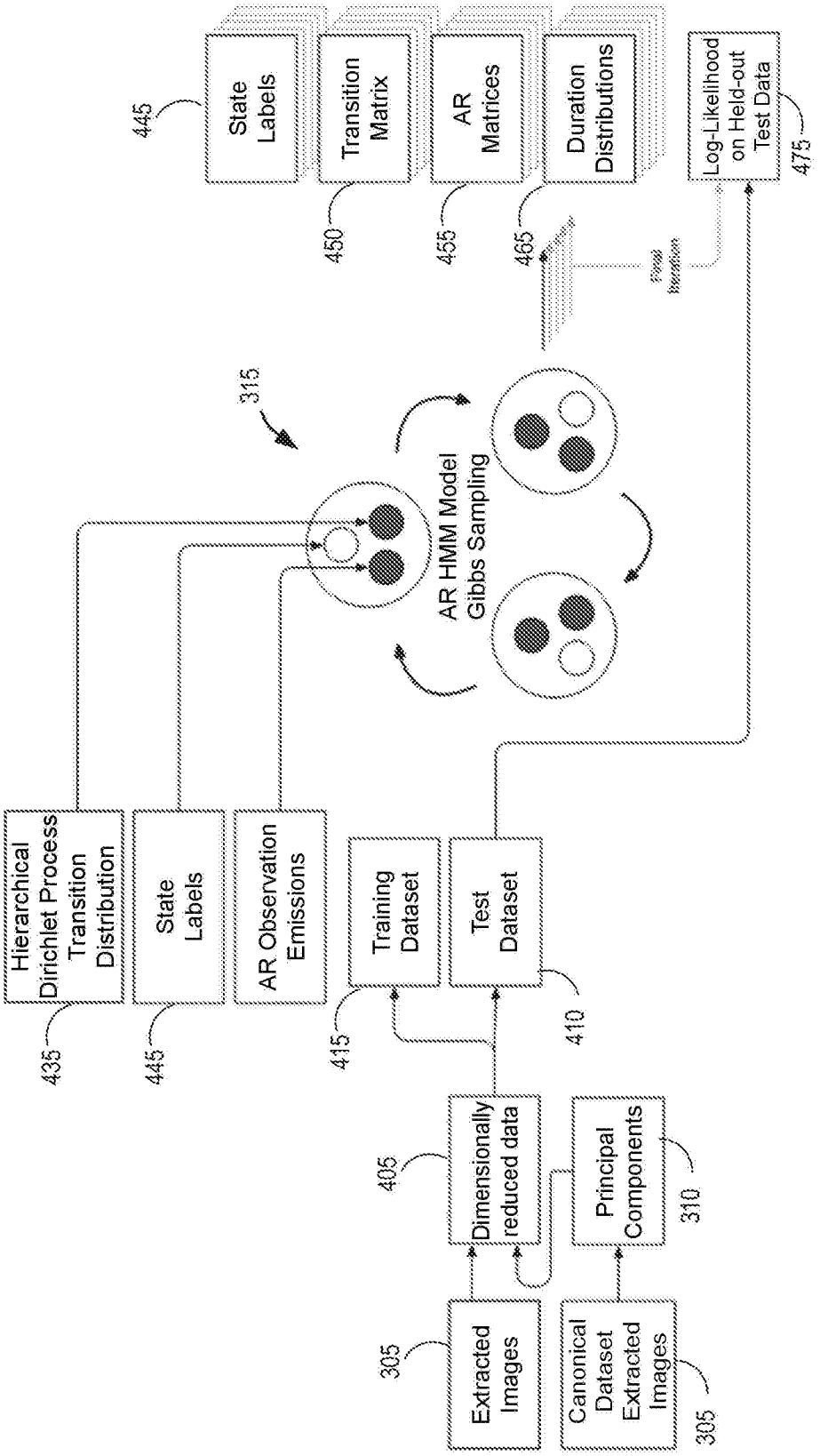

Extracted Images
305
Canonical Dataset Extracted Images
305
Dimensionally reduced data
405
Principal Components
310
Training Dataset
415
Test Dataset
410
Hierarchical Dirichlet Process Transition Distribution
435
State Labels
445
AR Observation Emissions
AR HMM Model Gibbs Sampling
315
State Labels
445
Transition Matrix
450
AR Matrices
455
Duration Distributions
465
Log-Likelihood on Held-out Test Data
475

FIG. 4

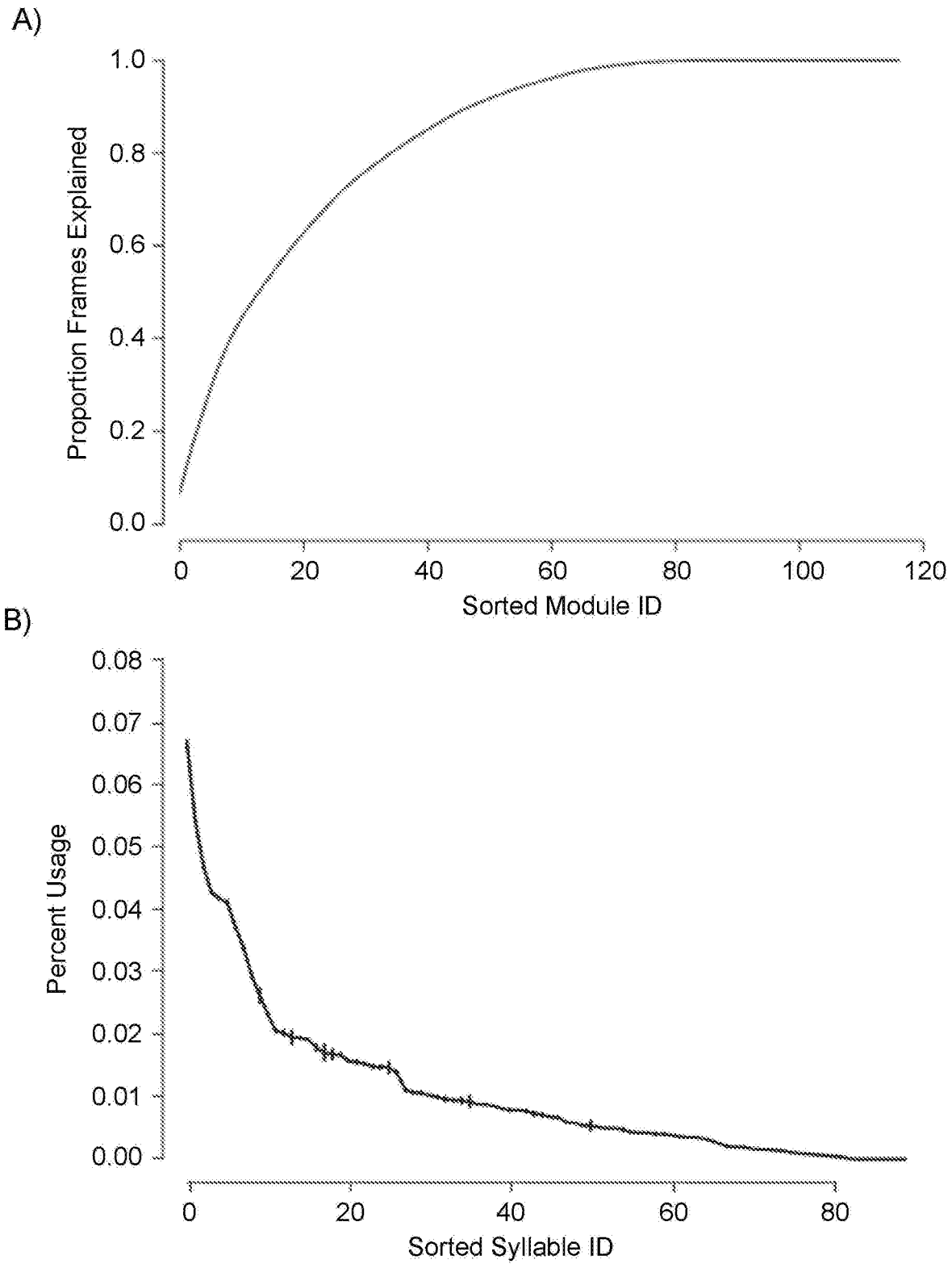

A)
1.0
0.8
0.6
0.4
0.2
0.0
Proportion Frames Explained
0
20
40
60
80
100
120
Sorted Module ID
B)
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Percent Usage
0
20
40
60
80
Sorted Syllable ID

FIG. 5A-5B

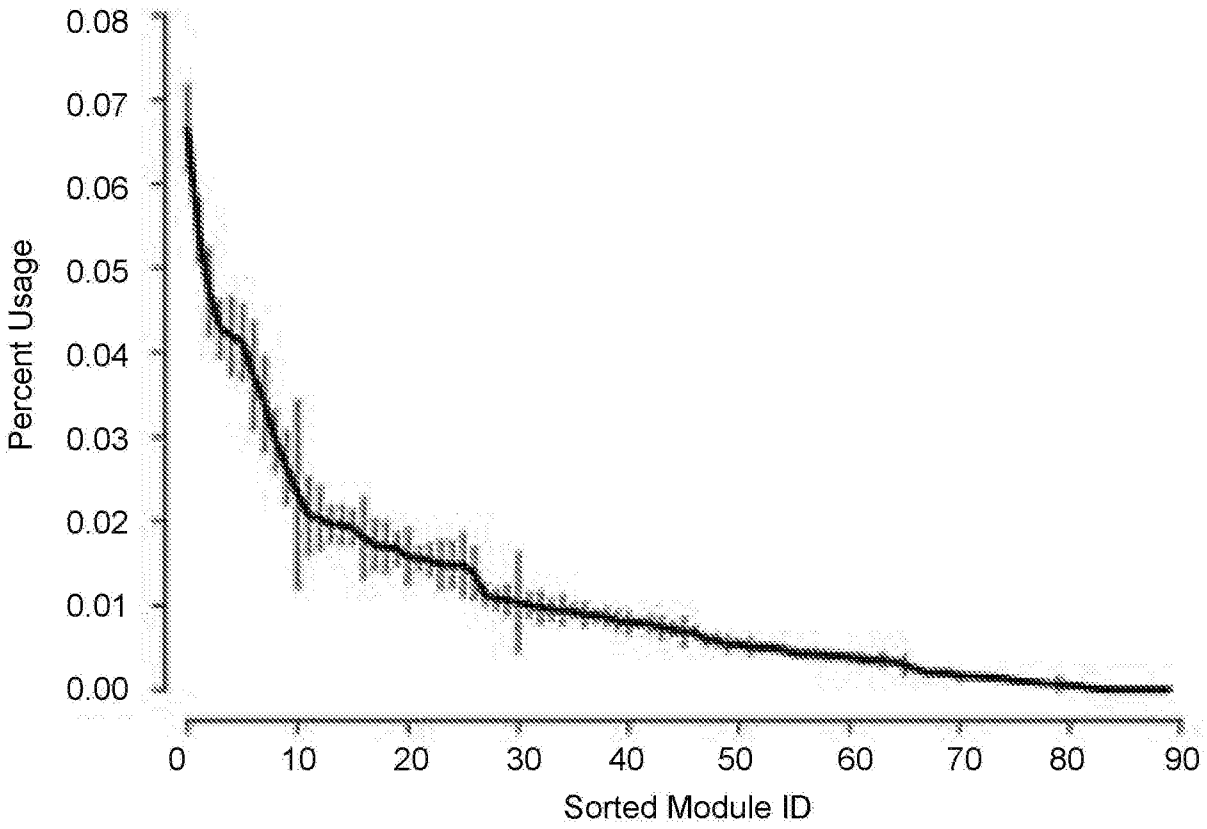

0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Percent Usage
0
10
20
30
40
50
60
70
80
90
Sorted Module ID

FIG. 6A

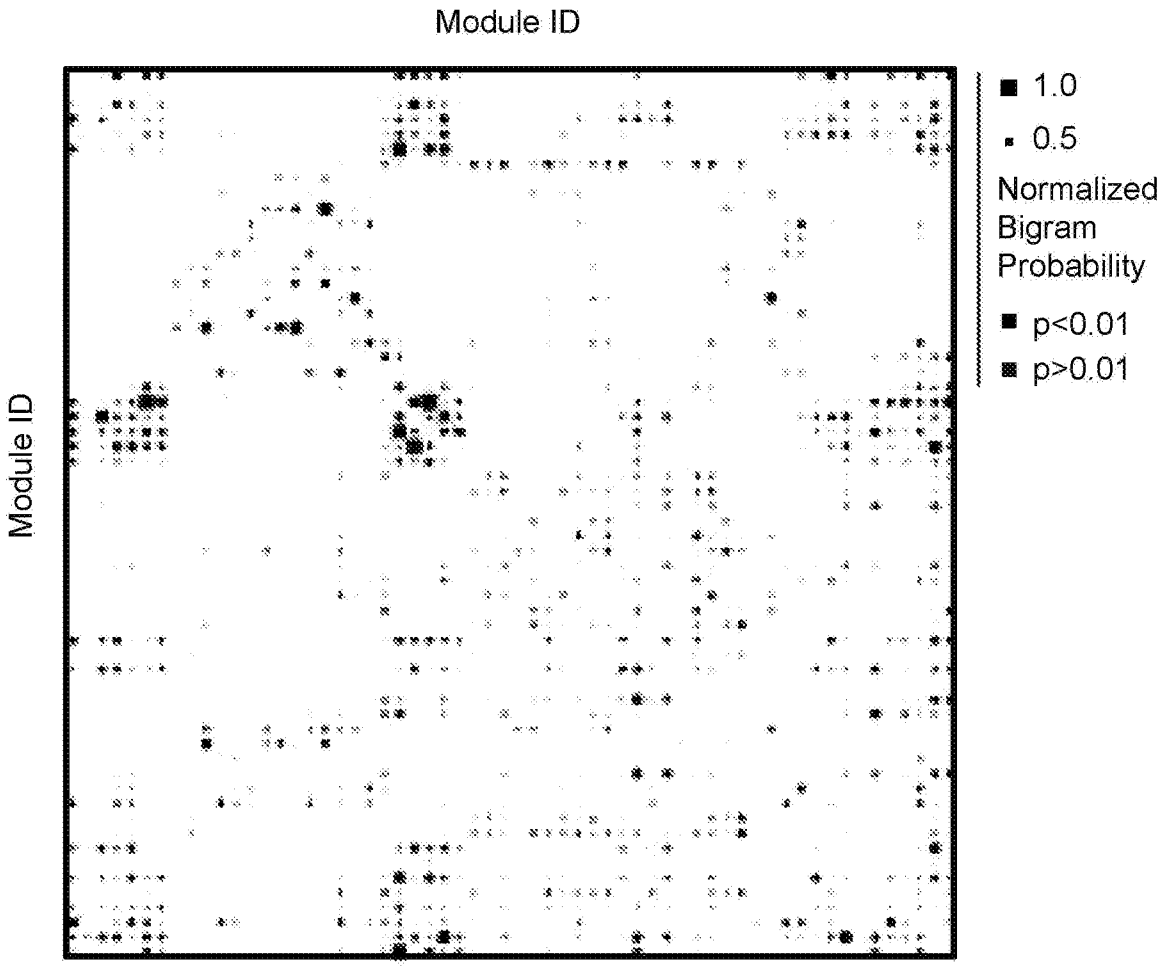

Module ID
Module ID
1.0
0.5
Normalized
Bigram
Probability
p<0.01
p>0.01

FIG. 6B

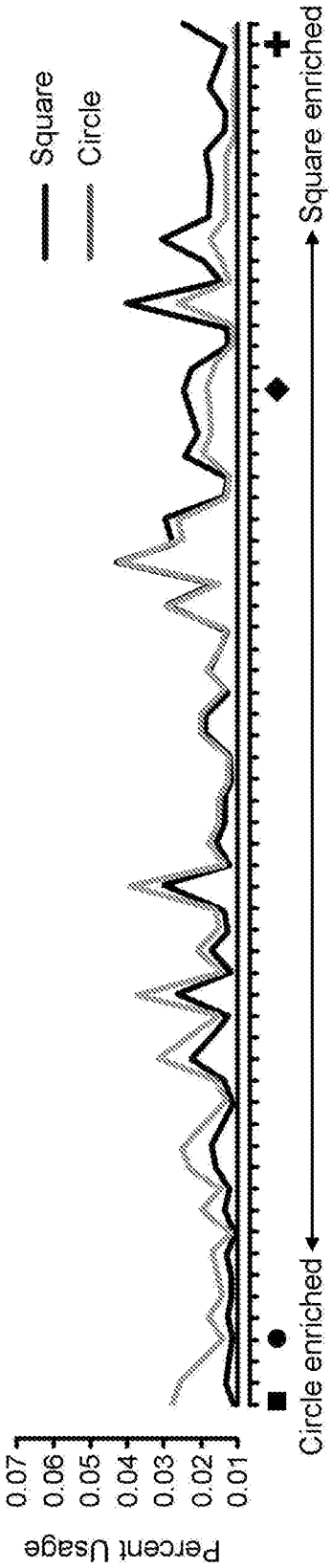

Square
Circle
Square enriched
Circle enriched
Percent Usage
0.07
0.06
0.05
0.04
0.03
0.02
0.01

