Uncertainty Quantification and Structure Discovery for Scalable Behavior Science

by

David S. Hayden

B.S., Arizona State University (2011)
M.S., Massachusetts Institute of Technology (2014)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author . . . . . .
Department of Electrical Engineering and Computer Science
May 20, 2021

Certified by . . . . . .
John W. Fisher III
Senior Research Scientist of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Uncertainty Quantification and Structure Discovery for Scalable Behavior Science

by

David S. Hayden

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2021 in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

ABSTRACT

Scientific analysis of motion and social interaction can identify animal models of human disease by relating genetics or neural activity to behavior. However, experiments are often limited in scope because they require vast quantities of expert annotation on private data. Attempts to automate aspects of behavior science typically have limited interpretability and lack uncertainty representation. Errors will go unrecognized without manual inspection and propagate to hypothesis tests, corrupting conclusions. In response, this dissertation develops principled Bayesian approaches to low-level behavior analysis that discover the articulated part structure of a moving object and quantify uncertainty in the motion of multiple objects. Uncertainty is used to identify possible errors and automatically schedule sparse annotations. We apply parts modeling and tracking to primate behavior data in experimental and observational settings, in one case contributing to the first evidence supporting the use of primate animal models in autism research. We additionally develop Marmoset100, a 100-hour RGB-Depth dataset of pairwise primate social interactions labeled with 25 high-level behaviors, and show that uncertainty representation in tracking estimates improves behavior classification.

The Nonparametric Parts Model (NPP) discovers structure by learning articulated parts decompositions in an unsupervised fashion by briefly observing objects moving in an image, depth, point cloud, or mesh sequences. NPP combines distributions on Lie groups with a Bayesian nonparametric prior to perform joint reasoning over an interpretable state-space model with nonlinear dynamics and state-dependent observation noise. In developing sampling-based inference for NPP, we discover a novel and efficient Gibbs decomposition for prior distributions on $\mathrm{SE}(D)$, the manifold of rigid transformations. We show that NPP learns intuitive part segmentations for diverse objects and enables both analysis and synthesis of relative part motion in the body frame.

The Joint Posterior Tracker (JPT) is a comprehensive Bayesian treatment of the general multi-object tracking problem that quantifies uncertainty in the motion of multiple objects. JPT uniquely performs asymptotically exact inference without gating heuristics or the combinatorial costs of exponential and factorial complexity. We develop novel Metropolis-Hastings proposals that reason over permutations of the latent space and enable efficient posterior mode hopping that correspond to possible confusion events. We show that JPT yields accurate uncertainty representation of data associations with high performance on standard metrics. Finally, we use posterior uncertainty to identify ambiguities in observed data and automatically schedule sparse human annotations that rapidly improve posterior estimates and reduce uncertainty.

Thesis Supervisor: John W. Fisher III
Title: Senior Research Scientist of Electrical Engineering and Computer Science

Acknowledgments

I thank my advisor, John, for seven years of vigorous technical debate, during which time I learned to appreciate what is quantifiable may not be immediately valuable, and what is valuable may not be immediately quantifiable. In particular, I came to the group eager to produce fast runtimes and high performance on existing benchmarks but, over time, found that good ideas prescribe evaluation (and not vice versa). The marathon 3–6 hour heated whiteboard sessions were memorable, and exactly what I sought from my advisor and MIT. Thank you also for actively cultivating a lab culture that was playful, communicative, open, and professional.

I commemorate my former and late advisor, Seth Teller, who enthusiastically supported my early work in wearable social interaction. Seth: your passion is missed by your students, community, and family but continues on in them. Special thanks goes to former 33x members including Sudeep Pillai, William Li, Matthew Walter, Stefanie Tellex, Javier Velez, Finale Doshi-Valez, and Bryt Bradley for making my early time at MIT a welcoming one. I would also pay tribute to Ross Finman and Sachi Hemachandra had they not been too busy deliberating amongst themselves. ☺

I thank my committee members Bob Desimone and Justin Solomon and additional biology collaborators Rogier Landman, Guoping Feng, Will Menegas, and Charles Jennings. Rogier: our many conversations over coffee and our repeated trials to balance variation in the data with robustness in our methods were enlightening. Bridging two worlds is never easy but we made real progress towards interpretable, principled automation in behavior science. Bob and Guoping: thank you for your time, patience, counsel, and insight. Justin: thank you for your insights on the Heat equation and HMC on Lie groups.

I also thank collaborators, labmates, and friends, especially Sue Zheng and Jason Pacheco, but also Chris Dean, Genevieve Flaspohler, Randi Cabezas, Julian Straub, and Guy Rosman for humoring my lunchtime conversation topics, tolerating my birthday questions, and generally contributing towards our joint enrichment. Jason: you may replace yearly birthday wisdom with another million-dollar scheme. Sue: oatmeal will always be a viable lunch, even when consumed daily. I also give special thanks to Fred Kjolstad, Mohammad Ghassemi, and Ross Finman for many compelling adventures, hypotheticals, and struggles.

The MIT, Harvard, and Cambridge communities are incredibly precious. I deeply value that more than 50 people contributed to Social Reading over the years by waking up on Saturday mornings to read outside their field of expertise at the Harvard Coop and share their learning at Border Cafe. May Social Reading outlive its original members, among them Colleen Silva-Hayden, Anna Aldric, Fred Kjolstad, Sebastian Claici, Catherine Le, Ed Chien, Leigh Martin, Lennart Husvogt, Trevor Campbell, Jaho King, Angela Xu, Carl Vondrick, Shraya Sharma, Ellen DeGennaro, Zoya and Alexei Bylinskii, Asad Lodhia, Xi Yin, Katya Anderson, Cole Franks, Violetta Medik, Genevieve Flaspohler, Nick Kalweit, Oscar Moll, Neta Batscha, and Brandon Morgan.

Crucially, I want to thank my wife, Colleen, for her constant encouragement and support. Pushing me to push my ideas to their conclusions and pursuing your own passions in education and Latin America continue to make me a better husband, researcher, and person. Bringing a sous vide steak and blowtorch to lab and throwing a birthday bash with a pool full of watermelons also help! Finally, I thank my mom, dad, step-dad, and brother for setting the example of doing what you love and maintaining what's important to you, regardless of practicality.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

"To reject one paradigm without simultaneously substituting another is to reject science itself."

— Thomas Kuhn

This dissertation develops Bayesian methods and theory that, although general, are specially motivated by scientific workflows where humans collaborate with probabilistic models in the principled collection and analysis of data at scale. Three themes are emphasized throughout:

1. Interpretable latent representations to support iterative model refinement and principled follow-on analysis,
2. Representation of uncertainty to automatically identify ambiguities in observed data and correct possible errors in inference,
3. Ability to identify structure that may not be known in advance, to support flexible modeling and knowledge discovery.

These themes are relevant to human-machine collaborations in any science, but this work focus on behavior science applied to primates. Primate behavior is difficult to observe relative to other species because their movements are rapid and varied. They include brachiation, climbing, and jumping, often behind partial or total occlusion in fully-3D environments (compared to e.g., mice, whose movements can often be effectively represented in 2D). Primate behavior is of special interest because their biology and social organization more closely match those of humans as compared to many other animals [18, 100, 20, 180]. In what follows, challenges and opportunities in behavior science are outlined (1.1), as are the above themes: interpretability (1.2), uncertainty representation (1.3), and discovery of structure (1.4). Contributions are then summarized and the organization of this dissertation outlined (1.5).

Thesis statement:
Principled, scientific analysis of behavior needs probabilistic models that reason over interpretable representations, explicitly quantify uncertainty, and are capable of discovering novel structure.

1.1 Scientific Analysis of Behavior

The controlled study of behavior in humans, animals or microorganisms examines relationships between genes, neural activity, and lifetime development [194]. It commonly involves scientists observing hundreds of hours of data (often video), annotating events on a clipboard as they occur. This process limits the scope and scalability of

behavioral studies because annotations are time-consuming, cannot be done in realtime or at all times, and requires expertise or privacy that prohibit crowdsourcing. Analysis is also limited to behaviors and phenomena that scientists can directly and reliably observe, and subject to disagreement in its interpretation, both within studies and especially across studies [120].

Behaviors of interest range from low-level position of one or more individuals over time to mid-level actions like `bite` and `jump` as well as high-level activities such as `courtship` or `aggression`. Position may be coarsely represented by body centroid or finely represented by subject-specific pose. Behaviors over time are traditionally collected by ethologists[1] into ethograms [195] that can be represented by state-space models (see Figure 1-1 for an example). Manual annotation of low-, mid-, or high-level behaviors can take months, often at rates that are two to three times slower than the rate of date input [7]. For example, a recent study annotated 542 hours of primate behavior over a three-month period [130].

[1] An ethologist is a scientist who studies behavior. An ethogram is a representation, often hierarchical, of behavior over time.

Machine learning and vision are increasingly playing a role in the study of behavior [7]. Much of their use is aimed at reducing the human burden of data collection by automatically inferring low- and high-level behaviors. But, many contemporary approaches are black-box, with limited interpretability and no model of uncertainty [134]. This presents a fundamental challenge to their use in scientific workflows, where follow-on analysis seeks to build knowledge, often based on principled hypothesis tests in experimental settings.

Duration	**Subjects**	**Behavior**
7.41	C1, M2	Chase/Flee
4.24	C1, M2	Groom
6.77	C1, M2	Mount
12.5	C1, M2	Play

Figure 1-1: Example simplified ethogram collected from Macaque studies in Chapter 5.3. The complete annotations are multiple thousands of lines long *per annotator*.

All models will make mistakes, but models with no uncertainty representation will be unable to provide users with an awareness that they may have done so. Inference results can be manually inspected or else assumed correct, but inspection will not scale to tens of thousands of hours of behavior data, and willfully accepting mistakes may corrupt analysis, leading to incorrect conclusions. Erroneous conclusions such as misclassification or incorrect recommendations are tolerable in lower-stakes domains such as content-based image retrieval but they pose serious problems in scientific settings.

The methods developed in this work can aid scientists in the study of behavior. They include unsupervised generative models that infer low-level behaviors—body motion and part articulation—as well as supervised models that infer higher-level behaviors from representations with differing levels of interpretability. Their results are validated on a variety of datasets, including novel primate behavior datasets.

1.2 Interpretable Modeling

Understanding what model inferences mean and why they occur requires interpretability. Interpretability is about the representation used to model a problem. This work makes a distinction between intermediate and final representations. Final representations are unknown val-

ues of interest such as classification or regression estimates. Intermediate representations are additional degrees of freedom used by a model to produce those estimates.

For example, a deep neural network for camera pose estimation [107] has an interpretable final representation (the rotation and translation of a camera), but its intermediate representation (convolutional weights) yield limited insights into why decisions are made. Post-hoc analyses such as saliency maps [187] and class-activation maps [232] can visualize prominent feature locations with respect to input or class by plotting the magnitude of loss gradients with respect to input or reprojecting output weights to previous convolutional layers, respectively. These analyses do not clarify what one might do to improve performance; the typical response, then, is to increase network or dataset size and try different optimizers. Although black-box approaches often achieve superior performance on classification and regression metrics, the limited insights they offer may not be adequate for scientific analysis. Neither are large, labeled, datasets always cost effective to create, particularly when each experiment may involve entirely new data.

1.2.1 A Bayesian Approach

An alternative, Bayesian approach is to explicitly model the process that generated observed data with a series of functional relationships, each of which is mathematically well-understood.[2] The rationale is thus: if each component of a model can be analyzed and interpreted in meaningful ways, then so too should the whole. In this approach, observations are assumed to be noisy estimates of some *latent* quantities that have fixed but unknown values. To reason about their value, latent quantities are modeled as random variables whose distributions are interpreted as a *prior* belief in their value. Combining the prior distribution of each latent variable, the functional relationships of the generating process, and the observed data yields a *posterior* distribution over the joint set of latent variables. Inference is the process of drawing samples or computing summary statistics of the posterior, which constitutes the model's updated belief in the value of each latent quantity.[3] Hierarchies of latent variables can be stacked so that the relationship between observed data and latent variables is defined indirectly, through other latent variables. Graphical models [114] can be used to visualize and encode dependencies between variables in a generative model.

Generative modeling does not guarantee interpretability. Variational Autoencoders [111] and Generative Adversarial Networks [76] define generating processes that are intractable or implicit. But they do so with many intermediary latent variables and transformations that collectively learn an unknown function whose behavior is difficult to analyze. Even Bayesian approaches can be rendered less interpretable through the use of *nuisance parameters*—intermediary latent variables whose values are deemed irrelevant to the problem, and may

[2] Understanding is about properties associated with a representation or algorithm and should not be confused with the imperative knowledge of how to perform some procedure.

[3] As Edwin Jaynes articulated, this model must be viewed as akin to a four-year old child: it will believe anything you tell it [98]. The model departs with the child in that it will reason consistently (where consistency is well-defined by a set of axioms) with whatever prior belief and observations it is provided. These properties make the Bayesian approach desirable in a scientific context where assumptions and conclusions must be clearly articulated and open to scrutiny.

be introduced to make computation of the posterior distribution more tractable rather than because they facilitate insight. Interpretability, then, is about how well the components of a model are understood. It can thus be improved by increasing mathematical understanding of model components (as in the above class-activation maps), or by using modeling components that are already mathematically well-understood. Clearly, interpretability is not a binary property; rather, it exists on a spectrum. Of note are mixed uses of more and less interpretable components, as in the combination of neural networks, including Variational Autoencoders and Generative Adversarial Networks, with graphical models [102, 121, 49]. The interpretability of an approach should be dictated by the problem being solved. Approaches in this work are motivated by scientific applications where inferences can be readily analyzed in support of knowledge discovery. As such, it emphasizes the use of well-understood model components.

1.2.2 Distributions on Expressive Spaces

It is difficult to accurately model the generating process of complex data using interpretable representations. Many physical processes are described with constraints such as unit determinant for rotation matrices [8], boundary constraints for population modeling [173], and differential constraints for nonholonomic motion [112]. Constraints are straightforward to encode in probabilistic models, but make it difficult to construct inference procedures that correctly sample from the posterior implied by the model. One approach is to improve inference techniques so that they can sample from posterior distributions with constraints [5, 150, 224], but most approaches can only effectively handle boundary discontinuities. Rejection sampling, where samples from a proposal distribution without constraints are drawn and rejected if they violate the constraints of the target distribution, is always an option, but they commonly fail to explore their targets [73]. Another approach is to define probability distributions on more complex spaces where constraints are implicitly satisfied.

Probability distributions are traditionally defined on simple vector spaces such as $\mathbb{R}^D$ or $\mathbb{Z}$. Distributions directly defined on more complex spaces exist, but often have limitations. For example, the inverse Wishart, defined on $\mathcal{P}(D)$,[4] is limited by a single, shared concentration parameter that controls variance in all dimensions [6]. New distributions can be defined on these spaces, but doing so often involves involved mathematical derivations [200].

[4] $\mathcal{P}(D)$ is the set of symmetric positive-definite matrices of dimension $D \times D$.

Lie groups and the manifolds they are defined on[5] provide a parsimonious representation of the degrees of freedom in a system and are interpretable owing to their long study [16]. Deriving a Lie group is involved, but their representation and manipulation is often simple. The operations they define can implicitly respect system constraints, including maintaining unit determinant when composing rotations and maintaining symmetric positive definite structure when composing

[5] Lie groups are sets that locally have Euclidean structure along with a differentiable binary operator whose inverse is also differentiable. They will be discussed in more detail in Chapter 2.6

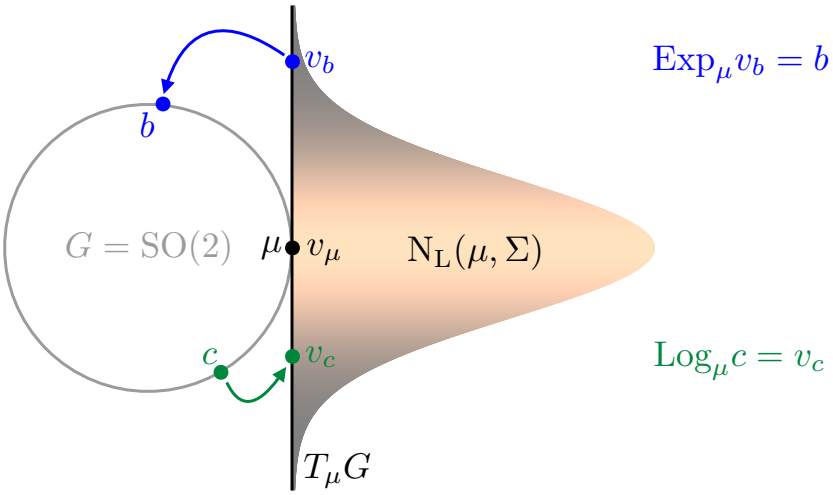


Figure 1-2: The exponential (blue) and logarithmic (green) maps on the Lie group of proper rotations in two dimensions, SO(2). The tangent space about μ, $T_\mu G$, can be identified with the Lie algebra $\mathfrak{g} = \mathfrak{so}(2)$.

covariance matrices. By placing distributions on general Lie groups, generative models can be constructed that have meaningful physical interpretations without incurring the inference challenges associated with handling explicit constraints.

A challenge to defining distributions on Lie groups is that many of them are do not form vector spaces; hence, standard distributions cannot be applied because notions of distance and norm do not immediately exist. Naive application of distributions to Lie groups would cause samples to be drawn that fail to respect group properties, thereby losing physical interpretability and correctness. However, associated with each Lie group G is a Lie algebra $\mathfrak{g}$ that is locally defined about the identity element of the group. Lie algebras forms a vector space in which standard distributions can be defined. Elements of the algebra can be locally mapped to elements of the group through (a typically nonlinear) exponential map, and elements of the group can be locally mapped to elements of the algebra by a corresponding logarithmic map. Most Lie groups of interest are Matrix Lie groups; that is, elements of the group can be expressed as a matrix with appropriate structure. In these cases, the exponential and logarithmic maps correspond to the well-understood matrix exponential and matrix logarithm [83] and the Lie algebra can be locally expanded about any element of the group through left- or right-translation.

To define a new distribution on the group, one need only apply the Lie group logarithmic map to the argument of a standard probability distribution. The support of the new distribution, then, is the group, but its arguments are mapped to the algebra, where all subsequent computation is performed. Arbitrary correlations between the degrees of freedom in a group can be modeled by using an appropriate distribution, such as the multivariate normal. Figure 1-2 shows an

example where a Gaussian distribution is defined on the Lie group of proper rotations in two dimensions. For rotations in higher dimensions, this same approach provides greater modeling flexibility than purpose-built rotation distributions such as the von Mises-Fisher [63], which cannot represent distinct per-dimension variances or correlation between dimensions. The benefit of taking this approach over defining a purpose-built distribution directly on the space of interest is that standard distributions can be defined in the algebra and take advantage of the already-understood exponential and logarithmic maps of a Lie group without having to perform involved derivations specific to that space.

This work develops several probabilistic models over Lie groups that are of interest to behavior analysis. In particular, $\mathrm{SE}(D)$ is used to model the rigid-body dynamics of an object with an unknown number of articulating parts, and $\mathcal{P}(D)$ is used to model dynamics on the *joint* kinematic states of multiple objects, in order to classify and analyze high-level behaviors. More broadly, probabilistic modeling on Lie groups is advocated as a promising and underutilized approach to generatively modeling complex processes.

1.3 Representation of Uncertainty

In a Bayesian framework, the prior distribution can be interpreted as a model's state of knowledge, or belief, concerning the values of each latent random variable. The posterior distribution is the model's updated belief in those values having accounted for observed data. Using distributions to represent belief enables comparison of how well models with differing priors or generating processes explain observed data, such as by using Bayes factors [142]. In a scientific setting, latent variables will correspond to quantities of interest, such as whether a particular gene affects the behaviors that an individual engages in. Representing belief with a distribution relaxes the need for a model to yield a de facto (and possibly incorrect) answer. Instead, the model can be interrogated with questions about the probability that a latent value falls within some range or set of values. I call this approach principled because the assumptions, data generating process, and probabilistic questions are all well-defined. In fact, Bayesian inference has been shown to be the only consistent extension of Aristotelian logic from boolean-valued propositions to propositions with degrees of plausibility [98]. Notably, Bayesian posteriors are uniquely determined given a generating process and prior distribution for each random variable, as well as a set of observed data.[6]

Bayesian posteriors can be difficult or impossible to represent completely. Exceptions exist for simple cases including discrete distributions with moderate numbers of states, continuous distributions with moderate numbers of modes, and distributions in the exponential family, where they can be perfectly summarized by a finite collection of

[6] But note that Bayesian posteriors are not usually invariant to reparameterizations of the prior [99].

sufficient statistics [163, 47, 115]. Many posteriors do not fall into these categories, however: they have no known analytic form, they exist in high dimensions that bar discretization, or they have unknown structure; for example, an unknown number of modes. Figure 1-5 shows examples where posterior distributions have an exponentially-growing number of states or a factorially-growing number of modes.

If a posterior cannot be completely represented, it can still summarized. A collection of location-scale parameters, one for each mode, is sufficient for distributions with known structure. Otherwise, a set of independently and identically distributed (IID) samples can be drawn. Collecting IID samples can in general be accomplished with Markov Chain Monte Carlo (MCMC) methods [85], where a random process is specially constructed so that successive samples from the process converge in distribution to the desired posterior. Monte carlo estimates computed on these samples converge to their true values as the number of IID samples grows. When posteriors are multimodal, many typical statistics, such as the moments of the distribution, become easy to misinterpret. The mean of a two-mode distribution may lie in a region of low probability, between the modes. The variance may be misinterpreted as broad uncertainty within a connected neighborhood of values when in actuality each mode is highly peaked and the space between them is substantial (Figure 1-3).

Multimodal posteriors denote ambiguity in the interpretation of observed data. For example, when two objects with similar appearance move indistinguishably close to each other and later depart, we would like a tracker to convey that the objects may or may not have crossed paths while they were close. The trajectory estimates will be similar in either case, but the assignment of trajectories to objects differs. Figure 1-4 represents how this ambiguity is conveyed in the multimodal posterior by the Joint Posterior Tracker, developed in Chapter 4.5.

We cannot hope to perfectly convey uncertainty in a posterior with large numbers of modes; any representation will necessarily form an underestimate. Hence, there will be ambiguity that the posterior accurately conveys—the known unknowns—and ambiguity that exists but is not conveyed—the unknown unknowns. Inability to convey all uncertainty does not obviate the value of representing a portion of it. The tracking example in Figure 1-5 have a multiplicative factor of $K!$ additional modes for every instance that K objects become indistinguishably close. Identifying even two ambiguous outcomes of each $K!$ factor is enough to draw attention to the region. We would like an automatic method for identifying such ambiguities without resorting to manual inspection and interpretation of the posterior, particularly when that posterior may exist of dozens or hundreds of hours of data, and may be represented by a set of hundreds to thousands of samples.

Posterior uncertainty can be summarized by entropy. Peaked posteriors have low entropy whereas uniform posteriors have high entropy. Intermediate values do not clarify whether a posterior has a single mode with broad probability mass or else multiple modes. Re-

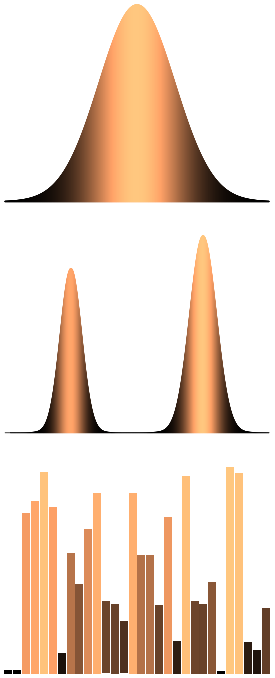


Figure 1-3: Representation of uncertainty is often simple to represent in continuous, unimodal distributions (top), but multimodal data, whether continuous (middle) or discrete (bottom), is non-trivial to represent, especially when there are an unknown number of modes.

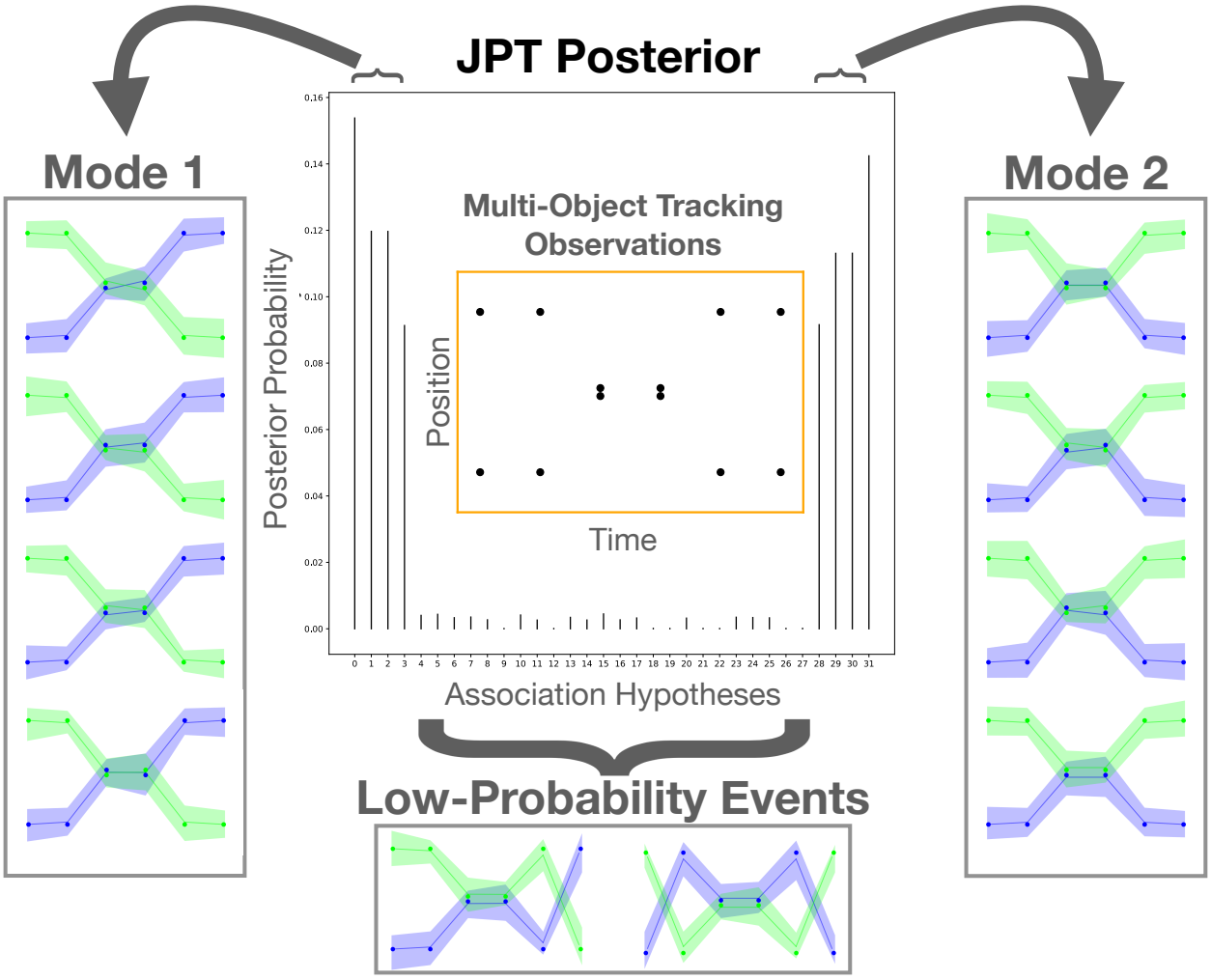


Figure 1-4: Multimodal posteriors convey ambiguity in observed data. Multi-object tracking observations over time (black points in orange box, middle) where $K = 2$ targets (green, blue) begin separated–briefly converge–then diverge. This ambiguity gives rise to $K!$ distinct posterior modes in the Joint Posterior Tracker (JPT), which grow exponentially with additional ambiguous regions. Here $K! = 2$: either the targets crossed (left) or not (right). JPT explores high-probability regions in its posterior and represents uncertainty in data association by a collection of posterior samples. Association events of high (left, right) and low (bottom) probability are visualized as observations colored according to their associated target and inferred trajectories are plotted with $\pm 3\sigma$ shading. Observe that *within* each mode, associations switch when targets are proximate but trajectory estimates remain similar whereas *between* modes, associations switch when targets are separated, causing large variation in trajectory estimates.

gardless, a series of hypothetical experiments can be proposed whose unknown outcomes can be simulated within the generative model. The results of each hypothetical experiment lead to a new posterior that is conditioned on observing its results. If some of the experiments lead to significant reductions in posterior entropy then we conclude that there is ambiguity in the original posterior and may choose to actually perform (rather than just simulate) one or more of the experiments, in batch or sequentially. This process can be repeated until entropy is no longer significantly reduced by further simulation of experiments. Doing so is equivalent to maximizing the mutual information between the original posterior and each considered experiment. Yet, absence of

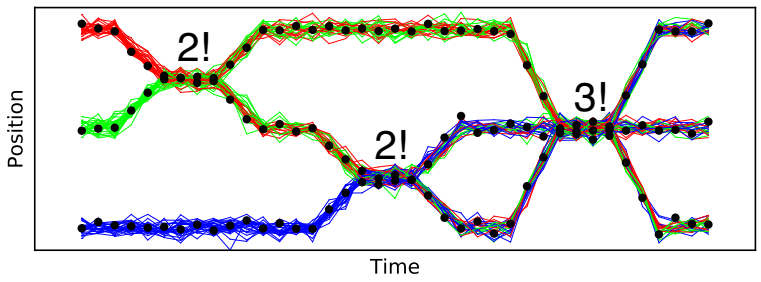


Figure 1-5: Posterior with an exponential number of states and factorial number of modes. At each time $t = 1, \ldots, T$ for $T = 39$ there are $N = 3$ observations to be uniquely associated to $K = 3$ objects. Then, there are $3!^T = 2.29e30$ posterior states. Assuming linear Gaussian dynamics for each object, there are a multiplicative $M!$ additional modes every time M objects become close. Here, there are $2!\ 2!\ 3! = 24$ posterior modes. Each one is visualized as a single set of red, green, and blue lines.

proof is not proof of absence: posterior ambiguity may still be present even when there is no uncertainty reduction from proposed experiments. It is crucial, then, that considered experiments exploit knowledge of both the problem being solved and the generative model they exist within.

Chapter 4.7 develops uncertainty reduction for multi-object tracking where it is common for objects to be confused with one another, usually because they move in proximity and have similar appearance. Common benchmarks show thousands of these *identity switching* errors for minutes-long sequences [136]. Multi-object tracking is vital for scaling behavior science, but will produce incorrect conclusions if trackers exhibit large numbers of identity switches. Collection conditions can sometimes be created where ambiguities are minimized, such as with the use of colored markings like collars and painted fur. But these markings can require weeks-long habituation processes *for each animal*, and are not always foolproof or available: subjects can remove specially-colored markers or keep them occluded for extended periods of time, long-term tracking may also encounter arrival events like animal birth, and not all can be specially-marked.[7]

In this work, experiments are questions to a human annotator about the assignment of measurements to objects in multi-object tracking. Their answers are modeled as a noisy oracle (i.e., they are correct with some probability). We show that a small number of automatically-scheduled yes/no questions can rapidly reduce posterior uncertainty and improve trajectory estimates. The framework for reducing posterior uncertainty through sequential Bayesian experiment design is introduced in Chapter 2.7 and developed for multi-object tracking in Chapter 4.7.

[7] Zebrafish cannot wear a colored vest or collar and are able to change appearance based on their surroundings.

1.4 Discovery of Structure

Reliably tracking motion over time is a starting point for studying behavior, but more granular measurements are needed, including parts modeling and behavior discovery. Typical approaches rely on hand-designed models that amount to strong prior information or large quantities of labeled training data in a supervised framework. These can yield precise results but are bound by advance interpretations of what is important to behavior modeling. This work emphasizes models that automatically discover structure that is not known in advance, including the number of articulating parts an object has (Chapter 3.4), the number of objects present in a scene (Chapter 4.5), and differences in behavior across experimental conditions (Chapter 5.3). I argue that models should not be limited by strong, advance specifications of expected behavior if that is the object of study.[8]

[8] Supervised approaches have a role to play in behavior science, but it should not be the only role.

Bayesian nonparametric (BNP) models discover structure by scaling the complexity of their representation based on observed data; hence, they can learn structure that humans are unaware of or unable to specify in advance. Structure is learned by specifying a prior over an infinite-dimensional parameter space for which only a finite subset of parameters are required to describe a finite observation set. As the number of observations grow, so too do the parameters required to describe them. BNP can be composed with distributions on expressive spaces so that discovered structure has physical-grounded interpretations. Figure 1-6 shows a time-varying nonparametric mixture model where the number of clusters, their individual and shared rigid transformation dynamics, and shape are learned from collections of observations over time. This is a simple application of the Nonparametric Parts Model (Chapter 3.4), whose latent representation be interpreted as learning the number and articulated pose of an object's parts over time, simply by observing that object in motion over time.

BNP are not approachable to non-expert users, including behavior scientists. Interpretation is made difficult by dense, measure-theoretic treatments of their underlying stochastic processes and implementation is made challenging because inference often requires manual derivation and implementation.[9] In particular, many BNP models require the solution to a posterior predictive integral over an infinite space that has a known, analytic form only when there is conjugate structure.[10] Understanding conjugate structure requires knowledge of a plethora of probability distributions while eschewing it requires special inference techniques.

[9] Gaussian processes [169] are an exception. Many machine learning packages implement basic forms, but users will still struggle to implement customized assumptions.

[10] Conjugate structure occurs when the composition of analytic distributions results in another analytically-known distribution.

Automatic inference engines exist for Bayesian models in the form of probabilistic programming languages. Many, including Stan [37], Tensorflow Probability [196, 57], and PyMC [176], compute a joint likelihood and its gradient from a generative specification and apply sophisticated HMC [146], NUTS [91], or ADVI [116] inference to exactly or approximately draw posterior samples. This approach separates modeling from inference by abstracting the generative model

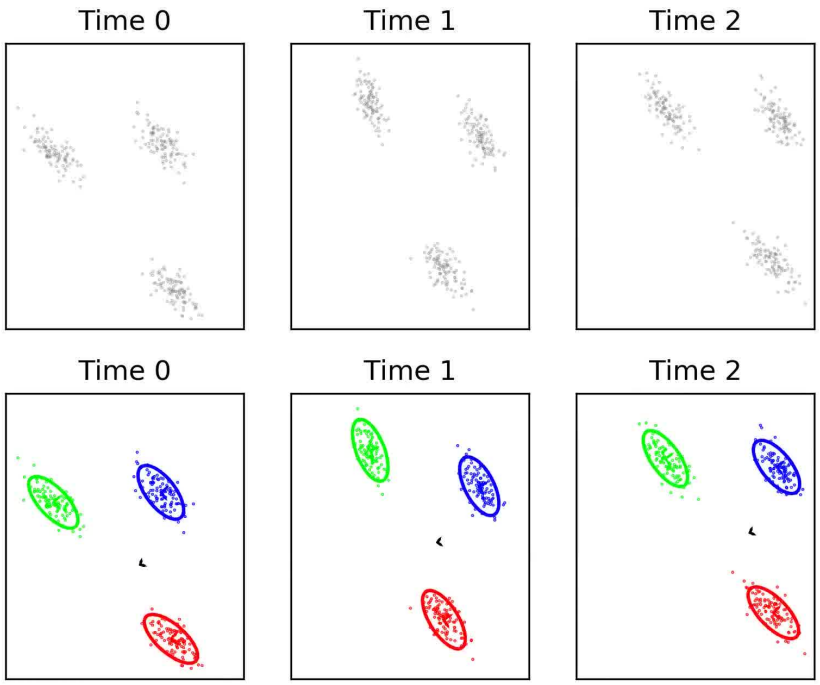


Figure 1-6: Discovering the number of components, and their dynamics—common and independent—in time-varying data. An unknown but fixed number of clusters evolve over time according to rigid-transformation dynamics. (**Top**): Observations over time. (**Bottom**): Posterior sample of the inferred number of clusters, their individual dynamics (colored frames of reference) and their shared dynamics (black frame of reference).

into function evaluations.[11] This separation makes it easier for users to explore and implement different models. Yet, exploiting conjugate structure or otherwise estimating posterior predictive distributions in BNP inference involves greater knowledge of the generative model than these inference engines have access to. Furthermore, inference on discrete and mixed continuous-discrete latent representations, common to BNP models, cannot be directly performed with HMC, NUTS, or ADVI, though several works exist to approximate discrete distributions with continuous distributions [129, 198], as well as to perform HMC-style sampling on discrete distributions [150]. Still, current approaches in gradient-based probabilistic programming inference is to marginalize over discrete distributions—something that is not always an option. BUGS and JAGS [36] use Gibbs sampling [74] inference and can take advantage of conjugate structure but cannot leverage more effective inference techniques, like HMC and NUTS, when available. Anglican [219] stands out in that it implements a common nonparametric prior, the Chinese Restaurant Process [165], as a primitive and can handle mixed continuous-discrete distributions by using Particle MCMC [10] inference, but it has a Lisp-based syntax that is inaccessible to non-expert users as well as most expert Bayesian users!