FIG. 6C

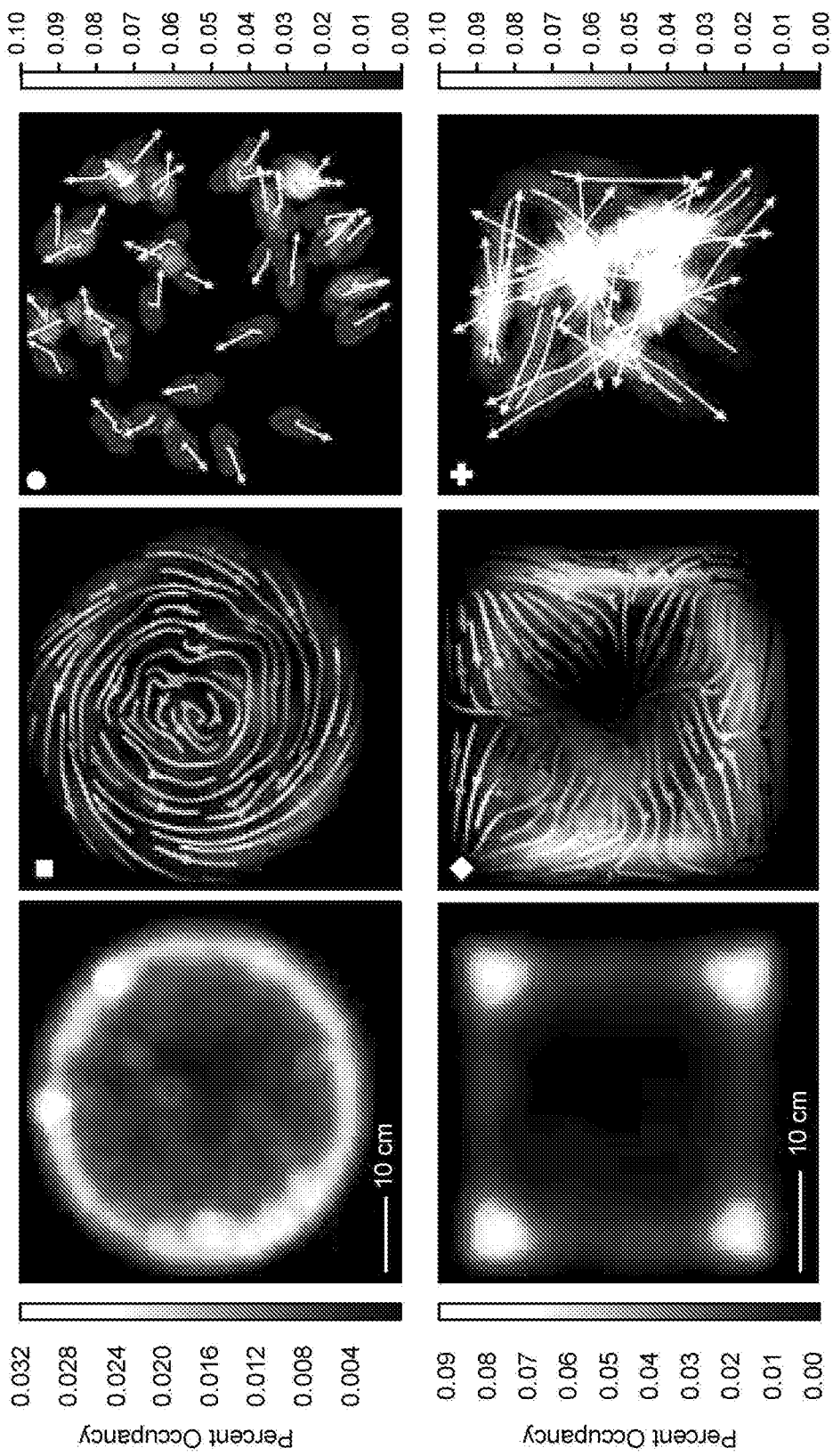

0.10
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
10 cm
0.032
0.028
0.024
0.020
0.016
0.012
0.008
0.004
Percent Occupancy
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Percent Occupancy

FIG. 6D-6E

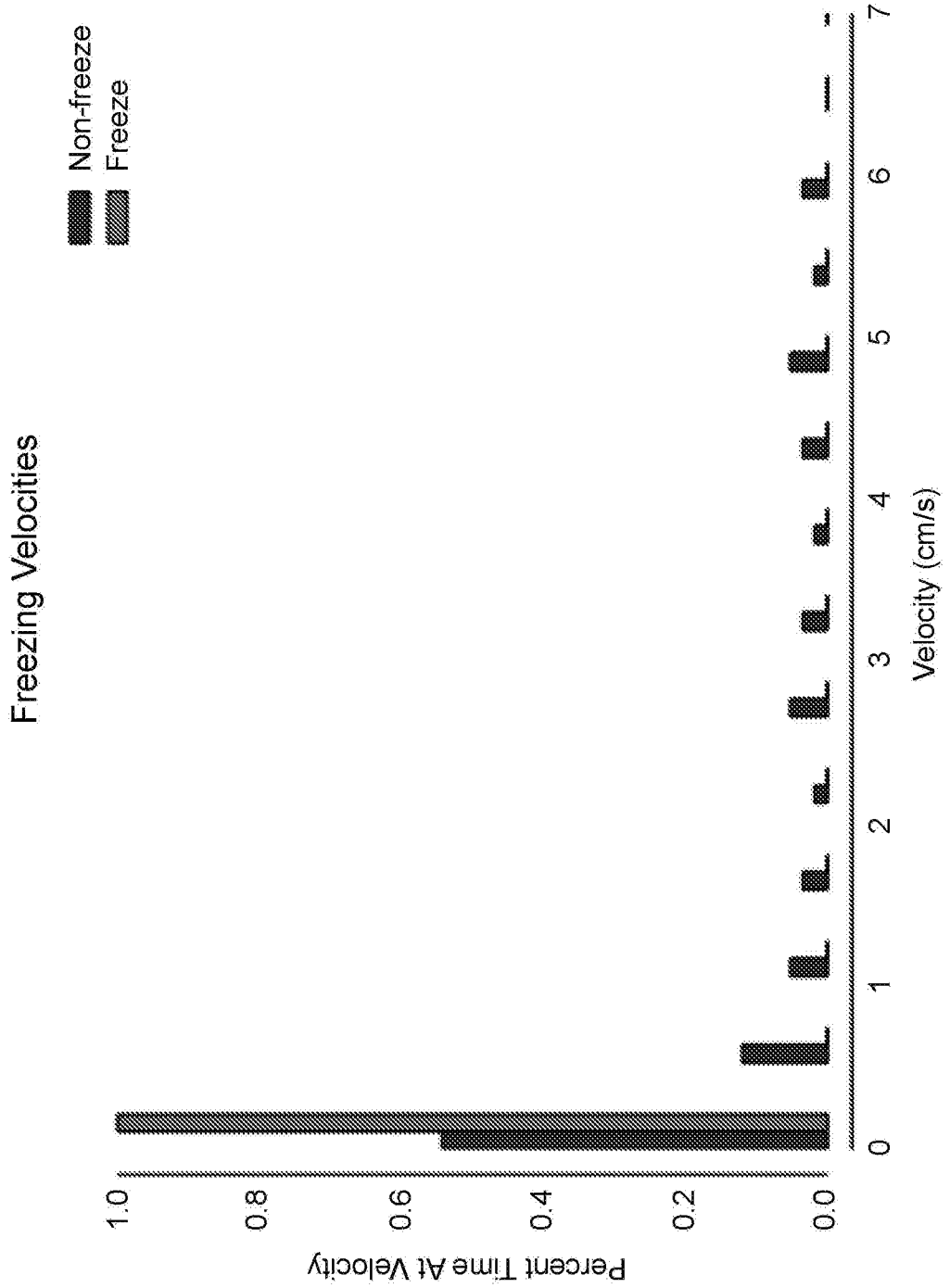
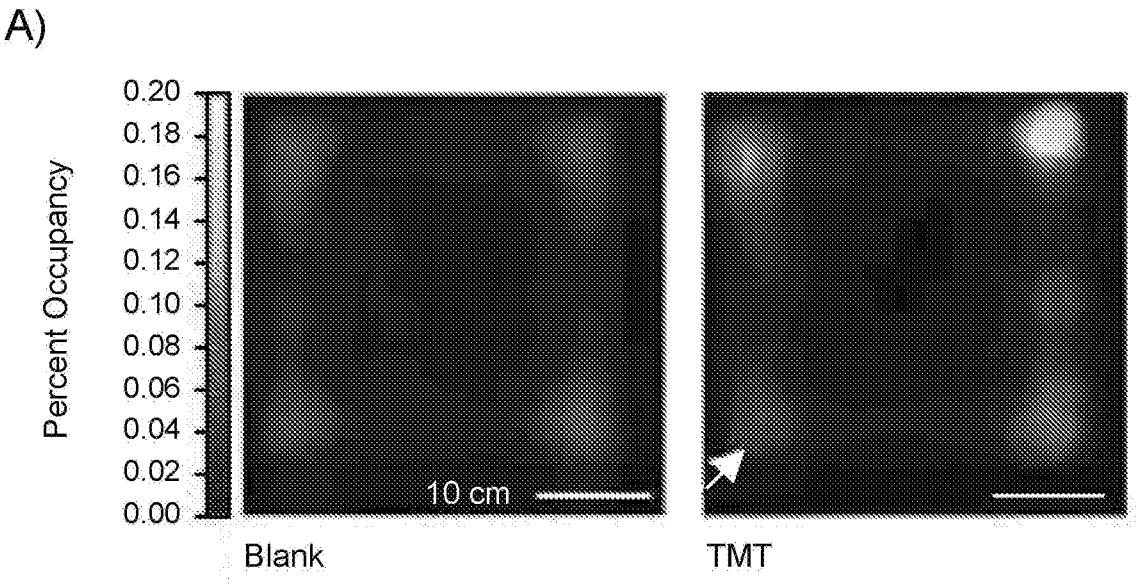

Freezing Velocities
Non-freeze
Freeze
Percent Time At Velocity
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6
7
Velocity (cm/s)

FIG. 7


A)
Percent Occupancy
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
10 cm
Blank
TMT

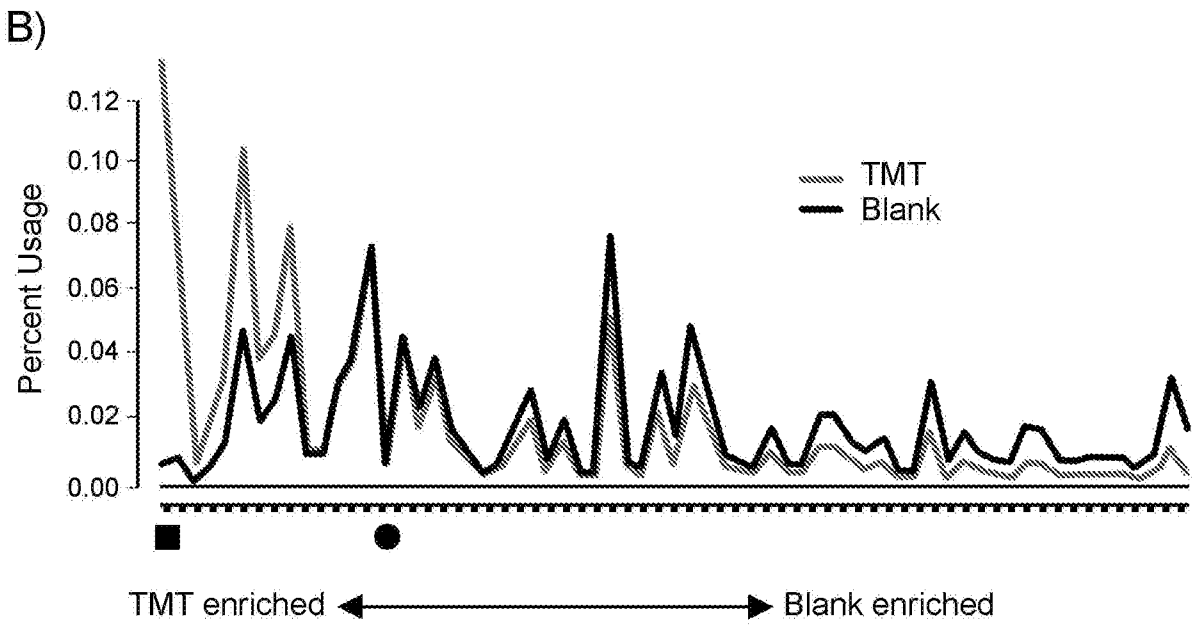

B)
Percent Usage
0.12
0.10
0.08
0.06
0.04
0.02
0.00
TMT
Blank
TMT enriched
Blank enriched

FIG. 8A-8B

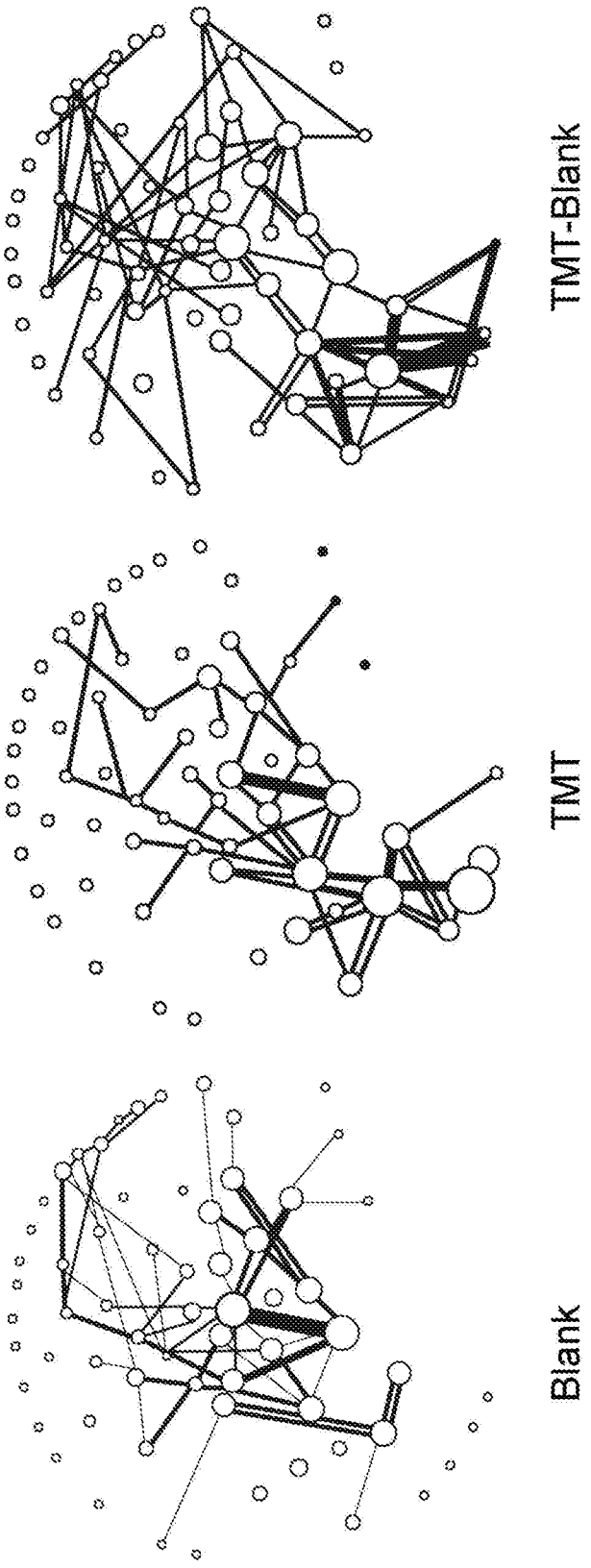
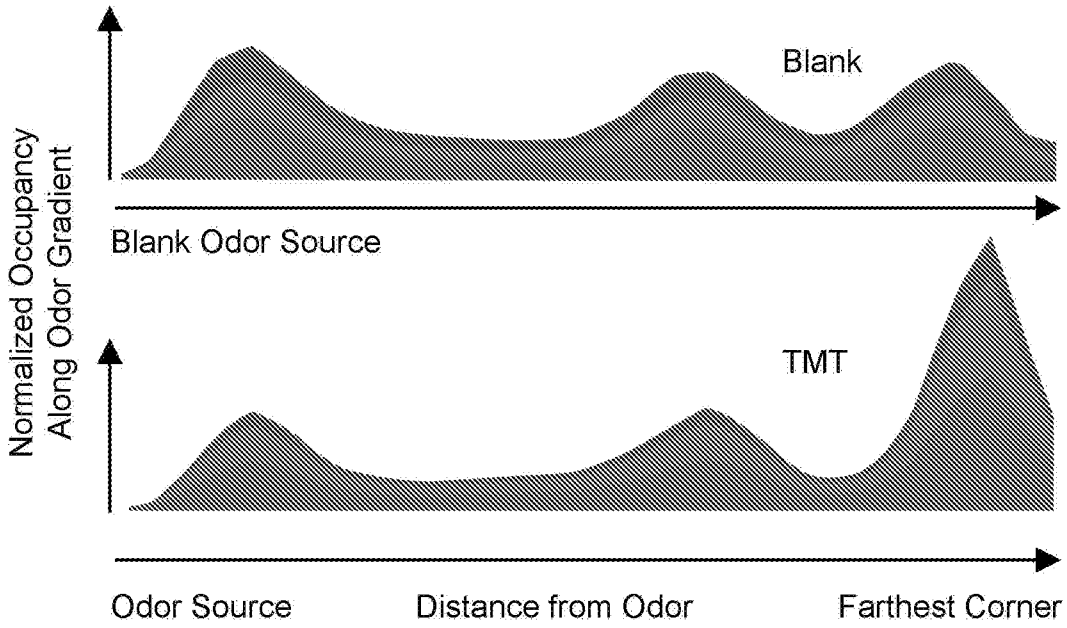