This work promotes Bayesian nonparametrics for behavior sci-

[11] Abstracting a generative model into likelihood and gradient evaluations is similar to how optimization packages abstract optimization problems to an objective and its gradient.

ence but acknowledges that, at present, they are difficult to use and implement. The Nonparametric Parts Model is a specific contribution with applications to behavior analysis. As part of deriving inference for that model, a novel Gibbs decomposition is discovered for Concentrated Gaussian priors with Gaussian likelihoods. Additionally, a simple Monte Carlo sampling technique is demonstrated to effectively approximate a complicated, non-analytic posterior predictive. It is hoped that these contributions motivate and simplify support for variable-dimension latent spaces in automatic inference engines, where knowledge of conjugate structures are exploited where possible.

1.5 Contributions and Overview

This dissertation combines Bayesian nonparametrics with probabilistic reasoning over manifolds and joint inference over time-evolving state-space models. Parts modeling, tracking, and both supervised and unsupervised behavior approaches are specifically treated. More generally, it advocates for interpretable Bayesian approaches to scalable behavior science.

Chapter 2 introduces Bayesian models and common sampling-based inference techniques, used throughout this work. Discrete and continuous state space models are introduced for handling time-evolving data. Both linear and nonlinear dynamics are supported through joint realizations of the latent space, as opposed to more commonly used filtering and smoothing approaches. Manifolds used in this work, and probabilistic reasoning on them, are then covered. Then, finite and infinite mixture models are discussed, as are their relation to state-space models. Finally, sequential Bayesian optimal experiment design is introduced as a general planning approach for reducing posterior uncertainty through a series automatically-scheduled questions.

Chapter 3 develops the Nonparametrics Parts Model (NPP), which learns the number, shape and motion of an object's parts by observing it in motion and assuming that its (unknown) parts orbit about a common, unknown body and evolve according to a random walk on $\mathrm{SE}(D)$ the manifold of D-dimensional rotations and translations. Evaluations show that NPP learns parts decompositions that accord with human intuition, enables analysis of part motion relative to a body frame of reference, and permits generative simulation of novel motion.

Chapter 4 develops the Joint Posterior Tracker (JPT), which performs long-term multi-object tracking of objects in motion with explicit uncertainty quantification. JPT outperforms baselines, both on traditional multi-object tracking metrics and on a novel uncertainty quantification metric. It also enables recovery from errors using human-in-the-loop corrections that are automatically scheduled by sequentially maximizing an objective based on information gain.

Chapter 5 relates parts modeling (Chapter 3) and tracking (Chapter 4) to the scientific analysis of behavior. In Chapter 5.3, we conduct

low-level behavior analysis of genetically mutant primates in an unsupervised, experimental setting. Automated tracking from our Nonparametric Extents Model (Chapter 5.3.1) saved scientists more than 250 hours of labeling effort. Tracks were provided to collaborators for analysis, and contributed to the first evidence for primate animal models of autism (Chapter 5.3.4).

Chapter 5.4 describes Marmoset100, a novel RGB-Depth dataset collected with collaborators. Marmoset100 consists of more than 1TB of data with 100 hours of tracked observational video on pairwise primate social interactions. Thirty hours of video are labeled for 25 behaviors (Chapter 5.5). We classify behaviors in a supervised, observational setting based on varying multi-object tracking representations and presence or absence of uncertainty (Chapter 5.5). We show that uncertainty representation in tracking estimates improves behavior classification and that higher-quality tracking with simple, point-based representations produces higher behavior classification performance than lower-quality tracking with complex, pose-based representations.

The work on nonparametric parts modeling is published in [87]. The work on multi-object tracking with uncertainty quantification and error recovery is available in [88] and featured in [231]. The work on unsupervised behavioral analysis is published in [233].

Chapter 2

Background

"Everything is hard before it is easy."
— Goethe

Foundations for the approaches used throughout this dissertation are introduced in this chapter. I begin by motivating Bayesian modeling as a generalization of deductive reasoning to plausible reasoning (2.1). Inference with focus on Markov Chain Monte-Carlo (MCMC) sampling methods is then introduced for Bayesian models (2.2). Common probability distributions used in this work are covered in (2.3), including an introduction to the Dirichlet Process. Coverage of the Dirichlet Process continues with its application to mixture models (2.4). Then, common models and probabilistic queries for time-evolving data are covered (2.5) as is probabilistic reasoning on manifolds and Lie groups (2.6). Last, Bayesian experiment design is covered (2.7).

2.1 Plausible Reasoning

Deductive reasoning is the repeated application of rules that relate propositions to conclusions. Rules are of the form, "if A then B," and they lead to conclusions such as, "A, therefore B," and "not B, therefore not A." All that can be said of propositions and conclusions is binary: either they are true or false. This precision and clarity is available and necessary in mathematics but insufficient in science and daily life [166]. Sensing an environment, navigating a crowded street, choosing a hypothesis, building a model, and accepting circumstantial evidence are all examples where information is limited and far from absolute, yet people regularly navigate these situations effectively. Designing agents to do the same requires a calculus not just about what is true and false, but also about what is more and less plausible.

One derivation of a system for plausible reasoning [98] begins with the following assumptions, where the last three are collectively defined as consistency:

1. Degrees of plausibility are represented by real numbers.
2. If a conclusion can be reached in more than one way then every possible way leads to the same conclusion.
3. All available evidence is used to form a conclusion.

4. Equivalent states of knowledge are represented by equivalent assignments of plausibility.

We desire a way to compute the plausibility of a compound proposition based on the plausibilities of simpler propositions. In particular, let AB denote the logical conjunction (and) of propositions A and B. Let $A \mid C$ denote the proposition A conditioned on the proposition C. What can be said of the plausibility of the proposition $AB \mid C$ in terms of A and B? It cannot be a function of A or B alone, because C would be unaccounted for. There are five quantities that condition on C,

$$(AB \mid C) \quad (A \mid C) \quad (A \mid BC) \quad (B \mid C) \quad (B \mid AC) \tag{2.1}$$

and eleven possible relationships between $AB \mid C$ and quantities that condition on C (see Figure 2-1). It can be exhaustively proven [197] that only two relationships satisfy our assumptions,

$$w(AB \mid C) = w(A \mid BC)\, w(B \mid C) = w(B \mid AC)\, w(A \mid C) \tag{2.2}$$

where $w(A)$ denotes the plausibility of proposition A. Further analysis reveals that w must be a positive, continuous, monotonic function. Two ranges of values for plausibility satisfy this: ∞ for completely implausible to 1 for completely plausible, or 0 for completely implausible to 1 for completely plausible. By convention, we choose the latter: $0 \leq w \leq 1$. Additional analysis of w derives,

$$w^m(A \mid B) + w^m(\neg A \mid B) = 1 \tag{2.3}$$

for any positive real m where $+$ denotes logical disjunction (or) and $\neg$ denotes logical negation. Defining $p(A) = w^m(A)$ in terms of Equations 2.2, 2.3, we have,

$$p(AB \mid C) = p(A \mid BC)\, p(B \mid C) = p(B \mid AC)\, p(A \mid C) \tag{2.4}$$

$$1 = p(A \mid B) + p(\neg A \mid B) \tag{2.5}$$

which are the standard product and sum rules of probability theory, with $0 \leq p(A) \leq 1$ the familiar notation for the probability of A. The product and sum rules form a complete set of rules for plausible reasoning in the same way that conjunction and negation form a complete set of rules for deductive reasoning. Most importantly, Bayes rule is easily derived from Equation 2.4 by assuming $C = \emptyset$:

$$p(A \mid B) = \frac{p(AB)}{p(B)} = \frac{p(B \mid A)\, p(A)}{p(B)} \tag{2.6}$$

The numerator is the joint distribution between A, B and the denominator is the evidence, which can also be expressed as the marginalization over A of the joint distribution, $p(A) = \int_B p(A, B)$.

In Bayesian modeling, A usually forms a set of latent random variables and B denotes a set of observations. Hence, $p(A)$ constitutes a

$F_1(v, w)$ $F_7(v, w, d)$
$F_2(v, d)$ $F_8(v, w, e)$
$F_3(v, e)$ $F_9(v, d, e)$
$F_4(w, d)$ $F_{10}(w, d, e)$
$F_5(w, e)$ $F_{11}(v, w, d, e)$
$F_6(d, e)$

$u = AB \mid C$ $v = A \mid C$ $w = A \mid BC$
$d = B \mid C$ $e = B \mid AC$

Figure 2-1: The eleven possible functional relationships for a product rule of probability. Single-argument possibilities are excluded because they can be proven incorrect by counterexample. Only F_3 and F_4 are consistent with our assumptions for plausible reasoning.

prior distribution over latent variables and $p(A \mid B)$ is the posterior conditioned on observations which are modeled with likelihood $p(B \mid A)$. Throughout this work, observations will be denoted by y and latent variables by other letters, most commonly x for continuous variables and z for discrete variables.

Bayesian modeling is one of several systems for plausible inference. It provides a well-defined mechanism for updating belief in the face of new evidence but is limited by the requirement that the conditional distribution for each random variable be precisely specified. As argued in Chapter 1, this is desirable and aids interpretability in scientific workflows. Other systems for plausible reasoning relax this constraint, allowing for the specification and combination of contradictory or partial evidence. They include Dempster-Shafer Theory and Belief Functions [183], imprecise probabilities [210], and fuzzy sets [225]. A recent tutorial relates several of these approaches for the interested reader [51].

2.2 Bayesian Inference

Constructing a Bayesian model involves defining a set of random variables and the distributions that generate them. The same model can be used to solve multiple problems, but the probabilistic queries will differ. In general, variables are distinguished by whether they are observed or latent, and some question is asked that involves inference—the updating of belief—over the latent variables conditioned on the observed variables.

Let $x \in \mathbb{R}^D$ be a set of latent random variables and $y \in \mathbb{R}^N$ be a set of observations. Two common inference tasks are to estimate the posterior distribution,

$$p(x \mid y) = \frac{p(y \mid x)\, p(x)}{p(y)} \tag{2.7}$$

and to estimate the posterior predictive distribution:

$$p(\hat{y} \mid y) = \int_x p(\hat{y} \mid x)\, p(x \mid y) \tag{2.8}$$

The posterior requires calculation of the integral,

$$p(y) = \int_x p(y \mid x)\, p(x) \tag{2.9}$$

which is similar in form to the posterior predictive (Equation 2.8), but is taken with respect to the prior. In general, neither integral is tractable. There may be too many states to integrate over or they may not have a known, analytic form. A prominent exception occurs when $p(x)$ and $p(y \mid x)$ have conjugate structure, so that their parametric forms are similar enough that their combination retains the same parametric

form, though with different parameters. Conjugacy most commonly occurs when the prior and the likelihood belong to the exponential family [208], though there are exceptions.[12] Conjugate structure provides an analytic posterior that has known moments and is easy to sample from. Analysis or summary of a posterior with analytic form, such as computing the most probable value along with a credible interval, computing quantiles, or computing measures of uncertainty like entropy can be straightforwardly accomplished.

[12] The uniform distribution is not in the exponential family yet has a conjugate prior: the Pareto distribution.

Many inference problems lack conjugate structure, most notably mixture models, which underpin much of the contributions in this work. In some cases, a MAP (maximum a posteriori) estimate suffices,

$$x_{\text{MAP}} = \operatorname*{argmax}_x p(x \mid y) = \operatorname*{argmax}_x p(y \mid x)\, p(x) \tag{2.10}$$

where the second equality follows because $p(y)$ is an unknown constant. The MAP estimate corresponds to the mode of a unimodal posterior, or to one of several modes in a multimodal posterior, but it may not be a representative sample![13] The MAP estimate can be computed with analytic form in simple cases but, in general, will require use of an optimization algorithm such as L-BFGS, Adagrad, RMSProp, or Adam [31, 113]. When there are multiple modes, the result of an optimization will correspond to one of the modes, but it may not actually be the mode with highest probability mass.

[13] The typical set of distribution $p(x \mid y)$ is the set of sequences $\{x_1, \ldots, x_S\}$ of IID draws whose entropy can be made arbitrarily close to $e^{-H(x|y)}$. The MAP estimate commonly lies outside the typical set, particularly in high dimensions. See [25] for more discussion.

MAP estimation is convenient because it can be performed without estimating an integral. It is limited because it summarize a posterior with a single point, eschewing any representation of uncertainty. One simple approach to estimating uncertainty that also avoids integrals is the Gaussian approximation,

$$H = \left.\frac{\partial^2 p(x \mid y)}{\partial x\, \partial x^\top}\right|_{x_{\text{MAP}}} \qquad p(x \mid y) \approx \text{N}\left(x \mid x_{\text{MAP}}, H^{-1}\right) \tag{2.11}$$

where H is the Hessian of the posterior estimated about x_{MAP}. The Gaussian assumption implies that the estimate will have a single mode. If $p(x \mid y)$ is actually multimodal, this will be a poor approximation. Note also that Gaussian approximation cannot be done for discrete or mixed continuous-discrete latent spaces.

2.2.1 Markov Chain Monte Carlo

When accurate uncertainty estimation is required and $p(y)$ is difficult to evaluate, we turn to the most general methods for posterior inference: Markov Chain Monte Carlo (MCMC). MCMC approaches iteratively sample from a specially-constructed stochastic process whose samples are eventually distributed according to $p(x \mid y)$. In this section, I show a general approach to constructing processes that sample from $p(x \mid y)$. Following, I provide specific sampling algorithms. The interested reader is referred to [70] for comprehensive treatment of Markov chain fundamentals in the context of MCMC.

To begin, let $\{x^s\}_{s=1}^{\infty}$ be a sequence of random variables where x^s is defined on some space $\mathcal{X}$. If $p(x^s \mid x^{1:s-1}) = p(x^s \mid x^{s-1})$, then we say that the sequence is a Markov chain. Let $T(x' \mid x, y)$ be the transition distribution that generates successive values on this Markov chain (where y are the fixed observations from Equation 2.7). We would like the values x^s to be distributed according to the posterior $p(x \mid y)$. Under mild assumptions,[14] a sufficient condition for this to occur is if detailed balance is satisfied,

$$T(x' \mid x, y)\, p(x \mid y) = T(x \mid x', y)\, p(x' \mid y) \tag{2.12}$$

for all $x, x' \in \mathcal{X}$. If this holds, then we say that the Markov chain with transition $T(x' \mid x, y)$ has $p(x \mid y)$ as its unique stationary distribution. Our goal, then, is to construct a transition distribution $T(x' \mid x, y)$ such that detailed balance holds. Define,

$$R(x' \mid x, y) = \frac{p(x' \mid y)\, q(x \mid x', y)}{p(x \mid y)\, q(x' \mid x, y)} \tag{2.13}$$

$$A(x' \mid x, y) = \min(1, R(x' \mid x, y)) \tag{2.14}$$

where $q(x' \mid x, y)$ is some *proposal distribution* from which a new state x' can be sampled given that the current state is x. We call R the Metropolis-Hastings ratio, and A the acceptance probability. Now, define the transition distribution as,

$$T(x' \mid x, y) = \tag{2.15}$$
$$\begin{cases} q(x' \mid x, y)\, A(x' \mid x, y) & \text{if } x' \neq x \\ q(x \mid x, y) + \int_{x' \neq x} q(x' \mid x, y)(1 - A(x' \mid x, y)) & \text{if } x' = x \end{cases}$$

where the first line captures the case where we sample $q(x' \mid x, y)$ and accept it with probability $A(x' \mid x, y)$ and the second line is the case where we transition to $x' = x$, which can occur either by sampling $q(x' \mid x, y) = q(x \mid x, y)$ when $x' = x$, or by proposing some other state x' with density $q(x' \mid x, y)$ and rejecting it with probability $1 - A(x' \mid x, y)$, integrated over all possible states $x' \neq x$. Suppose,[15]

$$p(x \mid y)\, q(x' \mid x, y) > p(x' \mid y)\, q(x \mid x', y) \tag{2.16}$$

then by dividing the LHS by the RHS, we get $R(x \mid x', y) > 1$, and so $A(x \mid x', y) = 1$ and $R(x' \mid x, y) = A(x' \mid x, y) < 1$. To show detailed balance, assume we have moved from state x to x' where $x' \neq x$. This is governed by the first case in Equation 2.15 so that:

$$T(x' \mid x, y) = q(x' \mid x, y)\, A(x' \mid x, y) \tag{2.17}$$

$$= q(x' \mid x, y)\, \frac{p(x' \mid y)\, q(x \mid x', y)}{p(x \mid y)\, q(x' \mid x, y)} \tag{2.18}$$

$$= \frac{p(x' \mid y)}{p(x \mid y)}\, q(x \mid x', y) \tag{2.19}$$

[14] $T(x' \mid x, y)$ must also be aperiodic (it doesn't get into state transition cycles), irreducible (can go from any state to any other in finite time), and not transient (it will return to the current state with probability one).

[15] The less than and equal cases can also be handled but we do not cover them.

Equation 2.18 follows because $A(x' \mid x, y) = R(x' \mid x, y)$. Combining Equation 2.17 with Equation 2.19, we have,

$$T(x' \mid x, y)\, p(x \mid y) = p(x' \mid y)\, q(x \mid x', y) \tag{2.20}$$

Now, the reverse transition is,

$$T(x \mid x', y) = q(x \mid x', y)\, A(x \mid x', y) \tag{2.21}$$
$$= q(x \mid x', y) \tag{2.22}$$

where Equation 2.22 follows from Equation 2.21 because we established that $A(x \mid x', y) = 1$. Substituting Equation 2.22 into Equation 2.20 we have,

$$T(x' \mid x, y)\, p(x \mid y) = p(x' \mid y)\, T(x \mid x', y) \tag{2.23}$$

hence, detailed balance is satisfied, implying that samples drawn according to $T(x \mid x')$ will eventually be distributed according to the posterior $p(x \mid y)$.

2.2.2 Metropolis-Hastings

Section 2.2.1 demonstrates how to construct a Markov chain that satisfied detailed balance so that the posterior is its unique stationary distribution. The transition function (Equation 2.15) is used by the Metropolis-Hastings algorithm to sample from $p(x \mid y)$. Observe that the posterior is evaluated in the transition function, but only as a ratio: $p(x' \mid y)/p(x \mid y)$ so that the normalization constants $p(y)$ cancel out. This construction makes it possible to sample from the posterior $p(x \mid y)$ when it can only be evaluated up to proportionality. We call $\tilde{p}(x \mid y) = p(y \mid x)\, p(x)$ the unnormalized posterior because $p(x \mid y) = \tilde{p}(x \mid y)/p(y)$. Algorithm 1 gives the Metropolis-Hastings (MH) algorithm, which can be used any time the generative model (prior and likelihood) are known.

Algorithm 1: The Metropolis-Hastings Algorithm

Input : x, y, proposal q, unnormalized density $\tilde{p}$
Output: x'

1 Sample $x' \sim q(x' \mid x, y)$
2 Evaluate $R(x' \mid x, y) = \frac{p(x'|y)\ q(x|x',y)}{p(x|y)\ q(x'|x,y)}$
3 Evaluate $A(x' \mid x, y) = \min(1, R(x' \mid x, y))$
4 Samlpe $u \sim \text{Unif}(0, 1)$
5 **if** $u < A(x' \mid x, y)$ **then** return x'
6 **else** return x

The one degree of freedom in Metropolis-Hastings is the proposal distribution $q(x' \mid x, y)$. When q depends on observations y, we say that it is a data-driven proposal, otherwise if $q(x' \mid x, y) = q(x' \mid x)$,

we say that it is an independent proposal. Determining proposal distributions that are general and efficient constitutes much of the contributions in the literature, including this dissertation. A special case of the MH algorithm was developed by [135] and later generalized by [85]. A fascinating history of the development of sampling algorithms is written in [172].

2.2.3 Gibbs Sampling

Let $x = \{x_d\}_{d=1}^{D}$ be the joint latent state and define $x_{-d} = x \setminus x_d$ as the set of all latent states except x_d. Then, omitting dependence on y, Gibbs sampling is useful when each of the D full conditional distributions $p(x_d \mid x_{-d})$ can be efficiently sampled from, either because they have analytic form or they are easier to sample than is the joint posterior $p(x \mid y)$. When this is the case, a special set of D proposal distributions can be used in Metropolis-Hastings. They require no tuning and are accepted with probability one. For x' a newly proposed joint state from current state x, let the d^{th} proposal have the form,

$$q(x' \mid x) = p(x'_d \mid x_{-d}) \delta_{x'_{-d} = x_{-d}} \tag{2.24}$$

where $\delta_{x'_{-d} = x_{-d}} = 1$ if all but the d^{th} variable are equal in the proposal and the previous state (and is 0 otherwise). Then, the MH ratio is,

$$\begin{aligned}
R(x' \mid x) &= \frac{p(x')\, q(x \mid x')}{p(x)\, q(x' \mid x)} && (2.25)\\
&= \frac{p(x'_d \mid x'_{-d})\, p(x'_{-d})\, p(x_d \mid x'_{-d}) \delta_{x_{-d} = x'_{-d}}}{p(x_d \mid x_{-d})\, p(x_{-d})\, p(x'_d \mid x_{-d}) \delta_{x'_{-d} = x_{-d}}} && (2.26)\\
&= \frac{p(x'_d \mid x'_{-d})\, p(x'_{-d})\, p(x_d \mid x'_{-d})}{p(x_d \mid x'_{-d})\, p(x'_{-d})\, p(x'_d \mid x'_{-d})} && (2.27)\\
&= 1 && (2.28)
\end{aligned}$$

where Equation 2.27 follows assuming that only the d^{th} variable of proposal x' is modified.

Gibbs sampling is a useful Metropolis-Hastings technique because it requires no special considerations for a proposal distribution: the proposal is implied by the model. One drawback is that Gibbs samplers can only move in a coordinate-aligned fashion, which can cause them to fail to explore posteriors where components of x are highly correlated. Gibbs sampling was developed in [74]; it is frequently used in models with mixed continuous-discrete latent spaces where efficiently drawing joint samples can be difficult. Finally, the full conditionals need not be able to be analytically sampled from: in a scheme known as Metropolis-Within-Gibbs, any Metropolis-Hastings method that targets $p(x_d \mid x_{-d})$ will maintain the originally-desired stationary distribution $p(x \mid y)$. The Gibbs sampling procedure is in Algorithm 2.

Algorithm 2: The Gibbs Sampling Algorithm

Input : $x = (x_1, \ldots, x_D), y$
Output: x'

Let $x' = x$
for $d \in 1, \ldots, D$ *in random order* **do**
  Sample $x'_d \sim p(x'_d \mid x'_{-d}, y)$
return x'

2.2.4 Slice and Beam Sampling

In slice sampling, an augmented distribution,

$$p(x, u \mid y) = \begin{cases} \frac{1}{p(y)} & \text{if } 0 \leq u \leq \tilde{p}(x \mid y) \\ 0 & \text{o.w.} \end{cases} \tag{2.29}$$

is defined, which has marginal:

$$\int_0^{\tilde{p}(x|y)} \frac{1}{p(y)} du = \frac{\tilde{p}(x \mid y)}{p(y)} = p(x \mid y) \tag{2.30}$$

To sample from $p(x \mid y)$ we thus sample from $p(x, u \mid y)$ and only consider the marginal $p(x \mid y)$. Similar to Gibbs sampling, this can be accomplished by repeatedly sampling each conditional:

$$p(u \mid x, y) = \text{Unif}(0, \tilde{p}(x \mid y)) \tag{2.31}$$
$$p(x \mid u, y) = \text{Unif}(\{x \ : u < \tilde{p}(x \mid y)) \tag{2.32}$$

The construction of $p(x, u \mid y)$ makes the first conditional straightforward to sample from. The second conditional is more challenging because we won't typically know the values x that fall below a given unnormalized density. When x is univariate and bounded, this set can be enumerated. When x is multivariate, then the conditional $p(x \mid u, y)$ can be broken into D dimensions, each treated as a univariate distribution. More generally, [145] defines several approaches that iteratively expand about a neighborhood of x so that the Markov chain remains invariant to the conditional $p(x \mid u, y)$. In this work, slice sampling is used to efficiently sample univariate rotation values from multimodal distributions. Though the sampler can get stuck in a single mode, a fixed series of MH proposals enumerate all other modes given a sample from one mode. See Chapter 3.5.3.

Slice sampling forms the basis for several exact samplers in nonparametric models, most notably the Dirichlet Process Mixture Model [209, 105, 71], where conjugacy is not required, and in infinite state space models, where it is called the beam sampler [203]. The striking feature of this approach is that it can exactly reason over an infinite number of atoms (or states) but it only ever considers a finite set of them in any individual sample. We sketch this approach for Dirichlet

Process Mixture Models, defined in Section 2.4.3.

Let $x = \left(\{\pi_k, \theta_k\}_{k=1}^{\infty}, \{z_n\}_{n=1}^{N}\right)$ and $y = \{y_n\}_{n=1}^{N}$ where π_k are an infinite collection of mixture weights, θ_k are an infinite collection of mixture parameters, and $z_n = k$ denotes that observation y_n is generated by mixture k. Then, the likelihood for the n^{th} observation can be written,

$$p(y_n \mid \pi, \theta) = \sum_{k=1}^{\infty} \pi_k \; f_k(y_n) \tag{2.33}$$

where $f_k(y_n)$ is the observation model for mixture k with parameters θ_k. For each n, augment Equation 2.33 with auxiliary variable $u_n \sim \text{Unif}(0, \pi_{z_n})$. Then,

$$\begin{aligned}
p(y_n, u_n \mid \pi, \theta) &= \sum_{k=1}^{\infty} \text{Unif}\,(u_n \mid 0, \pi_k) \;\; \pi_k \; f_k(y_n) && (2.34)\\
&= \sum_{k=1}^{\infty} \frac{1}{\pi_k} \delta_{0 \le u_n \le \pi_k} \;\; \pi_k \; f_k(y_n) && (2.35)\\
&= \sum_{k=1}^{\infty} \delta_{0 \le u_n \le \pi_k} f_k(y_n) && (2.36)\\
&\propto \sum_{\substack{k: \\ \pi_k \ge u_n}} f_k(y_n) && (2.37)
\end{aligned}$$

As in slice sampling, Equation 2.33 is the marginal of Equation 2.36 when integrating u_n out. Importantly, Equation 2.36 is non-zero for only a finite number of terms, which is made explicit in Equation 2.37. A sampler can thus be constructed where, at each iteration, there are only ever a finite number of atoms in the latent state. Algorithm 3 gives the procedure.

2.2.5 Reversible-Jump MCMC

Reversible Jump Markov Chain Monte Carlo (RJMCMC) extends the Metropolis-Hastings algorithm to a union space so that posterior inference can move between dimensions (such as by adding or removing a mixture model component). RJMCMC allows for model selection to be carried out as part of posterior inference.

Let u be a set of proposed random variables drawn according to $u \sim q(u \mid x, y)$. Let $x', u' = h(x, u)$ for some deterministic, invertible function h, where $x \in \mathbb{R}^D, x' \in \mathbb{R}^{D'}, u \in \mathbb{R}^U, u' \in \mathbb{R}^{U'}$ such that the input and output dimensions match: $D+U = D'+U'$. Then, [78] shows that the acceptance probability,

$$\min\left(1, \frac{p(x' \mid x, y)\; q(u \mid x', y)}{p(x \mid x', y)\; q(u' \mid x, y)} \left|\frac{\partial h(x, u)}{\partial (x, u)}\right|\right) \tag{2.38}$$

satisfies detailed balance with respect to posterior $p(x \mid y)$. The stan-

dard MH acceptance probability (Equation 2.14) is recovered when h has unit Jacobian determinant.

Algorithm 3: Slice Sampler for Dirichlet Process Mixtures

Input : $y, x = (z, \pi, \theta)$
Output: z', π', θ'

for $n \in 1, \ldots, N$ **do** Sample $u_n \sim \text{Unif}(0, \pi_{z_n})$
Let $u^* = \max_n u_n$
Let $K^* = |\pi|$ be the current number of components
/* Recover beta variables */
for $k \in 1, \ldots, K^*$ **do** Let $v_k = \pi_k / \prod_{i=1}^{k-1}(1 - v_i)$
Let $\beta^* = \prod_{k=1}^{K^*}(1 - v_k)$ be the remaining stick length
/* Instantiate additional components */
while $\beta^* < u^*$ **do**
    Let $K^* \leftarrow K^* + 1$
    Sample $v_{K^*} \sim \text{Beta}(1, \alpha), \theta_{K^*} \sim H$
    Let $\pi_{K^*} = v_{K^*}\ \beta^*$
    Let $\beta^* \leftarrow \beta^*(1 - v_{K^*})$
/* Sample new associations */
for $n \in 1, \ldots, N$ **do** Sample $p(z_n = k) \propto f_k(y_n)\delta_{\pi_k \geq u_n}$
for $k \in 1, \ldots, K^*$ **do**
    Let $n_k = |\{z_n \ : \ z_n = k\}|$
    if $n_k = 0$ **then**
        /* Remove empty components */
        $\theta \leftarrow \theta \setminus \theta_k$
        $\pi \leftarrow \pi \setminus \pi_k$
    else
        /* Sample θ_k posterior, update mixture weights */
        Sample $\theta_k \sim p(\theta_k \mid y, z)$
        Sample $\tilde{v}_k \sim \text{Beta}(1 + n_k, \alpha + \sum_{j=1}^{K^*} n_j$
        Let $\pi_k = \tilde{v}_k \prod_{i=1}^{k-1}(1 - \tilde{v}_i$
return $x' = (z, \pi, \theta)$

2.2.6 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) augments the latent space $x \in \mathbb{R}^D$ with randomly sampled momentum variables $u \in \mathbb{R}^D$ so that their joint distribution is,

$$p(x, u) = \frac{1}{Z}\ e^{-U(x|y)}\ e^{-K(u)} \tag{2.39}$$

where potential energy $U(x \mid y) = -\log \tilde{p}(x \mid y)$ and kinetic energy $K(u) = \text{N}(u \mid 0, M)$ so the $U(x \mid y) + K(u)$ define a Hamiltonian, which is simulated for L timesteps to propose a new augmented state (x', u')

that is accepted with probability:

$$\min\left(1, e^{H(x,u)-H(x^*,u^*)}\right) \tag{2.40}$$

Simulation requires the ability to evaluate $\frac{\partial U(x)}{x}$, the gradient of the negative log unnormalized posterior; hence, HMC can only sample from continuous distributions. When Hamiltonian dynamics can be simulated exactly, the acceptance ratio will always be one. Otherwise, simulation requires careful selection of an integrator, with the leapfrog integrator being a common choice. For more details and extensions, see [146, 25, 150, 91].

2.3 Probability Distributions

The generating processes of Bayesian models are composed of functional relationships between random variables that can be compactly described by conditional probability distributions. This section introduces probability distributions that are commonly used throughout this work, including their probability density function (PDF) if continuous or probability mass function (PMF) if discrete. First and second moments are conveyed without derivation, as are useful properties or operations used throughout this work. Pointers are given to how each distribution is used in this work and notation is kept as consistent as possible with their later use. For more thorough coverage, see [73, 142].

2.3.1 Multivariate Gaussian

For $x, \mu \in \mathbb{R}^D$, $\Sigma \in \mathcal{P}(D)$, the PDF of the Multivariate Gaussian is,

$$\mathrm{N}(x \mid \mu, \Sigma) = (2\pi)^{-N/2}|\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right) \tag{2.41}$$

where μ is the mean of the distribution, Σ is the covariance, and $|\Sigma|$ is the determinant of Σ. Gaussian marginals are themselves Gaussian, and so are Gaussian conditionals, implying that Gaussians are conjugate with themselves. In particular, if $p(y \mid x) = \mathrm{N}(y \mid x, R)$ and $p(x)$ is defined as in Equation 2.41 then,

$$p(x \mid y) = \mathrm{N}\left(x \mid \mu_{x|y}, \Sigma_{x|y}\right) \tag{2.42}$$

$$\Sigma_{x|y}^{-1} = \Sigma^{-1} + R^{-1} \tag{2.43}$$

$$\mu_{x|y} = \Sigma_{x|y}\left(R^{-1}y + \Sigma^{-1}\mu\right) \tag{2.44}$$

$$p(y) = \mathrm{N}(y \mid \mu, \Sigma + R) \tag{2.45}$$

Multivariate Gaussians are extensively used throughout this work, particularly as a likelihood for parts modeling and tracking.

2.3.2 Inverse-Wishart

For $\Sigma, S \in \mathcal{P}(D), \upsilon > D-1 \in \mathbb{R}$, the PDF of the Inverse-Wishart distribution is,

$$\mathrm{IW}(\Sigma \mid S, \upsilon) = \frac{|S|^{\upsilon/2}}{2^{(\upsilon D)/2}\Gamma_D(\upsilon/2)}|\Sigma|^{-(\upsilon+D+1)/2}e^{-\frac{1}{2}\mathrm{tr}(S\Sigma^{-1})} \tag{2.46}$$

where Γ_D is the multivariate gamma function,

$$\Gamma_D(\upsilon/2) = \int_{R\in\mathcal{P}(D)} e^{-\mathrm{tr}(R)}|R|^{\frac{\upsilon}{2}-\frac{D+1}{2}} \tag{2.47}$$

and tr() is the trace of its matrix argument. The Inverse-Wishart has moments,

$$\mathbb{E}[\Sigma] = S/(\upsilon - D - 1) \tag{2.48}$$

$$\mathrm{Var}[\Sigma_{ij}] = \frac{(\upsilon-D+1)S_{ij}^2 + (\upsilon-D-1)S_{ii}S_{jj}}{(\upsilon-D)(\upsilon-D-1)^2(\upsilon-D-3)} \tag{2.49}$$

where Σ_{ij}, S_{ij} are the $1 \le i, j \le D$ elements of matrices Σ, S. The Inverse-Wishart is conjugate to a Multivariate Gaussian with known mean and unknown covariance. In particular, let $p(y \mid \mu, \Sigma) = \mathrm{N}(y \mid \mu, \Sigma)$ and let $p(\Sigma)$ be defined as in Equation 2.46. Then,

$$p(\Sigma \mid y) = \mathrm{IW}\left(\Sigma \mid S + (y-\mu)(y-\mu)^\top, \upsilon + 1\right) \tag{2.50}$$

It is also common to use a joint Normal-Inverse-Wishart prior, which is conjugate to Multivariate Gaussian likelihoods with unknown mean and covariance. For $\kappa > 0$,

$$\mathrm{NIW}(\mu, \Sigma \mid \mu_0, \kappa, S, \upsilon) = \mathrm{N}\left(\mu \mid \mu_0, \kappa^{-1}\Sigma\right)\ \mathrm{IW}(\Sigma \mid S, \upsilon) \tag{2.51}$$

but this forces uncertainty in the mean μ to be proportional to uncertainty in the covariance. Alternatively, unconditionally independent priors can be specified for the mean and covariance of a Multivariate Gaussian likelihood. In this case, conjugacy is lost but the full conditionals for the mean and covariance are separately conjugate,[16] and so can be analytically sampled as part of a Gibbs sampler. Models in this work benefit from keeping the prior mean and prior covariance unconditionally independent; hence, the Normal-Inverse-Wishart prior is not used as commonly as separate Normal and Inverse-Wishart priors. Inverse Wishart distributions are used throughout this work as prior distributions on noise covariances for latent dynamics and observation noise.