FIG. 8C

D)


Blank
Normalized Occupancy Along Odor Gradient
Blank Odor Source
TMT
Odor Source
Distance from Odor
Farthest Corner

E)

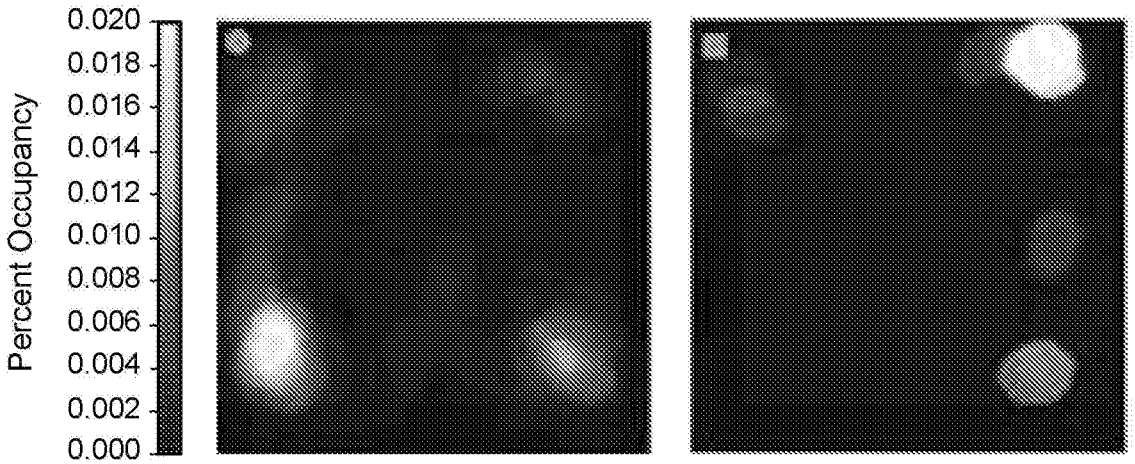

Percent Occupancy
0.020
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000

FIG. 8D-8E

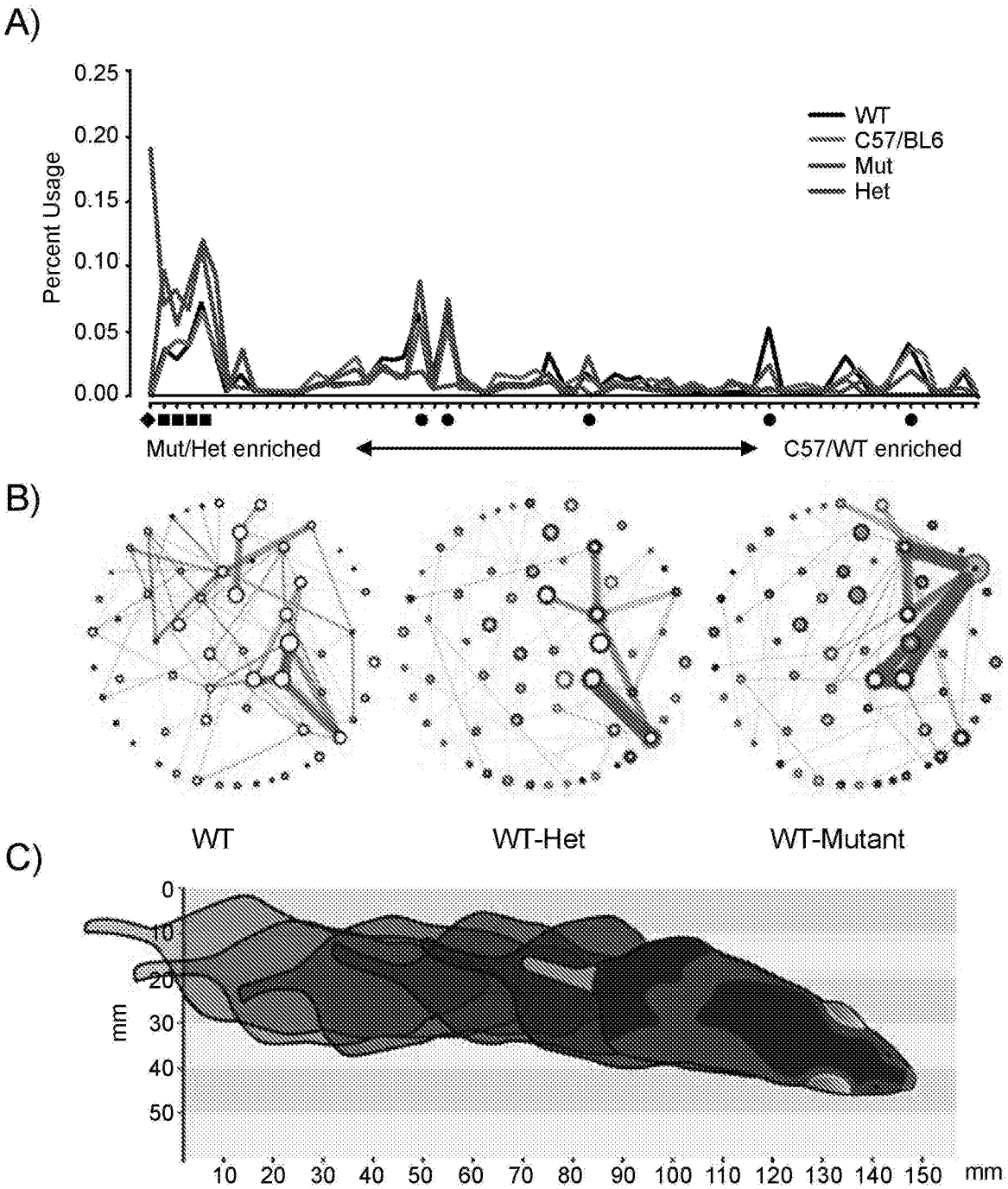

A)
0.25
0.20
0.15
0.10
0.05
0.00
Percent Usage
WT
C57/BL6
Mut
Het
Mut/Het enriched
C57/WT enriched
B)
WT
WT-Het
WT-Mutant
C)
0
10
20
30
40
50
mm
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 mm

FIG. 9A-9C

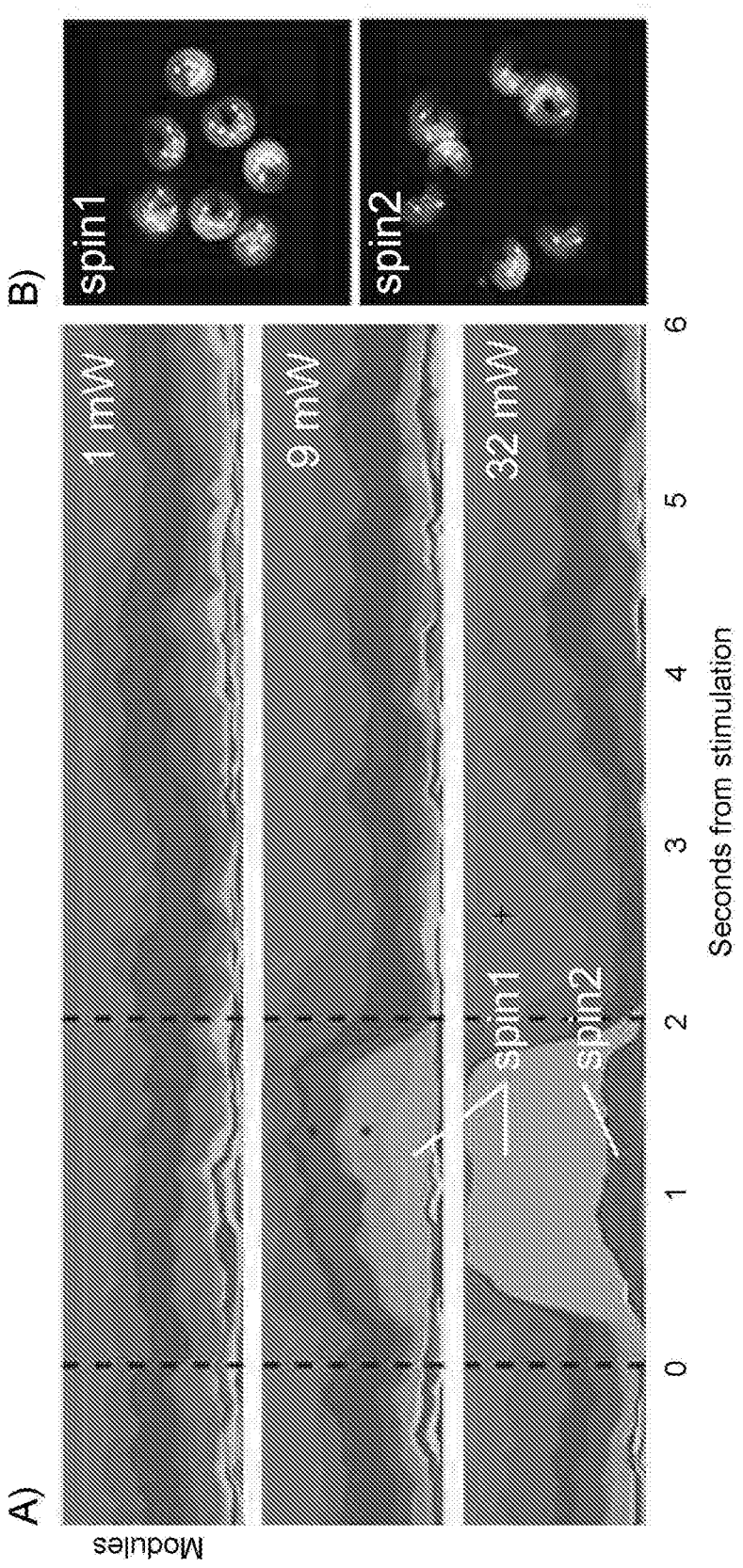

B)
spin1
spin2
A)
1 mW
9 mW
32 mW
spin1
spin2
Modules
0
1
2
3
4
5
6
Seconds from stimulation

FIG. 10A-10B

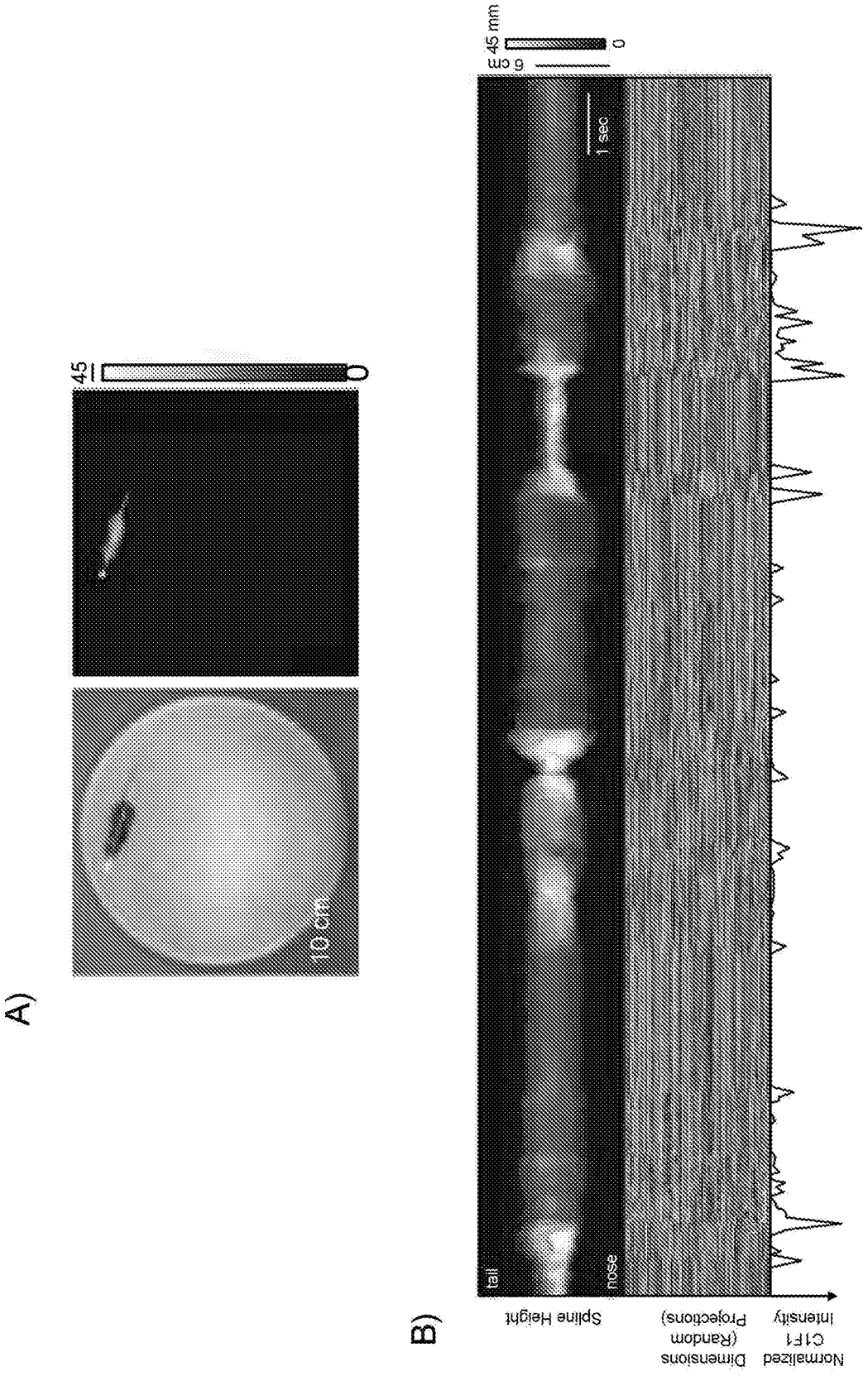

A)
45
0
10 cm
B)
6 cm
45 mm
0
1 sec
tail
nose
Spline Height
Dimensions (Random Projections)
Normalized CIF1 Intensity

FIG. 11A-11B

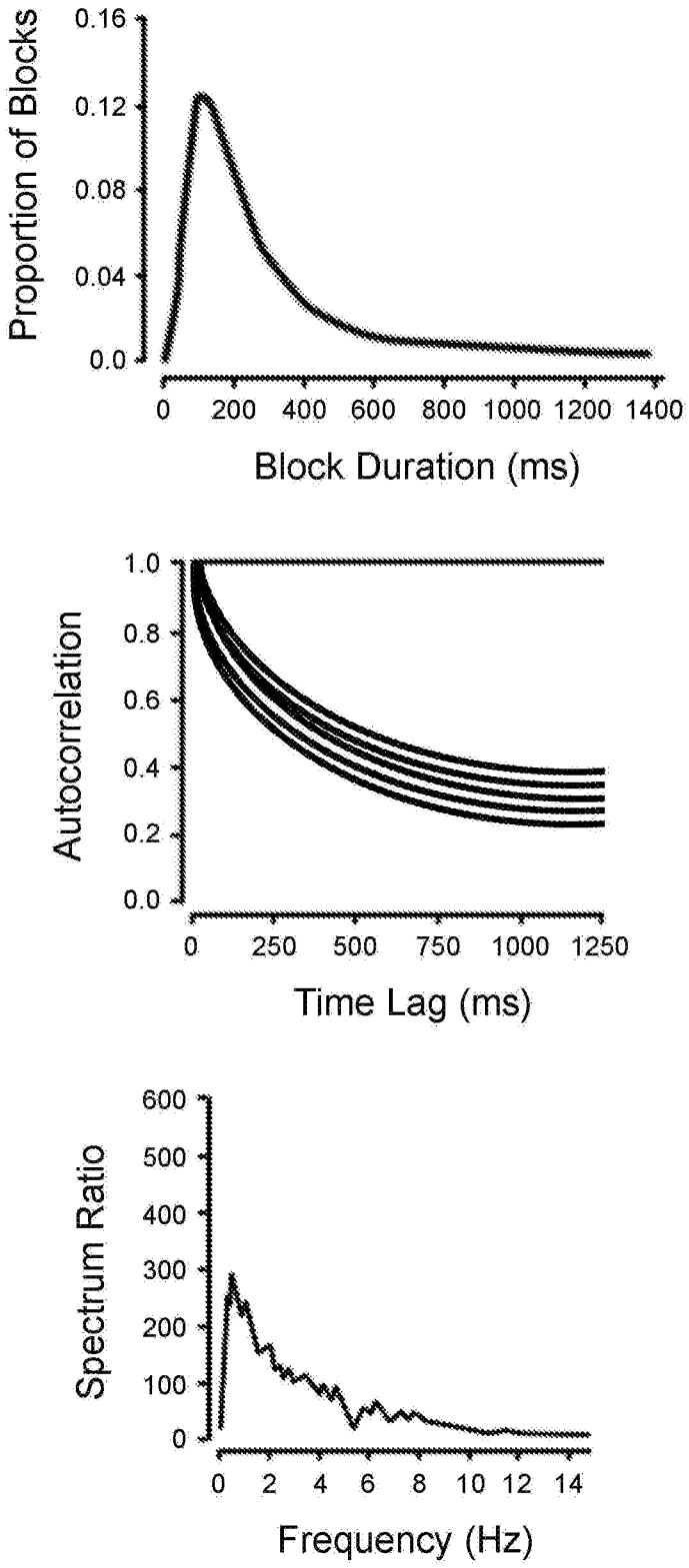

Proportion of Blocks
0.16
0.12
0.08
0.04
0.0
0 200 400 600 800 1000 1200 1400
Block Duration (ms)
Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
0 250 500 750 1000 1250
Time Lag (ms)
Spectrum Ratio
600
500
400
300
200
100
0
0 2 4 6 8 10 12 14
Frequency (Hz)

FIG. 11C

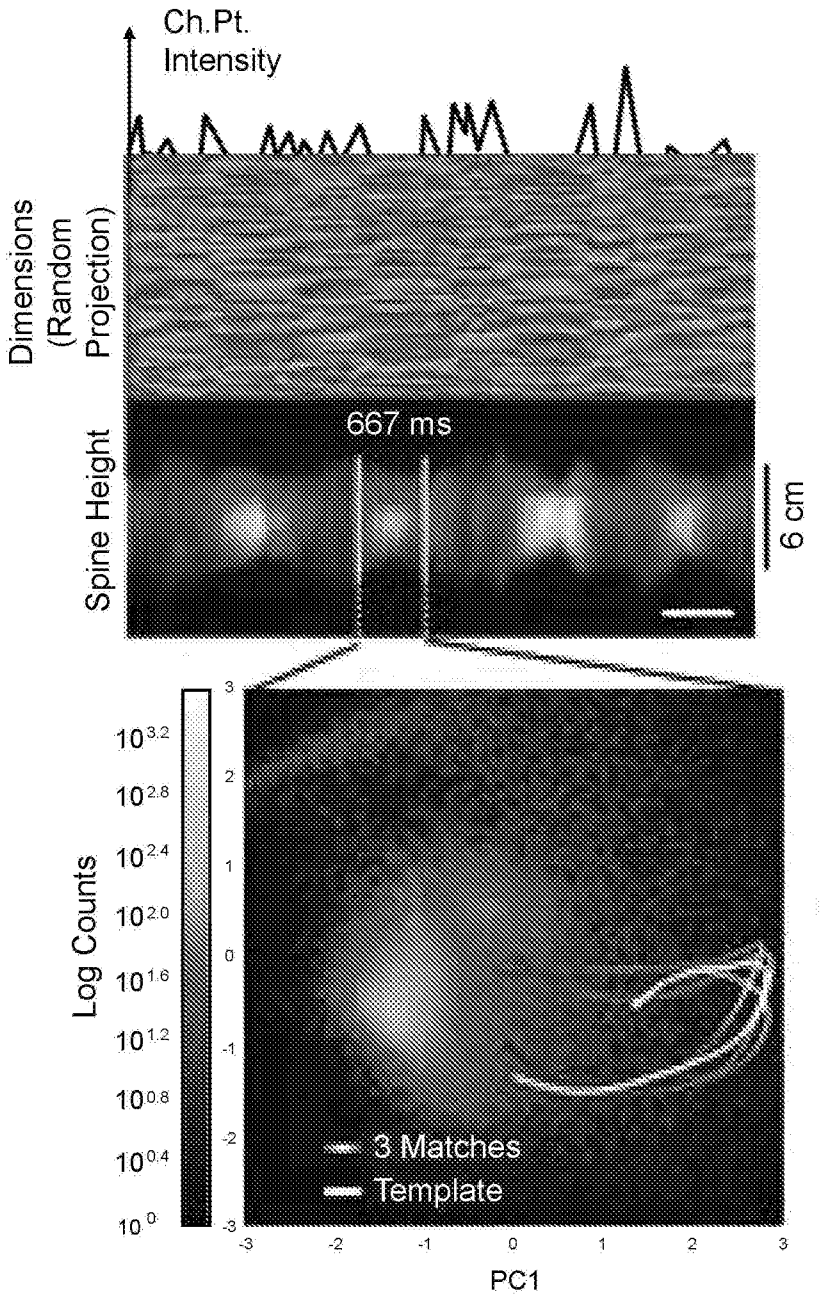

Ch.Pt.
Intensity
Dimensions
(Random
Projection)
667 ms
Spine Height
6 cm
Log Counts
$10^{3.2}$
$10^{2.8}$
$10^{2.4}$
$10^{2.0}$
$10^{1.6}$
$10^{1.2}$
$10^{0.8}$
$10^{0.4}$
10^{0}
3
2
1
0
-1
-2
-3
3 Matches
Template
-3
-2
-1
0
1
2
3
PC1

FIG. 12A

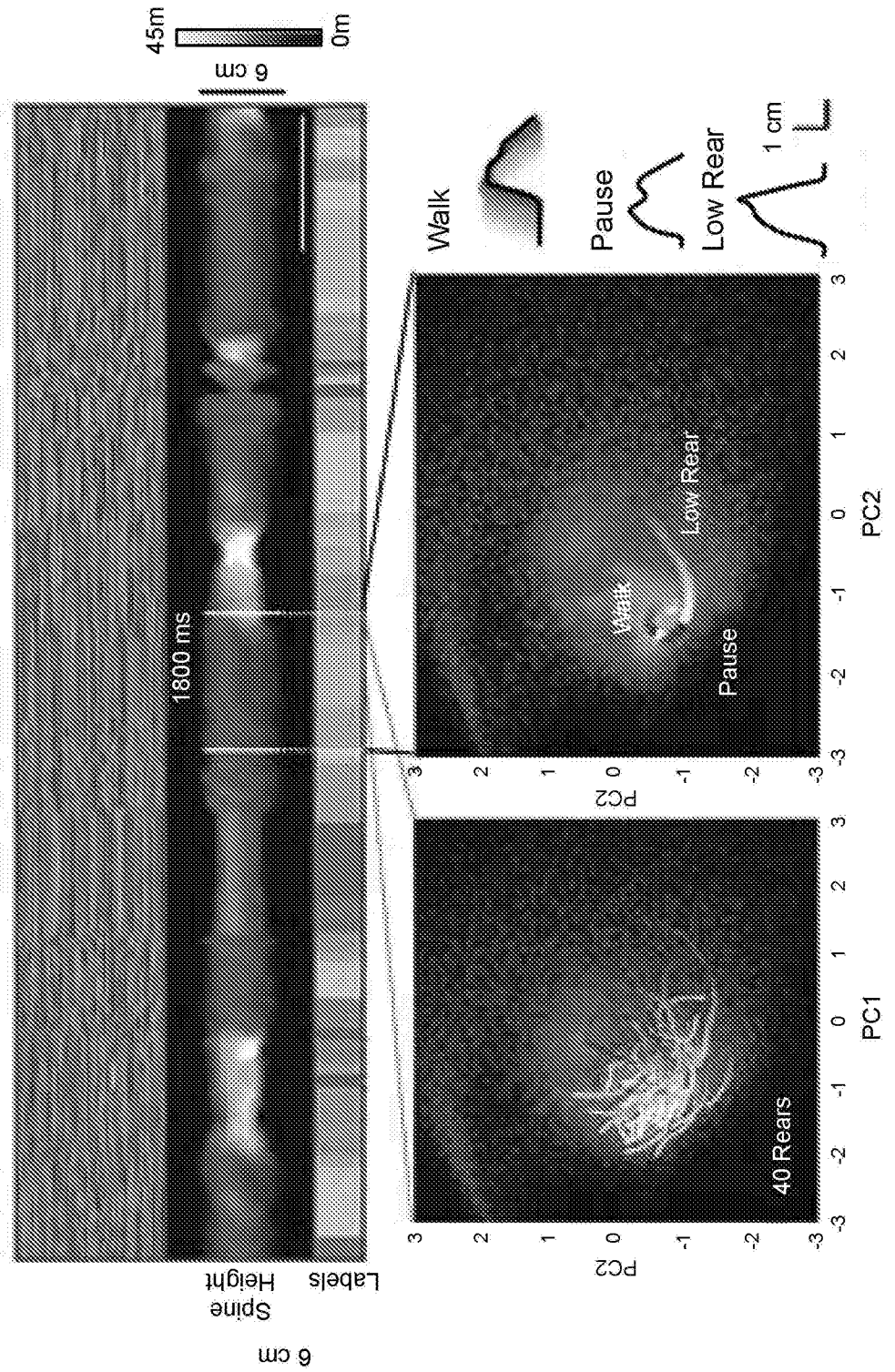

45m
0m
6 cm
1800 ms
Spine Height
Labels
6 cm
Walk
Pause
Low Rear
1 cm
40 Rears
PC1
PC2
Walk
Low Rear
Pause
PC2

FIG. 12B

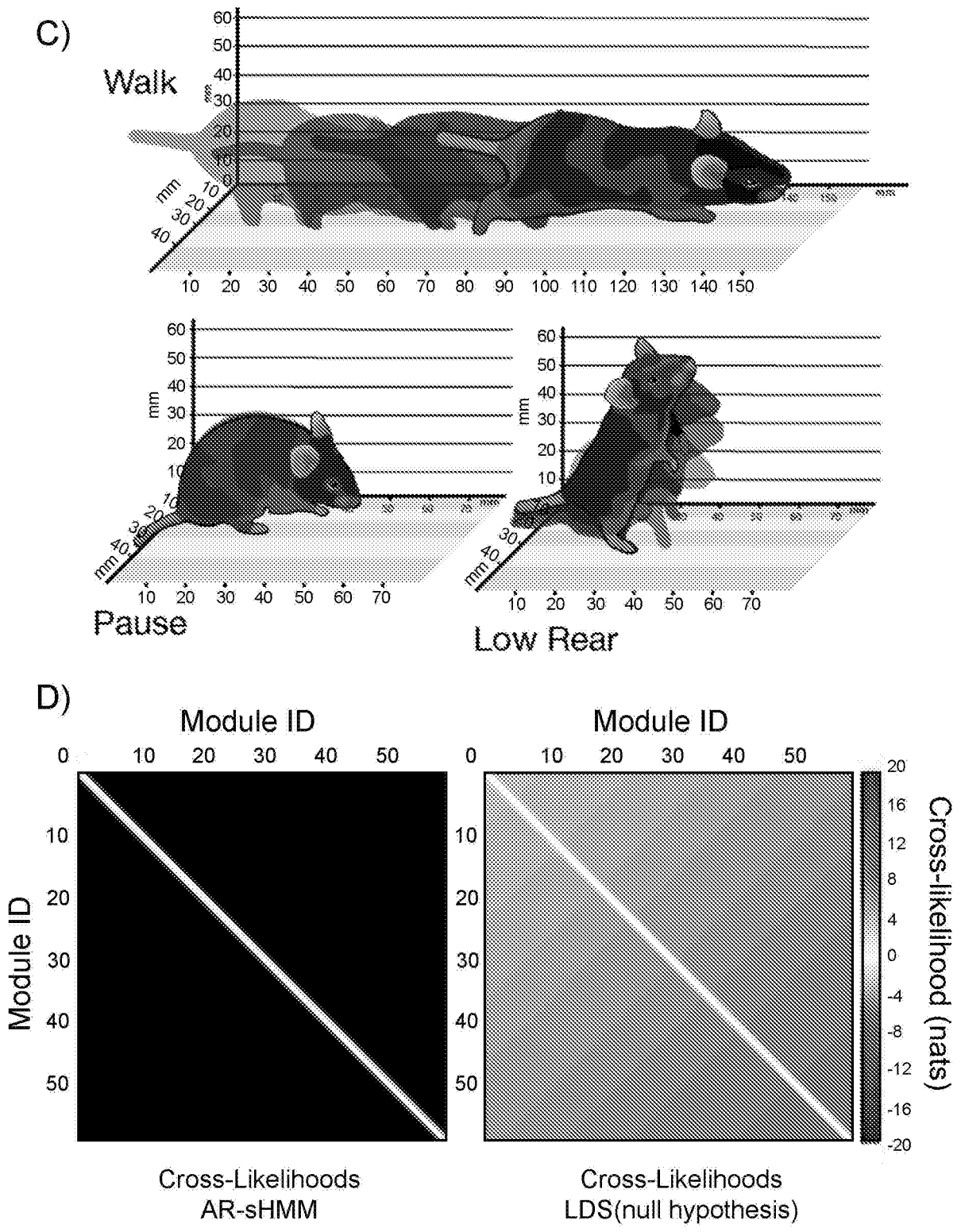

C)
Walk
Pause
Low Rear
mm
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
D)
Module ID
Module ID
Module ID
Cross-likelihood (nats)
Cross-Likelihoods
AR-sHMM
Cross-Likelihoods
LDS(null hypothesis)

FIG. 12C-12D

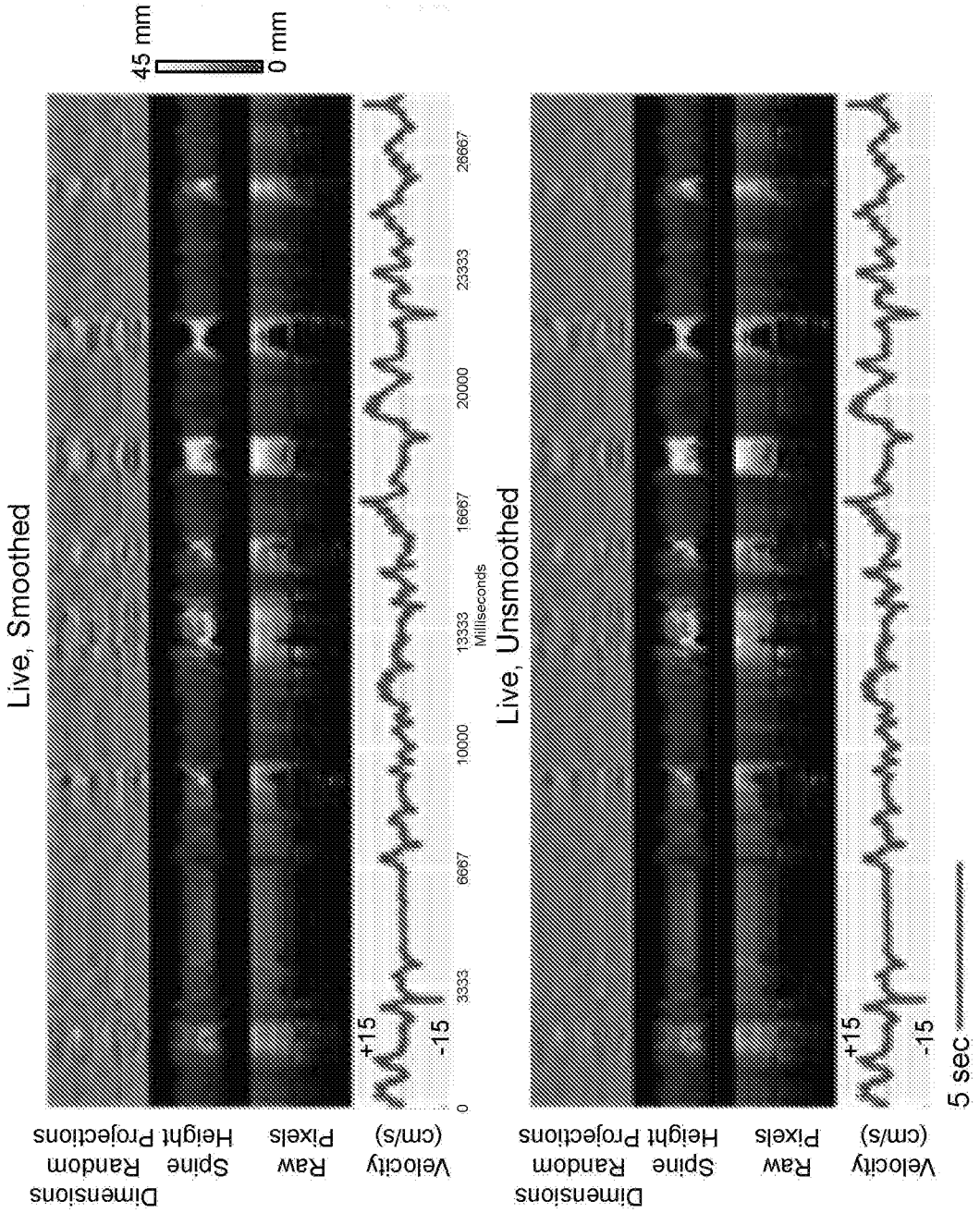

Live, Smoothed
45 mm
0 mm
Dimensions
Spine Random
Height Projections
Raw
Pixels
Velocity
(cm/s)
+15
-15
0
3333
6667
10000
13333
16667
20000
23333
26667
Milliseconds
Live, Unsmoothed
5 sec