[16] Also known as semi-conjugacy, which occur when the full conditionals of a collection of priors are separately but not jointly conjugate with the likelihood

2.3.3 Dirichlet and Beta

Let $\Pi_K = \left\{\pi_{1:K} : 0 \le \pi_k \le 1, \sum_{k=1}^K \pi_k = 1\right\}$ be the probability simplex in K dimensions. Then, the Dirichlet distribution with concentration

parameter $\alpha \in \mathbb{R}^K$ such that $0 \leq \alpha_k \leq 1$ has support on the probability simplex with PDF,

$$\text{Dir}(\pi \mid \alpha) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \tag{2.52}$$

where Γ is the univariate special case of the multivariate gamma function (Equation 2.47). The Dirichlet has moments,

$$\mathbb{E}[\pi_j] = \frac{\pi_j}{\sum_{k=1}^{K} \pi_k} \tag{2.53}$$

$$\text{Var}[\pi_j] = \frac{\pi_j - 1}{\sum_{k=1}^{K} \alpha_k - K} \tag{2.54}$$

for $1 \leq j \leq K$. The Dirichlet is a distribution over K-dimensional discrete distributions and is often used as a conjugate prior to the Multinomial and Categorical distributions, defined below.

When $K = 2$, the Dirichlet distribution is known as the Beta distribution with parameters $a = \alpha_1, b = \alpha_2$. While $\pi \in \Pi_2$, the Beta distribution's support is over $0 \leq \pi_1 \leq 1$ since $\pi_2 = 1 - \pi_1$ can be determined from π_1. The Beta PDF is:

$$\text{Beta}(\pi_1 \mid a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \pi_1^{a-1} (1 - \pi_1)^{b-1} \tag{2.55}$$

The Beta distribution is conjugate to Binomial distribution and marginals of the Dirichlet are beta-distributed,

$$\pi_k = \text{Beta}\left(\pi_k \mid \alpha_k, -\alpha_k + \sum_{k=1}^{K} \alpha_k\right) \tag{2.56}$$

In this work, the Dirichlet distribution is used as a prior distribution over mixture model weights for parts modeling and the Beta distribution is used in one of several constructions of the Dirichlet Process.

2.3.4 Binomial

For $n \geq 0 \in \mathbb{Z}$ the number of trials of a random binary event, each with independent probability of success $0 \leq p \leq 1$, the Binomial distribution models the number of successes k in n trials. It has PMF,

$$\text{Bin}(k \mid n, p) = \binom{n}{k} p^k (1-p)^{n-k} \tag{2.57}$$

where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the binomial coefficient. It has moments,

$$\mathbb{E}[k] = np \tag{2.58}$$

$$\text{Var}[k] = np(1-p) \tag{2.59}$$

The binomial distribution is used in this work to model random object arrivals and departures in multi-object tracking.

2.3.5 Multinomial and Categorical

Let $n \geq 0 \in \mathbb{Z}$ be the number of trials of a random event with K possible outcomes. Let $\pi \in \Pi_K$ be the K probabilities of each outcome, independent across trials. Let $z = (z_1, \ldots, z_K)$ model the counts of each outcome. Then, z is distributed according to the multinomial distribution, which has PMF:

$$\text{Mul}(z \mid n, \pi) = \frac{n!}{z_1! \cdots z_K!} \prod_{k=1}^{K} \pi_k^{z_k} \tag{2.60}$$

The multinomial has moments,

$$\mathbb{E}[z_k] = n\pi_k \tag{2.61}$$

$$\text{Var}[z_k] = n\pi_k(1 - \pi_k) \tag{2.62}$$

The multinomial is the generalization of the binomial distribution from two outcomes to $K \leq 2$. Additionally, when the there is only a single trial ($n = 1$), then we call it the categorical distribution,

$$\text{Cat}(z \mid \pi) = \text{Mul}(z \mid 1, \pi) \tag{2.63}$$

Categorical random variables are extensively used throughout this work to model associations of observations to objects or parts.

2.3.6 Poisson

For rate $\lambda \in (0, \infty)$, the Poisson distribution models the number of events a that occur within a fixed interval. It has PMF,

$$\text{Pois}(a \mid \lambda) = \frac{\lambda^a e^{-\lambda}}{a!} \tag{2.64}$$

with moments, $\mathbb{E}[a] = \text{Var}[a] = \lambda$. The Poisson is the limit of the Binomial distribution when the probability of success in each trial is $\frac{n}{\lambda}$ and the number of trials $n \to \infty$. In this work, the Poisson distribution is used to model random object arrivals and clutter detections in multi-object tracking.

2.3.7 Dirichlet Process

The Dirichlet Process (DP) is a family of stochastic processes[17] over the space of probability distributions. It is similar to the Dirichlet distribution in that its samples can be interpreted as discrete distributions,[18] but they exist in spaces more general than the probability simplex that Dirichlet samples are limited to. They were first developed in [60].

[17] Stochastic processes are distributions over function spaces. Probability distributions are functions defined on some space Ω that are everywhere positive and which integrate to unity.

[18] Technically, DP samples are discrete almost surely; that is, the set of exceptions has null probability.

The Dirichlet Process is a more complicated object than other distributions in this chapter. We begin by defining relevant notation and conditions for a random variable to be distributed according to the Dirichlet Process. Let $\alpha \in \mathbb{R}^+$ and H be any probability distribution with parameters ϕ and support over the space Ω. Let $A_1, \ldots, A_K$ be any finite partitioning of Ω:

$$\Omega = \bigcup_{k=1}^{K} A_k \qquad \text{such that } A_i \cap A_j = \emptyset \ \ \forall i \neq j \tag{2.65}$$

Let G be a discrete distribution, and for all k, let $G(A_k), H(A_k)$ denote the probability mass assigned to $A_k \subset \Omega$ under the distributions H, G, respectively. Then, we say G is distributed according to the Dirichlet Process with concentration α and base measure H if:

$$G(A_1), \ldots G(A_K) \sim \text{Dir}\left(\alpha H(A_1), \ldots, \alpha H(A_K)\right) \tag{2.66}$$

That is, for every finite partitioning of Ω, the marginals of G are jointly distributed according to the Dirichlet distribution whose k^{th} parameter is the probability mass of A_k under distribution H scaled by the DP concentration parameter α. Notationally, we write:

$$G \sim \text{DP}(\alpha, H) \tag{2.67}$$

For any $A \subset \Omega$, the first two moments of G are,

$$\mathbb{E}[G(A)] = H(A) \qquad \text{Var}[G(A)] = \frac{H(A)(1 - H(A))}{\alpha + 1} \tag{2.68}$$

so that the expected value of G is the expected value of H, and samples from G concentrate around its mean as α gets larger. Figure 2-3 depicts the Dirichlet-distributed marginals of the Dirichlet Process and Figure 2-2 depicts

$$G \sim \text{DP}\left(\alpha, H\right) \qquad y = (y_1, \ldots, y_N) \sim G \tag{2.69}$$

then the posterior on G is,

$$p(G \mid y_1, \ldots, y_N) = \text{DP}\left(\alpha + N, \frac{\alpha}{\alpha + N} H + \frac{1}{\alpha + N} \sum_{n=1}^{N} \delta_{y_n}\right) \tag{2.70}$$

where δ_{y_n} is the Dirac delta function, which can be interpreted as a distribution with unit mass at y_n. The base measure of the DP posterior interpolates between the base measure H weighted by $\alpha/(\alpha + N)$ and the empirical distribution of observations y weighted by $1/(\alpha + N)$. Clearly, the Dirichlet Process is conjugate with itself.

The generative process in Equation 2.69 is purely descriptive. It cannot be simulated so straightforwardly because G, although discrete,

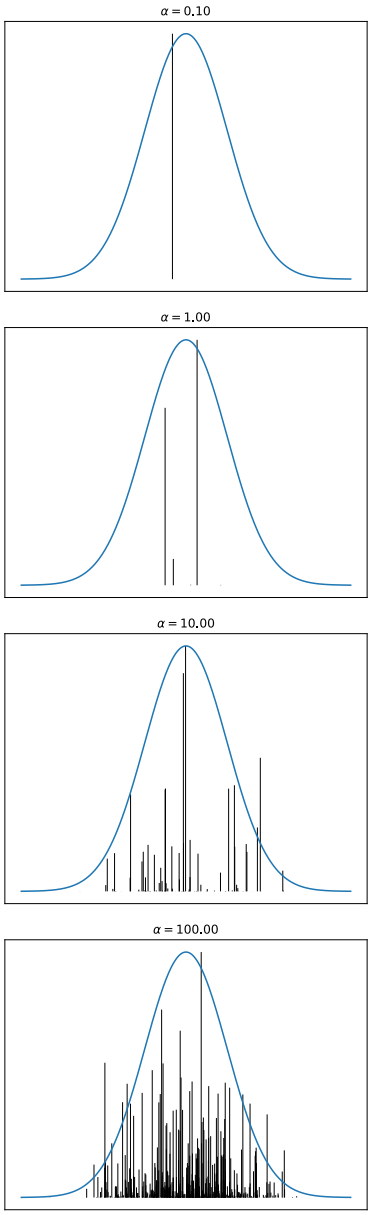


Figure 2-2: Visualizing unique draws from the Dirichlet Process, $G \sim \text{DP}(\alpha, H)$ for base measure $H = \text{N}(0, 1)$ and varying α. Larger α cause draws to concentrate on base measure H. Each unique draw is a line with height proportional to π_k in Equation 2.76.

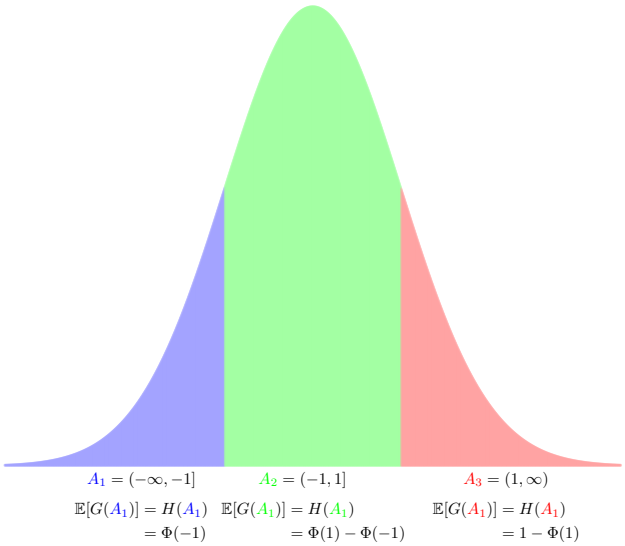


Figure 2-3: Visualizing marginals of the Dirichlet Process, $G \sim \text{DP}(\alpha, H)$ for base measure $H = \text{N}(0, 1)$ with CDF $\Phi(x) = \int_{-\infty}^{x} \text{N}(x \mid 0, 1)$ and any arbitrary partitioning $A_1, \ldots, A_K$ of $\Omega = \mathbb{R}$. The proportion of atoms in distribution G that are contained in region A_k is Dirichlet-distributed with corresponding parameter $\alpha H(A_k)$.

contains an infinite number of atoms. Several constructions exist that enable manipulations of the Dirichlet Process, however. The Blackwell-MacQueen Urn Scheme [27] derives the predictive distribution $p(y_{N+1} \mid y_{1:N}) = \int_G p(y_{N+1} \mid G)\, p(G \mid y_1, \ldots y_N)$ to be,

$$y_{N+1} \mid y_{1:N} \sim \frac{\alpha}{\alpha + N} H + \frac{1}{\alpha + N} \sum_{n=1}^{N} \delta_{y_n} \tag{2.71}$$

which can be easily sampled from: with probability $\frac{1}{\alpha+N}$ draw from the discrete distribution with atoms at $y_1, \ldots, y_N$, and with probability $\frac{\alpha}{\alpha+N}$ draw a new atom from H with parameters ϕ. For N large enough, there will be repeated draws so that $y_i = y_j$ for some $i \neq j$. This forms the basis for using the Dirichlet Process as a prior for nonparametric mixture models; draws that are repeated many times are interpreted as clusters with significant weight. See Chapter 2.4 for more discussion. Note also that the posterior base measure in Equation 2.70 and the predictive distribution for the $(N+1)^{\text{th}}$ observation are the same distribution. Thus, G cannot be sampled, but it does not need to be if we are actually interested in reasoning over its posterior or on observations y.

So far we have described properties of the Dirichlet Process, but it may be that no distribution exists which satisfies them. Using Equation 2.71, it can be shown that observations y are infinitely exchangeable,[19] in which case by de Finetti's theorem [52] that there exists a prior on G such that the sequence of observations y are iid draws. That prior is the Dirichlet Process.

In this work, the Dirichlet Process is used as a prior distribution

[19] Random sequence y is infinitely exchangeable if $p(y) = p(\sigma(y))$ where $\sigma(y)$ denotes any permutation of a finite number of terms in y.

on infinite mixture models, notably for parts modeling (3.4) and tracking (5.3.1). Additional treatment is given in Chapter 2.4 and a more thorough introduction is given by [192].

2.4 Nonparametric Mixtures

Mixture models can be interpreted as inferring group-level properties from population-level observations.[20] One example of group/population decompositions used in this work is time-varying point clouds of a moving object (population) and the articulated motion of that object's unknown parts (groups). Another example is video frames showing multiple objects in motion over time (population), the motion of individual objects (group), and the parts decomposition of each object (sub-groups). Throughout this dissertation, groups, clusters, and components are used interchangeably.

[20] Another common use of mixture models is density estimation.

2.4.1 Finite Mixtures

The simplest Bayesian mixture models are finite, meaning that they have a pre-determined number of groups, K. They can be described by the following generative model,

$$\pi \sim \mathrm{Dir}(\pi \mid \alpha) \qquad \theta_k \sim \mathrm{H}(\theta_k \mid \phi) \tag{2.72}$$

$$z_n \sim \mathrm{Cat}(z_n \mid \pi) \qquad y_n \sim \mathrm{F}\left(y_n \mid \theta_{z_n}\right) \tag{2.73}$$

where H is a prior distribution on group parameters with hyperparameters ϕ and F is some observation model with group parameters θ_k. Indices $n = 1, \ldots, N$ index observations and their associations while $k = 1, \ldots, K$ index groups and their parameters. One inference problem is to infer the joint distribution of mixture weights $\pi = \pi_1, \ldots, \pi_K$, associations $z = (z_1, \ldots, z_N)$, and cluster parameters $\theta = (\theta_1, \ldots, \theta_K)$ given observations $y = (y_1, \ldots, y_N)$ and hyperparameters α, ϕ. An equivalent generative model that is commonly expressed in the literature and can be the basis for confusion is,

$$\pi \sim \mathrm{Dir}(\pi \mid \alpha) \qquad \theta_n \sim G \qquad y_n \sim F(y_n \mid \theta_n) \tag{2.74}$$

where G is a discrete distribution with $k = 1, \ldots, K$ point masses, each centered at δ_{θ_k} with weight π_k where θ_k is sampled as in Equation 2.72. In this formulation, the associations are implicit and would typically be instantiated to perform inference. These two models can be represented by the graphical models in Figure 2-4.

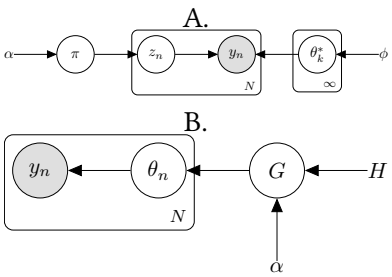


Figure 2-4: Two equivalent mixture model representations: A. emphasizes the associations z_n while B. emphasizes the base measure H.

2.4.2 Finite Mixture Inference

The posterior distribution,

$$p(\theta, z, \mu \mid y) \tag{2.75}$$

cannot be sampled from analytically, nor does an analytic formula exist for a MAP estimate. However, MAP estimation can still be iteratively performed using the Expectation-Maximization (EM) algorithm [50, 221, 17], which separates variables into two groups: parameters and missing data. EM alternates between inferring the posterior of the missing data ($z \mid y, \theta$ above) and computing point estimates for the parameters ($\pi, \theta \mid y, z, \phi, \alpha$ above). EM is a special case of variational Bayes approximate inference [103, 95, 75, 92] where parameter approximating distributions are point masses and the missing distributions are unconstrained [73]. More generally, exact inference can be performed using Gibbs Sampling (Chapter 2.2), which is particularly simple when F, H are conjugate (as when F is Gaussian and H is Normal-Inverse-Wishart). We will discuss sampling-based inference after covering nonparametric mixture models.

2.4.3 Mixture Models of Unknown Size

The number of groups K is not always known and it may not always be reasonable to assume that there are a fixed number of groups. One approach to reason over K is to separately perform inference over a set of finite mixture models, each with differing $K \in \mathcal{K}$. Models can then be compared using Bayes factors (if the evidence $p(y)$ can be estimated) or by approximate measures like the Bayesian Information Criterion (BIC) [179]. Another approach is to place a prior on the number of components and perform RJMCMC inference (Chapter 2.2) so that model selection is built into posterior inference. Any prior with support on $\mathbb{Z}^+$, such as the Uniform or Poisson distributions—can be used, but exploration of latent spaces with variable dimension remains a challenging problem when no special structure can be assumed [137].

Another approach to mixture modeling with an unknown number of components, used extensively in this dissertation, is to employ a Dirichlet Process prior on group parameters. In Chapter 2.3, we showed the Blackwell-MacQueen Urn construction, which derived the predictive distribution of observations from a Dirichlet Process. Observe from Equation 2.71 that there will be draws from the base measure with the same value, which holds even if H is continuous. Moreover, when drawing the $(N+1)^{\text{th}}$-observation, the probability of drawing a non-unique value is proportional to the number of times that value has already been drawn, $N/(\alpha + N)$, so that a small number of unique draws explain most of the observations as N grows. This motivates the stick-breaking construction [182] of the Dirichlet Process, sketched below.

Let H be any distribution with parameters ϕ and let $\alpha \in \mathbb{R}^+$. Define $\beta = \{\beta_k\}_{k=1}^{\infty}$ where $\beta_k \sim \text{Beta}(1, \alpha)$. Then, we say,

$$\pi \sim \text{GEM}(\alpha) \quad \text{iff} \quad \pi_k = \beta_k \prod_{i=1}^{k-1} (1 - \beta_k) \tag{2.76}$$

for all $k = 1, \ldots, \infty$. GEM stands for Griffifths, Engen, and McCloskey;

it's history is discussed in [164]. Notably, it is a common notation when working with Dirichlet Processes. The stick-breaking construction of the Dirichlet Process shows that if,

$$\pi \sim \text{GEM}(\alpha) \quad \theta_k^* \sim H\left(\theta_k^* \mid \phi\right) \quad G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*} \tag{2.77}$$

then $G \sim \text{DP}(\alpha, H)$. The θ_k^* are unique draws from base measure H. The number of times they have repeatedly been drawn is proportional to π_k. With this construction, it is straightforward to adopt the Dirichlet Process as a nonparametric prior on group parameters in an infinite mixture model. For $n = 1, \ldots, N$ and $k = 1, \ldots, \infty$,

$$\pi \sim \text{GEM}(\alpha) \qquad \theta_k^* \sim \text{H}\left(\theta_k^* \mid \phi\right) \tag{2.78}$$

$$z_n \sim \pi \qquad y_n \sim \text{F}\left(y_n \mid \theta_{z_n}\right) \tag{2.79}$$

which can equivalently be expressed as,

$$\pi \sim \text{GEM}(\alpha) \qquad \theta_n \sim G \qquad y_n \sim F\left(y_n \mid \theta_n\right) \tag{2.80}$$

where $G \sim \text{DP}(\alpha, H)$. These definitions closely correspond to Equations 2.72–2.74 and they also have the same graphical models, except that K is replaced with ∞. The existence of repeated draws is often referred to as the *clustering property* of the Dirichlet Process because they are what enable the DP to be a prior over infinite mixture models. Mixture models with a Dirichlet Process prior are often called infinite mixture models or Dirichlet Process Mixture Models (DPMM).

DPMM Inference would be intractable if an infinite number of group parameters θ_k^* and mixture weights π_k had to represented but, like observations from a Dirichlet Process sample, there are techniques that can reason exactly over an infinite number of items using only finite representations. We discuss these next.

2.4.4 Inference for Dirichlet Process Mixture Models

Early approaches to DPMM inference were Gibbs samplers that either approximated an infinite number of atoms by finite truncation, relying on the Dirichlet marginals of the DP prior as well as well as the property that the number of unique draws grows logarithmically with observed data (conditional methods), or they utilized the DP predictive distribution (Equation 2.71) along with a conjugate prior on group parameters [56, 127, 94] (marginal methods). In marginal approaches, mixture weights and group parameters are marginalized out in closed form, leaving a stationary distribution that consists of associations z, which are finite. Gibbs updates have the form,

$$p\left(z_n = k \mid z_{-n}, \phi, \alpha\right) \propto p(z_n = k \mid z_{-n}, \alpha)\, p(y_n \mid y_{-n}, z, \phi) \tag{2.81}$$

where the first term is a posterior predictive on associations,

$$p(z_n = k \mid z_{-n}, \alpha) = \begin{cases} \frac{N_{k\setminus n}}{N+\alpha} & \text{if } N_{k\setminus n} > 0 \\ \frac{\alpha}{N+\alpha} & \text{if } N_{k\setminus n} = 0 \end{cases} \tag{2.82}$$

which is of the same form as the DP predictive distribution (Equation 2.71) where $N_{k\setminus n}$ is the count of observations associated to group k not including observation n. The condition $N_{k\setminus n} = 0$ handles the case that the observation is explained by one of the (infinite) groups that does not yet have associations. Let $Y_{k\setminus n} = \{y_i \ : \ z_i = k \, , 1 \leq i \leq N, i \neq n\}$. Then, the second term is a posterior predictive for observation y_n conditioned on all observations,

$$p(y_n \mid y_{-n}, z, \phi) = p(y_n \mid Y_{k\setminus n}, \phi) \tag{2.83}$$

$$= \int_{\theta_k^*} F(y_n \mid \theta_k^*) \, H(\theta_k^* \mid Y_{k\setminus n}, \phi) \tag{2.84}$$

which has closed form for F, H conjugate, and is simply the prior predictive distribution if $Y_{k\setminus n} = \emptyset$, in which case $N_{k\setminus n} = 0$. The sampler proceeds by iteratively sampling each association z_n in a randomized order, instantiating a new association label anytime z_n is assigned to what was previously an empty cluster.

Exact samplers have been devised for the DPMM when conjugacy is not available, including retrospective sampling [156], slice sampling and auxiliary variable methods [209, 105, 143] and RJMCMC [79, 96, 42]. Approximate samplers include using Monte Carlo estimates of the prior and posterior predictives [214] and variational inference [28].

2.4.5 Identifiability

A challenge with sampling-based inference in mixture models is identifiability. In particular, $p(y \mid \theta, z) = p\,(y \mid \sigma(\theta), \sigma(z))$ where $\sigma(\theta)$ is any permutation of $\theta_1, \ldots, \theta_K$ and $\sigma(z)$ is the corresponding permutations on association indices k in $z_n = k$ for all $n = 1, \ldots, N$. In words, groups can be permuted with no change in the likelihood. Figure 2-5 depicts this graphically. This induces $K!$ modes into the posterior and its possible that two distinct posterior samples come from separate modes, meaning that their group parameters (and association labels) are permuted and so not directly comparable. This problem does not arise in EM or variational Bayes because, as optimization approaches, they lock onto a single mode.[21] Given a set of posterior samples from a mixture model, there are several approaches to identifying groups between samples: the simplest is to establish an ordering on group parameters post-hoc by relabeling the groups in ascending order of their D dimensional location parameters (assuming they have location parameters, as when the groups are Gaussian-distributed). Use of non-symmetric priors can also alleviate the problem when they can meaningfully be used [97]. Alternatively, a mapping can be computed between group

[21] Use of the output of multiple optimizations with different initializations can also cause identifiability problems in mixture model inference.

parameters by defining a suitable matching objective [38, 139].

2.5 State Space Models

State space models are used to reason about time-evolving data. Let there be $t = 1, \ldots, T$ discrete timesteps. At time t, there are observations y_t and latent variables x_t. Latent variables evolve from time t to $t+1$ according to dynamics f, and observations y_t are observed through observation model h. Both f, h typically include random innovation so that a common formulation is,

$$x_t = f(x_{t-1}, q_t) \qquad q_t \overset{iid}{\sim} p(q_t) \tag{2.85}$$

$$y_t = h(x_t, r_t) \qquad r_t \overset{iid}{\sim} p(r_t) \tag{2.86}$$

where $p(q_t)$ is some distribution on dynamics noise and $p(r_t)$ is some distribution on observation noise. If the latent space is discrete, this model is commonly called a Hidden Markov Model (HMM). If the latent state is continuous, f, h are linear, and $p(q_t), p(r_t)$ are zero-mean Gaussians, then it is commonly called a linear dynamical system (LDS). The functional specification of Equations 2.85–2.86 can be written in a the following probabilistic form for an LDS,

$$p(x_t \mid x_{t-1}) = \mathrm{N}(x_t \mid Fx_{t-1}, Q) \tag{2.87}$$

$$p(y_t \mid x_t) = \mathrm{N}(y_t \mid Hx_t, R) \tag{2.88}$$

where $x_t \in \mathbb{R}^{d_x}, y_t \in \mathbb{R}^{d_y}$, $Q \in \mathbb{R}^{d_x \times d_x}$ is the dynamics noise covariance, $R \in \mathbb{R}^{d_y \times d_y}$ is the observation noise covariance, $F \in \mathbb{R}^{d_x \times d_x}$ is the linear dynamics represented as a matrix, and $H \in \mathbb{R}^{d_y \times d_x}$ is the linear observation model represented as a matrix. This system depicts a order $M = 1$ Markov model for latent dynamics because the latent state only depends on the previous time's latent state. Higher-order $M > 1$ dependencies can be incorporated but are equivalently represented by a first-order model whose latent state and dynamics is augmented to account for the previous M states.

The models used in this dissertation have mixed continuous and discrete latent latent spaces, so the general term dynamical system is used to refer to any model that can be specified by Equations 2.85–2.86. These can, in general, be depicted by the graphical model in Figure 2-6.

The dynamics and observation models can take many forms. A form widely used in tracking applications is the random acceleration model where $d_x = 2d_y$ and,

$$F = \begin{pmatrix} I & \delta I \\ 0 & I \end{pmatrix} \qquad H = \begin{pmatrix} I & 0 \end{pmatrix} \qquad Q = \begin{pmatrix} 0 & 0 \\ 0 & q \end{pmatrix} \tag{2.89}$$

where $I, 0 \in \mathbb{R}^{d_y}$ are block identity or zero matrices, respectively, $\delta > 0$ indicates the time difference between discrete timesteps (usually $\delta =$

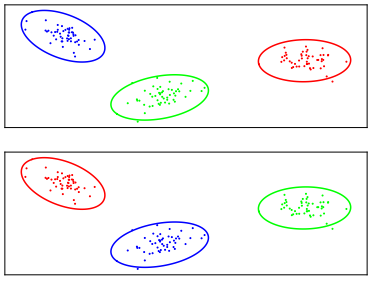


Figure 2-5: Mixture models likelihoods (and hence, posteriors) are not invariant to label permutations. The above samples have the same likelihood and posterior probability.

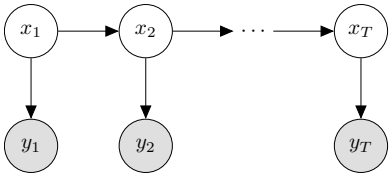


Figure 2-6: Graphical model of a dynamical system for times $t = 1, \ldots, T$. Latent states x_t evolve with dynamics f and measurements y_t are observed according to $h(x_t)$.

1), and $q \in \lceil_\dagger$. The latent state is interpreted as a collection of position values followed by a collection of velocity values and the observed state is simply the projection of the latent position values. Innovations Q occur only on velocity, hence the name random acceleration. More complicated dynamics can be modeled (such as for estimating bearing from radar observations), see [122] for a survey.

2.5.1 Filtering and Smoothing

Perhaps the most common task in dynamical models is to estimate,

$$p(x_t \mid y_{1:t}) \propto p(y_t \mid x_t)\, p(x_t \mid y_{1:t-1}) \tag{2.90}$$

$$= p(y_t \mid x_t) \int_{x_{t-1}} p(x_t \mid x_{t-1})\, p(x_{t-1} \mid y_{1:t-1}) \tag{2.91}$$

which is the filter distribution of x_t conditioned on all data observed up to time t. The predictive distribution $p(x_t \mid y_{1:t-1})$ is defined in terms of $p(x_{t-1} \mid y_{1:t-1})$, which is the filter distribution at $t-1$; hence, the filter can be computed in recursive fashion starting from $t = 1$. The integral in Equation 2.91 is s special case of the Chapman-Kolmogorov equation.[22] The process of recursively estimating $x_t \mid y_{1:t}$ for all t is called filtering. It can be exactly computed in closed-form when the latent space is finite and discrete with complexity $O(Td_x^2)$, as well as when there are linear Gaussian dynamics with complexity $O(Td_x^2 d_y^3)$. Filtering in a linear Gaussian system is called Kalman Filtering [106]. Filtering is widely used in state estimation problems that require real-time performance.

[22] In a discrete setting, Chapman-Kolmogorov gives the probability of reaching state j from state i in L steps. Here, $L = 1$.

More accurate state estimates can be made if realtime performance isn't required by incorporating not future as well as past information into the estimate of x_t. In these *batch* settings, all data $y = y_{1:T}$ are observed before inference. The smoothing distribution,

$$p(x_t \mid y_{1:T}) = \int_{x_{t+1}} p(x_t \mid y_{1:T}, x_{t+1})\, p(x_{t+1} \mid y_{1:T}) \tag{2.92}$$

$$= \int_{x_{t+1}} p(x_t \mid y_{1:t})\, p(x_{t+1} \mid y_{1:T}) \tag{2.93}$$

can be estimated, where the first term in Equation 2.93 is the filter distribution (Equation 2.91) and the second term is the smoothing distribution at time $t+1$. This can be computed by a first forward pass that computes and stores the filter distributions $p(x_t \mid y_{1:t})$ for all t, then, in a second backwards pass, recursively computes the smoothing (distribution Equation 2.93), which can be accomplished in closed form for HMMs and linear Gaussian models.

2.5.2 Joint Sampling

The filter and smoothing distributions reason over the *marginal* states x_t. Independent samples from each of the T marginals, $x_t \sim y_{1:T}$, can

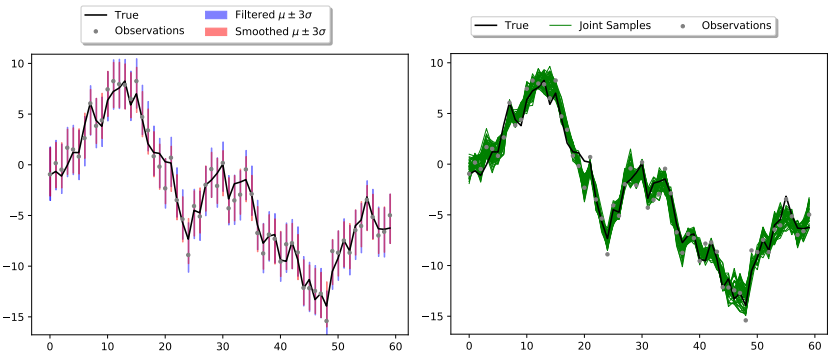


Figure 2-7: Marginal (filtered, smoothed) compared to joint state estimates.

be combined, but they may be jointly improbable or even impossible in some models. When dependence matters, inference can be performed on the joint distribution,

$$p(x_{1:T} \mid y_{1:T}) = \prod_{t=1}^{T} p(x_t \mid x_{t+1:T}, y_{1:T}) \tag{2.94}$$

$$= \prod_{t=1}^{T} p(x_t \mid x_{t+1}, y_{1:t}) \tag{2.95}$$

$$= \prod_{t=1}^{T} \frac{p(x_t \mid y_{1:t})\, p(x_{t+1} \mid x_t, y_{1:t})}{p(x_{t+1} \mid y_{1:t})} \tag{2.96}$$

$$= \prod_{t=1}^{T} \frac{p(x_t \mid y_{1:t})\, p(x_{t+1} \mid x_t)}{p(x_{t+1} \mid y_{1:t})} \tag{2.97}$$

where the RHS in Equation 2.94 follows from the product rule (Equation 2.4), the third line follows from Bayes rule (Equation 2.4), and the second and fourth lines simplify based on conditional independence. The denominator $p(x_{t+1} \mid y_{1:t})$ is another instance of the Chapman-Kolmogorov equation, and is constant due to conditioning on x_{t+1} in Equation 2.95. Figure 2-7 distinguishes filtered, smoothed, and joint estimates. This work emphasizes sampling from a joint posterior over all latent variables; hence, joint state estimates are favored over filtered or smoothed estimates.

Joint samples can be drawn using Forward-Filtering Backward Sampling (FFBS) by first computing and storing the filter distribution (Equation 2.91) over all times t, and then recursively sampling joint states starting from time T and working backwards. Like smoothing, this can be performed analytically for HMMs and for linear Gaussian systems.

The filter distribution cannot typically be represented in analytic form when there are nonlinear dynamics. In these cases, there are many approximate inference approaches that can be used to estimate the filter or smoothing distributions. These include the Extended Kalman Filter [201], Unscented Kalman Filter [211] and Particle Filter [53].

Joint samples can be drawn using Gibbs sampling of each condi-

tional, though surprisingly I show in Appendix A.1 that Gibbs sampling is not ergodic for the common case of a linear Gaussian system with random acceleration dynamics. If the whole state space is continuous, then joint samples can be drawn using HMC or NUTS. [9] proved that unbiased estimates of the sampled density could be used within MCMC proposals and the stationary distribution would be maintained. This led to a class of methods that use particle methods as proposals within MCMC, including Particle MCMC/Gibbs [10], and their extensions [124].

2.6 Lie Groups

In what follows, operations and properties of Matrix Lie groups that are necessary for probabilistic reasoning are covered. See [82] for a thorough introduction to Matrix Lie groups that does not require background in manifolds, [118] for a general treatment of manifolds including Lie groups, [43, 4] for numerical methods on Lie groups and manifolds, and [55] for an accessible tutorial on Lie groups.

A Lie group G is a continuous space equipped with a binary operator that satisfies closure, associativity, existence of identity, and existence of an inverse. A collection of bijective maps called charts,

$$\{(U_l, \rho_l)\}_{l=1}^{L} \quad \rho_l \ : \ U_l \to \mathbb{R}^K \quad U_l \subset G \quad G = \bigcup_{l=1}^{L} U_l \tag{2.98}$$

can be defined so that the coordinates of overlapping charts i, j are differentiable with respect to the coordinates of all other charts k, making G a differentiable manifold in addition to having group structure.