FIG. 13Ai

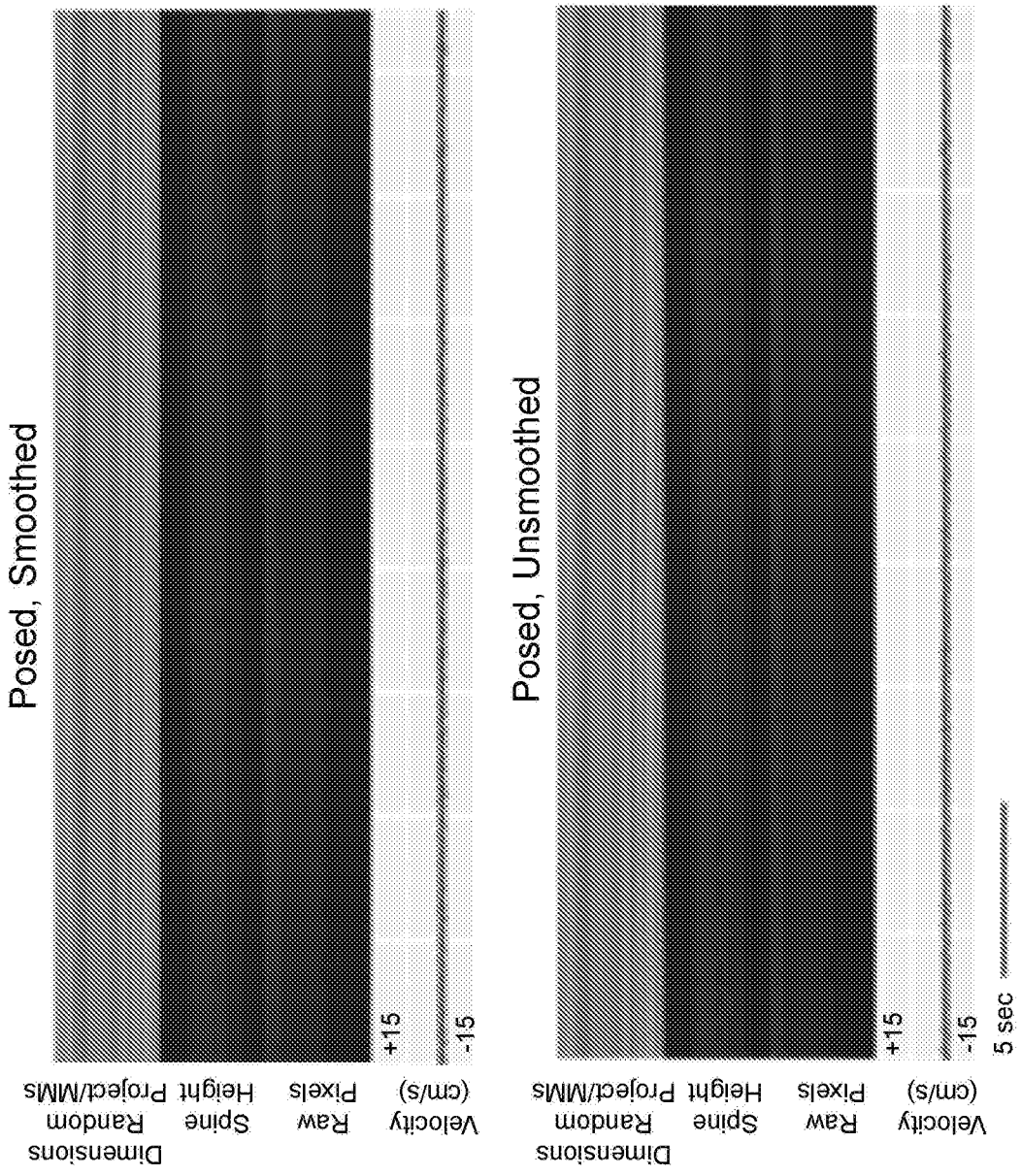

Posed, Smoothed
Posed, Unsmoothed
Velocity (cm/s)
Raw Pixels
Spine Height
Random Project/MMs
Dimensions
+15
-15
5 sec

FIG. 13Aii

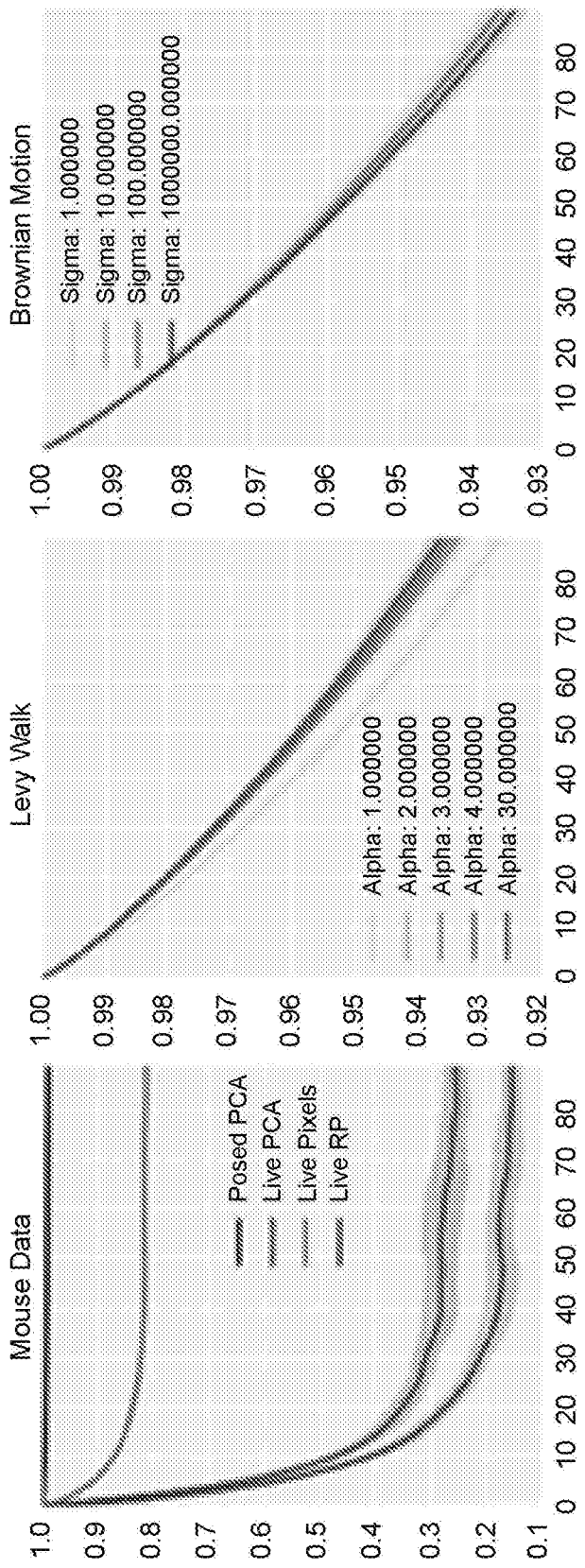

Mouse Data
Posed PCA
Live PCA
Live Pixels
Live RP
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0 10 20 30 40 50 60 70 80
Levy Walk
Alpha: 1.000000
Alpha: 2.000000
Alpha: 3.000000
Alpha: 4.000000
Alpha: 30.000000
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0.92
0 10 20 30 40 50 60 70 80
Brownian Motion
Sigma: 1.000000
Sigma: 10.000000
Sigma: 100.000000
Sigma: 1000000.000000
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0 10 20 30 40 50 60 70 80

FIG. 13B

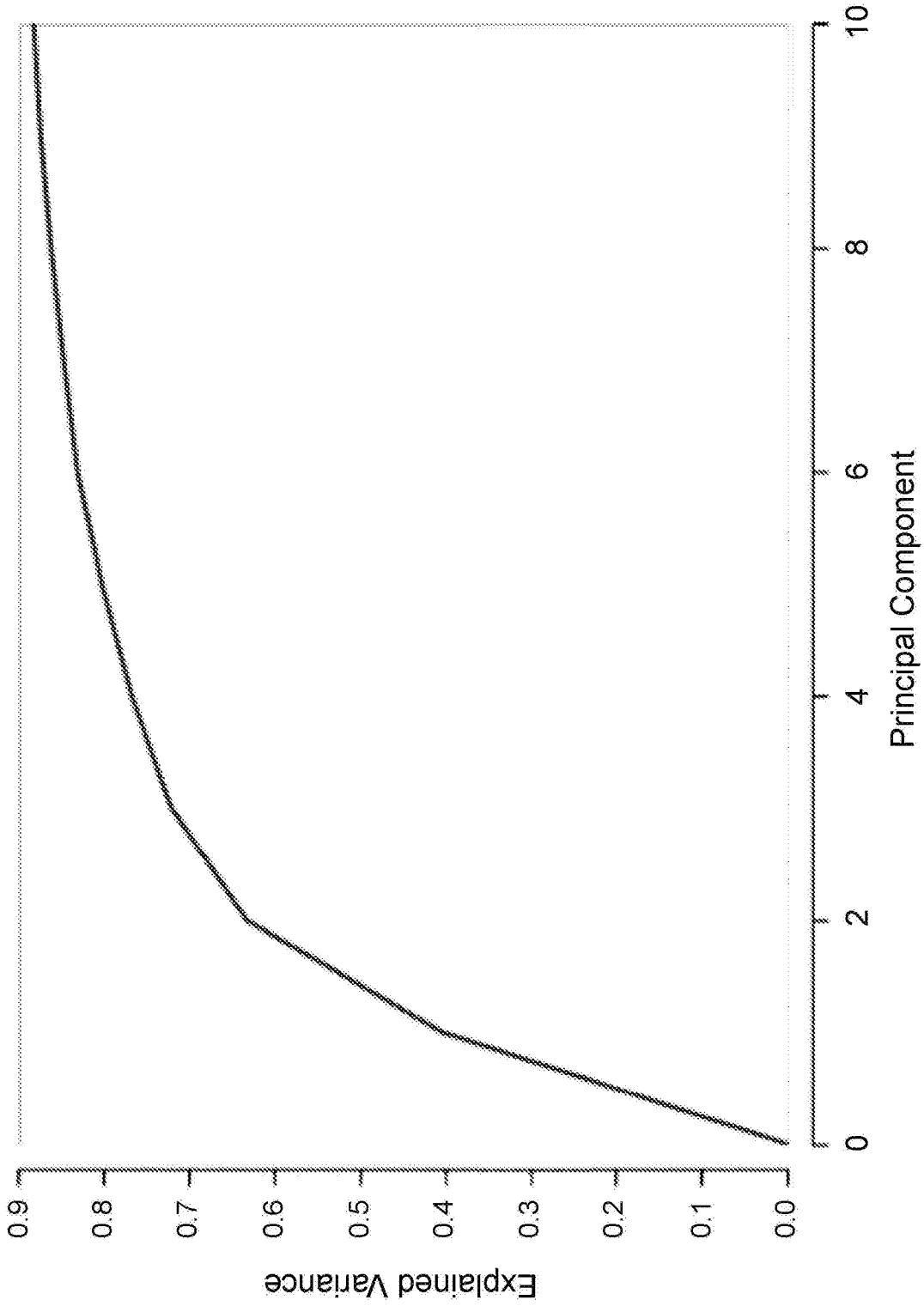
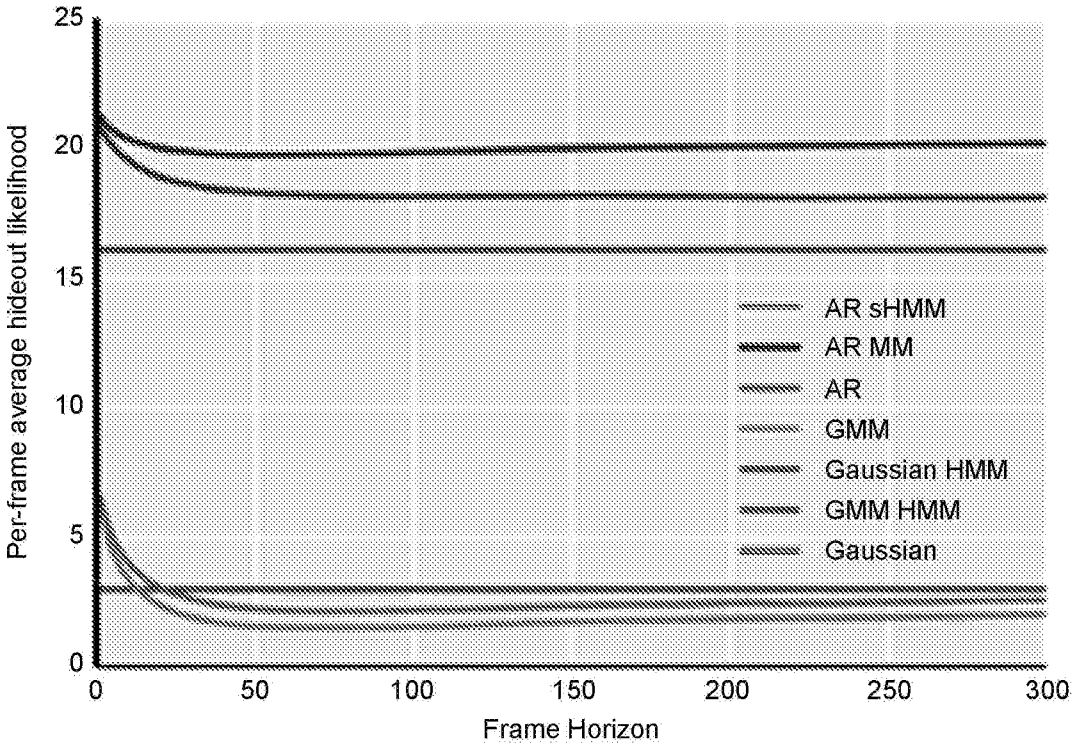

Explained Variance
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Principal Component
0
2
4
6
8
10

FIG. 14


Per-frame average hideout likelihood
25
20
15
10
5
0
0
50
100
150
200
250
300
Frame Horizon
AR sHMM
AR MM
AR
GMM
Gaussian HMM
GMM HMM
Gaussian

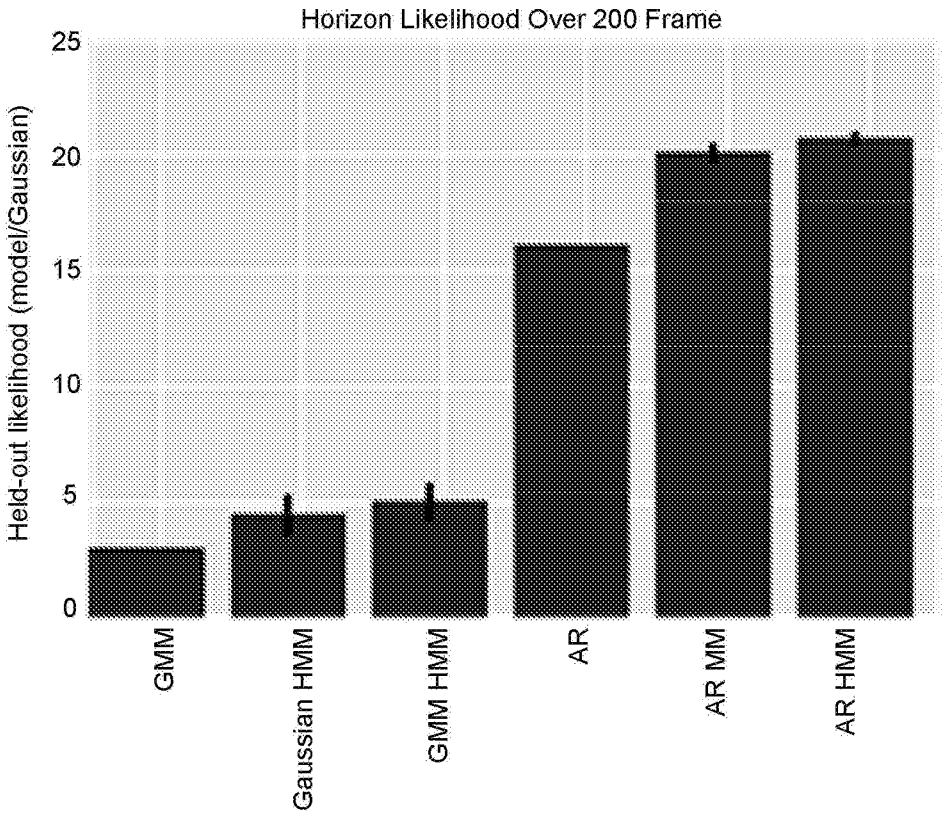

Horizon Likelihood Over 200 Frame
Held-out likelihood (model/Gaussian)
25
20
15
10
5
0
GMM
Gaussian HMM
GMM HMM
AR
AR MM
AR HMM

FIG. 15A

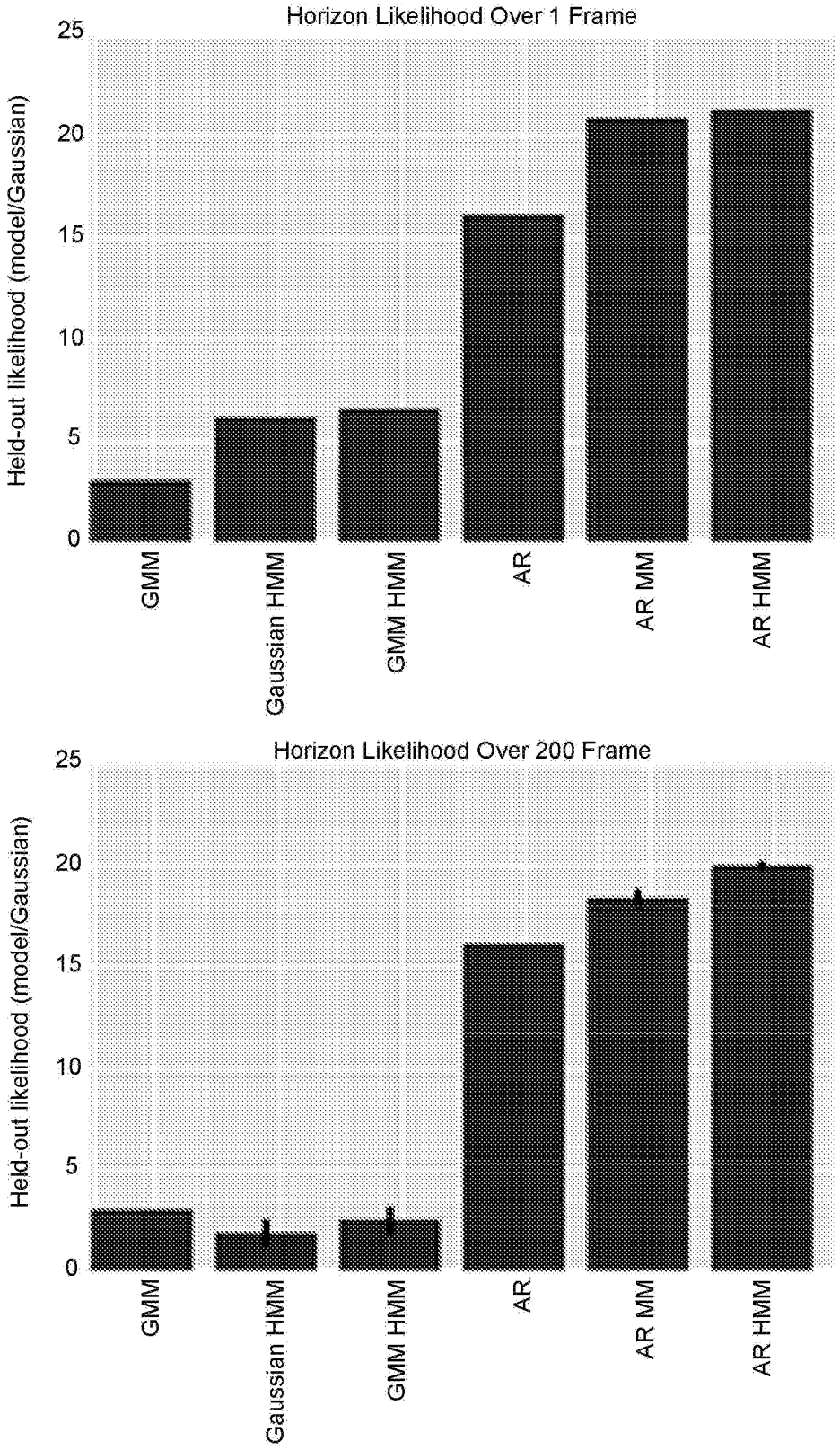

Horizon Likelihood Over 1 Frame
Held-out likelihood (model/Gaussian)
25
20
15
10
5
0
GMM
Gaussian HMM
GMM HMM
AR
AR MM
AR HMM
Horizon Likelihood Over 200 Frame
Held-out likelihood (model/Gaussian)
25
20
15
10
5
0
GMM
Gaussian HMM
GMM HMM
AR
AR MM
AR HMM

FIG. 15B

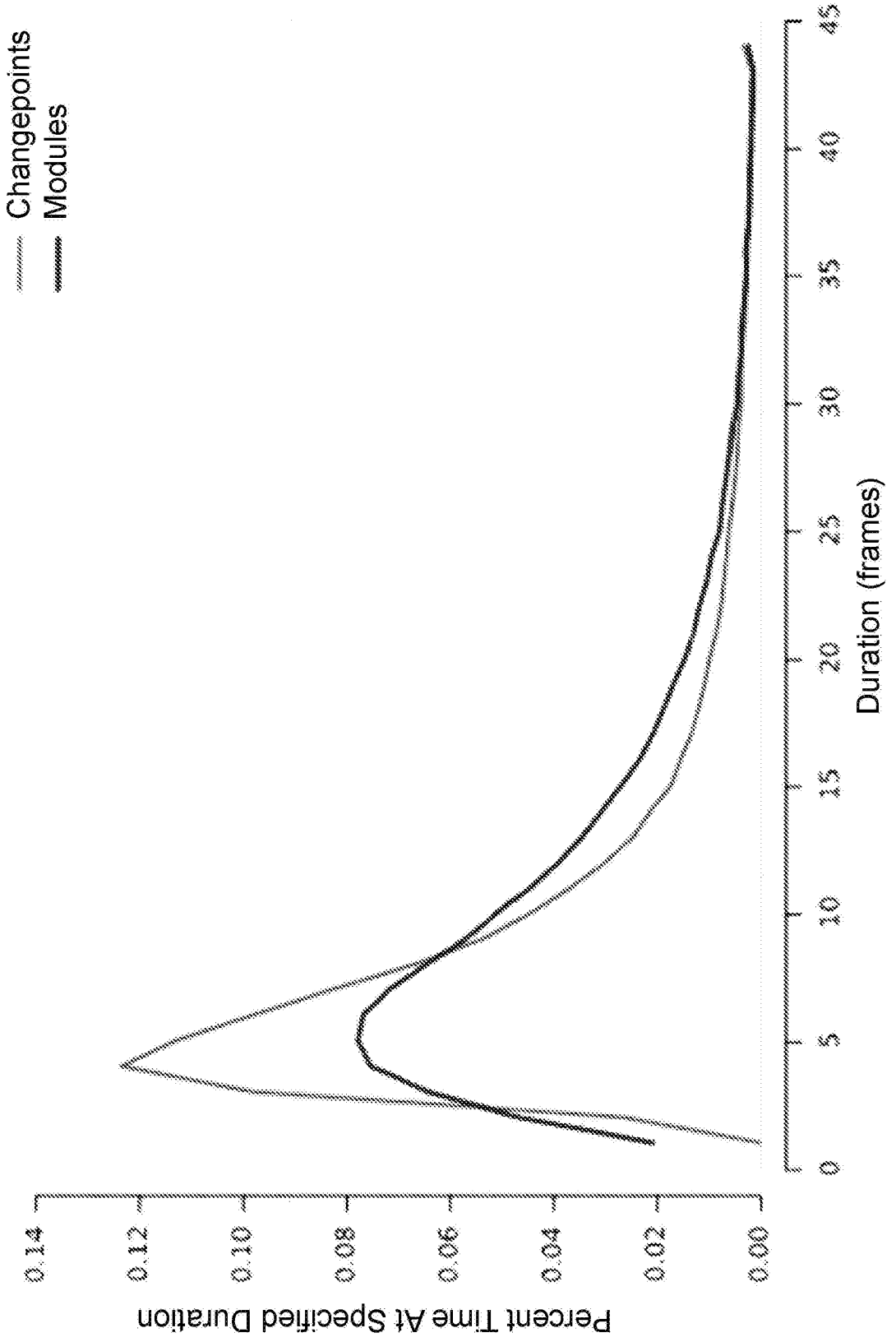

Changepoints
Modules
Percent Time At Specified Duration
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20
25
30
35
40
45
Duration (frames)

FIG. 16

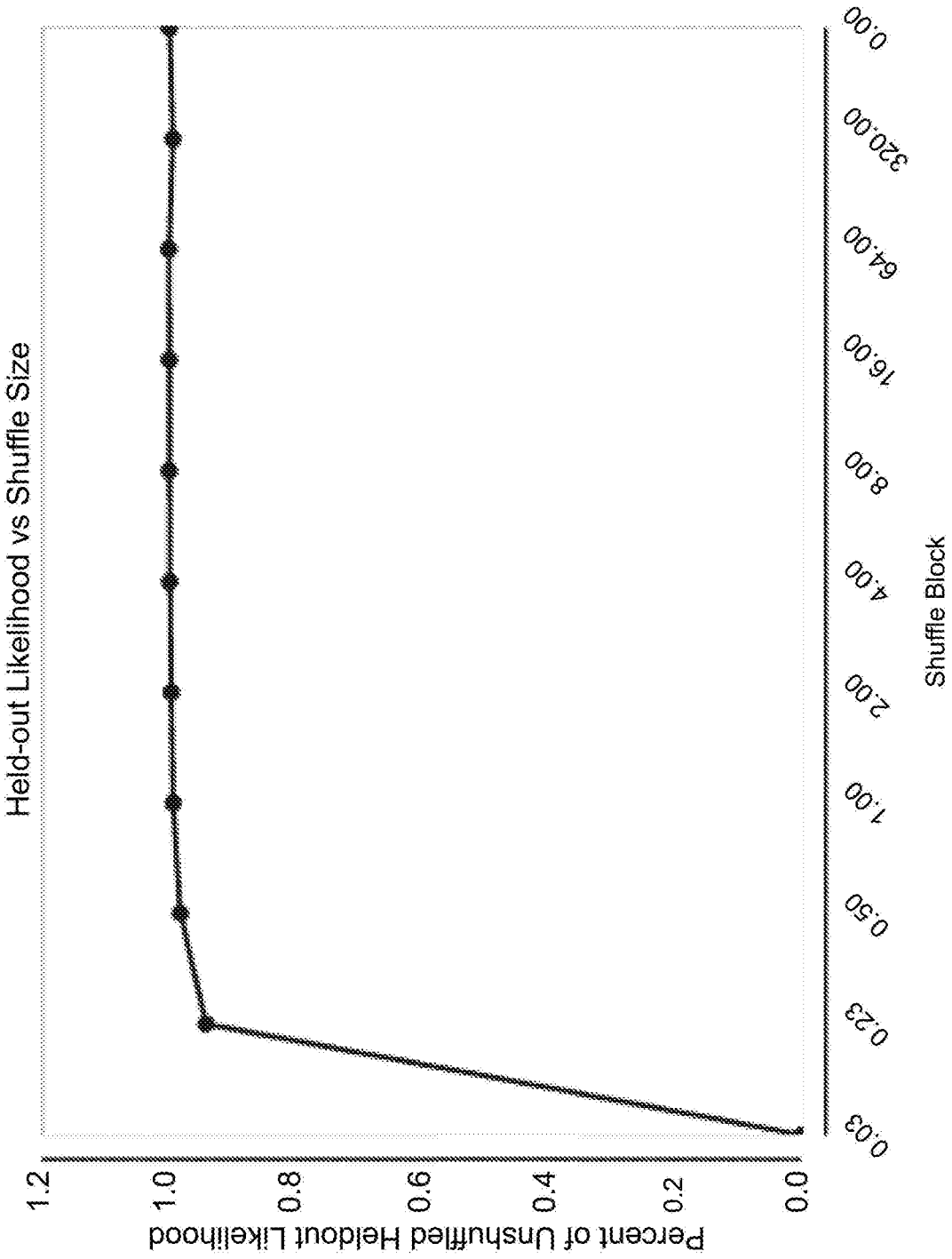

Held-out Likelihood vs Shuffle Size
Percent of Unshuffled Heldout Likelihood
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.03
0.23
0.50
1.00
2.00
4.00
8.00
16.00
64.00
320.00
0.00
Shuffle Block

FIG. 17

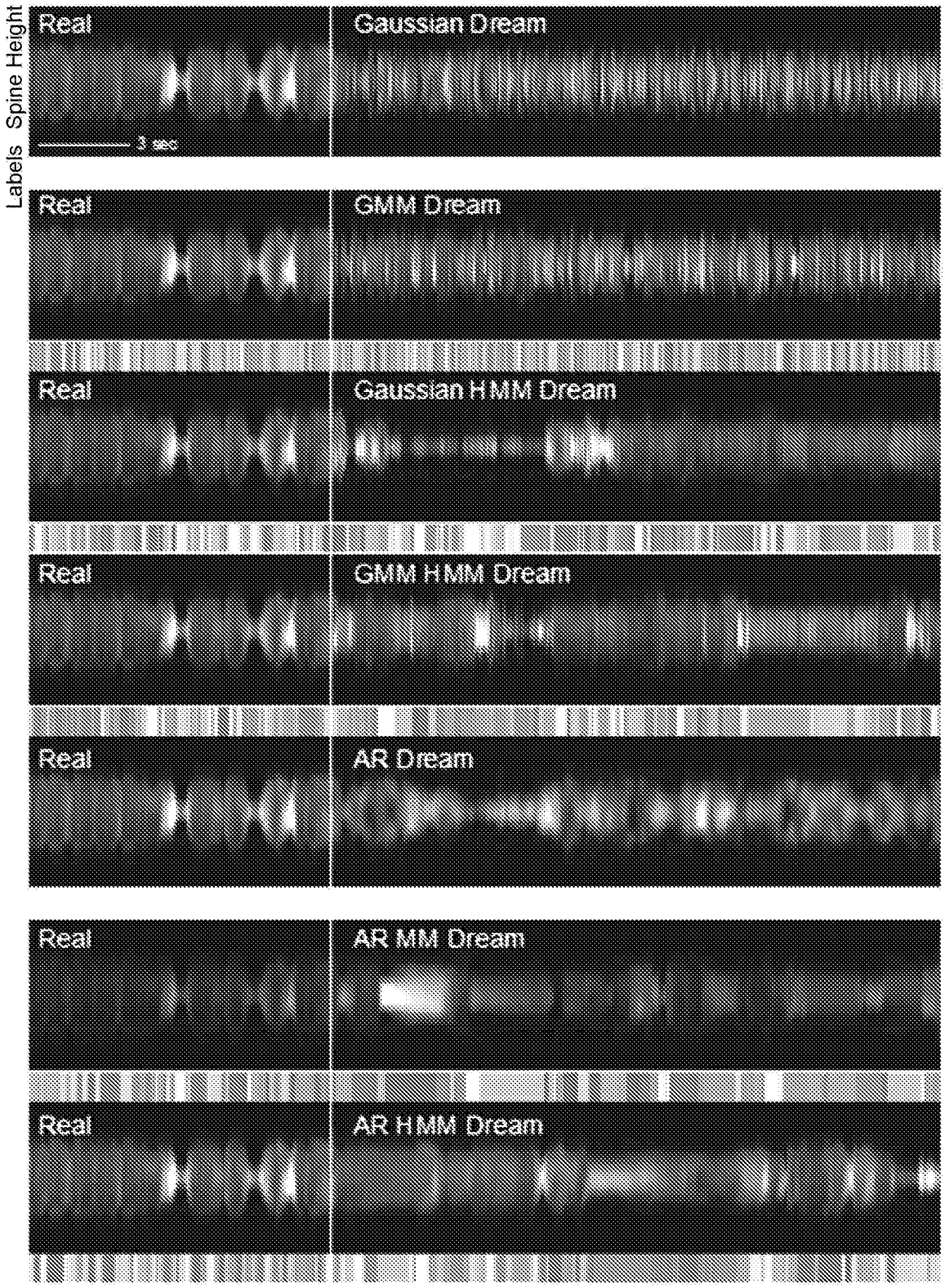

Spine Height
Labels
Real
Gaussian Dream
3 sec
Real
GMM Dream
Real
Gaussian HMM Dream
Real
GMM HMM Dream
Real
AR Dream
Real
AR MM Dream
Real
AR HMM Dream