This work uses Matrix Lie groups $\mathrm{SO}(D)$ and $\mathrm{SE}(D)$, the groups of proper rotations and rigid transformations in D dimensions, respectively. Elements of these groups can be represented as,

$$\mathrm{SO}(D) = \left\{ R \in \mathbb{R}^{D \times D} \ : \ |R| = 1, R^\top R = R R^\top = I \right\} \tag{2.99}$$

$$\mathrm{SE}(D) = \left\{ (R, d) \ : \ R \in \mathrm{SO}(D), \ d \in \mathbb{R}^D \right\} \tag{2.100}$$

where $|R|$ is the determinant of matrix R and I is the D-dimensional identity matrix. Elements of $\mathrm{SE}(D)$ can additionally be represented as block matrices,

$$\mathrm{SE}(D) = \left\{ \begin{pmatrix} R & d \\ 0 & 1 \end{pmatrix} \ : \ R \in \mathrm{SO}(D), d \in \mathbb{R}^D \right\} \tag{2.101}$$

where $0 \in \mathbb{R}^{1 \times D}$ and $1 \in \mathbb{R}^{1 \times 1}$. The group operation for any Matrix Lie group G is matrix multiplication so that for $b, c \in G$ we have $bc \in G$ and $cb \in G$, but $bc \neq cb$. Elements of G can be interpreted as a basis, also called a frame of reference, so that bc acts as a change of basis. Appendix A.5 expands on this interpretation since it is easily misunderstood.

Elements of G do not form a vector space, nor do are they equipped with a distance metric. This precludes the direct use of probability distributions. One can straightforwardly place independent distributions on each matrix element or on the vectorization of the matrix representation of each element, but pathologies quickly arise: sampled values will no longer be in G and likelihoods do not correspond with human intuition of closeness.

2.6.1 Lie Algebra and Tangent Space

Reasoning can instead be performed in a space defined locally about the group's identity element using the Lie group logarithm map,

$$\log_G \ : \ G \to \mathfrak{g} \qquad \log_G b = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} (b - I)^k \tag{2.102}$$

which is the standard matrix logarithm and converges for all Lie groups considered in this work.[23] The space $\mathfrak{g}$ is called the Lie algebra associated with Lie group G. Elements of G and $\mathfrak{g}$ can be represented by matrices with shared dimension $\mathbb{R}^{M \times M}$ but different structure. There is a bijective mapping from the matrix representation of elements of $\mathfrak{g}$ to a vector representation in $\mathbb{R}^K$ but $K \neq M$ in general. The Lie algebra $\mathfrak{g}$ forms a K-dimensional vector space and has a standard set of basis vectors $G_1, \ldots, G_K$, also called generators, which can be represented by elements of $\mathbb{R}^{M \times M}$. We use the matrix and vector representations of elements of $\mathfrak{g}$ interchangeably throughout this work to reduce notational overhead.

[23] The Lie group logarithm map $\log_G b$ converges whenever $||b - I||_F \leq 1$.

Elements in $\mathfrak{g}$ can be mapped back to G through the Lie group exponential map,

$$\exp_G \ : \ \mathfrak{g} \to G \qquad \exp_G c = \sum_{k=0}^{\infty} \frac{1}{k!} c^k \tag{2.103}$$

which is the standard matrix exponential and converges for all Lie algebras considered in this work. We would like to reason locally in a vector space defined about an arbitrary element $\mu \in G$ so that we can define a distance metric on G. Existence of an inverse makes this possible since $\mu^{-1}\mu = I$. Define the left-invariant Riemannian logarithm map and left-invariant Riemannian exponential map as:

$$\text{Log} \ : \ G \times G \to \mathfrak{g} \qquad \text{Log}_\mu b = \log_G(\mu^{-1} b) \tag{2.104}$$

$$\text{Exp} \ : \ G \times \mathfrak{g} \to G \qquad \text{Exp}_\mu v = \mu \ \exp_G v \tag{2.105}$$

We can alternatively define the right-invariant Riemannian logarithm and exponential maps:

$$\widetilde{\text{Log}}_\mu b = \log_G(b \ \mu^{-1}) \qquad \widetilde{\text{Exp}}_\mu v = \exp_G(v) \ \mu \tag{2.106}$$

In some Lie groups G, left-invariance and right-invariance will be equiv-

alent so that they are called bi-invariant. In the context of SE(3), the choice of left- or right-invariance matters a great deal because it turns out that SE(D) has no bi-invariant metric [157]. The consequence of this is that a distance metric can be defined that is consistent when measured from an observer's point of view (left-invariant), or consistent when measured from the body frame of reference (right-invariant), but a distance metric cannot be defined which is consistent in both contexts. The applications in this work reason about the articulated motion of a moving object from the point of view of a world frame of reference; thus, we choose left-invariance.

For left-invariant Riemannian log and exponential maps, the following holds locally about element $\mu \in G$,

$$\mathrm{Log}_\mu \left(\mathrm{Exp}_\mu v \right) = \log_G \left(\mu^{-1} \mu \exp_G v \right) = v \tag{2.107}$$

$$\mathrm{Exp}_\mu \left(\mathrm{Log}_\mu b \right) = \mu \exp_G \left(\log_G \left(\mu^{-1} b \right) \right) = \mu \mu^{-1} b = b \tag{2.108}$$

so that Log and Exp can be *locally* treated as inverses of one another. Parameterizing by $\mu \in G$, we say,

$$\mathrm{Log}_\mu \ : \ G \longrightarrow T_\mu G \tag{2.109}$$

$$\mathrm{Exp}_\mu \ : \ T_\mu G \longrightarrow G \tag{2.110}$$

where $T_\mu G$, is the tangent space of element $\mu \in G$, which is isomorphic with the Lie algebra $\mathfrak{g}$ so that $T_\mu G$ is also a vector space, with the same basis vectors, as $\mathfrak{g}$.

2.6.2 Riemannian Metrics and Distributions

We can define inner products on the tangent space. For $\mu \in G$ and $u, v \in T_\mu G$, any inner product of the form,

$$\langle u, v \rangle_\mu = u^\top Q v \quad Q = \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix} \in \mathcal{P}(K) \tag{2.111}$$

is a Riemannian metric, which induces a distance metric on G,

$$D(\mu, b) = \sqrt{\left(\mathrm{Log}_\mu b \right)^\top Q \left(\mathrm{Log}_\mu b \right)} \tag{2.112}$$

In this work, we use the scale-dependent left-invariant metric on SE(D). For $b, \mu \in$ SE(D) with matrix representations,

$$\mu = \begin{pmatrix} R_\mu & d_\mu \\ 0 & 1 \end{pmatrix} \qquad b = \begin{pmatrix} R_b & d_b \\ 0 & 1 \end{pmatrix} \tag{2.113}$$

the scale-dependent left-invariant metric on SE(D) is,

$$D(\mu, b) = \sqrt{a || \mathrm{Log}_{R_\mu} R_b ||^2 + e || d_\mu - d_b ||^2} \tag{2.114}$$

for $a, e \in \mathbb{R}^+$ chosen to trade off differences in rotations with differences in translation distance.

Given a metric, we can define standard statistical notions included expected value and covariance. For $x_1, \ldots x_N \in G$,

$$\mathbb{E}[x_1, \ldots, x_N] = \underset{\mu \in G}{\operatorname{argmin}} \sum_{k=1}^{K} D(\mu, x_k)^2 \tag{2.115}$$

$$\operatorname{Cov}(x_1, \ldots, x_N) = \frac{1}{N} \sum_{k=1}^{K} \left(\operatorname{Log}_\mu x_k\right) Q \left(\operatorname{Log}_\mu x_k\right)^\top \tag{2.116}$$

which have familiar forms to their analogues in Euclidean space. One distinction is that the expected value may have multiple local minima. When the solution $\mu \in G$ to Equation 2.115 is a local minima, it is called a Karcher mean. When it is a global minima, it is called a Fréchet mean.

Continuous location-scale probability distributions on Lie group G can be straightforwardly defined once Riemannian log and exponential maps are chosen.[24] Location parameters are defined on G and scale parameters are defined in $T_\mu G$. The distribution is given support on G by mapping its argument to $T_\mu G$. The left-invariant concentrated Gaussian is defined in this way. Let G be a Matrix Lie group with K degrees of freedom so that $x, \mu \in G$ can be represented by an element of $\mathbb{R}^{M \times M}$ and elements of $\mathfrak{g}$ (equivalently, $T_\mu G$) can be represented by matrices $\mathbb{R}^{M \times M}$ or vectors $\mathbb{R}^K$. Then the left-invariant concentrated Gaussian is,

$$\mathrm{N}_\mathrm{L}\left(x \mid \mu, \Sigma\right) = \mathrm{N}\left(\operatorname{Log}_\mu x \mid 0, \Sigma\right) \tag{2.117}$$

where $\Sigma \in \mathbb{R}^{K \times K}$, $0 \in \mathbb{R}^K$.

Following [55], I provide the degrees of freedom K, the square dimension of the matrix representations of group and algebra elements M, the Lie algebra generators $G_1, \ldots, G_K$ and the Lie group log and exponential maps for groups used in this work.

[24] Comment about other retractions, including pseudo-log.

2.6.3 Group Forms

$SO(2)$ The group of proper rotations in 2D (K=1, M=2). Elements $b \in G$ are 2×2 rotation matrices which rotate a point on the circle. The Lie algebra has one generator:

$$G_1 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{2.118}$$

For $v = \theta G_1 \in \mathfrak{g}$ where $\theta \in \mathbb{R}$, the log and exponential maps are,

$$b = \exp_G v = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{2.119}$$

$$v = \log_G b = \arctan(b_{21}, b_{11})\, G_1 \tag{2.120}$$

SE(2) The group of rigid transformations in 2D ($K = 3, M = 3$). Elements $b \in G$ rotate then translate 2D homogeneous points. The Lie algebra has three generators,

$$G_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.121}$$

$$G_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.122}$$

so that:

$$v = xG_1 + yG_2 + \theta G_3 = \begin{pmatrix} \theta G_3 & u \\ 0 & 0 \end{pmatrix} \in \mathfrak{g} \tag{2.123}$$

where $u = (x, y)$. Note that G_3 is the same generator as is used in SO(2). Let $G' = \mathrm{SO}(2)$ and denote $b = \exp_G v$ as,

$$b = \begin{pmatrix} R_b & d_b \\ 0 & 1 \end{pmatrix} \in G \tag{2.124}$$

where $R_b \in \mathrm{SO}(2)$ is a 2×2 rotation matrix, $d_b \in \mathbb{R}^2$, $0 \in \mathbb{R}^{1\times 2}$, $1 \in \mathbb{R}^{1\times 1}$. Then, the the log and exponential maps have the form:

$$b = \exp_G v = \begin{pmatrix} \exp_{G'}(\theta G_3) & Vu \\ 0 & 1 \end{pmatrix} \tag{2.125}$$

$$v = \log_G b = \begin{pmatrix} \log_{G'} R_b & V^{-1} d_b \\ 0 & 1 \end{pmatrix} \tag{2.126}$$

$$V^{-1} = \frac{1}{A^2 + B^2} \begin{pmatrix} A & B \\ -B & A \end{pmatrix} \tag{2.127}$$

$$A = \sin(\theta)/\theta \qquad B = (1 - \cos(\theta))/\theta \tag{2.128}$$

To compute the log map for SE(2), first compute the SO(2) log map for rotation component R_b, then compute V and apply it as a linear operator to d_b.

SO(3) The group of proper rotations in 3D ($K = 3, M = 3$). Elements $b \in G$ rotate 3D points on the sphere. The Lie algebra has three generators,

$$G_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \tag{2.129}$$

$$G_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.130}$$

so that $v = \sum_{k=1}^{3} \theta_k G_k \in \mathfrak{g}$. Let $\theta = (\theta_1, \theta_2, \theta_3)^\top$ and let $\gamma = +\sqrt{\theta^\top \theta}$. Then,

$$b = \exp_G v = I + \frac{\sin \gamma}{\gamma} v + \frac{1 - \cos \gamma}{\gamma^2} v^2 \tag{2.131}$$

$$v = \log_G b = \frac{\lambda}{2 \sin \lambda}(b - b^\top) \quad \text{for } \lambda = \arccos\left(\frac{\mathrm{tr}(b) - 1}{2}\right) \tag{2.132}$$

SE(3) The group of rigid transformations in 3D ($K = 6, M = 4$). Elements $b \in G$ rotate then translate 3D homogeneous points. The Lie algebra has six generators,

$$G_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad G_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.133}$$

$$G_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad G_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.134}$$

$$G_5 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad G_6 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.135}$$

so that for $x, y, z, \theta_1, \theta_2, \theta_3 \in \mathbb{R}$,

$$v = xG_1 + yG_2 + zG_3 + \theta_1 G_4 + \theta_2 G_5 + \theta_3 G_6 \in \mathfrak{g} \tag{2.136}$$

Let $u = (u_1, u_2, u_3)^\top$, $\theta = (\theta_1, \theta_2, \theta_3)^\top$ and $G' = \mathrm{SO}(3)$. Then,

$$b = \exp_G v = \begin{pmatrix} R_b & d_b \\ 0 & 1 \end{pmatrix} \tag{2.137}$$

$$v = \log_G b = \begin{pmatrix} \log_{G'} R_b & V^{-1} d_b \\ 0 & 0 \end{pmatrix} \tag{2.138}$$

$$R_b = I + A\theta_\times + B\theta_\times^2 \tag{2.139}$$

$$d_b = Vu \tag{2.140}$$

$$V^{-1} = I - \frac{1}{2}\theta_\times + \frac{1}{\lambda^2}\left(1 - \frac{A}{2B}\right)\theta_\times^2 \tag{2.141}$$

$$A = \sin(\lambda)/\lambda \qquad B = (1 - \cos(\lambda))/\lambda \tag{2.142}$$

$$\theta_\times = \sum_{k=4}^{6} \theta_k G_k \qquad \lambda = \sqrt{\theta^\top \theta} \tag{2.143}$$

To compute the log map for SE(3), first compute the SO(3) log map for rotation component R_b, then compute V and apply it as a linear operator to d_b.

2.7 Bayesian Experiment Design

Bayesian Experiment Design (BED) is an approach to decision making where there is a generative model for what would occur once the decision has been made. With this model, the results of possible experiments can be reasoned over without actually performing the experiment, such as by marginalizing over or sampling possible outcomes. An experiment can then be chosen based on a desirable criteria.

In particular, design $d \in \mathcal{D}$ is chosen and observations $a \in \mathcal{A}$ are observed. The space $\mathcal{D}$ is a space of possible experiments to run—including any parameters—and the space $\mathcal{A}$ is the space of possible experimental observations or results. The design is chosen based on the optimization of some utility function. Designs can be sequentially chosen in a greedy manner so that at round $l = 1, \ldots, L$, design d_l is chosen and observations a_l are observed. Greedy selection does not yield optimal utility over all L rounds in general, but is more tractable. In this work, the space of designs $\mathcal{D}$ is discrete and the utility function considered is mutual information so that at the l^{th} round, the observation model for design d is,

$$p_d(a_l \mid x, y, D_{l-1}) \tag{2.144}$$

where $D_{l-1} = \{a_l, d_l\}_{i=1}^{l-1}$ is the set of previous decisions and outcomes. The utility function is mutual information between random variable a_l and latent state x conditioned on observations y and decision/outcome pairs D_{l-1},

$$I_d(a_l; x \mid y, D_{l-1}) = \mathbb{E}\left[\log \frac{p_d(a_l, x \mid y, D_{l-1})}{p_d(a_l \mid y, D_{l-1})\, p_d(x \mid y, D_{l-1})}\right] \tag{2.145}$$

$$= \mathbb{E}\left[\log \frac{p_d(a_l \mid x, y, D_{l-1})}{p_d(a_l \mid y, D_{l-1})}\right] \tag{2.146}$$

$$= \mathbb{E}\left[-\log p_d(a_l \mid y, D_{l-1})\right] - \tag{2.147}$$

$$\mathbb{E}\left[-\log p_d(a_l \mid x, y, D_{l-1})\right] \tag{2.148}$$

which is a difference of entropies that quantifies reduction in posterior uncertainty. In the l^{th} round, the design $d = d_l$ is chosen that corresponds to,

$$a_l^* = \underset{a_l}{\operatorname{argmax}}\, I_d(a_l; x \mid y, D_{l-1}) \tag{2.149}$$

and $D_l = D_{l-1} \cup (d_l^*, a_l^*)$. Analytic solutions are often not available for Equation 2.148. This work uses sampling-based Monte Carlo estimates for approximation. For a review of Bayesian Experiment Design with emphasis on tractable, linear designs, see [40]. Recent sampling-based or variational approaches include [231, 64, 154].

Chapter 3

Nonparametric Parts Modeling with Lie Group Dynamics

The world is full of moving objects comprised of articulating parts. Examples include the arms, legs, and tails of animals. Despite the wide range and complexity of such objects, humans have a remarkable ability to accurately discern both the number of articulating parts and their relation to the whole with few observations [148]. I seek to develop reasoning methods and algorithms that mimic this ability.

This work builds an unsupervised, articulated parts model for a general object simply by observing that object in motion. It eschews the need for labeled training data. Not all objects have the same number of parts, nor are their parts of a common shape, size, or appearance. Hence, no advance specification of the number of parts can be designated and assumptions on physical part properties must be limited. Object motion can be sensed in many ways including video or depth cameras as well as 3D capture setups that generate dynamic point clouds or meshes in a common coordinate system. For maximum flexibility, the Nonparametrics Parts Model proposed in this chapter seamlessly supports observation models from diverse sensors in 2D or 3D. Figure 3-1 shows examples of the articulating parts learned from objects in motion for 2D and 3D data.

Learning an articulated parts model with limited prior knowledge of the object is principally motivated by the study of animal motion, which stands to benefit from methods that infer motion decompositions. In particular, there is interest in how low-level motion composes to higher-level motion [217], and how those motions can predict higher-level behavior [7]. Furthermore, whereas humans may naturally segment an object into parts based on motion that is familiar to them (arms, legs, wings), it is conceivable that part decompositions which depart from human intuition may make for better descriptions of, or predictors for, behavior.

Object and part motion can be described in many ways. One straight-

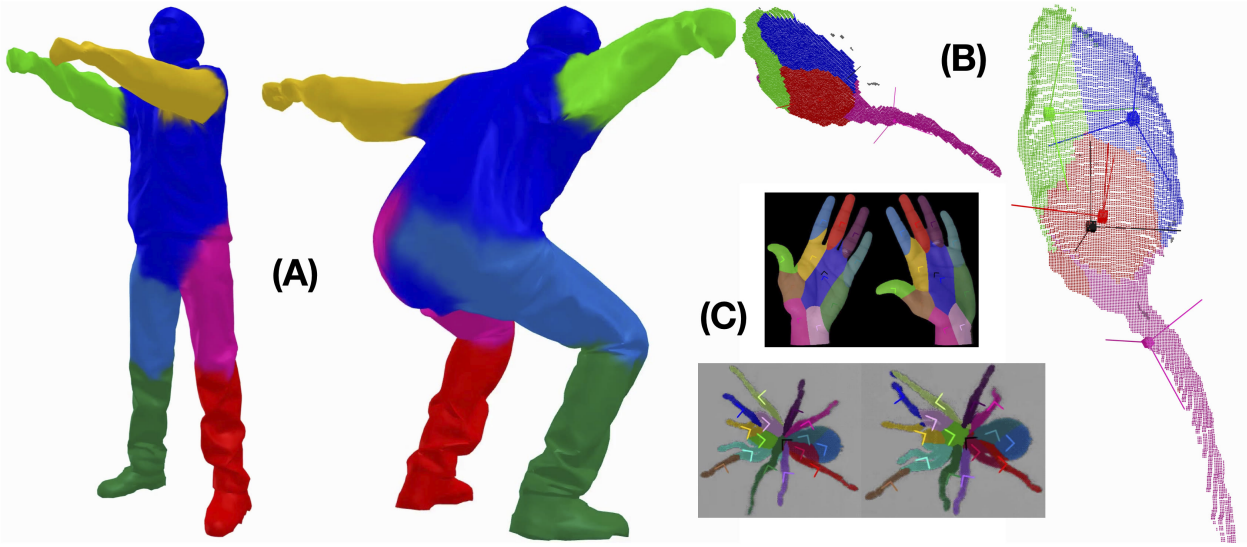


Figure 3-1: The number, rotation, translation, and shape of an object's parts are learned from a small number of observations of that object in motion. Motion of the body and parts is parameterized by the Lie group of rigid transformations in 3D or 2D. Supported data sources include sequences of meshes / point clouds (A, human), depth data (B, marmoset), and 2D images (C, hand, spider).

forward representation is to describe it as the set of pixels, points or mesh faces that are associated to each part at each time. This is a thorough description, but is less interpretable by humans than is a description containing the pose of each part at each time. Manifolds provide a parsimonious representation of the degrees of freedom in a system and are interpretable owing to their long study [16]. Whereas many methods learn a manifold on which observations exist [213, 175], this work instead begin with a representation that employs manifolds that have well-understood properties. Body and part motion is represented using SE(D), the Lie group of rotations and translations. SE(D) is appropriate for describing articulated motion because it can describe any rigid transformation [213].

3.1 Approach

This work develops an unsupervised parts model. It assumes that object-associated observations arrive in batch over all times. Observations may be in $D = 2$ or $D = 3$ dimensions depending on whether they are pixels from an RGB camera, unprojected depth measurements from an RGB-D sensor, or XYZ point clouds or meshes from a 3D capture setup. Input is observed in a world coordinate frame that is aligned with the sensor. Observations are modeled as being Gaussian-distributed about the center of one of an unknown number of parts. Observations are generated in their respective part frames, but observed in a common world frame. A key inference challenge is to determine appropriate body and part transformations that meaningfully explain how subsets

of observations engage in shared transformations.

Articulated object motion is modeled as time-varying body and part frames represented by elements of SE(D). Body dynamics are modeled as a random walk on SE(D) whereas part dynamics are modeled as independent, stabilized random walks centered about an unknown canonical part frame. Canonical parts are distributed about the body frame according to a concentrated Gaussian distribution on SE(D). Body and part dynamics have inverse-Wishart [218] distributed noise covariances that are interpreted as existing in the tangent space of the previous time's body or part frame. Conditioned on observations over all time, Gibbs sampling [74] inference proceeds by iteratively sampling body and part translations and rotations from their respective full conditional distributions. Body and part transformations are described as manifold-valued elements that do not exist in a vector space whereas their innovations are described as random variables existing in tangent spaces that do form vector spaces. Hence, it is necessary to define a Riemannian metric on SE(D) and use the Riemannian logarithm and Riemannian exponential maps to compose body or part transformations with their random innovations.

A Dirichlet Process prior is placed on the part associations of observations over all time. Doing so enables reasoning over an unknown number of parts but complicates inference because reasoning occurs over an infinite number of currently-uninstantiated parts. Specifically, computing the probability of association for each observation to an unidentified part requires an integration over all possible rigid transformations that could have generated the observation at that time. This integral has no analytic form. Instead, it is approximated with a constant derived from Monte Carlo sampling.

Chapter 3.2 summarizes contributions related to the Nonparametric Parts Model. Chapter 3.3 develops a naive parts model without Lie group dynamics; it motivates their use and aids intuition for the Nonparametric Parts Model. Chapter 3.4 develops the Nonparametric Parts Model and Chapter 3.5 develops inference. Chapter 3.6 provides experimental results and discussion. Finally, Chapter 3.7 provides related works.

3.2 Contributions

Object motion can be ambiguous from a single view. For example, spider legs observed by a camera from above may be occluded and foreshortened, making it unclear whether or not they have crossed. Such mutually exclusive outcomes can be represented by a distribution on part motion. Reasoning over distributions on manifolds requires additional care when the manifold does not form a vector space. Not only is the motion of a given part ambiguous, but so too are the number of parts. Both challenges are addressed by nontrivially combining Bayesian nonparametric models, specifically the Dirichlet Process [60],

with distributions on manifolds. The proposed Nonparametric Parts Model is well-suited to the properties of articulated objects in motion: parts persist across time, their motion is described by rigid transformation dynamics, and a distribution is maintained over an unknown number of parts. Novel Gibbs decompositions of inferring translations and rotations in posterior distributions on SE(D) with concentrated Gaussian [213] priors are derived. I show that translation conditionals have analytic form under a concentrated Gaussian prior and a multivariate Gaussian observation model (Chapter 3.5). Efficient, bounded Slice sampling [145] inference is used for rotation conditionals that is aware of multimodal structure in the posterior.

NPP is the first parts model that can infer an unknown number of articulating parts in 2D or 3D from a single sensor without requiring a body model a priori or resorting to markers, dyes, annotations, or sensors being placed on the object. No correspondences between object observations over time are assumed, nor are observations assumed to be present for each part at all times. I demonstrate that parts decompositions can be learned from short sequences of object motion by validating NPP on 2D and 3D sequences containing different object types (Chapter 3.6). Additionally, the parts in one data sequence transfer to other data sequences of the same object type (but different instance) (Chapter 3.6.5).

In addition to learning a parts decomposition, NPP enables analysis of part motion by separately inferring body and part contributions to motion. Observations associated to an object can be segmented over time based on the integrated motion of their associated part (Chapter 3.6.3). This facilitates understanding by showing, for example, that the observations corresponding to the legs of a spider experience more motion relative to its body frame than does the thorax, without advance knowledge of spider legs or thoraxes. Finally, NPP enables synthesis of novel articulated body and part motion based on parameters learned from observing the object in motion (Chapter 3.6.4).

3.3 A Naive Parts Model

We begin construction of the Nonparametric Parts Model by first developing our intuition with a naive parts model that has a known number of parts with linear Gaussian dynamics on $\mathbb{R}^D$. Much of the generative model and challenges we encounter in the naive parts model will motivate and transfer over to the Nonparametric Parts Model, which models dynamics on the Lie group of rigid transformations SE(D) and does not assume a known number of parts.

Let $t = 1, \ldots, T$ index time. A simple random walk for a body centroid $x_t \in \mathbb{R}^D$ can be constructed as, for all times t,

$$x_t \sim f(x_{t-1}) = x_{t-1} + q_t \qquad q_t \sim \mathrm{N}(0, Q) \tag{3.1}$$

where Q is its driving noise covariance. This evolves over time accord-

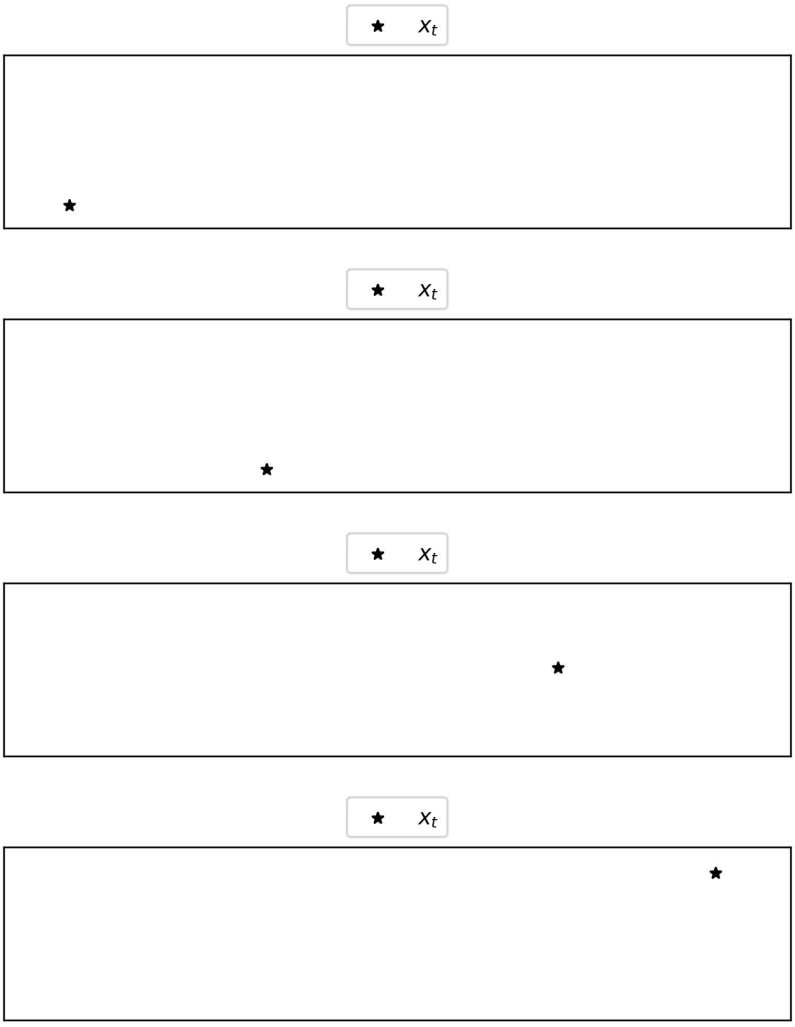


Figure 3-2: A random walk on $\mathbb{R}^2$ for body x_t. See Equation 3.1.

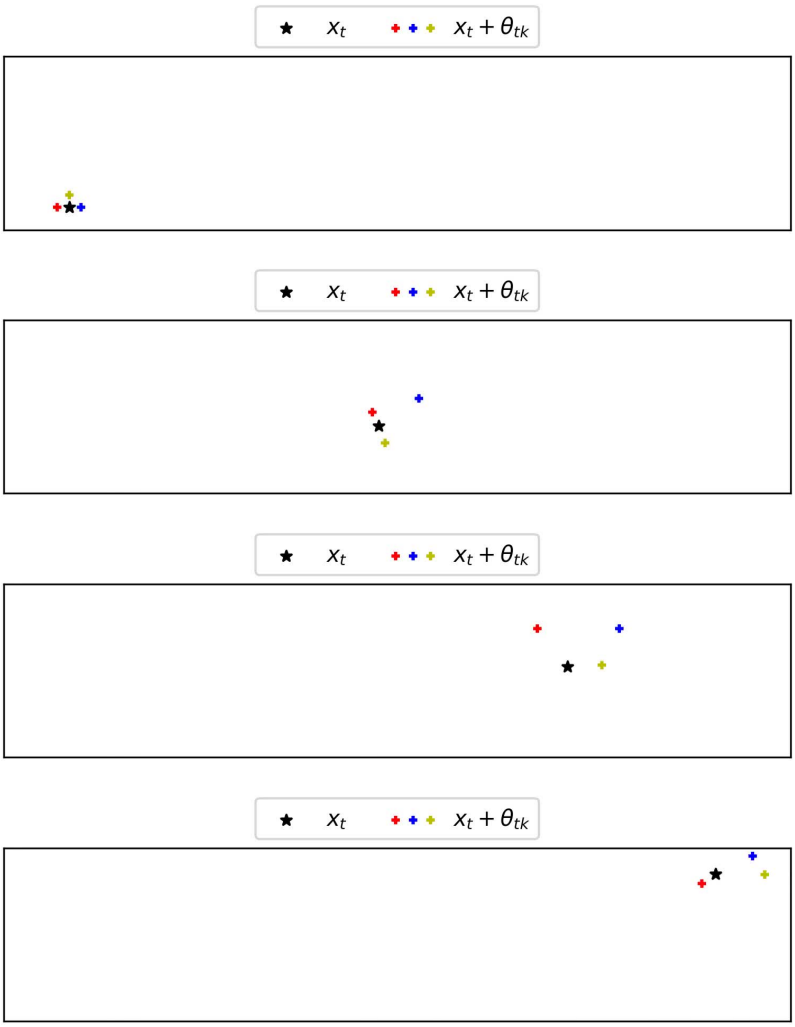


Figure 3-3: A random walk on $\mathbb{R}^2$ for body x_t and parts θ_{tk}. See Equations 3.2–3.3.

ing to Figure 3-2. We can add $k = 1, \ldots, K$ parts that evolve relative to the body so that the model is, for all times t and parts k,

$$x_t \sim f(x_{t-1}) = x_{t-1} + q_t \qquad q_t \sim \mathrm{N}(0, Q) \tag{3.2}$$

$$\theta_{tk} \sim g(\theta_{(t-1)k}) = \theta_{(t-1)k} + s_{tk} \qquad s_{tk} \sim \mathrm{N}(0, S_k) \tag{3.3}$$

where θ_{tk} is the k^{th} part's centroid at time t and S_k is its driving noise covariance. This system evolves according to Figure 3-3. Observe that because the parts are specified relative to the body they must be transformed by the body's centroid to be observed in world coordinates. The world coordinates for part k at time t are $x_t + \theta_{tk}$.

A problem with the model in Equations 3.2–3.3 is that parts can wander arbitrarily far from the body. This is physically implausible because object parts tend to remain proximate. Instead, we would prefer that they tend to remain proximate. Constraints can be imposed that enforce nearness, but they would be object-specific and compli-

cate inference. Instead, Appendix A.3 shows how a stabilized random walk can be constructed so that over all times t it has an expected value of 0 and an asymptotic, bounded variance equal to its driving noise variance.

A stabilized random walk can be straightforwardly implemented in the above formulation because the parts are interpreted relative to the body: design the parts (Equation 3.3) to evolve with a stabilized random walk so that they have a suitably small driving noise variance[25] and an expected value their expected value equal to zero in part coordinates, which is the body centroid in world coordinates. Designing in this way would encourage the object to be compact, with all its parts "folded in" near its center of mass. Instead, we add a "canonical part" centroid ω_k that is some offset from the body and fixed over all time. Then the part transformations θ_{tk} are reinterpreted as existing in the canonical part frame so that their zero expected value means that they will stay near their canonical part centroid in world coordinates. This encourages parts to have distinct locations but allows them to overlap when there is supporting evidence. This model is, for all times t and parts k,

[25] The driving noise variance should be small enough that the part does not wander off but not so small that the part cannot move about or cross over other parts in the body frame.

$$x_t \sim f(x_{t-1}) = x_{t-1} + q_t \qquad q_t \sim \mathrm{N}(0, Q) \tag{3.4}$$

$$\omega_k \sim \mathrm{N}(0, W_k) \tag{3.5}$$

$$\theta_{tk} \sim g(\theta_{(t-1)k}) = A\,\theta_{(t-1)k} + B\,s_{tk} \qquad s_{tk} \sim \mathrm{N}(0, S_k) \tag{3.6}$$

where W_k is the covariance of the k^{th} part's translation from the body centroid and matrices $A, B \in \mathbb{R}^{D\times D}$ control the smoothness of the stabilized random walk (with greater smoothness for $a \to 1$),

$$A = \mathrm{diag}\left(\sqrt{a}, \ldots, \sqrt{a}\right) \qquad B = \mathrm{diag}\left(\sqrt{1-a}, \ldots, \sqrt{1-a}\right) \tag{3.7}$$

Observe from Figure 3-4 that this model stabilizes the parts so that they do not wander too far from the body. This is a consequence of their asymptotic covariance being designed to be equal to S_k.