FIG. 18

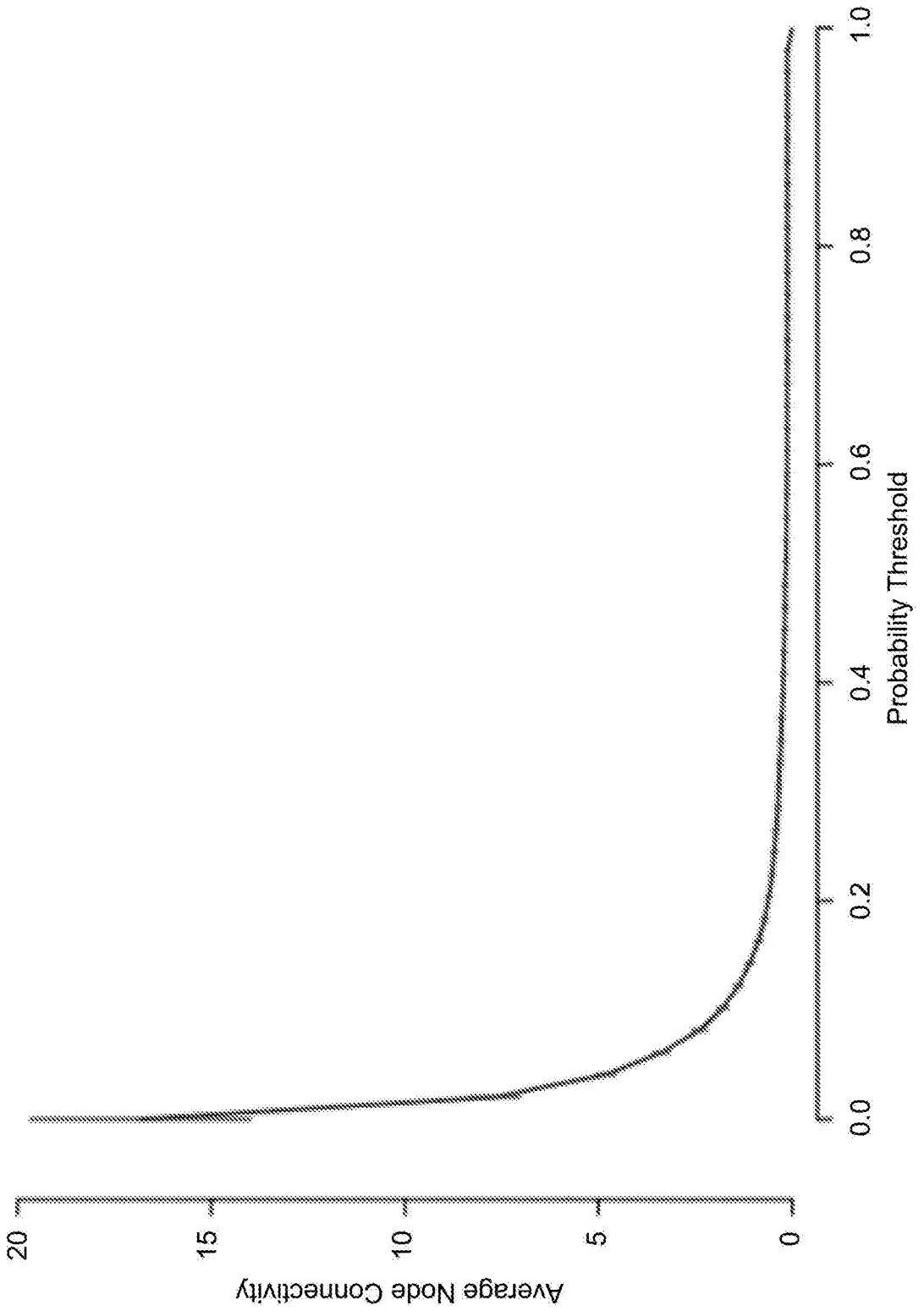

Average Node Connectivity
20
15
10
5
0
0.0
0.2
0.4
0.6
0.8
1.0
Probability Threshold

FIG. 19

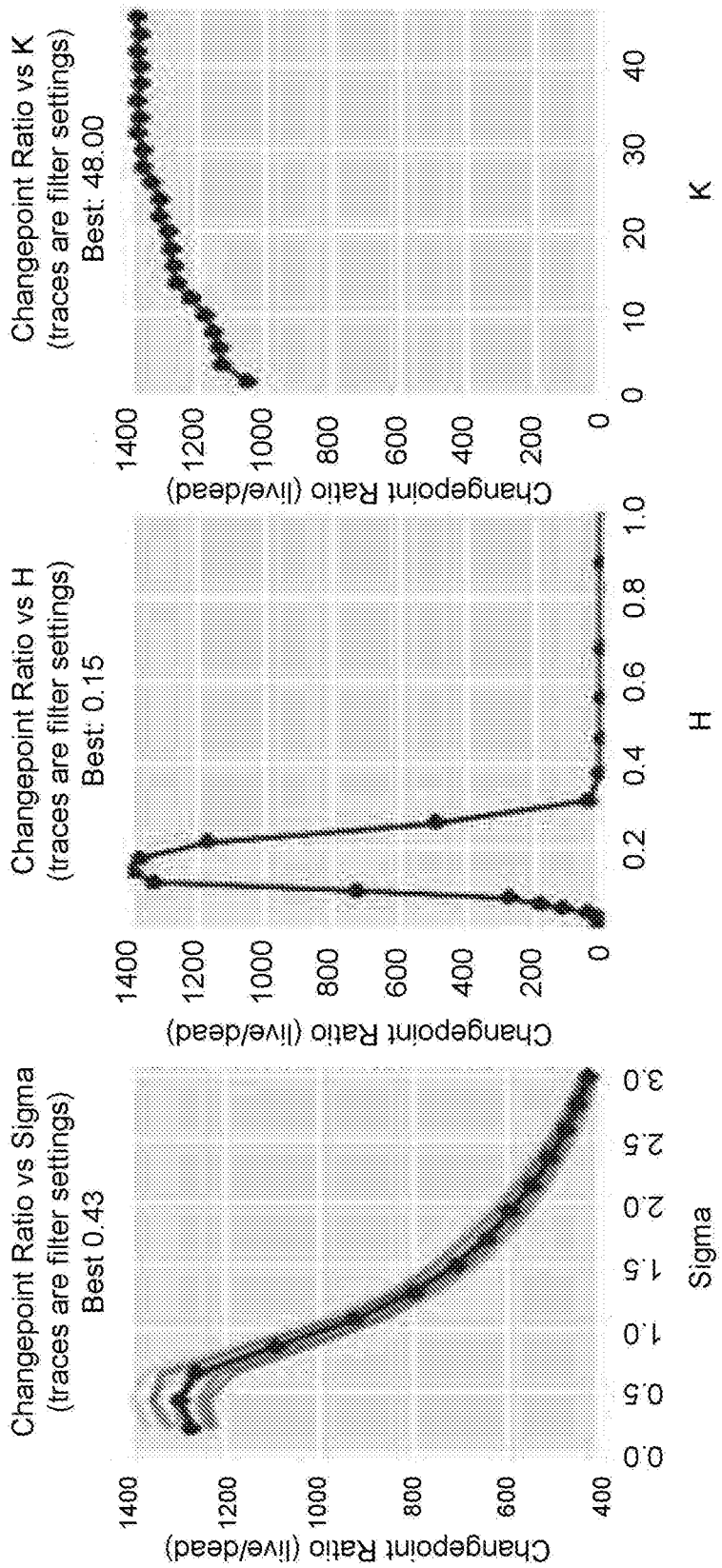

Changepoint Ratio vs Sigma
(traces are filter settings)
Best 0.43
Changepoint Ratio (live/dead)
1400
1200
1000
800
600
400
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Sigma
Changepoint Ratio vs H
(traces are filter settings)
Best: 0.15
Changepoint Ratio (live/dead)
1400
1200
1000
800
600
400
200
0
0.2
0.4
0.6
0.8
1.0
H
Changepoint Ratio vs K
(traces are filter settings)
Best: 48.00
Changepoint Ratio (live/dead)
1400
1200
1000
800
600
400
200
0
0
10
20
30
40
K

FIG. 20

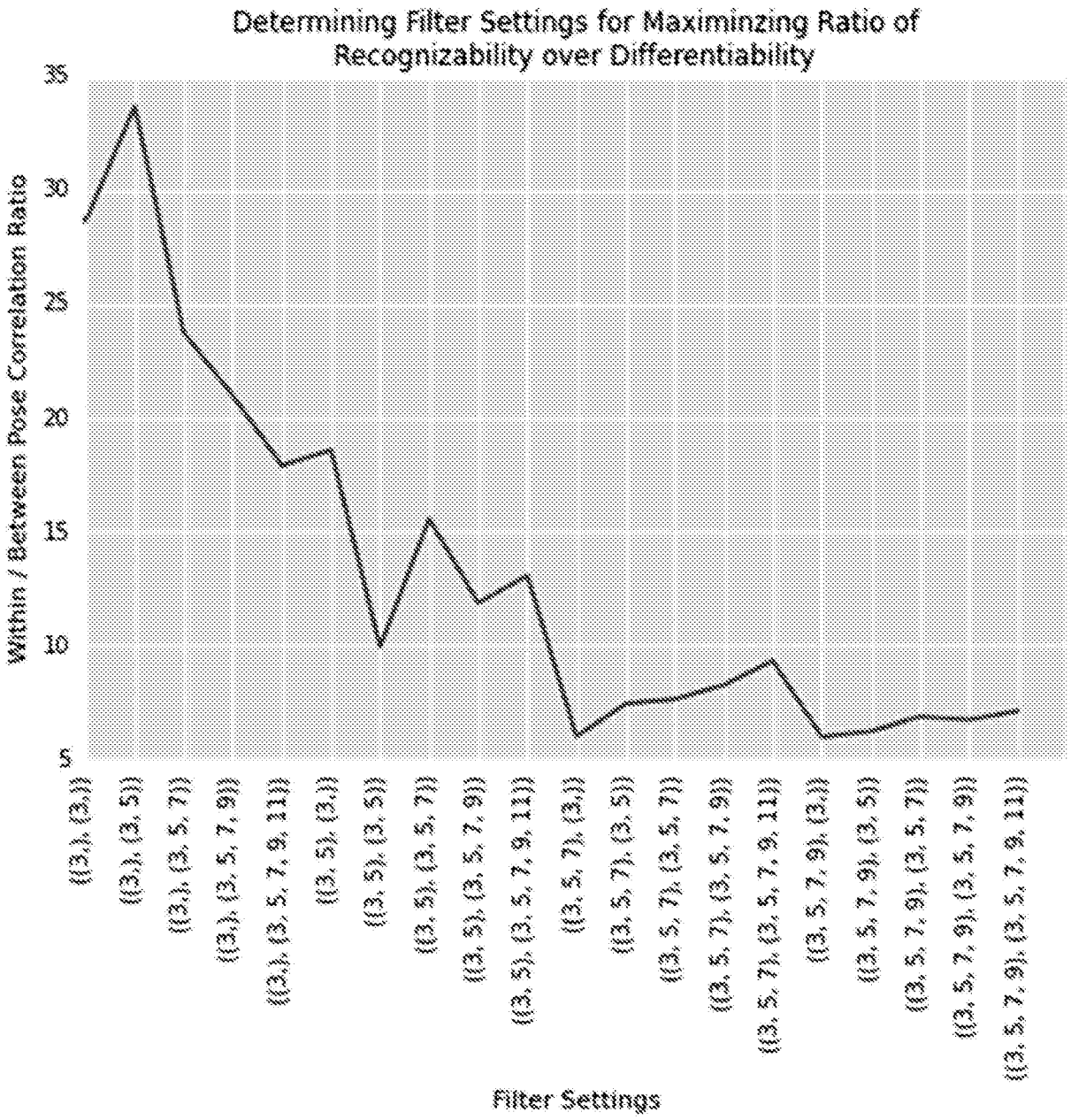

Determining Filter Settings for Maximinzing Ratio of Recognizability over Differentiability
Within / Between Pose Correlation Ratio
35
30
25
20
15
10
5
{{3,}, {3,}}
{{3,}, {3, 5}}
{{3,}, {3, 5, 7}}
{{3,}, {3, 5, 7, 9}}
{{3,}, {3, 5, 7, 9, 11}}
{{3, 5}, {3,}}
{{3, 5}, {3, 5}}
{{3, 5}, {3, 5, 7}}
{{3, 5}, {3, 5, 7, 9}}
{{3, 5}, {3, 5, 7, 9, 11}}
{{3, 5, 7}, {3,}}
{{3, 5, 7}, {3, 5}}
{{3, 5, 7}, {3, 5, 7}}
{{3, 5, 7}, {3, 5, 7, 9}}
{{3, 5, 7}, {3, 5, 7, 9, 11}}
{{3, 5, 7, 9}, {3,}}
{{3, 5, 7, 9}, {3, 5}}
{{3, 5, 7, 9}, {3, 5, 7}}
{{3, 5, 7, 9}, {3, 5, 7, 9}}
{{3, 5, 7, 9}, {3, 5, 7, 9, 11}}
Filter Settings

FIG. 21

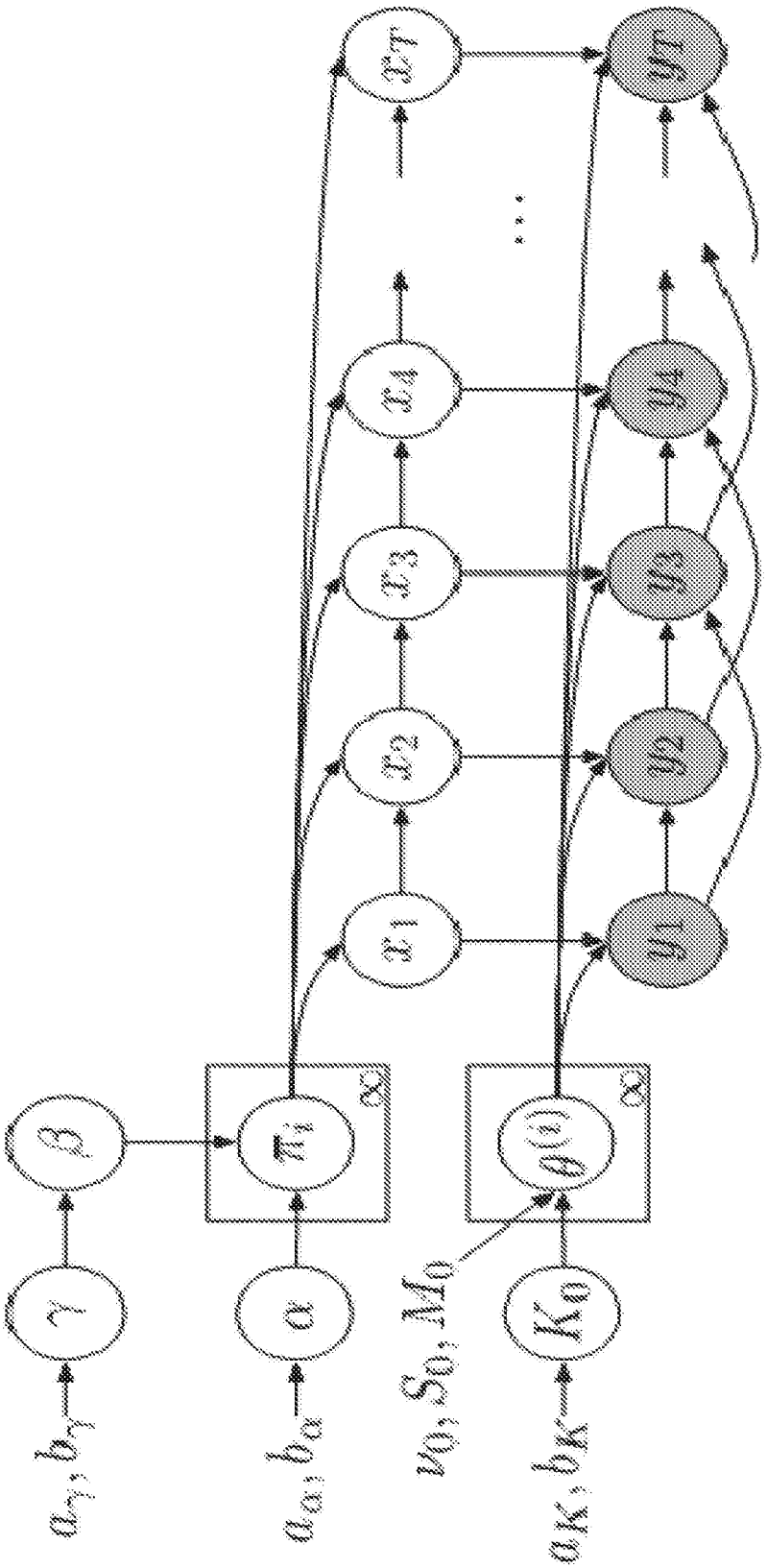

x_T
y_T
x_4
y_4
x_3
y_3
x_2
y_2
x_1
y_1
β
π_1
$\theta^{(1)}$
γ
α
K_0
a_γ, b_γ
a_α, b_α
v_0, S_0, M_0
a_K, b_K