Finally, body and part centroids are not usually observed directly; instead, indirect observations $y_t = \{y_{tn}\}_{n=1}^{N_t}$ are measured such that y_{tn} is generated by part k if association $z_{tn} = k$. The final naive parts model is, for all times t, parts k, and observations n,

$$x_t \sim f(x_{t-1}) = x_{t-1} + q_t \qquad q_t \sim \mathrm{N}(0, Q) \tag{3.8}$$

$$\omega_k \sim \mathrm{N}(0, W_k) \tag{3.9}$$

$$\theta_{tk} \sim g(\theta_{(t-1)k}) = A\,\theta_{(t-1)k} + B\,s_{tk} \qquad s_{tk} \sim \mathrm{N}(0, S_k) \tag{3.10}$$

$$z_{tn} \sim \mathrm{Cat}(z_{tn} \mid \pi) \tag{3.11}$$

$$\begin{aligned} y_{tn} &\sim h(x_t, \omega_k, \theta_{(t-1)k})^{\delta_{z_{tn}=k}} \\ &= x_t + \omega_k + \theta_{tk} + \epsilon_{tn} \quad \epsilon_{tn} \sim \mathrm{N}(0, E_k) \end{aligned} \tag{3.12}$$

where $\pi = (\pi_1, \ldots, \pi_K)$ are mixture weights that sum to one such that π_k is proportional to the number of observations that part k tends to generate. Covariance E_k defines the typical range in which observa-

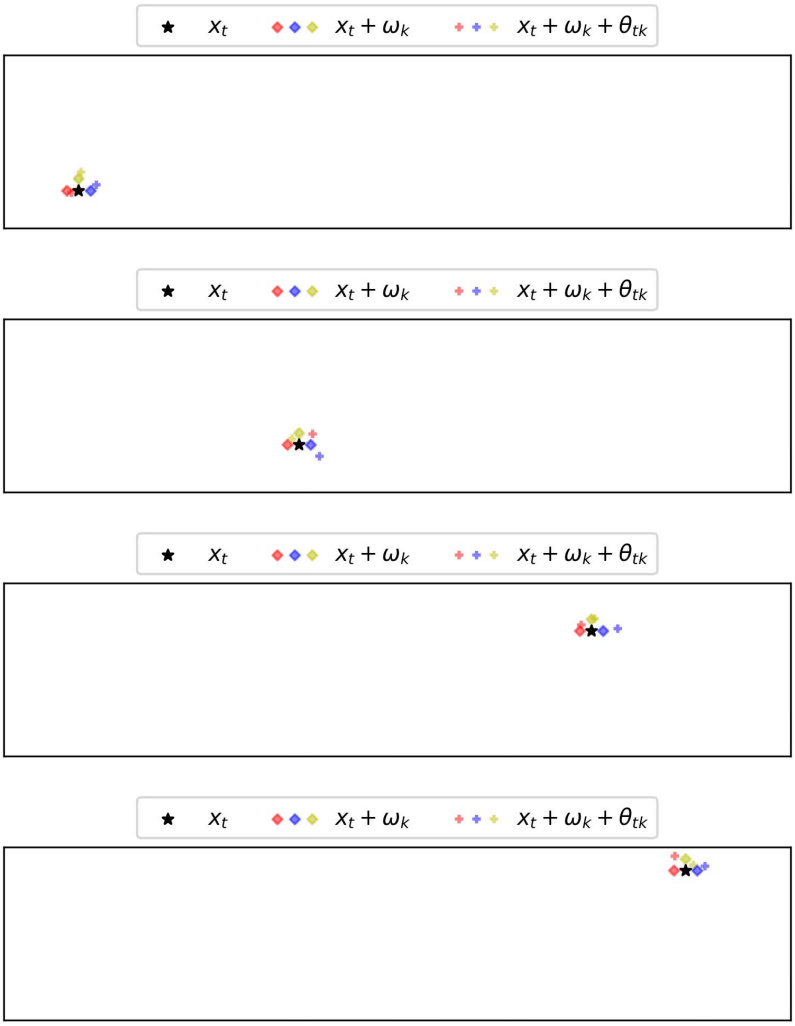


Figure 3-4: A stabilized random walk on $\mathbb{R}^2$ for parts θ_{tk}. See Equations 3.4–3.6.

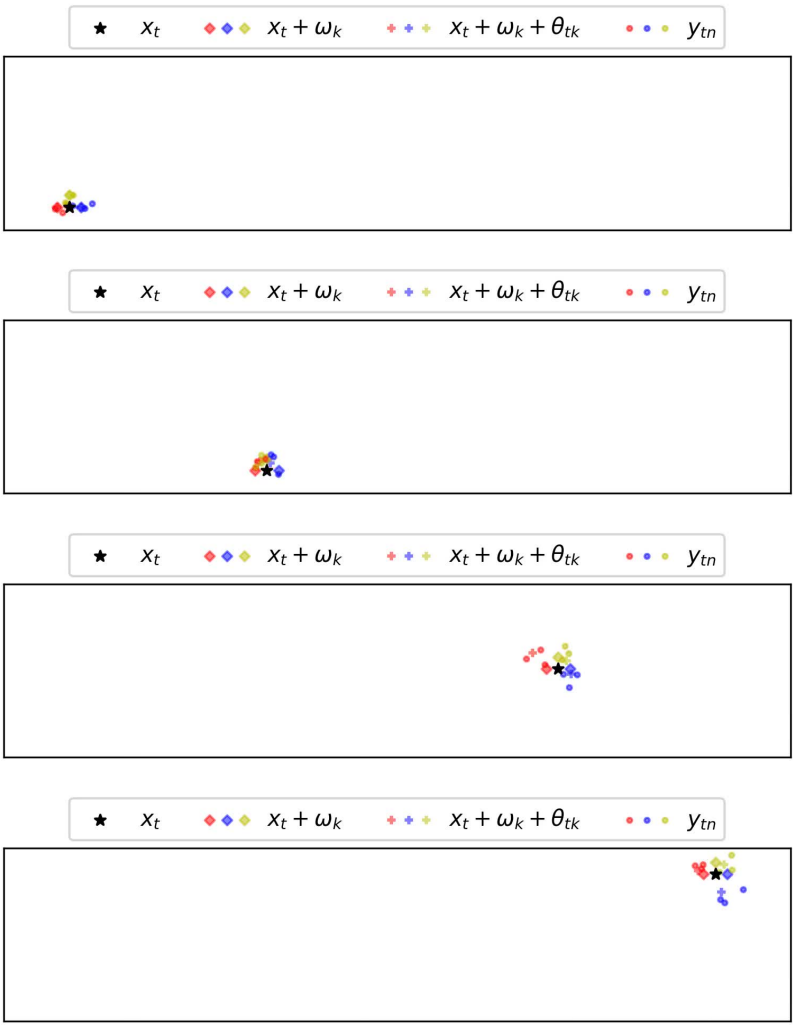


Figure 3-5: A complete naive parts model. See Equations 3.8–3.12.

tions associated to part k are generated. This model is visualized in Figure 3-5. The naive parts model has the following parameters:

$$\{K, Q, \pi, \{W_k, S_k, E_k\}_{k=1}^{K}\} \tag{3.13}$$

It captures that parts persist over time, that observations need not be observed from all parts at all times, that parts do not wander too far from a typical location near the body, and that parts can have different typical rates of motion compared to each other and the body. We retain these properties when building the Nonparametric Parts Model while also addressing two key limitations of the naive parts model:

1. Parameters must be determined for each observed object, which would involve a costly grid search or empirical Bayes approach. The Nonparametric Parts Model instead places prior distributions on all parameters (Equation 3.13), most notably on the num-

ber of parts K using a Dirichlet Process.

2. The only valid body and part motions are translations. This fails to capture articulated motion. The Nonparametric Parts Model changes the latent space from $\mathbb{R}^D$ to SE(D), the Lie group of rigid transformations. The dynamics models f, g and observation model h undergo significant changes though their interpretation and algebraic form are remarkably similar to Equations 3.8–3.12.

Following, we develop the Nonparametric Parts Model.

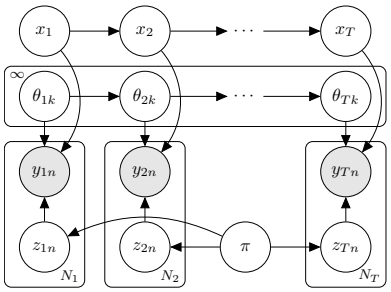


Figure 3-6: Simplified graphical model for an unknown number of time-varying parts $\{\theta_{tk}\}_{t=1,k=1}^{T,\infty}$ coupled by shared dynamics $\{x_t\}_{t=1}^{T}$. Observations y_{tn} are generated by part k if $z_{tn} = k$. Stick weights $\{\pi_k\}_{k=1}^{\infty}$ influence the observation counts for each part. Priors $\{\alpha, H_x, H_\theta, H_\omega, H_S, H_E, H_W, H_Q\}$ and latent parameters $\{Q, \omega_k, S_k, W_k, E_k\}$ are omitted for clarity.

3.4 Nonparametric Parts Model

Let $t = 1, \ldots, T$ index time, $k = 1, \ldots, \infty$ index parts, and $n = 1, \ldots, N_t$ index observations at time t. Most generally, the Nonparametric Parts model (Figure 3-6) takes as its sole input observations $\{y_t\}_{t=1}^{T}$ where the t^{th} batch $y_t = \{y_{tn}\}_{n=1}^{N_t}$ contains N_t observations with unknown correspondence. There is a global (body) dynamic with time-varying parameters x_t and time-fixed parameter Q. There are an unknown number of components (parts) with time-varying parameters θ_{tk} and time-fixed parameters $\{\omega_k, S_k, E_k, W_k\}$. Stochastic dynamics models f, g and stochastic observation model h are, for each t, k, n,

$$x_t \sim f(x_{t-1}, Q) \qquad \theta_{tk} \sim g(\theta_{(t-1)k}, \omega_k, S_k) \tag{3.14}$$

$$y_{tn} \sim h(x_t, \theta_{tz_{tn}}, \omega_{z_{tn}}, E_{z_{tn}}) \tag{3.15}$$

where $z_{tn} = k$ indicates that observation y_{tn} was generated by component k. A prior probability of association is given by the infinite discrete distribution of stick weights π (for $\alpha > 0$) implied by the Dirichlet Process when used as a prior for mixture models:

$$z_{tn} \sim \pi = p(z_{tn} \mid \pi) \qquad \pi \sim \text{GEM}(\alpha) = p(\pi) \tag{3.16}$$

The GEM distribution implies a Dirichlet Process prior with concentration α and a base measure whose density is the product of densities of the component parameters $\{\theta_{tk}, \omega_k, S_k, E_k, W_k\}$ for a single k and all times t.

To specialize for object and parts modeling we must further specify the domain of random variables $\{y_{tn}, x_t, \theta_{tk}, \omega_k, S_k, E_k, W_k, Q\}$, the form of priors $\{H_x, H_\theta, H_\omega, H_S, H_E, H_W, H_Q\}$ and the forms of stochastic dynamics and observation models $\{f, g, h\}$.

Variable	Description
$G = \text{SE}(D)$	Lie group
$x_t \in G$	Body frame
$\omega_k \in G$	Part canonical frame
$\theta_{tk} \in G$	Part per-time frame
$y_{tn} \in \mathbb{R}^D$	Observation
$z_{tn} \in \mathbb{Z}$	Association
$Q \in \mathcal{P}(M)$	Body driving noise
$W_k \in \mathcal{P}(M)$	Part dispersion
$S_k \in \mathcal{P}(M)$	Part driving noise
$E_k \in \mathcal{P}(D)$	Part observation noise
$\alpha > 0$	Concentration
$t \geq 1$	Time index
$k \geq 1$	Part index
$n \geq 1$	Observation index
$M \in \{3, 6\}$	DoF of G
$D \in \{2, 3\}$	Dimension of y_{tn}

Table 3.1: Nonparametric Parts Model notation.

3.4.1 Body and Parts

Let $G = \text{SE}(D)$ for dimension $D \in \{2, 3\}$. We seek to infer a parts decomposition of an articulating object by directly observing it in motion. Specifically, we model the inputs $y_{tn} \in \mathbb{R}^D$ as being random collections of points sampled *within* the object as it moves across time.

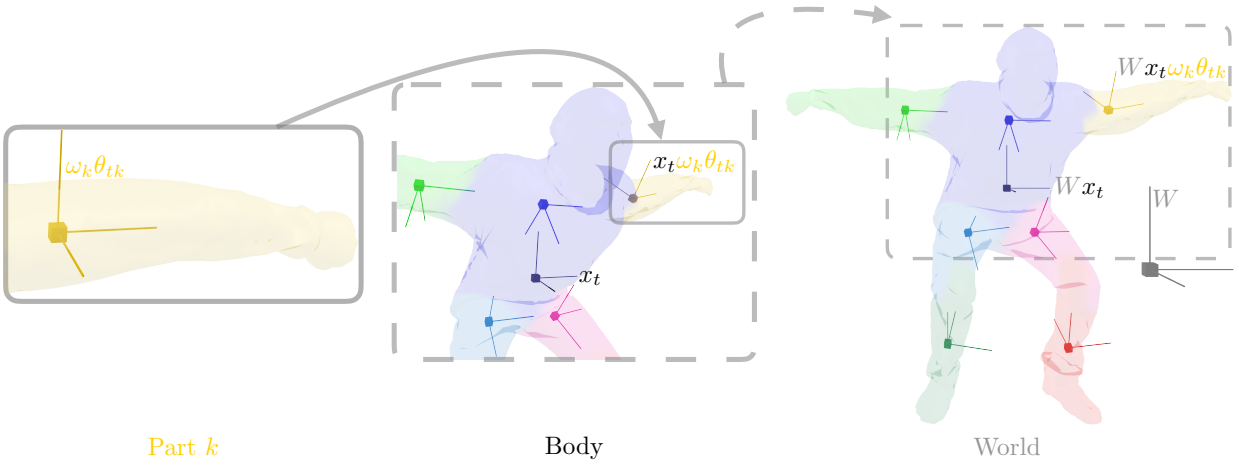


Figure 3-7: The frames that comprise an object in the Nonparametric Parts Model at time t. Per-time body frames x_t are rigid transformations from world frame W. Each part k contains a time-fixed canonical part frame ω_k and a per-time part frame θ_{tk}. ω_k are a rigid transformation from body frame x_t while θ_{tk} are a rigid transformation from ω_k. Using stabilized random walk dynamics, each per-time part frame θ_{tk} is designed to transform smoothly over time but remain near the origin of their respective canonical part frame ω_k.

Variable (including no) observations are supported at each time, and no correspondence between observations is assumed. Diverse inputs are supported, including foreground pixels of 2D image sequences, unprojected points from depth sequences, and 3D point clouds sampled within mesh sequences.

We assume part persistence–an object does not gain or lose parts over time. We also assume that parts move smoothly through space but remain close (in an L2 sense) to a common body which also moves smoothly. The relation between body and part motion can be modeled in many ways: one naive extreme would be to model them as floating bodies with linear dynamics (as we did with the naive parts model), while the other extreme would be to model them as existing in a skeletal network of joints. Linear dynamics fail to capture part articulation while skeletal networks are overly restrictive.

We take a middle ground: parts $\{\theta_{tk}, \omega_k, S_k, E_k, W_k\}$ are modeled as floating bodies that rotate and translate smoothly through space about a body frame $x_t \in G$, but whose origins tend to remain near the origin of a canonical part frame $\omega_k \in G$ through stabilized random walk dynamics. Canonical part frames are close to the body frame and remain fixed across time but parts also have per-time frames $\theta_{tk} \in G$. Parts are not fixed in their spatial extent; instead, they have a probabilistic, ellipsoidal shape model governed by Gaussian covariance E_k. Part dynamics are governed by covariance S_k and body dynamics are governed by covariance Q. The dispersion of canonical part frames about the body frame is governed by covariance W_k. Figure 3-7 graphically depicts how body and part frames compose. Table 3.1 summarizes important notation.

3.4.2 Dynamics

Body frames x_t and parts evolve independently, but are implicitly coupled through the observation model. In particular, the body frame stochastic dynamics model is:

$$x_t \sim \mathrm{N_L}(x_t \mid x_{t-1}, Q) = p(x_t \mid x_{t-1}, Q) \tag{3.17}$$

Object dynamics are a non-linear random walk on G whose noise covariance Q exists in the tangent space about the body frame at the previous time. Canonical part frames ω_k are dispersed about the body frame with covariance W_k,

$$\omega_k \sim H_\omega = \mathrm{N_L}(\cdot \mid I, W_k) = p(\omega_k \mid W_k) \tag{3.18}$$

where $I \in G$ is the identity element (no translation or rotation) and covariance W_k can be thought to (implicitly) exist in the tangent space of x_t. Each part has per-time dynamics θ_{tk} with driving noise covariance S_k governed by:

$$\theta_{tk} = \begin{pmatrix} \mathrm{Exp}_{R_{\theta_{(t-1)k}}} \phi_{tk} & A\, d_{\theta_{(t-1)k}} + B\, m_{tk} \\ 0 & 1 \end{pmatrix} \tag{3.19}$$

with constants A, B as in Equation 3.7. Exp in Equation 3.19 is the Riemannian exponential for $\mathrm{SO}(D)$. $\phi_{tk} \in \mathrm{so}(D)$ is a vector in the tangent space of $R_{\theta_{(t-1)k}}$. Part translation driving noise m_{tk} and rotation driving noise ϕ_{tk} are jointly distributed:

$$(m_{tk}, \phi_{tk}) \sim \mathrm{N}(0, S_k) \tag{3.20}$$

As discussed in the naive parts model and in Appendix A.3, appropriately chosen coefficients of matrices A, B ($a = 0.95$) cause the asymptotic covariance of the part translation $d_{\theta_{tk}}$ to equal the covariance of translation driving noise m_{tk}. This form enables parts to transform smoothly, but never too far from their canonical location, and mitigated part confusion during inference. Equations 3.19 and 3.20 combine to give the per-time part dynamics,

$$p(\theta_{tk} \mid \theta_{(t-1)k}, S_k) \tag{3.21}$$

which can be evaluated by solving for m_{tk}, ϕ_{tk} in Equation 3.19 and evaluating Equation 3.21. Simulation is also straightforward: Sample m_{tk}, ϕ_{tk} according to Equation 3.21 and compute the next part according to Equation 3.20.

All driving noise covariances are drawn from Inverse-Wishart distributions, where we note that our model supports arbitrary correlations between translation and rotation for object, canonical part, and

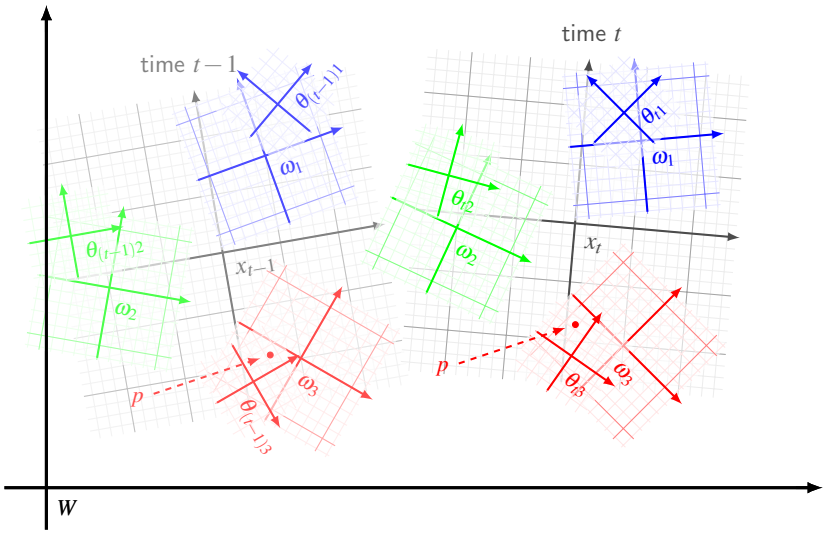


Figure 3-8: Nonparametric Parts Model generation of observations.. The body frames x at times $t-1$ and t, along with the (fixed) canonical part transformations ω_k and per-time part transformations θ_{tk} for times $t-1$ and t. Parts are visualized by color (red, green, blue). The point p with coordinates in the time t, part k frame of reference has world coordinates $x_t\omega_k\theta_{tk}p$. Much of the inference challenge in the Nonparametric Parts Model is in determining transformations $x_t, \omega_k, \theta_{tk}$ for all times t, k given that points points are only observed in world coordinates.

part transformations:

$$Q \sim H_Q = \mathrm{IW}(\cdot|\upsilon_{Q_0}, \Lambda_{Q_0}) = p(Q) \tag{3.22}$$

$$S_k \sim H_S = \mathrm{IW}(\cdot|\upsilon_{S_0}, \Lambda_{S_0}) = p(S_k) \tag{3.23}$$

$$W_k \sim H_W = \mathrm{IW}(\cdot|\upsilon_{W_0}, \Lambda_{W_0}) = p(W_k) \tag{3.24}$$

And initial body and part frames are drawn according to:

$$x_1 \sim H_x = \mathrm{N_L}(\cdot|x_0, \Sigma_x) = p(x_1 \mid x_0) \tag{3.25}$$

$$\theta_{1k} \sim H_\theta = \mathrm{N_L}(\cdot|\theta_0, \Sigma_\theta) = p(\theta_{1k} \mid \theta_{0k}) \tag{3.26}$$

3.4.3 Observation Model

Input y_{tn} is assumed to be in world coordinate system W, which is assumed to be aligned with the sensor's coordinate system (hence, W has no rotation or translation and is henceforth omitted). Parts generate observations in their respective part coordinate systems and are mapped to world coordinates via θ_{tk}, ω_k and the body frame x_t. That is, part k generates point $e_{tn} \sim \mathrm{N}(0, E_k)$ which is then mapped to world coordinates $\tilde{y}_{tn} = x_t\omega_k\theta_{tk}\tilde{e}_{tn}$ if $z_{tn} = k$ (where $\tilde{(\cdot)}$ is a homogeneous projection of $(\cdot)$). The transformation is linear in $\tilde{e}_{tn}$ allowing straightforward mean and variance computations of the homogeneous point in world coordinate $\tilde{y}_{tn}$, yielding the following observation model,

$$\tilde{y}_{tn} \sim \mathrm{N}(\tilde{y}_{tn}|x_t\omega_k\theta_{tk}\tilde{0}_{\mathbb{R}}, x_t\omega_k\theta_{tk}\tilde{E}_k\theta_{tk}^\top\omega_k^\top x_t^\top)^{\delta_{z_{tn}=k}} \tag{3.27}$$

where $\tilde{0}_{\mathbb{R}}$ is the homogeneous zero vector in $\mathbb{R}^D$ and $\tilde{E}_k$ is a degenerate block covariance matrix E_k with a zero row and column (a covariance in homogeneous coordinates). Without homogeneous coordinates this is,

$$y_{tn} \sim \mathrm{N}(y_{tn} \mid \mu_{tk}, \Sigma_{tk})^{\delta_{z_{tn}=k}} = p(y_{tn} \mid x_t, \omega, \theta_t, z_{tn}) \tag{3.28}$$

$$\mu_{tk} = R_{x_t} R_{\omega_k} \left(d_{\theta_{tk}} + d_{\omega_k} \right) \tag{3.29}$$

$$\Sigma_{tk} = R_{x_t} R_{\omega_k} R_{\theta_{tk}} E_k R_{\theta_{tk}}^\top R_{\omega_k}^\top R_{x_t}^\top \tag{3.30}$$

The observation model is parameterized by state-dependent noise. While simple, it accommodates image plane observations in 2D, depth observations in 2.5D and XYZ observations in 3D. Incorporating additional terms (*e.g.*, appearance) is straightforward, but were not needed for our purposes. As with most generative models, robustness to missing data (common for depth sensors) is handled seamlessly. Figure 3-8 visually depicts the relationship between points generated in part k's coordinate system at time t and the world coordinates that are observed.

The observation covariance Σ_{tk} for y_{tn} is some rotation of E_k for $z_{tn} = k$ due to the composition of body and part frames. Consequently, E_k is constrained to be diagonal (i.e., axis-aligned) so as to avoid ambiguity. While the use of E_k implies a probabilistic, ellipsoid part shape model, its primary function is to yield robust associations z_{tn} of observations to parts. Here, we use the following prior:

$$E_k \sim H_E = \mathrm{IW}(\cdot | \upsilon_{E_0}, \Lambda_{E_0}) = p(E_k) \tag{3.31}$$

3.5 Inference

The Nonparametric Parts Model is a Dirichlet Process Mixture with a base measure whose density is,

$$p(\omega, \theta, W, S, E) = \tag{3.32}$$
$$\prod_k p(W_k)\, p(S_k)\, p(E_k)\, p(\omega_k \mid W_k) \prod_t p(\theta_{tk} \mid \theta_{(t-1)k}, S_k)$$

when evaluated for a single k. Combining the base measure with the stick-breaking prior on π, the shared body frame dynamics (x, Q), and the observation model for y yields the joint posterior for the Nonparametric Parts Model,

$$p(x, \omega, \theta, Q, W, S, E, z, \pi \mid y) = \tag{3.33}$$
$$\frac{1}{Z}\, p(\omega, \theta, W, S, E)\, p(\pi)\, p(Q)$$
$$\prod_t p(x_t \mid x_{t-1}) \prod_n p(z_{tn} \mid \pi)\, p(y_{tn} \mid x_t, \omega, \theta_t, z_{tn})$$

where prior parameters are omitted and Z is an intractable normalization constant . We sample from this complicated posterior using

Markov Chain Monte Carlo (MCMC) inference that exploits Gaussian statistics in the tangent space for efficient updates while simultaneously respecting the geometry of the Lie group. This is accomplished by sampling from the full conditional distributions of each latent variable, grouped in order of discussion,

$$(x_t, \theta_{tk}, \omega_k) \qquad z_{tn} \qquad (\pi, E_k, S_k, W_k, Q) \tag{3.34}$$

where $t = 1, \ldots, T, k = 1, \ldots, \infty, n = 1, \ldots, N_t$ and omitted leading subscripts are taken to mean joint dependence (i.e., $y = \{y_t\}_{t=1}^T$ and $y_t = \{y_{tn}\}_{n=1}^{N_t}$). Inference complexity is linear in the number of observations and parts. In our experiments, chains were generally mixed after about 300 samples, with approximately 1 minute per sample being the worst-case timing for any data we tested on.

In the sequel we sketch the sampling of body transformations x_t. Full details are in Appendix A.2, along with sampling of the canonical parts ω_k and part transformations θ_{tk} which take a similar form. We also discuss sampling part associations z_{tn}, which are conjugate except when sampling assignments to the base measure. The conditionals in the third grouping (π, E_k, S_k, Q) can be sampled analytically due to conjugate priors. Parts $\{\theta_{tk}, \omega_k, S_k, W_k, E_k\}$ can be sampled in parallel across k and z_{tn} can be sampled in parallel across t, n.

Following, we employ the notational convention that any element $b \in \mathrm{SE}(D)$ has rotation matrix $R_b \in \mathrm{SO}(D)$ and translation vector $d_b \in \mathbb{R}^D$. Then, the body and per-time part dynamics can be represented as block matrices with components,

$$x_t = \begin{pmatrix} R_{x_t} & d_{x_t} \\ 0 & 1 \end{pmatrix} \quad \theta_{tk} = \begin{pmatrix} R_{\theta_{tk}} & d_{\theta_{tk}} \\ 0 & 1 \end{pmatrix} \tag{3.35}$$

and similarly for ω_k.

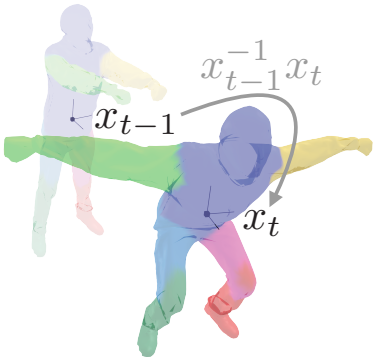
3.5.1 Lie Group Dynamics Decompositions

We exploit the Lie algebra to develop an efficient Gibbs sampler for dynamical terms $\{x_t, \omega_k, \theta_{tk}\}$. For example, the operation $x_{t-1}^{-1} x_t$ transforms the body frame at time t into that of the body frame at time $t-1$ (Fig. 3-9, top). This operation is an element of $SE(D)$:

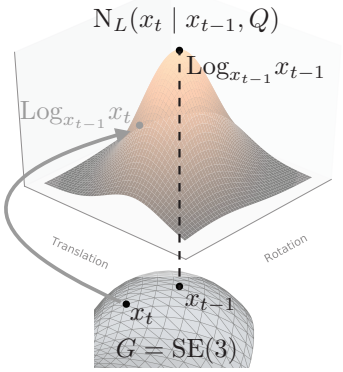
$$x_{t-1}^{-1} x_t \triangleq \begin{pmatrix} R_{x_{t-1}^{-1}, x_t} & d_{x_{t-1}^{-1}, x_t} \\ 0 & 1 \end{pmatrix}, \tag{3.36}$$

where $R_{x_{t-1}^{-1}, x_t} = R_{x_{t-1}}^T R_{x_t}$ and $d_{x_{t-1}^{-1}, x_t} = R_{x_{t-1}}^T (d_{x_t} - d_{x_{t-1}})$. Elements in the frame x_t are mapped to the tangent space of x_{t-1} via the Riemannian Log map (Figure 3-9, bottom):

$$\mathrm{Log}_{x_{t-1}} x_t \triangleq \log_G(x_{t-1}^{-1} x_t) = \begin{pmatrix} V^{-1}_{x_{t-1}^{-1}, x_t} d_{x_{t-1}^{-1}, x_t} \\ \phi_{x_{t-1}^{-1}, x_t} \end{pmatrix} \tag{3.37}$$



Body Frame Projection



Tangent Distribution

Figure 3-9: **Top:** Object dynamics of the body frame x_t at time t are projected into body frame coordinates at time $t-1$ by the Lie group operation $x_{t-1}^{-1} x_t$. **Bottom:** The projection is in $SE(3)$ with Gaussian statistics in the tangent space of x_{t-1}. The figure notionally depicts two degrees of freedom, whereas $SE(3)$ has 6 degrees of freedom.

The first entry $V^{-1}_{x^{-1}_{t-1},x_t} d_{x^{-1}_{t-1},x_t}$ are tangent space coordinates of translation and the second entry $\phi_{x^{-1}_{t-1},x_t}$ is a rotation vector. The invertible linear operator $V^{-1}_{x^{-1}_{t-1},x_t}$ is computable from rotation $R_{x^{-1}_{t-1},x_t}$ (or from $\phi_{x^{-1}_{t-1},x_t}$). This is well-defined for $x^{-1}_{t-1}x_t$ sufficiently close to identity and consistent with small incremental motions.

3.5.2 Translation Conditionals

Recall that Equations (3.36) and (3.37) map x_t to the tangent space of x_{t-1}. When conditioned on rotation, this mapping is linear in the translation component d_{x_t}. This observation, combined with Gaussian statistics in the tangent space, yields closed-form Gibbs updates for translation. To see this, observe that the distribution over dynamics in the tangent space is (Figure 3-9, bottom),

$$\mathrm{N}_L(x_t \mid x_{t-1}, Q) = \mathrm{N}\left(\begin{pmatrix} Cd_{x_t} + u \\ \phi_{x^{-1}_{t-1},x_t} \end{pmatrix} \middle| 0, Q\right) \tag{3.38}$$

where $C = V^{-1}_{x^{-1}_{t-1},x_t} R^{\top}_{x_{t-1}}$ and $u = -V_{x^{-1}_{t-1},x_t} R^{\top}_{x_{t-1}} d_{x_{t-1}}$. Conditioned on rotation R_{x_t} and previous body frame x_{t-1}, the corresponding rotation vector $\phi_{x^{-1}_{t-1},x_t}$ and matrix $V_{x^{-1}_{t-1},x_t}$ are fixed quantities. This renders C and u computable and yields a Gaussian conditional distribution for d_{x_t}. This conditional constitutes our prior belief about d_{x_t} given R_{x_t}, x_{t-1} and covariance Q. Similar logic allows us to derive a Gaussian conditional on d_{x_t} given future transformation x_{t+1}. These can be analytically combined to provide a Gaussian distribution for $d_{x_t} \mid R_{x_t}, x_{t-1}, x_{t+1}$. Because this is Gaussian, and the observation model is also a product of Gaussians whose parameters are known given $\{\omega_k, E_k, \theta_{tk}\}_{k=1}^{\infty}$ and $\{z_{tn}\}_{n=1}^{N_t}$, it follows that the posterior on d_{x_t} is also Gaussian, and analytically computable.

In contrast, sampling of rotation parameters lacks a closed form. We utilize univariate slice sampling [145] for the full conditional of each rotation parameter, along with a fixed number of MCMC proposals to correct for known rotational symmetries.

3.5.3 Rotation Conditionals

We perform univariate slice sampling [145] to sample from the rotation full conditionals of body x_t, canonical part ω_k and part transformation θ_{tk}. This is straightforward because the Lie algebraic coordinates of each transformation decompose into a set of univariate coordinates corresponding to translation, and a set corresponding to rotation. Given this decomposition sampling is straightforward: each rotation coordinate is sampled, holding all others fixed. The task is furthered simplified because the bounds of $\pm\pi$ can be imposed.

One complication is that the distribution is multi-modal because the observation likelihood is invariant to 180° rotations. There are two such modes in SE(2) and four in SE(3). Given one mode, all others can

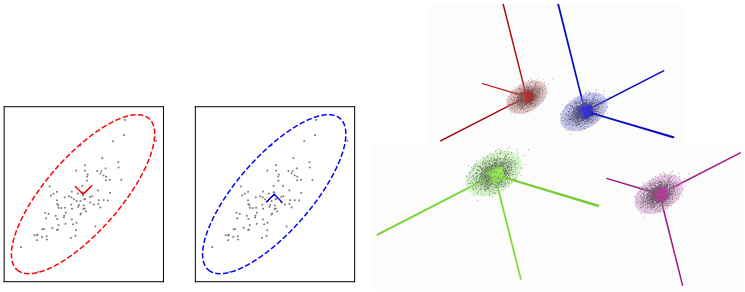


Figure 3-10: Nonparametric Parts likelihoods are invariant to two rotation symmetries in SE(2) (left) and four rotation symmetries in SE(3) (right). Notice that the colored observation covariances cover the same volume when drawn at fixed standard deviations, implying that Mahalanobis distances of part covariances to observations will be equal for each symmetric rotation. While dynamics will typically favor one mode over others, the slice sampler sometimes locks onto the wrong mode. The remedy is a fixed number of MCMC proposals which, given a mode, can enumerate and propose all other modes.

be enumerated by inverting any subset of the columns of the sampled rotation matrix such that the determinant remains +1 (as opposed to −1 for an inversion of an odd number of columns). Figure 3-10 visually demonstrates these symmetries for SE(2) and SE(3) Although the dynamics will typically penalize one mode over others, it sometimes happens that the slice sampler locks onto a particular mode. The solution is simple: we propose a fixed number of MCMC samples, one for each enumerated mode. This is of minimal cost because there is only one other mode in SE(2) and three other modes in SE(3).

When sampling rotation full conditionals, we use characteristic width $w = 0.01\pi$ and a maximum of 10 doubling iterations. Ten samples are drawn, then the MCMC proposals for rotation symmetries are proposed starting from the final sample.

3.5.4 Part Associations

The conditional distribution for a single assignment to an existing part $k \geq 1$ is given by,

$$p(z_{tn} = k \mid y_{tn}, x_t, \omega, \theta_t, \pi, E) \propto \pi_k \, p(y_{tn} | x_t, \omega_k, \theta_{tk}, E_k) \tag{3.39}$$

where π_k is the stick weight of part k. Conversely, association to a new part is given by,

$$p(z_{tn} = -1 \mid y_{tn}, x_t, \pi) \propto \tag{3.40}$$
$$\pi_* \int p(y_{tn} \mid x_t, \omega_*, \theta_{t*}, E_*) \, p(\omega_*, \theta_{t*}, E_*) \, \mathrm{d}(\omega_*, \theta_{t*}, E_*)$$

where π_* is the stick weight corresponding to the base measure (i.e., all uninstantiated parts). This is not analytic in our model, but can be effectively approximated by Monte Carlo sampling of parts (need only be done once) or approximation by a constant (since the predic-

tive distribution of parts will be broad, but centered at x_t). We obtain satisfactory results with both approaches.

3.5.5 Conjugate Conditionals

We show that driving noise covariance Q for body frame of reference x_t is a product of an Inverse Wishart prior with a product of multivariate Gaussian likelihoods, yielding analytic sampling updates by conjugacy. The same reasoning holds for part transformation driving noise covariances $\{S_k\}_{k=1}^K$.

The posterior distributions for Q is:

$$p(Q \mid x_{1:T}) \propto \mathrm{IW}(Q \mid \cdot) \prod_{t=1}^{T} \mathrm{N}_{\mathrm{L}}\left(x_t \mid x_{t-1}, Q\right) \tag{3.41}$$

$$= \mathrm{IW}(Q \mid \cdot) \prod_{t=1}^{T} \mathrm{N}\left(\begin{pmatrix} V^{-1}_{x_{t-1}^{-1} x_t} d_{x_{t-1}^{-1} x_t} \\ \phi_{x_{t-1}^{-1} x_t} \end{pmatrix} \middle| 0, Q \right) \tag{3.42}$$

The terms inside the product are all computable given $x_{1:T}$, so this is an Inverse-Wishart multiplied by a product of Gaussians. In this case, the posterior is conjugate to the prior, yielding Inverse Wishart updates (see [73], Appendix A). The same form and reasoning applies for S_k, hence samples can also be analytically drawn for each S_k.