FIG. 22

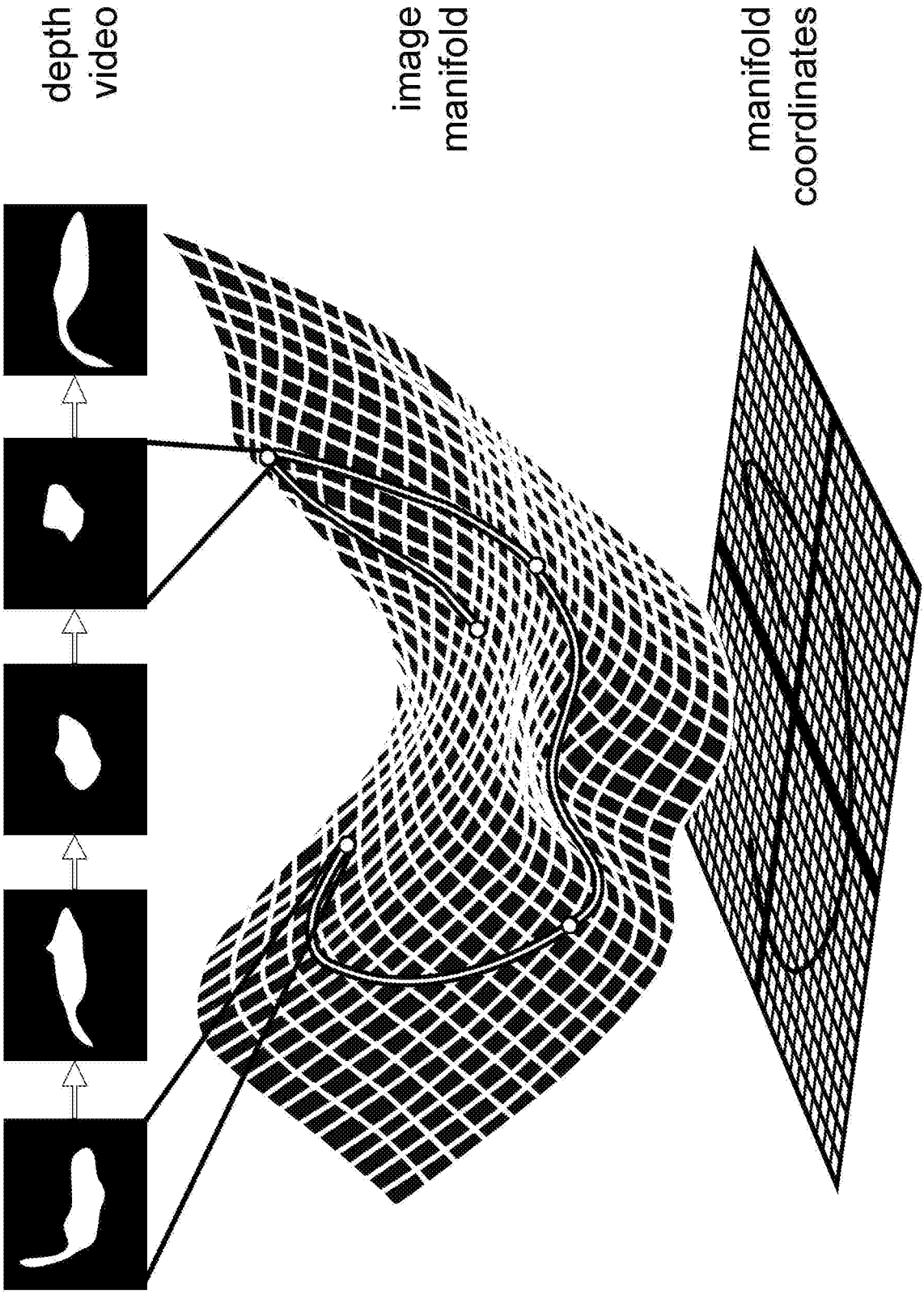
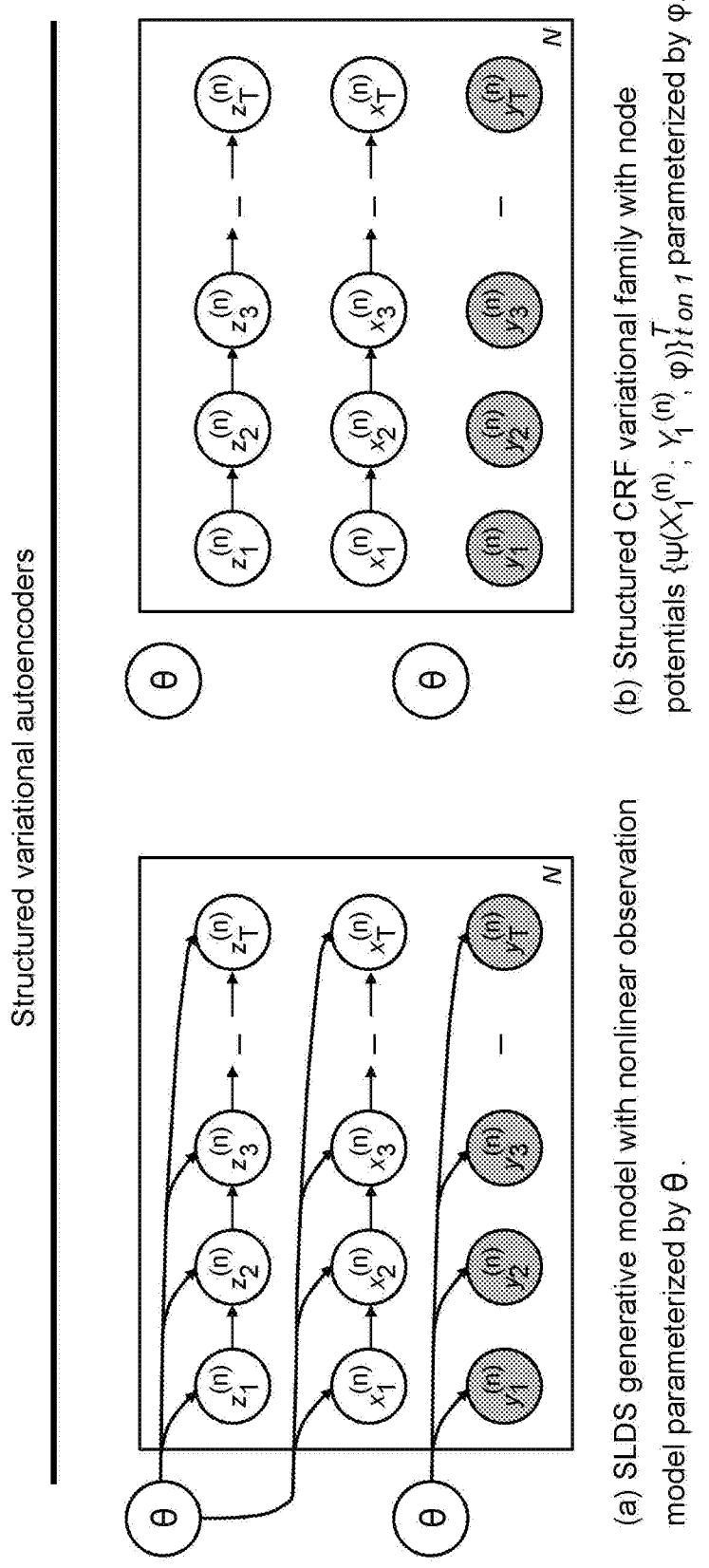


FIG. 23



(a) SLDS generative model with nonlinear observation model parameterized by θ.

(b) Structured CRF variational family with node potentials $\{\psi(X_1^{(n)}, Y_1^{(n)}, \varphi)\}_{t=1}^{T}$ parameterized by φ.

FIG. 24

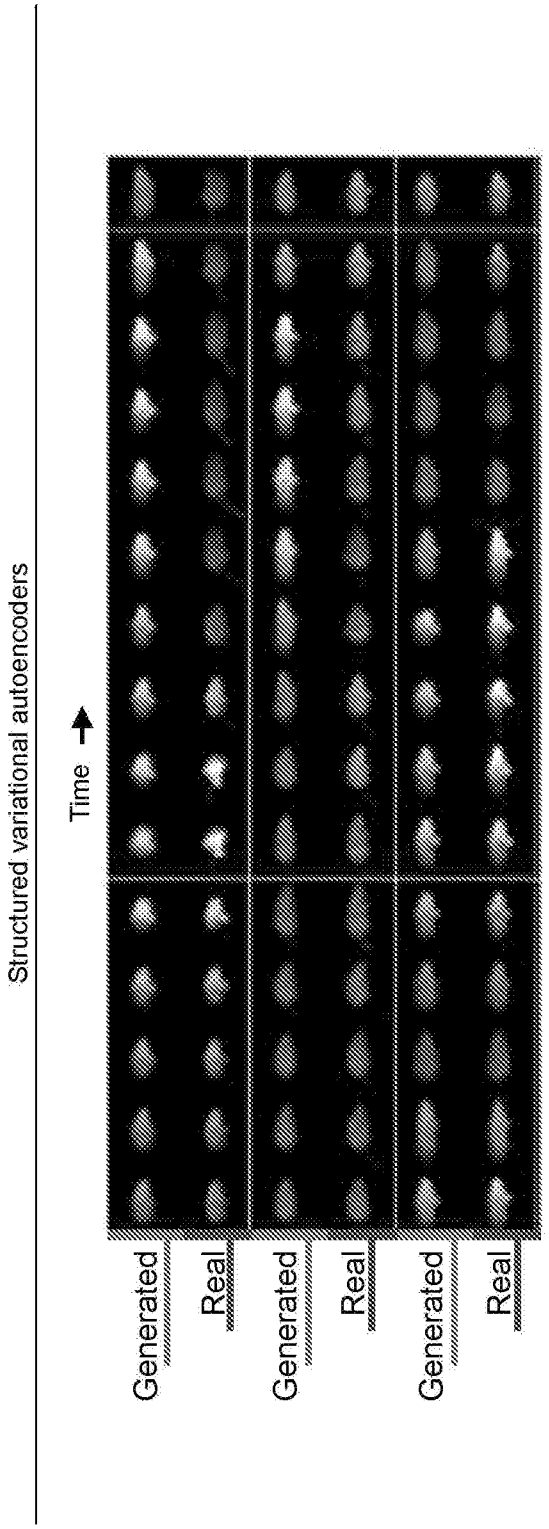


Figure 4. Generative video completion

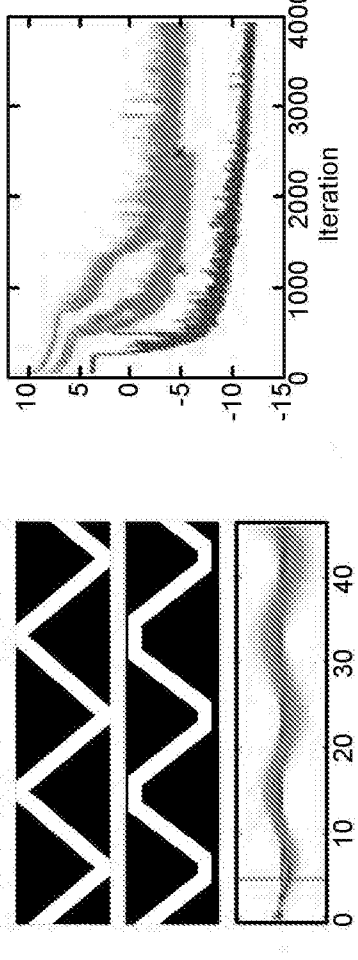


Figure 6. Comparison on the dot problem of natural gradient updates (blue) and standard gradient updates (orange).

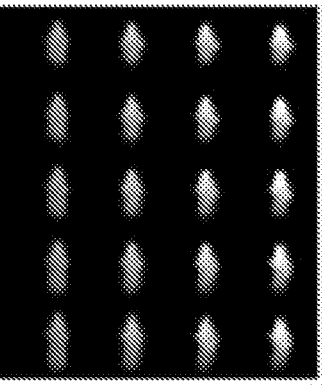


Figure 7. 2D grid in mouse image manifold coordinates.

Figure 5. Filtering and prediction for ID bouncing dot

FIG. 25

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/022781

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - A61B 5/11; G06T 7/00; G06T 7/20 (2017.01)
CPC - G06K 9/00335; A61B 1/00; A61B 5/00; A61B 5/1116; H04N 13/0203 (2017.02)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC - 382/110; 382/100 (keyword delimited)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2008/0152192 A1 (ZHU et al) 26 June 2008 (26.06.2008) entire document	1-5, 7
Y	US 2004/0141635 A1 (LIANG et al) 22 July 2004 (22.07.2004) entire document	1-5, 7
Y	US 2012/0056800 A1 (WILLIAMS et al) 08 March 2012 (08.05.2012) entire document	5
P,A	Johnson et al, "Composing graphical models with neural networks for structured representations and fast inference", 30th Conference on Neural Information Processing Systems NIPS 2016, 2016. Retrieved on 19 June 2017. Retrieved from the internet <URL: https://papers.nips.cc/paper/6379-composing-graphical-models-with-neural-networks-for-structured-representations-and-fast-inference.pdf> entire document	6
A	WO 2015/030689 A2 (ALTUN) 05 March 2015 (05.03.2015) entire document	1-7
A	WO 2013/170129 A1 (PRESIDENT AND FELLOWS OF HARVARD COLLEGE) 14 November 2013 (14.11.2013) entire document	1-7
A	US 2015/0146939 A1 (PRESIDENT AND FELLOWS OF HARVARD COLLEGE) 28 May 2015 (28.05.2015) entire document	1-7
A	US 2011/0032334 A1 (RAVEENDRAN et al) 10 February 2011 (10.02.2011) entire document	1-7
A	US 2012/0078097 A1 (WANG et al) 29 March 2012 (29.05.2012) entire document	1-7

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 June 2017	10 JUL 2017
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300	Authorized officer Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/022781

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☒ Claims Nos.: 8, 9
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
Claims 8, 9 are not drafted in accordance with Rule 6.2(a).

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:
See extra sheet(s).

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:
1-7

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/022781

Continued from Box No. III Observations where unity of invention is lacking

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fees must be paid.

Group I, claims 1-7, drawn to a method for analyzing the motion of a subject to separate it into modules.
Group II, claims 10-14, drawn to pre-processing three dimensional video data.

The inventions listed as Groups I-II do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons: the special technical feature of the Group I invention: assigning the at least one set of modules to a category that represents a type of animal behavior as claimed therein is not present in the invention of Group II. The special technical feature of the Group II invention: using a multi-layer perceptron (MLP) to output pose dynamics data, wherein the pose dynamics data represents a pose of the subject for each aligned frame through a three dimensional graphical space; processing the aligned frames to temporally segment the pose dynamics data into separate sets of sub-second modules wherein each of the sub-second module in a set of modules exhibits similar pose dynamics; and displaying a representation of each of the sets of modules that occur with a frequency above a threshold in the three dimensional video data as claimed therein is not present in the invention of Group I.

Groups I and II lack unity of invention because even though the inventions of these groups require the technical feature of a method for analyzing the motion of a subject to separate it into modules, the method comprising: processing three dimensional video data that represents the motion of the subject using a computational model to partition the video data into at least one set of modules, this technical feature is not a special technical feature as it does not make a contribution over the prior art.

Specifically, US 2011/0032334 A1 (RAVEENDRAN et al) 10 February 2011 (10.02.2011) teaches a method for analyzing the motion of a subject to separate it into modules, the method comprising: processing three dimensional video data that represents the motion of the subject using a computational model to partition the video data into at least one set of modules (Paras. 159-161).

Since none of the special technical features of the Group I or II inventions are found in more than one of the inventions, unity of invention is lacking.