Part observation covariances E_k have the form:

$$p(E_k \mid \omega_k, \{x_t, \theta_{tk}, \{y_{tn}, z_{tn}\}_{n=1}^{N_t}\}_{t=1}^{T}) \propto \tag{3.43}$$

$$\mathrm{IW}(E_k \mid \cdot) \prod_{t=1}^{T} \mathrm{N}\left(\left(x_t \omega_k \theta_{tk}\right)^{-1} \tilde{y}_{tn} \mid \tilde{0}, \tilde{E}_k \right)^{\mathbb{I}(z_{tn}=k)}$$

As above, this posterior is also Inverse Wishart.

3.5.6 Data-Dependent Priors

Results in the evaluation (Chapter 3.6 were computed by using data-dependent priors that are similar in spirit to those used for static Dirichlet Process Mixture Models. All Inverse Wishart priors (for Q, $\{S_k, E_k\}_{k=1}^{\infty}$) were set to ten degrees of freedom, making the prior weak in the sense that it accounts for 10 pseudo-observations (among tens to thousands of observations incorporated into the posterior).

The Inverse Wishart scatter matrix prior for Q was set so that the expected per-timestep body rotation was 0.25 radians ($\approx 15°$) and expected per-timestep body translation was the mean absolute difference between time-adjacent pairs of observation sets.

The Inverse Wishart scatter matrix prior for S_k was set so that the expected per-timestep part rotation was 0.025 radians ($\approx 1.5°$) and expected per-timestep part translation was the mean absolute difference between time-adjacent pairs of observation sets (expected translation for parts and body are the same under the prior).

The Inverse Wishart scatter matrix prior for part observation covariances E_k was set to 0.1 times the mean observation set variance.

The prior for the initial body transformation was set to identity mean rotation with mean translation equal to the mean of the first observation set. The initial body transformation covariance was set diagonal and broad, so that π radians were within one standard deviation of rotation covariance, and body translation variances were set equal to the variance of the first observation set.

Canonical part transformations ω_k were set to identity mean transformation with π radians being within one standard deviation of rotation covariance, and canonical part translation variances set equal to the variance of the first observation set.

3.6 Evaluation

We compare quantitatively and qualitatively to nonparametric and parametric baselines in Chapter 3.6.1. We present results on dynamic mesh data in Chapter 3.6.2 and on chaotic double pendulum data in Chapter 3.6.3. We additional look at the posterior motion of parts and demonstrate object segmentation based on relative part motion in 3.6.3. We show synthesize motion from a learned representation in Chapter 3.6.4 and transfer of learned representations to a novel dataset in Chapter 3.6.5.

3.6.1 Quantitative Comparison

We examine *part discovery* performance on three object motion datasets and compare to manually-annotated ground-truth. We emphasize that annotations are not incorporated into the inference procedure. We refer to the datasets as `hand`, `spider`, and `marmoset`. `hand` and `spider` are 2-D image data, while `marmoset` is 3D data unprojected from a depth camera. Inference utilizes 12−44 frames (depending on the dataset) and results are compared to five manually-annotated ground-truth frames (where ground truth is the number of parts and their segmentations, discussed below). In each dataset, parts have nearly indistinguishable appearances and none of the compared methods use an appearance model. Consequently, part discovery is achieved via analysis of motion dynamics. Inputs only contain foreground (i.e., background is removed), as is done in related works [126].

We report multi-object tracking and segmentation (MOTS) metrics [206], which measure how well the part associations overlap with groundtruth part segmentations (MOTSA, sMOTSA, MOTSP) and how stable the part associations are over time (IDS). These metrics are intended for segmenting multiple objects, but we repurpose them to segment multiple parts of a single object. Comparisons are with IoU 0.3.

Figure 3-13 shows example ground-truth segmentations for each dataset used for quantitative comparison. Ground-truth was hand-labeled, and the number of parts were chosen at the granularity supported by

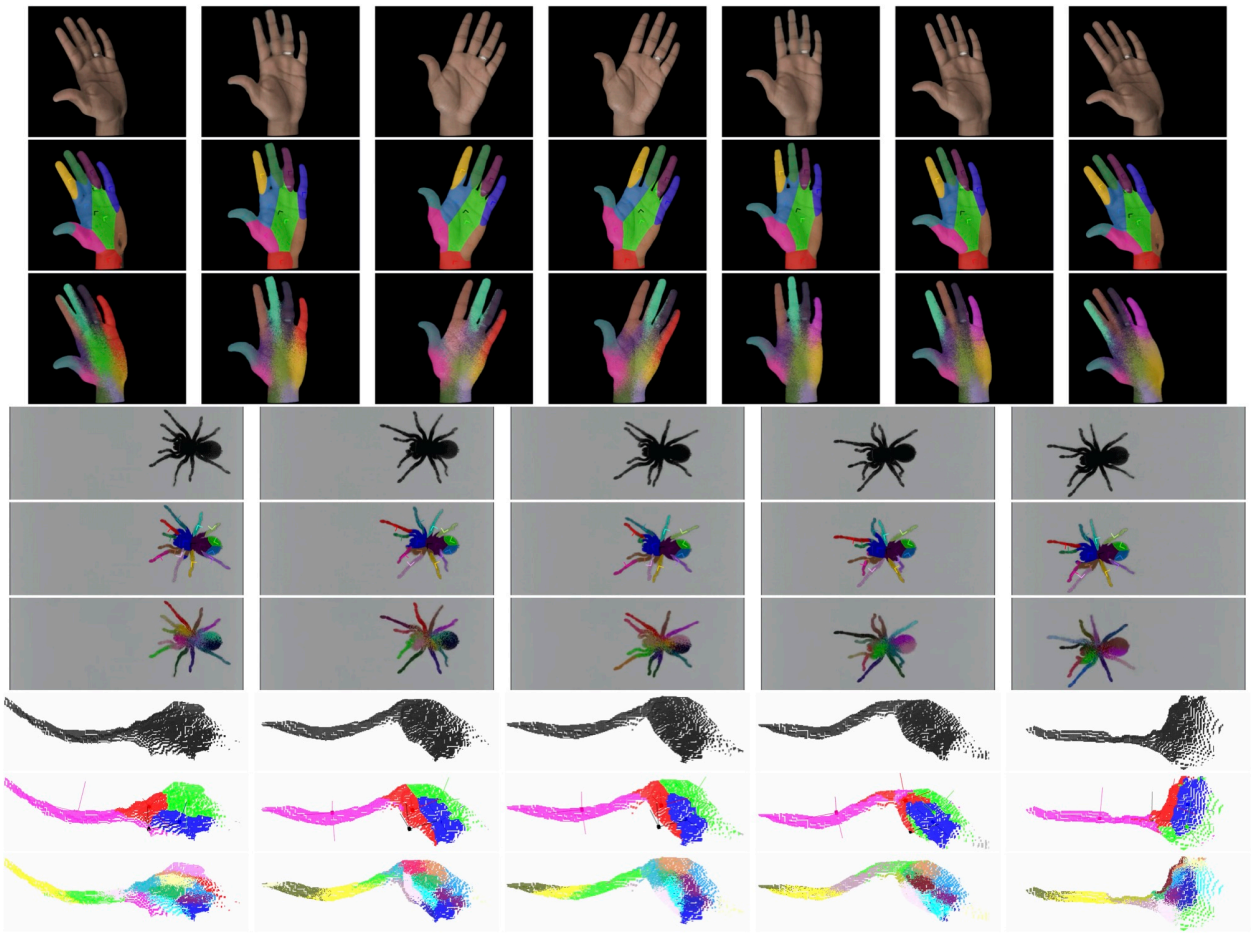


Figure 3-11: Example part associations in hand, spider and marmoset. For each sequence, example frames from the original video are shown (*top-row*) with part associations and object/part coordinate frames overlaid from our method (*middle-row*) and baseline NPE associations (bottom-row). Parts estimated by our method are largely consistent over time, even for the highly-articulated spider legs.

the dataset (e.g. marmoset has head, body and tail but not hands or feet because they were not visible from the top-down RGB-D views).

We compare against two baselines: the Bayesian nonparametric model of [233] (discussed in Chapter 3.7), which we call the nonparametric extents model npe, and a parametric modification of [233], so that it is given the advantage of knowing the true number of parts. We call this the parametric extents model pe. Neither npe nor pe consider part *persistence* over time (as we do), so for these methods we use the Hungarian algorithm [141] to compute part correspondences between pairs of timesteps on the distance (in the body frame) of component means.

Taken together, our model, and the two baselines, constitute an ablation study in which we consider unknown number of parts with Lie group dynamics, and unknown / known number of parts, without Lie group dynamics. In all cases, we compute mean and standard deviation of MOTS statistics on 100 samples taken from a Markov chain of

Dataset	Method	IDS	MOTSA	MOTSP	sMOTSA
hand	ours	**0.00 ± 0.00**	**2.79 ± 0.30**	**0.71 ± 0.01**	**1.34 ± 0.24**
	npe	4.45 ± 1.84	1.93 ± 0.8	0.51 ± 0.01	−4.2 ± 0.78
	pe	4.03 ± 2.11	1.57 ± 0.44	0.47 ± 0.01	−0.33 ± 0.37
spider	ours	**5.14 ± 1.49**	**3.44 ± 0.25**	**0.55 ± 0.02**	**1.26 ± 0.18**
	npe	19.6 ± 2.88	−4.4 ± 0.92	0.51 ± 0.01	−6.72 ± 0.9
	pe	17.28 ± 3.06	1.73 ± 0.31	0.52 ± 0.01	−0.24 ± 0.27
marmoset	ours	1.24 ± 0.65	1.39 ± 0.89	**0.49 ± 0.02**	−0.47 ± 0.71
	npe	3.18 ± 1.28	−32.44 ± 2.78	0.35 ± 0.01	−34.06 ± 2.72
	pe	**0.43 ± 0.51**	**3.86 ± 0.17**	0.48 ± 0.00	**1.77 ± 0.17**
average	ours	**2.12 ± 0.71**	**2.54 ± 0.48**	**0.58 ± 0.02**	**0.71 ± 0.38**
	npe	9.07 ± 2.0	−11.63 ± 1.5	0.46 ± 0.01	−15.0 ± 1.47
	pe	7.25 ± 1.89	2.39 ± 0.31	0.49 ± 0.01	0.39 ± 0.27

Table 3-12: Quantitative comparison of Nonparametric Parts Model. (ours) with nonparametric baseline npe and parametric baseline pe using MOTS metrics. Lower IDS is better, higher MOTSA, MOTSP, and sMOTSA is better. Best-performing method is emboldened.

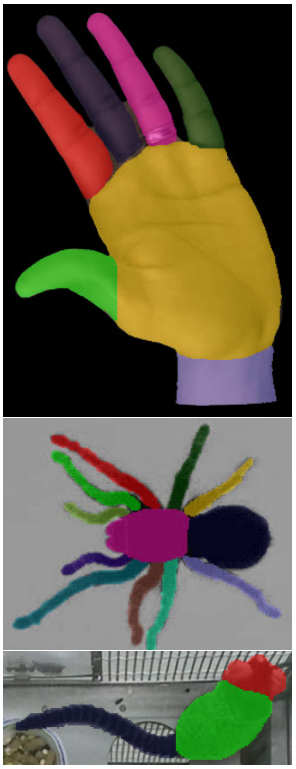


Figure 3-13: Groundtruth segmentations used for Nonparametric Parts Model.

1000 samples, use data-dependent priors (specified in Chapter 3.5.6), and set concentration parameter $\alpha = 0.1$. Table 3-12 shows quantitative results while Figure 3-11 show qualitative comparisons between our method and the baseline.

Our model outperforms the nonparametric baseline in all datasets and metrics. The pe baseline (which benefits from knowing the number of parts in the groundtruth) outperforms our method on label switches (IDS) and overall quality (sMOTSA) on the 3D marmoset data. This is largely due to noisy data from the depth sensor generating observations from the background that are distant from the object, but not so distant as to be relegated to the base measure. We see very little ID switching (IDS) and relatively high precision (MOTSP) in our model, which we attribute to the canonical parts ω_k enforcing that each part transformation θ_{tk} move stably. Visually, part assignments correspond best to ground-truth parts that are extremities (fingers, legs, tails), but tend to oversegment large object interiors (palms, bodies). We attribute this to the ellipsoidal observation model but find that, for the purposes of part analysis, it has no obvious negative impact.

3.6.2 Dynamic Mesh Segmentation

We apply our method to the squat1 sequence in the articulated mesh dataset of [204], decomposing the mesh sequence into parts as shown in Figure 3-14. Note that legs are segmented into two parts each, while arms are segmented into one part. This is consistent with the movement in this sequence where the legs bend, but the arms are held straight. Minor artifacts appear when the lower-left leg (red) has small numbers of associations above the knee when the person is squatting, but not when standing straight up. Qualitatively, the results conform

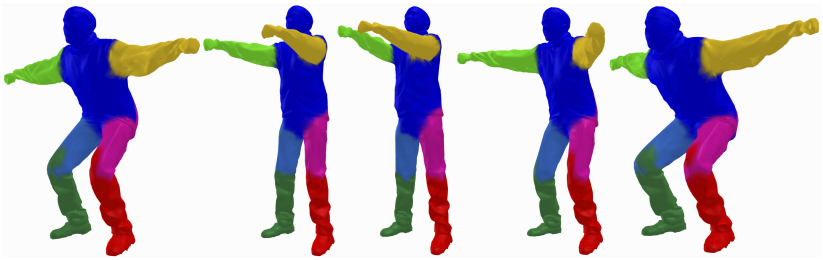


Figure 3-14: Dynamic mesh segmentation. By using points sampled inside a mesh as the input to our nonparametric parts model, then computing associations to mesh vertices, our model can learn parts and dynamics from mesh data. Additional views in Figure 3-1.

to human part interpretation.

3.6.3 Motion Analysis

We show how our model facilitates novel object / part analysis. Beginning with Figure 3-15, we visualize part diagrams for `hand` and `spider`. Dotted ellipses show the observation noise model E_k for each part (in the object frame), while solid ellipses show the covariance for that part's translation across time. Because the part translation covariances are spatially separated, the model resists label switching between parts because they tend to stay proximate to their canonical frame. We observe that the part translation covariances are tight for the hand, but horizontally smeared for the spider–this is expected, because the fingers moved very little in `hand` compared to the legs in `spider`.

One analysis that our model enables is the comparison of part motions in the body frame (i.e. motion not from the object moving, but from its parts). By integrating each part's motion over time within the body frame we can determine which areas of an object experience high or low *relative* motion. Figure 3-16 shows that, for `spider`, the legs are able to be segmented from other parts due to their rapid motion.

We also show that our model can segment the rapid, chaotic motion of a double pendulum. Partial confusion of assignments occurs when the pendulum is folded on itself (e.g., third image in first row) but part transformations are effectively maintained through areas of confusion.

Figure 3-15: Part posteriors for *hand* and *spider*. Dotted ellipses are the mean part covariance, solid ellipses visualize the part posterior location covariance. Points are observed part locations used for the posterior updates. The leg locations of spider are smeared due to their articulation whereas the fingers of the hand are concentrated.

3.6.4 Motion Synthesis

In Figure 3-18, we sample new part motions from the model after all parameters have been sampled from the `spider` dataset. Specifically, we generate new body transformations, in which the spider is subject to constant velocity and no rotation. Part transformations θ_{tk} are seeded with inference results then resampled from their full conditionals. Observations are taken from a single frame of the original video, projected into their respective part coordinate systems according to

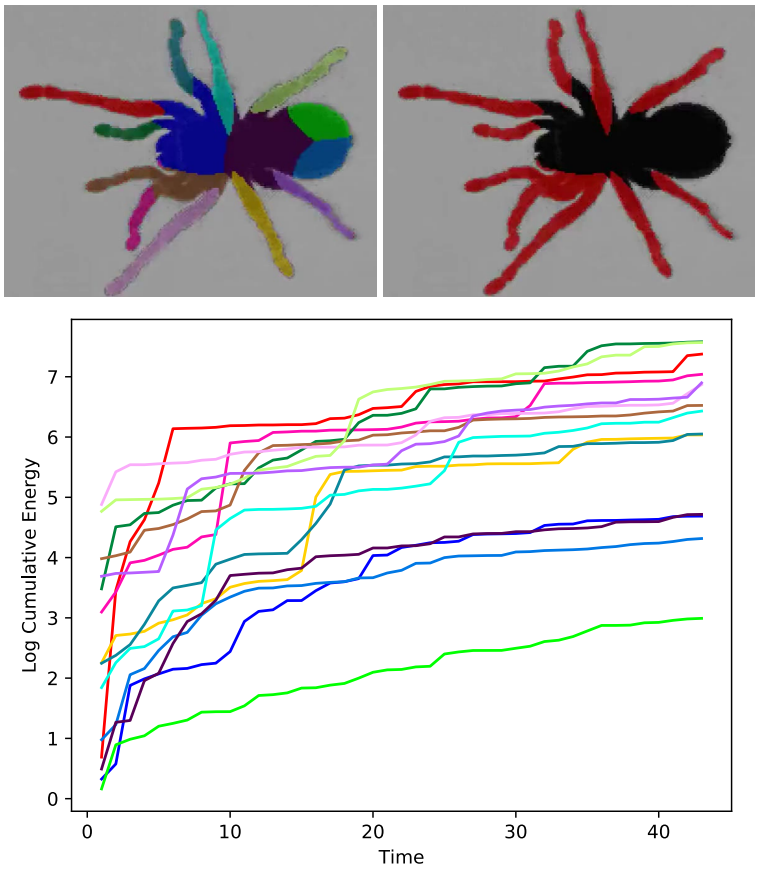


Figure 3-16: Nonparametric Parts object segmentation based on relative part motion over time. Whereas the parts nearest the body center exhibit little motion (in the body frame), the extremities of `spider` exhibit large amounts of motion. (Top-Left): Part associations. (Top-Right): Part segmentation based on motion energy. (Bottom): Log cumulative part motion energy across time (color-coordinated to associations).

the inferred part assignments, then reprojected to new world coordinates using the newly synthesized body and part transformations at each time. We stress that these are novel part motions and that they can be generated for arbitrary durations and body paths.

We observe that parts close to the spider's center exhibit relative stability, and the legs demonstrate the expected rhythmic walking motion. The pedipalps (the two front appendages) display implausible 'baton twirling-like' rotations, however. This is because these parts undergo foreshortening and occlusion in the original dataset. Since occlusion is not explicitly handled by our model in SE(2), inference permitted large rotations to explain observations on the pedipalps as they go from visible to not visible and vice versa (causing label switches along the way). Nevertheless, the spider's basic walking motion remains recognizable.

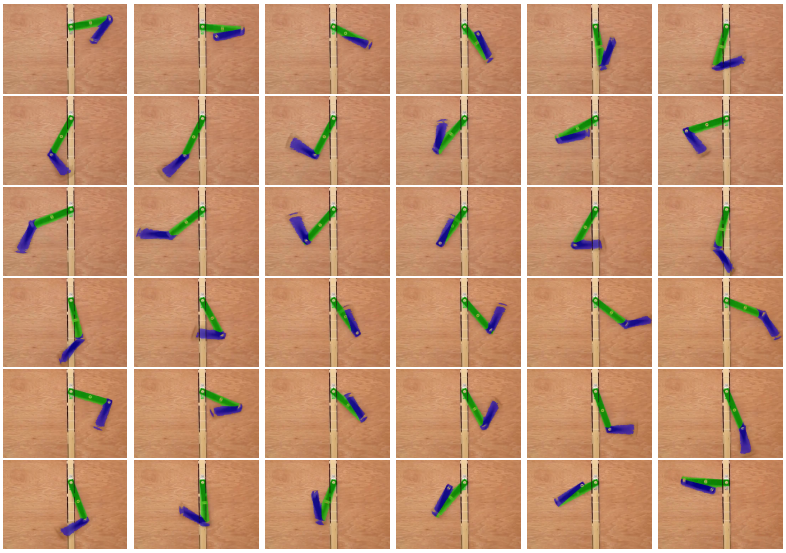


Figure 3-17: Nonparametric Parts Model segments a double pendulum.

3.6.5 Generalization

We demonstrate that our model can reason about the motion and parts of different instances of the same type of object across multiple videos. This is accomplished by assuming that the number of parts and the canonical part transformations, ω_k, are shared by similar objects, but that the motion parameters are distinct. In this experiment, we sample all parameters (including number of parts) from a video containing one instance of an object. In the second video, we restrict sampling to associations z_{tn}, body transformations x_t and part transformations θ_{tk}. Figure 3-19 shows RGB-D data projected into 2D for two videos; all model parameters are initially sampled in the video of the top row, then body and part transformations are sampled in the second video.

We note that part assignments correspond reasonably across videos. By reasoning in 3D, our model accommodates scale changes within and across videos, such as when the object is closer or further from the camera. While we do see some migration of part locations on the torso, this is due to the proximity of the respective ω_k's combined with sufficiently free motion dynamics. Regardless, torso parts remain associated to the torso, and the tail is consistently segmented.

3.7 Related Works

This work draws on body/parts models, Bayesian nonparametric dynamical models and Lie groups. Each contain a rich literature so we highlight only the most relevant details. Importantly, we are aware of no work that models body and part motion over time with Lie group dynamics, that is also unsupervised and nonparametric in the parts.

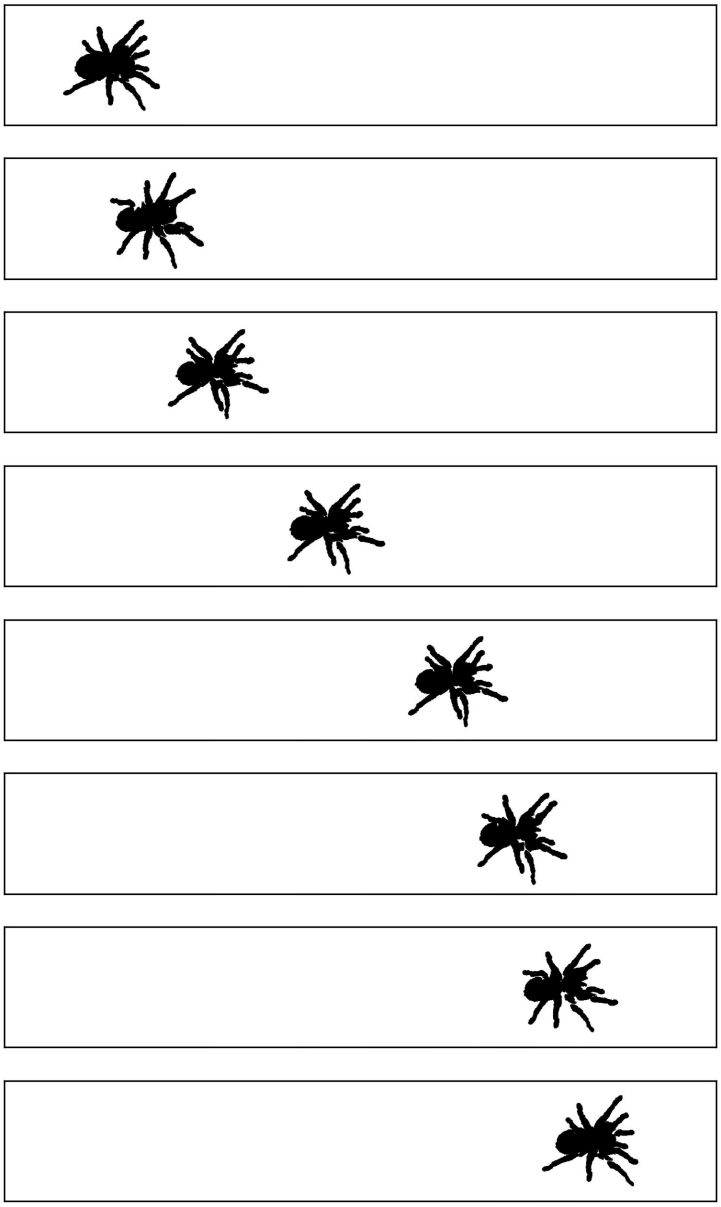


Figure 3-18: Novel body and part motions sampled from nonparametric parts model after being observing `spider` data. Body frame is subjected to constant velocity while part transformations are sampled.

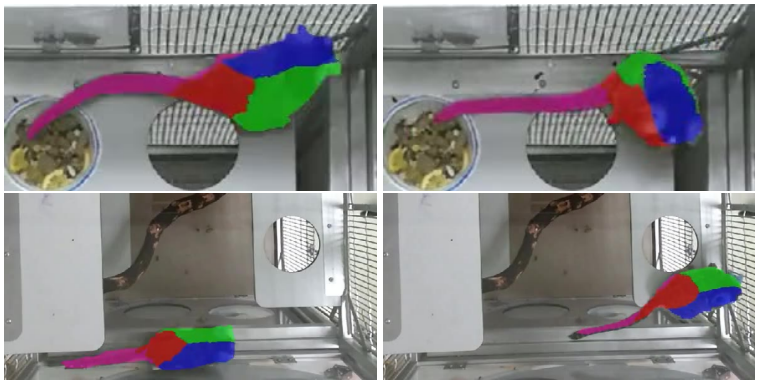


Figure 3-19: Example of how our model can infer results on multiple videos of the same type, but different instances, of an object in motion. These results were computed on RGB-D data but are visualized in 2D.

3.7.1 Body and Part Models

The many treatments of part-based modeling begin with the pioneering work on human models of pictorial structures [62] and cardboard people [104]. Later work on deformable parts models [59] removes the need to define object-specific part configurations. Building on the success of offline analysis, real-time human pose tracking is now possible as well [185, 67]. All of these methods require specifying the number of parts. More detailed shape and pose models have been developed for a variety of objects, using a combination of known body models, mesh representations and sophisticated collection schemes including multiple cameras, IMUs, lasers and/or specially-painted targets [131, 234, 29, 167].

Unsupervised methods [223, 126, 174, 234, 235] have significant restrictions such as working for only 2D or only 3D data, or requiring annotated landmarks or point correspondences. In contrast, our unsupervised method works for 2D and 3D inputs, requires only a single sensor observing an object in motion, requires no distinctive or annotated object markings, and no observation correspondences.

3.7.2 Lie Groups

Our work relies on the Lie group SE(D), the space of rigid transformations, for representing body and part motion. Lie groups have been used extensively in robotics and computer vision tasks such as SLAM [34], navigation [125], and parts-based models [32, 68, 86]. Defining observation models for Lie groups is challenging since the group is not a vector space. As such, notions of distance (and therefore distributions) require special care [157, 228], *e.g.*, simple additive noise models violate the group topology. Our approach defines a distribution (Gaussian) in the tangent plane about an element of the group [213]. Most works that model dynamics with SE(D) perform inference with approximate filters or smoothers, commonly the EKF [30] or UKF [34]. One exception

that does full posterior inference is [189], though that work is not a dynamical model. See [55] for an accessible introduction to Lie groups, and [83] for a more thorough introduction.

3.7.3 Nonparametric Models

Sequential models extending the well-known Dirichlet process (DP) [12] include the HDP-HMM [193], sticky HDP-HMM [66], infinite HMM [19], and infinite factorial HMM (ifHMM) [69]. Each of these permit an infinite number of states, but are restricted to discrete labels. Extensions to continuously-varying latent states include the HDP-SLDS [65], dynamic HDP [171], mixture of DPs [54], and the evolutionary HDP [229]. While each has the desirable property of shared global dynamics, none capture component persistence allowing new atoms at each time instance. This is undesirable for parts modeling as objects do not tend to acquire and lose parts over time and nonparametric priors already risk creating duplicate parts [66].

Closely related is the infinite factorial dynamical model [202], a continuous extension of the ifHMM which only permits shared global binary on/off states, and the Transformed Dirichlet Process [191], a DP allowing multiple groups of observations to share the same set of atoms (but with no dynamics). Most relevant, and what we use for comparison, is the Bayesian nonparametric model of Zhou et al. [233], a linear dynamical model where parts are independently sampled from a Dirichlet process at each time (but with no part persistence or Lie group representation).

3.8 Conclusion

We have demonstrated that our nonparametric representation of kinematic bodies infers meaningful part decompositions of objects in an unsupervised way, by simply observing them in motion (Chapters 3.6.1, 3.6.2). Our Lie group representation (Chapter 3.4) constrains articulations of moving parts to physically plausible kinematic states, without the requirement of object-specific knowledge such as skeletal structures. Part decompositions can be learned on very short sequences, and generalize to other datasets and instances of the same object type (Chapter 3.6.5). In contrast to methods that rely on extensive training data and/or object-specific 2D/3D models, we demonstrate robust analysis by direct observation of single instances of an object, without distinct visual part appearance.

Our model simplifies inference and motion analysis while suggesting straightforward extensions. Part persistence ensures that the representation of parts persists over a video sequence, even if parts become occluded. Hierarchical extensions over multiple videos of similar objects, or multiple videos and multiple objects, would be robust to part occlusions in any single video. Explicit models of part shape may

avoid over-segmenting large body regions and visual appearance modeling will help part segmentation when objects have visually distinct parts. Modeling gravity and friction may enable estimation of physical object properties such as mass and inertia. Finally, incorporating distributions on manifolds into probabilistic programming languages would expedite development and enable an entirely new class of models with representations that could be interpreted by users without intimate knowledge of Lie groups and Riemannian manifolds.

Chapter 4

Multi-Object Tracking with Uncertainty Quantification

"There are known unknowns ... and there are unknown unknowns"

— Donald Rumsfeld

In multi-object tracking the trajectories of an unknown number of objects are estimated from noisy observations over time. Assigning observations to objects is known as the *data association problem*. The well-known multidimensional assignment formulation [168] of data association provides a constrained objective function in which each observation is assigned either to an object or to clutter and no object is assigned more than one observation at any time. It permits arbitrary object arrivals and departures and assumes that there is a fixed cost for each association hypothesis. The complexity of the multidimensional assignment formulation is factorial in the number of observations at each time and exponential in the number of timesteps, making it NP-hard [26]. The multidimensional assignment objective is defined and related to other approaches in Chapter 4.3.

Most approaches to multi-object tracking solve an objective function, such as the multidimensional assignment objective, using optimization [84, 119, 109, 226, 35]. While this can provide automated and sometimes fast inference with the help of sophisticated solvers, it yields a single, point estimate solution to the data association problem. Yet, ambiguities commonly occur in multi-object tracking data, such as when two targets with similar appearance and kinematic state approach one other and then diverge (see Figure 1-4 for an example). A tracker that incorrectly estimates the targets as crossing when they did not (or vice versa) has committed an identity switching error. Identity switches manifest as multiple modes in the objective function, but an optimization-based solution will only identify one mode.

All approaches to multi-object tracking reason over data associations, but *few* explicitly represent uncertainty, much less make it available for subsequent tasks. Traditional applications in security [21], surveillance [152], sensor networks [177] and robotic localization [149] favor real-time performance. More recently, there is increased interest in the use tracking for follow-on decision making and analysis including the creation of gold-standard datasets [207], sports analytics [61],

study of animal behavior [233] and cellular dynamics [13]. Such applications benefit significantly from *accurate* representations of uncertainty to inform subsequent analysis.

State-of-the-art tracking algorithms accumulate thousands of identity switching errors on short sequences as shown in the MOT Challenge benchmarks [136]. These errors limit the utility of multi-object tracking in follow-on analysis and decision making because tracking errors will propagate. In sports analysis, tracking errors will lead to faulty calculation of player attributes. Worse, in scientific applications, identity switching can lead to support for incorrect conclusions.

Errors cannot be avoided in multi-object tracking, but an accurate representation of uncertainty can at least highlight where they are likely to have occurred. Yet, very few approaches to multi-object tracking are formulated in a way that permits uncertainty quantification. Of the ones that do, they either do not solve the general multi-object tracking problem (batch reasoning, unknown number of objects, arbitrary arrival/departure, clutter), they do not use exact inference, or they are limited by gating heuristics that restrict the association hypotheses they can represent. I design the first fully-Bayesian model that addresses the general multi-object tracking problem which explicitly quantifies uncertainty, uses efficient, exact inference, and is free of gating heuristics.

4.1 Approach

This work develops the Joint Posterior Tracker (JPT), a generative, Bayesian model on associations and trajectories for the general multi-object tracking problem. JPT supports arbitrary dynamics and observations models, including linear or nonlinear models in arbitrary dimension. It supports an unknown number of objects with a uniform prior over permutations of association labels in the range $[0, \infty)$. Arbitrary object arrival and departure times, and missed or false detections—collectively called event counts—are modeled with prior distributions on the number of these events that occur at each time. Constraints are placed on associations such that event counts can be uniquely identified from an association hypothesis. JPT adopts additional constraints used in the multidimensional assignment formulation: that every observation must be assigned to an object or clutter, and no object can have more than one observation assigned to it at any time. These constraints are straightforward to specify in the generative model, but make inference challenging. JPT reasons over a joint posterior of trajectories and associations; thus, it departs from the multidimensional assignment formulation because it does not assign a fixed cost to each association hypothesis. A fixed cost can be constructed for the multidimensional assignment formulation by marginalizing over all latent trajectories for a given association hypothesis.

Given the posterior distribution implied by JPT's generative model,

I construct Metropolis-Hastings inference [85] to draw posterior samples. This is difficult due to the existence of constraints and the exponential and factorial scaling of the number of possible association hypotheses. The dynamics and observation models parameterize a forward model, which inference uses where possible to aid reasoning over permutations of object-object and object-clutter associations. Directly using the forward model enables rejection-free inference in some cases, and endows JPT with efficient uncertainty quantification.

I evaluate JPT against sampling- and optimization-based trackers, both on traditional metrics and on a novel uncertainty quantification metric that matches a set of multi-object tracking samples to a known set of modes using a track comparison metric and discrete optimal transport. I collect a novel scientific behavior dataset consisting of long-term marmoset (a type of primate) movements with many partial and full occlusions (15k timesteps, 2 objects, 25k observations). i evaluate JPT using traditional CLEAR MOT tracking metrics [23] on the Marmoset dataset, as well as a Soccer dataset (1.5k timesteps, 22 objects, 12k observations). A synthetic dataset is created to evaluate uncertainty quantification. Batch Markov Chain Monte Carlo Data Association (MCMCDA) [153] is the primary baseline for JPT because it can, in principal, represent uncertainty, addresses the general multi-object tracking problem, and uses exact, sampling-based inference, modulo gating heuristics on the maximum spatial and maximum temporal distance between associations to a single track. Figure 4-1 compares JPT and MCMCDA uncertainty quantification; Chapter 4.9 further analyzes uncertainty quantification.

Uncertainty quantification is the focus of this work. Appearance modeling, which is required for performant comparisons on modern tracking benchmarks, is not treated. Instead, JPT is compared to a modern variant of Multiple Hypothesis Tracking (MHT) [109], which was shown to achieve state-of-the-art performance on modern tracking benchmarks when paired with deep appearance features.[26] MHT is compared to as a way to demonstrate the typical point estimate approach of multi-object trackers and to ground the CLEAR MOT tracking metrics calculated on each dataset. In our tracking comparisons, no approach makes use of appearance modeling.

[26] Since publication, MHT performance has been superseded by end-to-end deep neural network approaches. We do not compare to these approaches because appearance cannot be straightforwardly separated from data association.

Finally, I demonstrate that JPT's uncertainty quantification enables rapid improvement of track quality by reducing uncertainty that arises from ambiguous data association events. This is accomplished using Sequential Bayesian Optimal Experiment Design, in which possible human annotations are modelled as experiments that could be performed. The experiment which maximizes mutual information [24] (equivalently, minimizes JPT posterior uncertainty) is iteratively chosen in greedy fashion. Annotations are pairwise questions of the form "do observations y_t and $y_{t'}$ belong to the same object or not?"

4.2 Contributions

This work develops the Joint Posterior Tracker (Chapter 4.5), the first fully Bayesian multi-object tracker that addresses the general (i.e., batch) multi-object tracking problem (Chapter 4.3), uses exact inference, and is not limited by gating heuristics. JPT emphasizes joint uncertainty over association hypotheses and object trajectories that, combined with exact inference, permits representation of multiple possible outcomes as discrete samples from a Markov chain whose limiting distribution is the JPT posterior (Equation 4.10).

The JPT posterior is well-defined but it can only be evaluated up to proportionality. Furthermore, constraints imposed by solving the general multi-object tracking problem (Equation 4.4) complicate posterior inference. To enable efficient sampling from the JPT posterior, I construct MCMC proposals that reason over permutations of object-object and object-clutter associations (Chapter 4.6). Notably, I generalize the method of [144], which constructs a discretized grid in observation space and efficiently reasons over permutations on that grid by sampling from the joint distribution of a specially-constructed Hidden Markov Model. Whereas their method is explicitly limited to the observation space (else it loses detailed balance), I extend their approach to permutations in the latent space; specifically, the latent space of trajectories and associations. Doing so enables efficient exploration of potential identity switching errors while retaining detailed balance (Chapter 4.6.1, Appendix A.4)

Finally, I show that JPT explores posterior modes much more completely and efficiently (Chapter 4.9.4) with superior performance on standard multi-object tracking metrics (Chapter 4.9.5) on scientific and sports datasets as compared to baselines. Using JPT's accurate representation of uncertainty, I demonstrate automatic scheduling of a small number of disambiguations that facilitate rapid improvement in trajectory quality with a consequent reduction in posterior uncertainty (Chapters 4.9.6, 4.7).

4.3 Multidimensional Assignment

Multi-object tracking can be formulated in several common ways: as a set partitioning problem [15], a set packing problem [140, 230], a maximum-weight independent set problem [155] or a multidimensional assignment problem [168]. A review of these formulations is conducted by [45], who shows that the multidimensional assignment formulation is capable of representing a broader set of modeling assumptions; namely, it is not limited to pairwise terms as the set-packing, network-flow solutions [162, 22] are. We next define the multidimensional assignment problem, as it is most similar to how JPT is formulated.

Consider $t = 1, \ldots, T$ timesteps with corresponding observation sets $y = \{y_1, \ldots, y_T\}$ where the time-t observation set $y_t = \{y_{tn}\}_{n=1}^{N_t}$

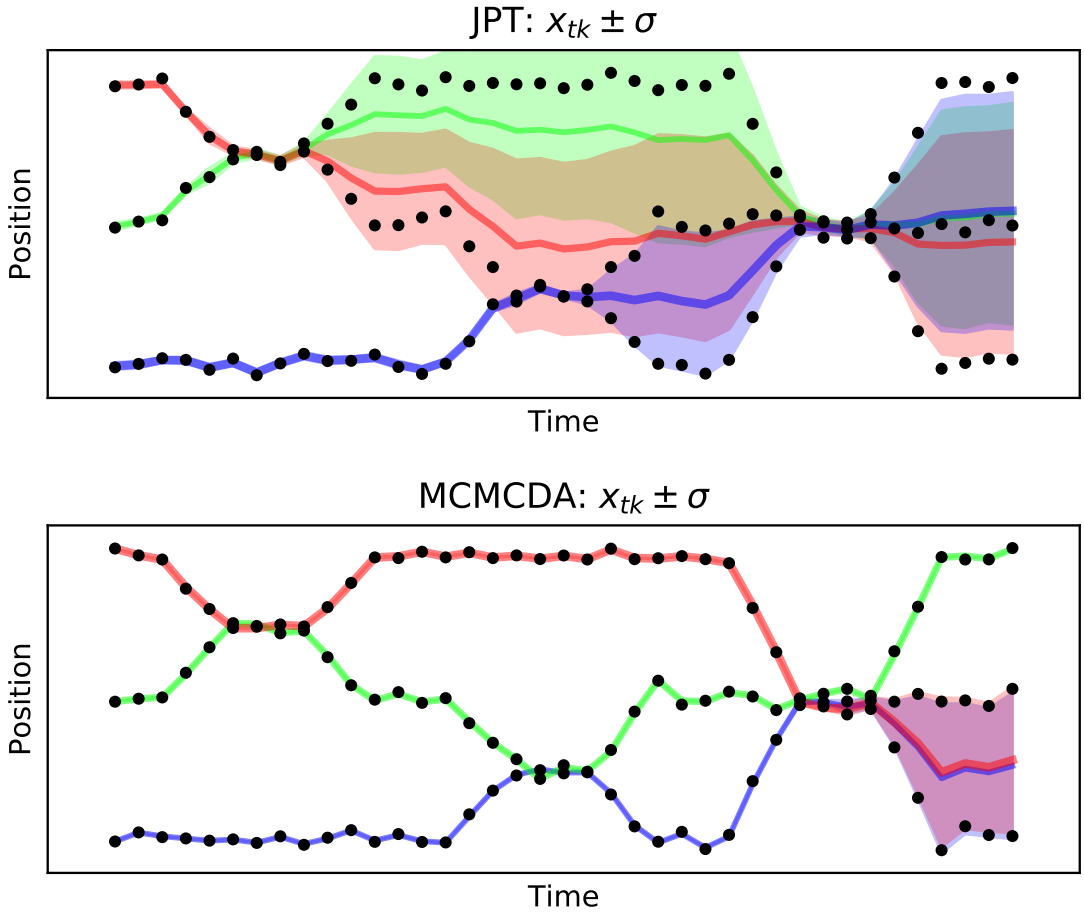


Figure 4-1: Mode capture of JPT and MCMCDA posterior trajectories. Trajectories ± one SD marginalized over sampled associations in a multi-object tracking scenario. JPT (**Top**) correctly captures the uncertainty from all ambiguous region while MCMCDA (**Bottom**) finds only one.

has N_t observations. Hence, y_{tn} is the n^{th} observation at time t. Let $I_t = \{0, 1, \ldots, N_t\}$ be an index set into y_t, where 0 indicates a false positive or missing detection. Define $\mathcal{P} = I_1 \times I_2 \times \cdots \times I_T$ as the set of paths through all index sets such that every path is length-T and has at least one non-zero index. Interpret a path with a single non-zero index as a false-positive. Interpret a path with two or more non-zero indices as an object. Define $\gamma(i_1, \ldots, i_T)$ as a fixed, real cost for path $(i_1, \ldots, i_T) \in \mathcal{P}$ where $i_t \in I_t$, and $B(i_1, \ldots, i_T)$ is a boolean variable signifying whether path $(i_1, \ldots, i_T)$ is included in a solution. Then the multidimensional assignment problem is to find the $B(i_1, \ldots, i_T)$ that

minimizes:

$$\begin{aligned}
\min \quad & \sum_{i_1=0}^{N_1} \sum_{i_2=0}^{N_2} \dots \sum_{i_T=0}^{N_T} \gamma(i_1, \dots, i_T)\, B(i_1, \dots, i_T) \\
\text{subject to} \quad & \\
& \sum \sum_{I \setminus i_t} \dots \sum B(i_1, \dots, i_T) = 1 \\
& B(i_1, \dots, i_T) \in \{0, 1\} \\
& \forall i_t = 1, \dots, N_t, \forall t = 1, \dots, T
\end{aligned} \tag{4.1}$$

The objective sums over the costs of all included paths in the solution. For each observation there is a constraint enforcing that it be claimed by exactly one path included in the solution (equivalently, that an observation is uniquely associated either to clutter or a distinct object). MHT [109] and JPDA [84] are deterministic solutions whereas MCMCDA [153] is a stochastic solution to the multidimensional assignment problem. The number of possible paths grow exponentially with T and factorially at each time with N_t. Solving this exactly is NP-hard [26, 158], forcing the above approaches to use gating heuristics such as a maximum distance between object locations and observations, a maximum distance between pairwise object locations, or a maximum number of consecutive missing detections.

We note that JPT represents a departure from the multidimensional assignment formulation because it does not assign a fixed cost to each association hypothesis. While a fixed cost could be constructed–such as by using smoothed state estimates or marginalizing out all trajectories–our focus is to explore and represent joint uncertainty in trajectories and data associations. JPT is not the only work to depart from a traditional multi-object tracking objective [11].

4.4 Related Works

We consider multi-object tracking approaches from the perspective of three capabilities. First, do they process measurements one frame at a time (single-scan, [181, 108, 33]), multiple frames at a time (multi-scan, [109]) or all at once (batch, [35, 153, 212]). Second, do they yield point estimates (as in optimization), [109, 84, 212] or represent multiple explanations (sampling or variational methods, [181, 199]). Third, do they utilize gating heuristics that restrict possible hypotheses [109, 153]. JPT is a batch, sampling-based tracker with no gating heuristics that reasons over a joint distribution of an unknown number of objects, their trajectories and the association of objects to observations.

Recent approaches to multi-object tracking emphasize sophisticated appearance, motion or shape modeling in an optimization-based framework [84, 119, 109, 226, 35]. They forego representing association uncertainty, providing a single point-estimate, and thus do not permit

	JPT	MCMCDA	Var	BP
Uncertainty	✓	✓	✓	✓
Exact Posterior	✓	✓		
General MOT	✓	✓	✓	

Table 4-2: Batch multi-object trackers that quantify uncertainty in data association. Joint Posterior Tracker (JPT): our method; MCMCDA: [153]; Variational Tracker (Var): [199]; Belief Prop (BP) Tracker: [216]

recovery from errors. The best of appearance models will still suffer from association errors in complex scenes.

Monte-Carlo approaches including [181], [108], [33] represent uncertainty via sampled realizations, but each are single-scan filters that do not incorporate future information.

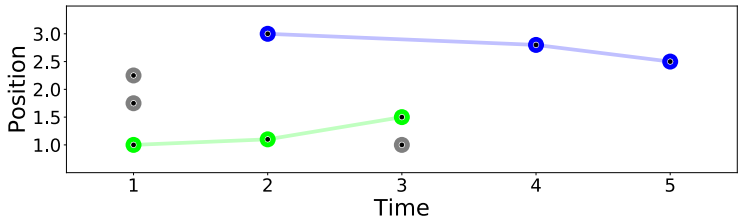
Random Finite Set (RFS) tracking methods propagate a Bayesian filter distribution defined on random sets. Data association is implicit in RFS but can be made explicit (as it is in JPT) by including track labels; when this occurs, they are called Labeled Random Finite Sets. Relations between RFS and explicit data association is an active areas of research [215, 44, 39]. The RFS *filter* models *marginal* states conditioned on past data—it does not model future information or *joint* state over time, as JPT does. RFS propagation induces super-exponential complexity necessitating heuristics such as truncation to K-best solutions, producing inaccurate UQ. A recently proposed smoothing RFS [205] models *marginal* states only and also uses K-best heuristics. Unlike RFS filters, MCMCDA samples from a posterior defined on the **general MOT problem** (batch tracking.

Table 4-2 summarizes related work that represents some aspect of uncertainty, such as marginal uncertainty using belief propagation [216] and approximate uncertainty using variational methods [199]. The former treats a fixed number of objects, while the latter samples from a variational approximation eliminating theoretical guarantees.

Markov-Chain Monte Carlo Data Association (MCMCDA) and its variants [153, 21, 72] is most closely related to JPT. MCMCDA can be run as a filter or in batch. Batch MCMCDA and JPT are both Bayesian treatments of the general multi-object tracking problem, but exhibit fundamental differences that we summarize here and expand on in Chapter 4.8:

1. MCMCDA defines a *marginal* posterior over associations that induces a *marginal* posterior on trajectories at *each* time whereas JPT defines a *joint* posterior over associations and trajectories over *all* time.

2. MCMCDA inference requires tuned gating heuristics that create excess objects if set too low or reduce inference to a random search if set too high and in all cases limits its ability to represent the configuration space of possible object associations. In contrast, JPT inference uses no gating heuristics and is instead made



Time	x_t	z_t	a_t	f_t	d_t	λ_t
1	1.0, $\emptyset$	1, 0, 0	1	2	0	0
2	1.3, 3.0	1, 2	1	0	1	0
3	1.5	0, 1	0	1	1	1
4	$\emptyset$, 2.7	2	0	0	1	0
5	$\emptyset$, 2.5	2	0	0	1	1

Figure 4-3: JPT's latent representation. (**Top**): Observations y colored by their association (green/blue for either of two objects, gray for clutter) and connected by sampled trajectories. (**Bottom**): Trajectories x, associations z and counts $M = \{a_t, f_t, d_t, \lambda_t\}_{t=1}^{T}$. Objects are observed at distinct arrival and departure times. Trajectories are length-T, padded by $\emptyset$ before arrival and after departure. Trajectory states with missing detections (blue at $t = 3$) are marginalized over.

efficient by using proposals based on its forward model. Unlike MCMCDA, JPT can represent any association hypothesis.

3. both MCMCDA and JPT represent posterior uncertainty, but in Chapter 4.9 we show that JPT quantifies uncertainty much more accurately than MCMCDA.

To our knowledge, batch MCMCDA is the only tracker that treats the general (i.e., batch) multi-object tracking problem, samples from its posterior exactly (modulo gating heuristics), and can, in principle, represent posterior uncertainty. As such, we use *batch* MCMCDA as a baseline for comparison to JPT.

4.5 Joint Posterior Tracker

The objective of multi-object tracking is to partition a set of observations across time into collections of objects such that every observation is uniquely assigned to one and only one object. The most general MOT formulation allows for clutter (false-positives), missed detections, unknown number of objects, and arbitrary object arrival and departure times. There are many formulations for multi-object tracking. In Chapter 4.3 we outlined several and defined the common multidimensional assignment formulation as it is most closely related to JPT.

JPT defines a joint distribution on trajectories and assignments. To reason over trajectories and assignments, we must define a generative model for observations $y = \{y_1, \ldots, y_T\}$ over all times $1, \ldots, T$ where $y_t = \{y_{tn}\}_{n=1}^{N_t}$. Vector-valued observation y_{tn} is the n^{th} observation at

time t and has dimension D_y. N_t is the total number of observations at time t. Table 4.1 summarizes notation by JPT.

Associations z define a partition of y into objects and clutter. Trajectories x are the latent states of objects over all times. JPT emphasizes accurate representations of posterior uncertainty. For clarity of exposition, we only model object position and velocity. However, shape or appearance models can be incorporated without modification of *any* equation in this work.[27] Following, Figure 4-3 can be used to ground definitions of JPT's latent representation and generative model.

[27] For example, linear Gaussian dynamics can be placed on appearance features, when available. In videos, this could take the form of deep appearance features extracted from a region around each observation as done in [109].

4.5.1 Event Counts $p(M)$

JPT explicitly models clutter (false-positives), missed detections and arbitrary arrival and departure times for an unknown number of objects. At time t, counts of new object arrivals a_t, clutter observations f_t, object detections d_t and object departures λ_t are modeled as

$$\begin{aligned} a_t &\sim \text{Pois}(a_t \mid \lambda_b) \quad & f_t &\sim \text{Pois}(f_t \mid \lambda_f) \\ d_t &\sim \text{Bin}(d_t \mid e_{t-1}, p_d) \quad & \lambda_t &\sim \text{Bin}(\lambda_t \mid d_t, p_\lambda) \end{aligned} \tag{4.2}$$

where $e_0 = d_0 = 0$ and $e_t = e_{t-1} + a_t - \lambda_t$ are counts of existing objects (those that arrived at time $t' \leq t$ and have not yet departed). Prior parameters λ_b, λ_f are the new object arrival and false alarm rates and p_d, p_λ are the detection and departure probabilities for existing objects. Every object is assumed to be observed at least twice: when it arrives and when it departs. Denote the set of all event counts as $M = \{M_1, \ldots, M_T\}$ where $M_t = \{a_t, f_t, d_t, \lambda_t\}$. From Equation 4.2, the generative model for latent counts M is,

$$\begin{aligned} p(M) &= \prod_{t=1}^{T} p(M_t \mid M_{t-1}) \\ &= \prod_{t=1}^{T} p(a_t)\, p(f_t)\, p(d_t \mid e_{t-1})\, p(\lambda_t \mid d_t) \end{aligned} \tag{4.3}$$

Variable	Description
$x_{tk} \in \mathbb{R}^{d_x}$	Trajectory
$y_{tn} \in \mathbb{R}^{d_y}$	Observation
z_{tn}	Association
M_t	Event counts a_t, d_t, f_t, λ_t
$a_t \geq 0$	Object arrivals
$d_t \geq 0$	Object detections
$f_t \geq 0$	Clutter
$\lambda_t \geq 0$	Object departures
a_l	Annotation
$t \geq 1$	index for time
$k \geq 1$	index for objects
$n \geq 1$	index for observations
$l \geq 0$	index for annotation
$\sigma(x)$	Trajectory permutation
$\sigma(z)$	Association permutation

Table 4.1: Joint Posterior Tracker notation.

4.5.2 Associations $p(z \mid M)$

JPT represents the association of each observation to an integer-labeled object or clutter. We denote the association of observation y_{tn} by the latent random variable $z_{tn} \in \mathbb{Z}^+ \cup \{0\}$ where $z_{tn} = k > 0$ if y_{tn} is associated to target k at time t and $z_{tn} = 0$ if y_{tn} is associated to clutter. An association hypothesis is the set of all associations $z = \{z_1, \ldots, z_T\}$ for $z_t = \{z_{tn}\}_{n=1}^{N_t}$.

Conditioned on event counts M, association hypotheses z have a uniform prior over the space of possible associations subject to constraints that enforce that all observations are either associated to an object or clutter (satisfied by definition of z_{tn}), that an object claim at most one observation at each time t (first constraint in Equation 4.4)

and that associations be consistent with event counts M (remaining constraints):

$$p(z \mid M) \propto 1 \text{ if } \begin{cases} |\{n : z_{tn} = k\}| \leq 1 \quad \forall k > 0, \forall t \\ f_t = |\{n : z_{tn} = 0\}| \quad \forall t \\ a_t = |\{k > 0 : z_{tn} = k \text{ and } z_{t'n} \neq k \text{ for all } t' < t\}| \quad \forall t \\ \lambda_t = |\{k > 0 : z_{tn} = k \text{ and } z_{t'n} \neq k \text{ for all } t' > t\}| \quad \forall t \\ a_t + d_t = |\{n : z_{tn} > 0\}| \quad \forall t \end{cases} \tag{4.4}$$

Association hypotheses that do not satisfy these constraints have zero probability. As noted, the space of possible associations is exponential in time T and factorial in the number of observations N_t at each time t [158]. Reasoning over this large space is *the* fundamental challenge of data association. Incorporating the constraints above adds additional complexity as discussed in Chapter 4.6.

4.5.3 Dynamics $p(x \mid z)$ and Observations $p(y \mid x, z)$

We denote the trajectory of object $k > 0$ at time t by the latent random variable $x_{tk} \in \mathbb{R}^{D_x}$ with state dimension D_x and the set of *all* trajectories by $x = \{x_1, \ldots, x_T\}$ where $x_t = \{x_{tk}\}_{k=1}^{K(z)}$ with $K(z)$ being the number of objects in hypothesis z. Every object has a length-T latent trajectory, but is represented by $x_{tk} = \emptyset$ for any time before its arrival or after its departure. We define the dynamics model for objects $1, \ldots, K(z)$ as:

$$p(x \mid z) = \prod_{t=1}^{T} p(x_t \mid x_{1:t-1}, z) = \prod_{t=1}^{T} \prod_{k=1}^{K(z)} p(x_{tk} \mid x_{t'k}) \tag{4.5}$$

where $t' = t - 1$ and the observation model as

$$p(y \mid x, z) = \prod_{t=1}^{T} p(y_t \mid z_t, x_t) = \prod_{t=1}^{T} \prod_{n=1}^{N_t} p(y_{tn} \mid z_{tn}, x_t) \tag{4.6}$$

As is commonly done, we specialize Equations 4.5, 4.6 to a linear Gaussian system yielding dynamics

$$p(x_{tk} \mid x_{t'k}) = \begin{cases} \mathrm{N}(x_{tk} \mid Fx_{t'k}, Q) & \text{if } x_{t'k} \neq \emptyset \\ \mathrm{N}(x_{tk} \mid \mu_0, \Sigma_0) & \text{o.w.} \end{cases} \tag{4.7}$$

The first line is a linear Gaussian system with system model F and noise covariance Q. The second line specifies a shared prior on trajectories with prior parameters μ_0, Σ_0 that are typically set to be broad over the observation space. We marginalize over missed detections (i.e., times when an object has already arrived but has no association) which has a closed-form expression for linear Gaussian dynamics.

The observation model becomes

$$p(y_{tn} \mid z_{tn} = k, x_t) = \begin{cases} \mathrm{N}(y_{tn} \mid Hx_{tk}, R) & \text{if } k > 0 \\ \mathrm{N}(y_{tn} \mid \mu_{\mathrm{FP}}, \Sigma_{\mathrm{FP}}) & \text{o.w.} \end{cases} \tag{4.8}$$

where the first line is a linear Gaussian system with observation projection H and observation noise covariance R. The second line specifies the model for clutter detections with prior parameters $\mu_{\mathrm{FP}}, \Sigma_{\mathrm{FP}}$, typically set to be broad.

4.5.4 Joint Distribution

Finally, the joint posterior over trajectories x, associations z and counts M given observations y is,

$$p(x, z, M \mid y) = \frac{p(y \mid x, z, M)\, p(x, z, M)}{p(y)} \tag{4.9}$$

$$= \frac{1}{Z}\, p(y \mid x, z)\, p(z \mid M)\, p(x \mid z)\, p(M) \tag{4.10}$$

where each term on the RHS is respectively given by Equations 4.6, 4.4, 4.5, and 4.3. The intractable normalization constant $Z = p(y)$ is not needed for inference. We develop a Metropolis-Hastings sampler for this non-trivial posterior in Chapter 4.6.

4.6 Inference

Sampling from the posterior distribution in Equation 4.10 is complicated by the constraints of Equation 4.4 and because the number of association hypotheses grows exponentially in T and factorially in N_t. With no analytic form and computationally infeasible enumeration of all hypotheses, we turn to the Metropolis-Hastings (MH) algorithm [85].

The MH algorithm enables sampling from intractable distributions by constructing a Markov chain whose unique stationary distribution is the desired distribution. Samples from this chain converge in distribution to the desired distribution, regardless of starting state. MH constructs transition distributions q^* that maintain detailed balance,

$$\frac{p(x', z', M' \mid y)}{p(x, z, M \mid y)} = \frac{q^*(x', z', M' \mid x, z, M, y)}{q^*(x, z, M \mid x', z', M', y)} \tag{4.11}$$

resulting in a chain where Equation 4.10 is a stationary distribution. MH accepts a proposed sample (x', z', M') from an arbitrary proposal distribution q with probability $\min(1, R)$ where the normalizers cancel,

$$R = \frac{p(x', z', M' \mid y)}{p(x, z, M \mid y)}\, \frac{q(x, z, M \mid x', z', M', y)}{q(x', z', M' \mid x, z, M, y)}. \tag{4.12}$$

Metroplis-Hastings is a conceptually simple algorithm to implement.

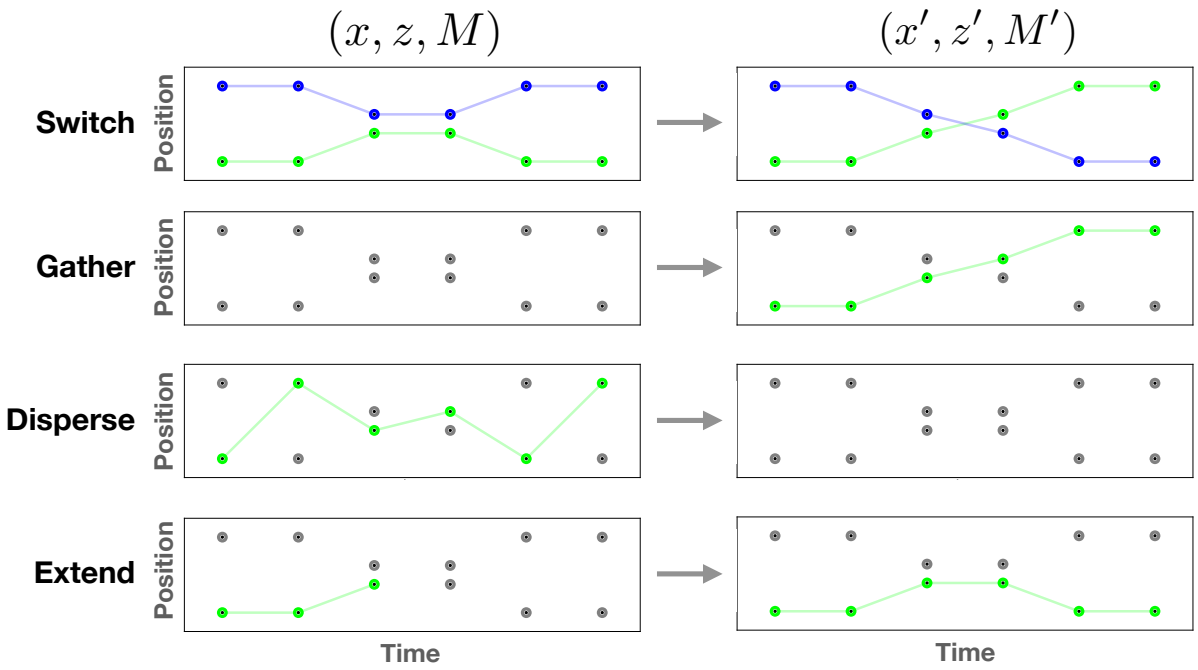


Figure 4-4: Examples of each JPT proposal. Left column is the input state (x, z, M) and right column is the output state (x', z', M'). Switch proposals can reason over many objects, but are shown here for two. Black points are observations y, encircled in the color of their association (green or blue for objects; grey for clutter). Example trajectories x are visualized as lines colored according to their associated object.

The difficulty comes in designing proposal distributions that effectively and efficiently explore the posterior. Following, we design proposals that rapidly explore high-probability regions in the JPT posterior, hopping between different modes (later quantified in Chapter 4.9). We then describe closed-form Gibbs sampling of joint trajectories conditioned on associations in Chapter 4.6.5.

We design Metropolis-Hastings proposals that make large moves in the latent space (including mode hopping) by reasoning over permutations of the latent state over time. JPT proposals are data-dependent—they make use of the observations y and current state (x, z, M) in proposing next state (x', z', M'). While data-dependent proposals introduce additional design complexity, here they avoid random exploration and the use of gating heuristics while retaining tractability. Broadly, JPT proposals reason over assignment and trajectory permutations between existing objects (Switch 4.6.1), between a new object and clutter (Gather 4.6.2, Disperse 4.6.4) or between existing objects and clutter (Extend, 4.6.3). The Switch proposal is a novel generalization of [144] and contributes most to JPT's exploration of posterior modes; thus we elaborate on it more extensively than for the other proposals. Pictorial examples for each proposal transitioning from state (x, z, M) to state (x', z', M') are shown in Figure 4-4.

In the following, Hastings ratios do not require the determinant Jacobian used in RJMCMC inference because batch inference fixes the latent dimension even though the number of objects is unknown. There

are N latent tracks, each of length T, for N total observations over all time and T timesteps. Each track has 2+ associations (object), 1 association (clutter) or 0 associations (empty). All observations must be explained by objects or clutter so the posterior may have empty tracks, but cannot have unexplained observations (e.g., it cannot have all empty tracks but it can have all clutter tracks).

4.6.1 Switch Proposal

Switch proposals consider possible trajectory and associations permutations between existing objects, and are sampled according to JPT's dynamics and observation models (Equations 4.5, 4.6). They cause rapid exploration of different posterior modes when objects are kinematically ambiguous. Strikingly, the Switch proposal is in many cases automatically accepted ($R_{\text{switch}} = 1$), similar to a Gibbs sampler.

Algorithm 4: Switch Proposal

Input : x, z, M, y
Output: x', z', M'

1 Let $x' = x,\ z' = z$
2 Sample object set $\mathcal{K} \subset \{1, \ldots, K(z)\}$ s.t. $|\mathcal{K}| \geq 2$
3 Define switch times $\tau = \{t : z_{tn} = k \text{ for any } k \in \mathcal{K}\}$
4 Set permutations σ_t as the identity permutation on $(1, \ldots, K(z))$ for any $t \notin \tau$
5 **for** $t \in \tau$ *in order* **do**
6 Sample valid permutation $p(\sigma_t \mid \sigma_{1:t-1}) \propto p\left(\sigma_t(x_t) \mid \sigma_{1:t-1}(x_{1:t-1})\right)\ p\left(y_t \mid \sigma_t(x_t)\right)$
7 Let $x'_t = \sigma_t(x_t),\ z'_t = \sigma_t(z_t)$
8 Compute counts M' from z'
9 **if** $M' = M$ *or* $rand(0, 1) < \min(1, R_{switch})$ **then**
10 return x', z', M'
11 **else**
12 return x, z, M

Following Algorithm 4, the Switch proposal samples uniformly at random a subset $\mathcal{K}$ of existing objects $\{1, \ldots, K(z)\}$ such that $2 \leq |\mathcal{K}| \leq \bar{\mathcal{K}}$ (Line 2) for $\bar{\mathcal{K}}$ a maximum size, discussed below.

Let σ_t be a *valid permutation* on objects $\{1, \ldots, K(z)\}$ at time t. Valid permutations do not permute objects outside of $\mathcal{K}$: for all $k \notin \mathcal{K}, \sigma_t(k) = k$. With slight abuse of notation, let $\sigma_t(x_t)$ and $\sigma_t(z_t)$ respectively represent the trajectory values and associations permuted according to σ_t. So for time t, the trajectory value x_{tk} (possibly an uninstantiated value) and association (possibly none) of object k become the trajectory value and association of object $\sigma_t(k)$. Define $\sigma_{1:t}(x_{1:t})$ over times $1, \ldots, t$ as $x'_{1:t}$ where $x'_t = \sigma_t(x_t)$.

The Switch proposal only considers permutations at times when at least one object $k \in \mathcal{K}$ has been observed. Let τ be all such times

(Line 3). For any time $t \notin \tau$, set σ_t as the identity permutation, $\sigma_t(k) = k$ (Line 4).

For increasing time $t \in \tau$, iteratively sample permutation σ_t conditioned on the previously-sampled permutations $\sigma_{1:t-1}$ with probability proportional to the product of the observation and dynamics models (Equations 4.5, 4.6) evaluated with the appropriate swaps in trajectory and association values imposed by permutations $\sigma_{1:t}$ (Line 6). There are $|\mathcal{K}|!$ possible values for σ_t at each time t, but we find $\bar{\mathcal{K}} = 7$ balances efficient computation and posterior exploration.

After sampling σ_t for all $t \in \tau$, we compute new counts M' from the permuted associations z' (Line 8) and the Hastings ratio (Line 10) between (x', z', M') and (x, z, M), noting that Switch proposals are their own reverse move.

$$R_{\text{switch}} = \frac{p(x', z', M' \mid y)}{p(x, z, M \mid y)} \times \frac{q_{\text{switch}}(x, z, M \mid x', z', M', y)}{q_{\text{switch}}(x', z', M' \mid x, z, M, y)} \tag{4.13}$$

$$= \frac{p(z' \mid M') \prod_{t=1}^{T} \frac{1}{Z} p(M'_t \mid M'_{t-1}) p(x'_t \mid x'_{1:t-1}) p(y_t \mid x'_t, z'_t)}{p(z \mid M) \prod_{t=1}^{T} \frac{1}{Z} p(M_t \mid M_{t-1}) p(x_t \mid x_{1:t-1}) p(y_t \mid x_t, z_t)} \times \tag{4.14}$$
$$\frac{\prod_{t=1}^{T} \frac{1}{Z_t} p(\sigma_t^{-1}(x'_t) \mid \sigma_{1:t-1}^{-1}(x'_{1:t-1}) p(y_t \mid \sigma_t^{-1}(x_t), \sigma_t^{-1}(z_t))}{\prod_{t=1}^{T} \frac{1}{Z_t} p(\sigma_t(x_t) \mid \sigma_{1:t-1}(x_{1:t-1}) p(y_t \mid \sigma_t(x_t), \sigma_t(z_t))}$$

$$= \frac{p(z' \mid M') \prod_{t=1}^{T} p(M'_t \mid M'_{t-1}) p(x'_t \mid x'_{1:t-1}) p(y_t \mid x'_t, z'_t)}{p(z \mid M) \prod_{t=1}^{T} p(M_t \mid M_{t-1}) p(x_t \mid x_{1:t-1}) p(y_t \mid x_t, z_t)} \times \tag{4.15}$$
$$\frac{\prod_{t=1}^{T} p(x_t \mid x_{1:t-1}) p(y_t \mid x_t, z_t)}{\prod_{t=1}^{T} p(x'_t \mid x'_{1:t-1}) p(y_t \mid x'_t, z'_t)}$$

$$= \frac{\prod_{t=1}^{T} p(M'_t \mid M'_{t-1})}{\prod_{t=1}^{T} p(M_t \mid M_{t-1})} \tag{4.16}$$

where Equation 4.14 substitutes in the values for each term in the ratio, defining σ_t^{-1} as the inverse permutation of σ_t and Z_t as the normalizer for the sampled σ_t at time t (equal to 1 if $t \notin \tau$). Equation 4.15 substitutes $\sigma_t(x_t)$ for x'_t and $\sigma_t^{-1}(x'_t)$ for x_t (similarly for $\sigma_t(z_t)$). It also cancels common normalizers Z for the joint ratio and Z_t at each time t for the proposal ratio. Equation 4.16 cancels all terms related to the dynamics and observation models, and also cancels $p(z' \mid M')$ with $p(z \mid M)$ under the assumption that no object $k \in \mathcal{K}$ was rendered invalid by having fewer than two observations. That can easily be detected and automatically rejected or entirely avoided by defining valid permutations to require the first two observed times for any $k \in \mathcal{K}$ to not be permutable.

The Switch proposal is always accepted ($R_{\text{switch}} = 1$) whenever $M' = M$. This occurs in several situations: when the number of objects are known in advance, when objects are assumed never to depart, when there are no missing observations and when all $k \in \mathcal{K}$ are observed at max τ. In many scientific and sports analytics applications, it is common for subjects to never depart. When these conditions don't hold, the event counts and Hastings ratio are efficiently evaluated (lin-

ear in time T and parallelizable) by only considering terms where the counts M', M differ. The Switch proposal scales linearly in time T and factorially in $|\mathcal{K}|$. In practice, we limit the subset size $|\mathcal{K}| \leq \bar{\mathcal{K}}$.

In Appendix A.4, we show that Switch proposals generalize the Extended HMM proposals of [144] by proposing a discretization that depends on the current latent state (in their nomenclature, JPT "pool states" are permutations of x, z). In their work, sampled discretizations (or pool states) cannot depend on the current latent state, else detailed balance is lost. In contrast, Switch proposals depend on the current latent state and maintain detailed balance.

4.6.2 Gather Proposal

Following Algorithm 5, the Gather proposal considers the formation of a new object $k = 1 + K(z)$ (Line 2) from the set of clutter-associated observations $\{y_{tn} : z_{tn} = 0\}$. Its reverse move is Disperse (Chapter 4.6.4). Let τ_0 be the set of times t with at least one clutter association (Line 3). For increasing $t \in \tau_0$, assignments to object k are iteratively sampled either among observations that are currently associated to clutter or, with probability $\delta = 0.01$, no clutter association to allow for missing observations. In the former, an association $z'_{tn} = k$ is first sampled among all clutter observations with probability proportional to Line 6 where $t' = t - 1$ and marginalization occurs between states with missing associations.

Algorithm 5: Gather Proposal

Input : x, z, M, y
Output: x', z', M'
1 Let $x' = x,\ z' = z$
2 Let $k = 1 + K(z)$
3 Define gather times $\tau_0 = \{t : z_{tn} = 0 \text{ for any } 1 \leq t \leq T\}$
4 **for** $t = \min \tau_0, \ldots, \max \tau_0$ **do**
5 **if** $rand(0,1) < \delta$ **then** continue
6 Sample $p(z'_{tn} = k) \propto p(y_{tn} | x_{t'k}, z'_{tn} = k)\, \mathbb{I}(z_{tn} = 0)$
7 Sample $p(x'_{tk} \mid x'_{t'k}, z'_{tn} = k) \propto p(x'_{tk} \mid x_{t'k})\, p(y_{tn} \mid x'_t, z'_{tn} = k)$
8 Compute counts M' from z'
9 **if** $rand(0,1) < \min(1, R_{gather})$ **then** return x', z', M'
10 **else** return x, z, M

The Gather proposal has ratio,

$$R_{\text{gather}} = \frac{p(x', z', M' \mid y)}{p(x, z, M \mid y)} \times \frac{q_{\text{disperse}}(x, z, M \mid x', z', M', y)}{q_{\text{gather}}(x', z', M' \mid x, z, M, y)} \tag{4.17}$$

$$= \frac{p(z' \mid M')\ \prod_{t=1}^{T} p(M'_t \mid M'_{t-1})\, p(x'_t \mid x'_{1:t-1})\, p(y_t \mid x'_t, z'_t)}{p(z \mid M)\ \prod_{t=1}^{T} p(M_t \mid M_{t-1})\, p(x_t \mid x_{1:t-1})\, p(y_t \mid x_t, z_t)} \times \tag{4.18}$$

$$\frac{(K(z)+1)^{-1}}{\prod_{t \in \tau_0} \omega_t} \tag{4.19}$$

where $\omega_t = \delta$ if $z'_{tn} \neq k$ for any n else,

$$\omega_t = \frac{1}{Z_t} p(y_{tn} \mid x_{t'k}, z'_{tn} = k)\, p(x'_{tk} \mid x'_{t'k}, z'_{tn} = k)(1 - \delta) \qquad (4.20)$$

per Gather algorithm Lines 5–7. All dynamics and observation model terms cancel in the posterior ratio for objects other than k, but terms remain for observations that were previously clutter and are now associated to object k and counts $M' \neq M$.

4.6.3 Extend Proposal

The Extend proposal is similar to the Gather proposal but rather than consider permutations between clutter associations and a new object, it considers permutations between clutter associations and an existing object. This allows an existing object to resample associations.

Algorithm 6: Extend Proposal

Input : x, z, M, y
Output: x', z', M'

1 Let $x' = x,\ z' = z$
2 Sample $k \in \{1, \ldots, K(z)\}$
3 Define extend times $\tau_k = \{t : z_{tn} \in \{0, k\}$ for any $1 \leq t \leq T\}$
4 **for** $t = \min \tau_k, \ldots, \max \tau_k$ **do**
5 **if** $rand(0, 1) < \delta$ **then** continue
6 Sample $p(z'_{tn} = k) \propto p(y_{tn} | x_{t'k}, z'_{tn} = k)\, \mathbb{I}(z_{tn} \in \{0, k\})$
7 Sample
  $p(x'_{tk} \mid x'_{t'k}, z'_{tn} = k) \propto p(x'_{tk} \mid x_{t'k})\, p(y_{tn} \mid x'_t, z'_{tn} = k)$
8 Compute counts M' from z'
9 **if** $rand(0, 1) < \min(1, R_{extend})$ **then** return x', z', M'
10 **else** return x, z, M

Following Algorithm 6, randomly sample object k from existing objects (Line 2) and iterate over all times $t \in \tau_k$ with an association to clutter $z_{tn} = 0$ or to the current object $z_{tn} = k$ (Line 3). As in the Gather proposal, skip a resampling of assignments at time $t \in \tau_k$ with probability δ. Otherwise, sample an association then a trajectory value (Lines 6-7) with definitions as in Gather, except that it is possible for $z'_{tn} = z_{tn}$ for some times t (resample the same assignment it had).

By automatically rejecting any Extend proposal that leaves a object with fewer than two observations, we can ensure that Extend proposals are always their own reverse move. As in Gather, the observation and dynamics terms cancel in the posterior ratio for all objects other than k, but an accept/reject step must still be computed, and is of similar form to the Gather proposal. This proposal has linear complexity in time T and the number of observations N_t at each time.

4.6.4 Disperse Proposal

Following Algorithm 7, the Disperse proposal simply chooses an existing object at random (Line 2), removes all its associations by setting them to clutter (Line 3) and deletes the trajectory values for that object (Line 4). It is the reverse move for the Gather proposal. Hence,

$$R_{\text{disperse}} = R_{\text{gather}}^{-1} \tag{4.21}$$

where R_{gather} is defined in Equation 4.18. As in the Gather proposal, an accept/reject step is required. Disperse has constant complexity.

Algorithm 7: Disperse Proposal

Input : x, z, M, y
Output: x', z', M'
1 Let $z' = z$
2 Sample $k \in \{1, \ldots, K(z)\}$
3 Set $z'_{tn} = 0$ for all t, n such that $z_{tn} = k$
4 Let $x' = x \setminus \{x_{tk}\}_{t=1}^{T}$
5 Compute counts M' from z'
6 **if** $rand(0, 1) < \min(1, R_{disperse})$ **then** return x', z', M'
7 **else** return x, z, M

4.6.5 Joint Trajectory Sampling with Missing Data

Joint sampling of trajectories from the full conditional,

$$p(x \mid z, M, y) = p(x \mid z, y) \tag{4.22}$$

constitutes a fifth MH proposal in the form of a Gibbs sampler where M is dropped due to independence. Jointly sampling trajectories $x \mid z, y$ differs from typical filter- and smoothing-based approaches, which only provide marginal distributions at each time. If there are no missing observations, then the full conditional can be sampled as,

$$p(x \mid z, y) = \prod_{k=1}^{K} \prod_{t=1}^{T} \frac{p(x_{tk} \mid y_{1:t}^{k})\, p(x_{(t+1)k} \mid x_{tk})}{p(x_{(t+1)k} \mid y_{1:t}^{k})} \tag{4.23}$$

where $p(x_{tk} \mid y_{1:t}^{k})$ is the filter distribution of x_{tk} and $y_{1:t}^{k} = \{y_{t'n} : z_{t'n} = k \text{ and } t' \leq t\}$. Sampling from this posterior is similar to smoothing [170], except that the backwards pass draws joint samples. Inference can be done in parallel over objects and, in the linear Gaussian case, is in closed form with complexity linear in T. For other (possibly non-linear) dynamics or observation models, any procedure that leaves the joint distribution invariant may be used.

In the case of missing observations, we marginalize over the intervening latent states, realizing samples only at times where an object has an association. Thus, the distribution for x_{tk} under the joint (the numerator of Equation 4.23), assuming that the most recent previous

association occurred at time $\overleftarrow{t} \leq t$ and most recent future association at time $\vec{t} > t$, is:

$$p(x_{tk} \mid y^k_{1:\overleftarrow{t}})\, p(x_{\vec{t}k} \mid x_{tk}) = \int p(x_{(\overleftarrow{t}+1:t)k} \mid y^k_{1:\overleftarrow{t}}) dx_{(\overleftarrow{t}+1:t-1)k} \int p(x_{(t+1:\vec{t})k} \mid x_{tk}) dx_{(t+1:\vec{t}-1k)} \tag{4.24}$$

We emphasize that the first term in Equation 4.24 integrates over past missing states. If there is an association at time t (i.e., $\overleftarrow{t} = t$), then there is no integration to carry out in the first term and so it simplifies to $p(x_{tk} \mid y^k_{1:t})$. Similarly, the second term in Equation 4.24 integrates over future missing states. If $\vec{t} = t + 1$ then there is no integration to carry out in the second term and so it simplifies to $p(x_{(t+1)k} \mid x_{tk})$. Hence, when $\vec{t} = t + 1$ and $\overleftarrow{t} = t$, we recover the numerator of Equation 4.23.

4.7 Uncertainty Reduction

Tracking is increasingly used in scientific applications where manual observation does not scale and cannot be crowdsourced due to privacy or expertise [7]. Datasets can be over 500 hours long [130]. Tracks are used as input to hypothesis tests about differences in behavior with respect to genetics or neural activity [233]. Most trackers provide point estimates and incur many identity switch errors on benchmark datasets like the MOT Challenge [136]. Errors propagate to hypothesis tests and corrupt conclusions. We do not expect the Joint Posterior Tracker to be error-free; instead, we have designed it to accurately quantify posterior uncertainty, which enables conclusions to be correctly weighted and can be used for additional tasks.

We extend the Joint Posterior Tracker so that it can automatically locate ambiguities in the data by finding posterior samples with conflicting interpretations. Ambiguities can be sequentially resolved by asking a series of questions to an oracle. Arguably the most common ambiguities in multi-object tracking occur from identity switches, where two or more targets become confused. If identity switches can be resolved by a series of automatically-scheduled questions then we expect that posterior uncertainty would diminish with each question while estimates of track quality would improve when compared to a groundtruth. To build automatic track refinement into the Joint Posterior Tracker, we define an annotation model and use sequential Bayesian Experiment Design to reason over which questions to ask. Questions are posed as possible experiments (also called designs) to perform, with the best experiment being identified by the optimization of a utility function. We optimize mutual information (MI), which quantifies the expected reduction in posterior uncertainty that results from the answer of a single question.

Let $\kappa = (t, n)$ be the time and observation indices that uniquely identifies observation y_{tn}. A design then corresponds to a tuple of these index pairs $d = (\kappa_1, \kappa_2)$. We abuse notation and let $\kappa_1(d) = (t_1, n_1)$ indicate the first pair in design d such that $y_{\kappa_1(d)} = y_{t_1 n_1}$ and $z_{\kappa_1(d)} = z_{t_1 n_1}$ and likewise for $\kappa_2(d)$. The annotation indicates if two observations $y_{t_1 n_1}, y_{t_2 n_2}$ belong to the same or different objects – $a_l(y_{t_1 n_1}, y_{t_2 n_2}) =$ 1 or 0 respectively – and is correct with probability $p_a = 0.99$. This event corresponds to whether the assignments $z_{t_1 n_1}, z_{t_2 n_2}$ share the same non-zero value – recall $z_{tn} = 0$ indicates clutter. After accounting for the annotation noise and design, we have the following annotation likelihood

$$p_d(a_l = 1 \mid x, y, z, M) = p_d(a_l = 1 \mid z_{\kappa_1(d)}, z_{\kappa_2(d)}) = \tag{4.25}$$
$$\begin{cases} 0.99 & \text{if } z_{\kappa_1(d)} = z_{\kappa_2(d)} \text{ and } z_{\kappa_1(d)} > 0 \\ 0.01 & \text{o.w.} \end{cases}$$

When conditioned on just the two assignments $z_{\kappa_1(d)}, z_{\kappa_2(d)}$, the annotation is independent of the remaining variables in the model; this yields the first equality in Equation 4.25. This conditional is added to the joint distribution $p(x, y, z, M)$, yielding an augmented generative model that now includes annotations.

Mutual information between the annotation a_l and the latent trajectories x conditioned on the observations y and past annotations $D = \{a_{1:l-1}, d_{1:l-1}\}$ is given by,

$$I_d(a_l; x \mid y, D) = \mathbb{E}\left[\log \frac{p_d(a_l, x \mid y, D)}{p_d(a_l \mid y, D) p_d(x \mid y, D)}\right] \tag{4.26}$$
$$= \mathbb{E}\left[\log \frac{p_d(a_l \mid x, y, D)}{p_d(a_l \mid y, D)}\right] \tag{4.27}$$

We use a greedy approach to Sequential BED, wherein we select the highest MI design within each round of BED. While myopic, this approach avoids the complexity associated with searching for an optimal policy. Thus, at the l^{th}–round of sequential BED, we seek

$$d_l = \arg\max_d I_d(a_l; x \mid y, D). \tag{4.28}$$

We typically cannot evaluate MI in closed form and instead resort to Monte Carlo estimation using M samples drawn from the posterior $\{a_l^m, x^m, z^m\}_{m=1}^M \sim p(a_l, x, z \mid y, D)$:

$$\tilde{I}_d = \frac{1}{M} \sum_{m=1}^{M} \log \frac{p_d(a_l^m \mid x^m, y, D)}{p_d(a_l^m \mid y, D)} \tag{4.29}$$

To evaluate the likelihoods in Equation 4.29, first we expand them as,

$$p_d(a_l^m \mid x^m, y, D) = \sum_z p_d(a_l^m \mid z, x^m, y, D)\, p(z \mid x^m, y, D) \tag{4.30}$$

$$= \sum_{z_{\kappa_1(d)}, z_{\kappa_2(d)}} p_d(a_l^m \mid z_{\kappa_1(d)}, z_{\kappa_2(d)}) \tag{4.31}$$

$$p(z_{\kappa_1(d)}, z_{\kappa_2(d)} \mid x^m, y, D)$$

$$p_d(a_l^m \mid y, D) = \sum_z p_d(a_l^m \mid z, y, D)\, p(z \mid y, D) \tag{4.32}$$

$$= \sum_{z_{\kappa_1(d)}, z_{\kappa_2(d)}} p_d(a_l^m \mid z_{\kappa_1(d)}, z_{\kappa_2(d)}) \tag{4.33}$$

$$p(z_{\kappa_1(d)}, z_{\kappa_2(d)} \mid y, D)$$

Equation 4.31 can be evaluated exactly because we can obtain,

$$p(z_{\kappa_1(d)}, z_{\kappa_2(d)} \mid x^m, y, D) \tag{4.34}$$

through enumeration of all pairwise assignments conditioned on the sampled trajectories x^m and observations y. Equation 4.33, on the other hand, requires $p_d(a_l^m \mid y, D)$ which is intractable, so we again use Monte Carlo estimation,

$$\hat{p}_d(a_l^m \mid y, D) = \frac{1}{M} \sum_{m'=1}^{M} p_d(a_l^m \mid z_{\kappa_1(d)}^{m'}, z_{\kappa_2(d)}^{m'}). \tag{4.35}$$

Our MI estimator is then,

$$\hat{I}_d = \frac{1}{M} \sum_{m=1}^{M} \log \frac{p_d(a_l^m \mid x^m, y, D)}{\hat{p}_d(a_l^m \mid y, D)} \tag{4.36}$$

where $p_d(a_l^m \mid x^m, y, D)$ is given in Equation 4.31 and $\hat{p}_d(a_l^m \mid y, D)$ is given in Equation 4.35.

4.8 JPT Compared to MCMCDA

Both batch MCMCDA and JPT are Bayesian treatments of the general multi-object tracking problem, but exhibit fundamental differences. First, batch MCMCDA defines a *marginal* posterior over associations. Each association event in MCMCDA induces a *marginal* distribution over trajectories at each time. In contrast, JPT defines a *joint* posterior over associations and trajectories over all times.

Second, both batch MCMCDA and JPT use exact MCMC inference, but MCMCDA samples uniformly at random from a graph that connects observations based on spatiotemporal gating heuristics defined in Equation 27 (page 492) and visualized in Figure 6 (page 490) of [153]. These heuristics, d and $\bar{v}$ are limits on the maximum number of times an object can go unobserved and the maximum distance an object can

travel between times. If set too low, these heuristics will cause MCMCDA to create many new objects for what is actually a single object. If set too high, MCMCDA inference will devolve into a random search that fails to converge (observed based on the grid search conducted in Chapter 4.9.3). Besides requiring tuning for good performance, MCMCDA heuristics limit its ability to represent the configuration space of object associations.

In contrast, JPT inference uses no heuristics and can represent the entire configuration space; it is efficient because it reasons locally over permutations in object associations using its forward model. Specifically, the Gather, Extend, and Disperse proposals do not depend on the number of objects. Gather and Extend have linear complexity in time and observation count whereas Disperse has constant complexity. The Switch proposal depends factorially on the number of objects present, but any single Switch move is kept manageable by only considering subsets of objects. Repeated Switch moves may cover all objects without incurring the cost of doing so all at once. Trajectory sampling has linear complexity in time and the number of objects. Chapter 4.9.4 directly shows evidence of JPT's efficient traversal of object-association space; in particular, it uncovers many more posterior modes than MCMCDA for the same number of posterior samples.

Finally, both batch MCMCDA and JPT can represent posterior uncertainty. But as we show in Chapter 4.9.4, JPT quantifies uncertainty much more accurately than MCMCDA.

4.9 Evaluation

Our experiments are designed to highlight the importance of accurate uncertainty quantification and the tracking performance of JPT when compared to existing methods. We begin by describing the datasets used in Chapter 4.9.1, including a novel primate dataset. We then explain the dynamics and observation models common to each method in Chapter 4.9.2. Our efforts to make the comparison as favorable to MCMCDA as possible are discussed in 4.9.3.

Following, we compare JPT to batch MCMCDA in terms of mode exploration in Chapter 4.9.4. In Chapter 4.9.5 we show that JPT outperforms both MCMCDA and an optimization-based tracker (MHT) [109] on large science and sports datasets.[28] Lastly, we use the more accurate uncertainty quantification of JPT to generate targeted queries to a noisy oracle (e.g., a human annotator) yielding significant improvement to trajectory quality (4.9.6). Throughout, JPT and MCMCDA are run for 5 replicates (Markov chains), each time drawing 2000 samples and discarding the first half as burn-in. For fair comparison, no method uses appearance information.

[28] MHT was chosen for comparison because it is a classical approach and also because it has known performance on the MOT Challenge when paired with deep appearance features. MHT's performance in Chapter 4.9.5 grounds the performance of other baselines on our datasets.

4.9.1 Datasets

K33 is a synthetic dataset containing ambiguous object crossing events. There are no clutter detections and all objects are detected at all times. The true outcome is randomly sampled from one of the 24 modes.

Marmoset contains two primates interacting in a laboratory environment over long periods of time where there are many total and partial occlusion events, as well as occasional clutter detections. Noisy observations are generated as the centroid of the detections from a trained Mask-RCNN neural network [90] and groundtruth accomplished by human annotation that correctly maintains object identities throughout the sequence. As a result, trackers must correctly re-identify objects that have been occluded to avoid getting penalized.

Soccer observations are the unassociated centers of players and referee. Groundtruth does not maintain track identities when players go out of frame. Consequently, re-identification of objects after a total occlusion (e.g., moving out of frame) are not rewarded.[29] It is evaluated in chunks of 20 frames according to the protocol in [199].

[29] This accords with MOT Challenge datasets whose groundtruths also do not re-identify objects that go out of frame and return.

4.9.2 Dynamics and Observation Models

All experiments use a random acceleration model with a 2D observation space. Hence, the latent space is 4-dimensional: 2 for position, and 2 for velocity. The dynamics and observation noise covariances are set according to the expected value of data-dependent Inverse-Wishart priors based on the per-time variances of the observations and the between-time, nearest-neighbor distances of observations.

Appearance modeling could be added to JPT without modifying any equation by augmenting the observation and latent spaces with appearance information, such as from deep neural network features or histograms counts of pixel colors. Uncertainty quantification is the focus of this work; hence, appearance was not used for any of the compared methods.

4.9.3 MCMCDA Gating Heuristic Grid Search

The MCMCDA baseline contains two gating heuristics: thresholds on $\bar{v}$, the maximum L2 spatial and d, the maximum L1 temporal distances between two observations belonging to the same track. Although they can be removed by setting the thresholds very high, this causes the inference procedures of MCMCDA to devolve into random exploration, severely damaging performance according to CLEAR MOT metrics. To make the comparison with JPT as competitive as possible, we performed a grid search over each gating threshold, as well as providing it with knowledge of the true number of objects or not, and restricted JPT/MCMCDA comparisons so that MCMCDA only used the parameters with best performance as measured by the CLEAR MOT multi-object tracking accuracy (MOTA) metric.

For MCMCDA on the K33 dataset, the best-performing spatial gating threshold was $\bar{v} = 6$ and the best temporal gating threshold was $d = 1$. For Marmoset and Soccer, the best-performing spatial gating threshold was $\bar{v} = 20$ and best-performing temporal gating threshold was $d = 6$. In all cases, MCMCDA performed best without knowledge of the true number of objects because this knowledge could limit its ability to explore by creating excess objects that it later destroyed. In some cases, it would also cause MCMCDA to be severely penalized by occlusion events that persisted for longer than its temporal gating threshold as it could either represent the object before or after the occlusion, but not both.

Increasing the gating thresholds to very large numbers caused random exploration of low-probability events in the MCMCDA posterior due to the way its inference is constructed. Specifically, MCMCDA precomputes a sparse graph of paths between observations that respect its gating thresholds. It then computes Metropolis-Hastings proposals that randomly sample from this graph on the assumption that the thresholds were set to encourage likely associations. Thus, MCMCDA inference has a fundamental limitation: either gating thresholds are set tight and some true association hypotheses are excluded, or they are set loose and random exploration occurs.

4.9.4 Representation of Posterior Uncertainty

Consider the K33 dataset in Figure 4-5 (Top). Three objects begin separated but become ambiguous after each of three confusion events (yellow shading). Observe that for k ambiguously proximate objects, there will be $k!$ possible outcomes. Figure 4-5 (Bottom) shows the $24 = 2!\ 2!\ 3!$ modes that a tracker would ideally explore in this dataset.

We investigate whether JPT and MCMCDA effectively explore the 24 modes of the K33 dataset by observing posterior trajectory variance for each Markov chain in Figure 4-1. JPT captures the uncertainty from each ambiguous region; for example, red and green cross in some posterior samples but not in others. In contrast, MCMCDA is *overconfident*: it fails to represent any uncertainty in either of the first two ambiguous regions, and is only partially successful in the final region. Not only is MCMCDA overconfident, it is wrong as will be shown by tracking metrics in (4.9.5).

We quantify mode exploration by matching each posterior sample from JPT and MCMCDA to the nearest of the 24 likely outcomes. Details of this matching procedure are below. Figure 4-6 shows histograms of modes matched by JPT and MCMCDA in different Markov chains. JPT represents every outcome within each Markov chain while MCMCDA captures at most 2 but usually 1 outcome in a single Markov chain. Noting that the ideal distribution over the 24 matched modes would be uniform, we compare total variation (L1) distance between that and the empirical distributions of matched modes for JPT and MCMCDA (Figure 4-7), plotted as a function of sample count. While

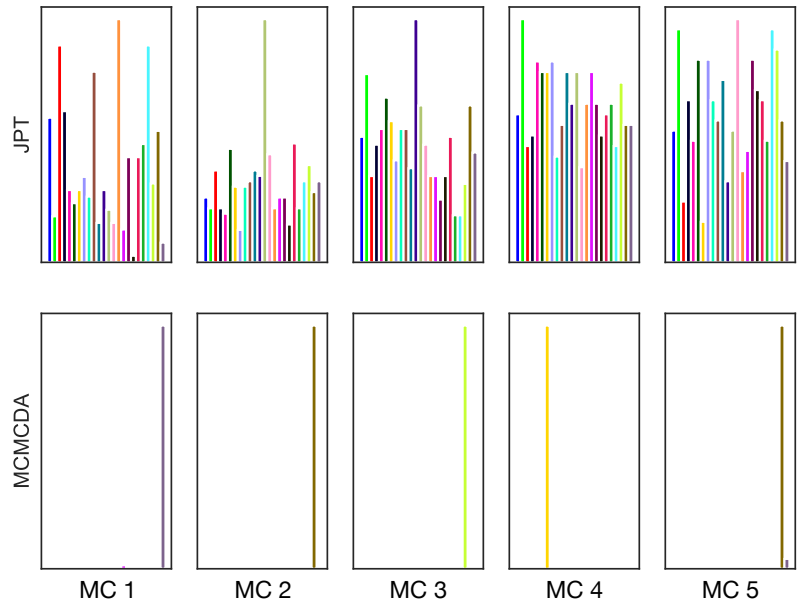


Figure 4-6: Histograms of the modes explored by JPT (**Top**) and MCMCDA (**Bottom**) in 5 Markov chains. JPT explores all modes in each chain while MCMCDA gets stuck in 1–2.

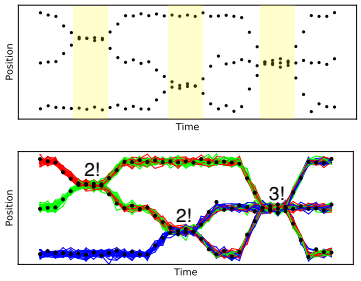


Figure 4-5: The K33 dataset. Observations over time (**Top**) with yellow shading for ambiguous regions and a joint trajectory sample from the $24 = 2!\ 2!\ 3!$ posterior modes (**Bottom**), each reflecting a possible outcome.

the total variation resulting from JPT's is nonzero, it is significantly lower than that of MCMCDA. Examination of Figure 4-1 (Top) reveals that JPT exhibits a moderate bias between upper and lower paths after the first ambiguous region. Nevertheless, JPT represents all modes whereas MCMCDA rarely represents more than one.

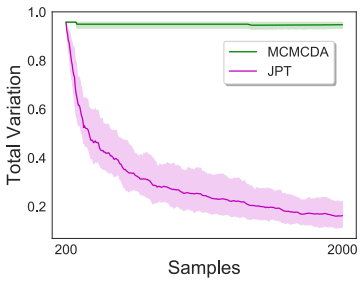


Figure 4-7: Total variation distance between the true distribution of modes on K33, and the histograms of matched modes for JPT and MCMCDA samples.

Computing Distances Between Association Hypotheses To match an inferred set of trajectories to another set of trajectories, we begin with the Spatiotemporal Linear Combine (STLC) Distance of [184], which compared favorably in [190]. Briefly, STLC evaluates trajectories on both their L2 spatial and L1 temporal alignment; it supports uneven sampling rates and arbitrary trajectory start/end times. It is a similarity measure that ranges from $[0, 2]$, but we convert it to a cost by inverting the limits.

Given STLC as an object-to-object cost, we define a distance between multi-object tracking association hypotheses by using discrete optimal transport [188], where the cost matrix is filled with the STLC costs of each object pair between the two samples. Note that this supports arbitrary numbers of objects in each sample. This distance was then used in determining mode representation in posterior samples of JPT and MCMCDA in the Uncertainty Quantification experiments and again in the Uncertainty Reduction experiments, where we demonstrated that planned annotations rapidly reduce the distance of JPT samples to the groundtruth.

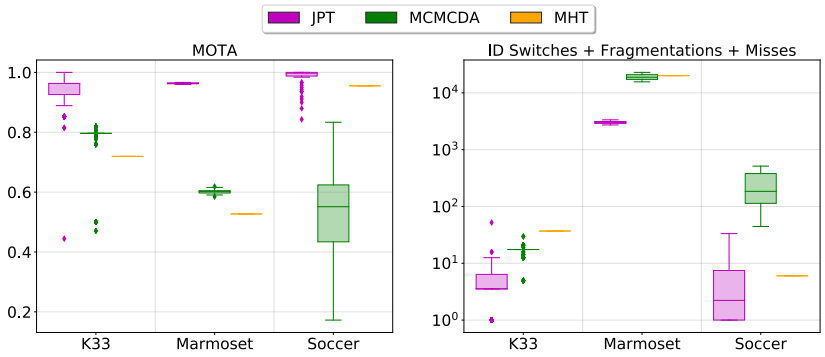


Figure 4-8: CLEAR MOT metrics for JPT, MCMCDA and MHT on datasets K33, Marmoset and Soccer. (**Left**), higher is better; (**Right**), lower is better.

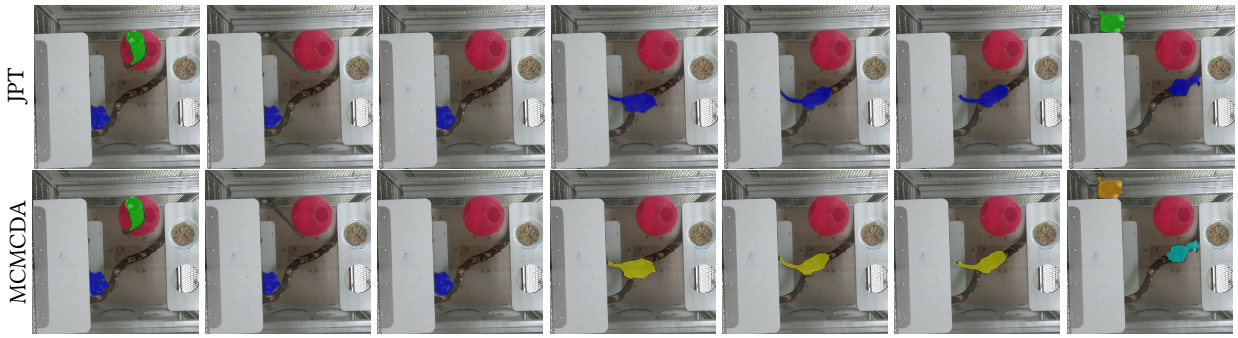


Figure 4-9: Example tracking on Marmoset for JPT (**Top**) and MCMCDA (**Bottom**). The upper primate (green, first column) goes under the shelf (second column) and is correctly re-identified when it emerges (last column) by JPT but not MCMCDA. The lower primate (blue, first column) rapidly traverses the branch, with stable associations for JPT but not MCMCDA.

4.9.5 Performance on Real and Synthetic Data

We compare tracking performance of JPT, MCMCDA and an optimization-based tracker [109] (MHT) on three datasets: the K33 dataset (39 timesteps, 3 objects, 117 observations), a scientific dataset Marmoset of primate interactions (15k timesteps, 2 objects, 25k observations), and the sports dataset Soccer (1.5k timesteps, 22 objects, 12k observations). Marmoset contains many long-term occlusions from primates going under shelves or into their nest while Soccer contains many player-player occlusions. More details of each dataset are in Chapter 4.9.1.

Figure 4-8 shows performance as evaluated by standard CLEAR MOT [23] metrics that account for identity switches (objects are confused), fragmentations (two inferred objects explain one actual object) and misses (failure to correctly assign an observation to some object). Multi-object tracking accuracy (MOTA) is a summary statistic commonly used to compare trackers,

$$\text{MOTA} = 1 - \frac{\sum_{t=1}^{T} \text{FN}_t + \text{FP}_t + \text{ID}_t}{\sum_{t=1}^{T} \text{GT}_t} \tag{4.37}$$

where FN, FP, ID, and GT are misses, false positives, identity switches, and true positives, respectively. Boxplots are provided for JPT and MCM-

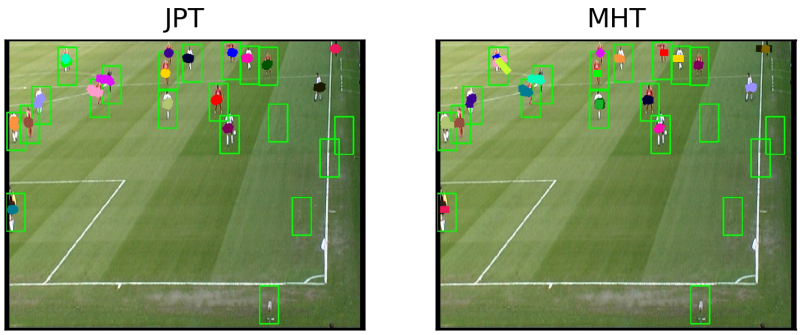


Figure 4-10: Example tracks for 20 frames on Soccer for JPT (**Left**) and MHT (**Right**). Both track players reasonably well but MHT has more fragmentations (multiple tracks corresponding to a single object, e.g., the top-left player).

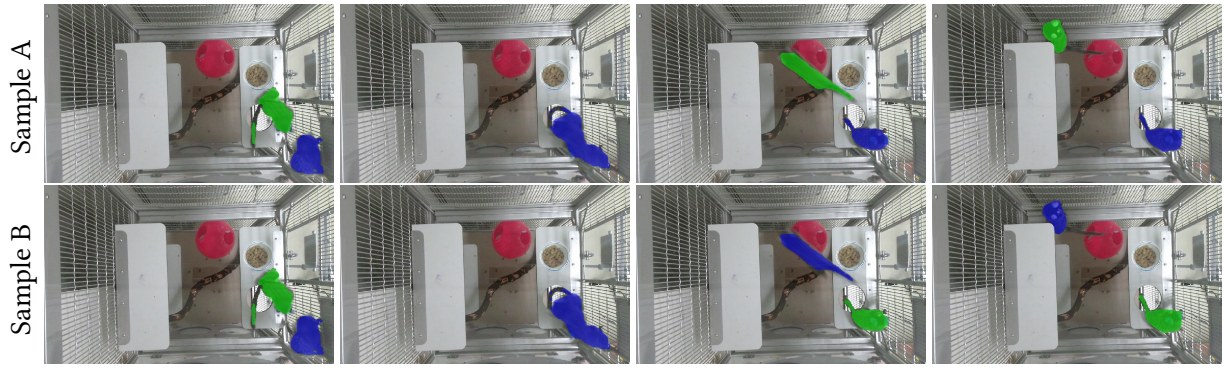


Figure 4-11: Two JPT samples (A, B) showing ambiguity captured due to missed detections (column 2) when objects are close.

CDA posterior samples but, being deterministic, MHT only provides a point estimate. JPT outperforms MCMCDA and MHT on all datasets and metrics with notably fewer identity switches, fragmentations and misses. Given the MCMCDA gating heuristic, we report its *best-scoring* MOTA from a grid search over parameter values. Details of the grid search are in Chapter 4.9.3. Figure 4-9 demonstrates that JPT can re-identify objects in Marmoset after long-term occlusions because it is not limited by gating heuristics as MCMCDA is. Figure 4-10 demonstrates that JPT maintains more stable tracks than does MHT on Soccer where there are dense numbers of objects.

4.9.6 Automatic Reduction of Posterior Uncertainty

We demonstrate JPT in an active-annotation application. The goal is to curate initial trajectories with a small number of targeted annotations to achieve high-quality trajectories, e.g., as might be required for long-term motion analysis. Targeted annotations will eliminate modes, but *only* if they are identified by the inference procedure to begin with. As we note in the first experiment, MCMCDA generally finds a single mode while ascribing near certainty to that mode, i.e., there is no mechanism for identifying and removing potential ambiguities. Supe-

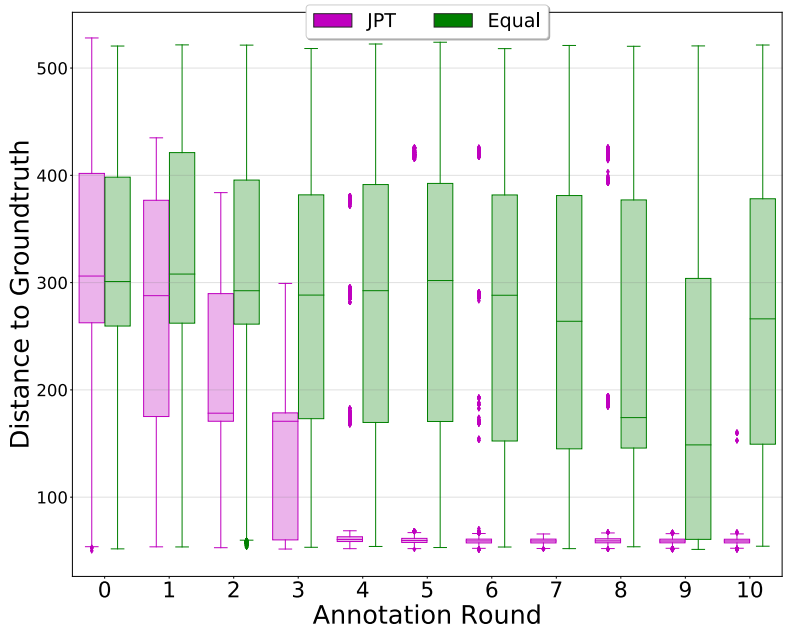


Figure 4-12: Reduction in JPT posterior uncertainty from scheduled annotations. JPT (magenta) asks informative questions that rapidly remove ambiguities in tracking estimates. In contrast, annotations scheduled with a poor model of uncertainty (green) fail to reduce posterior uncertainty.

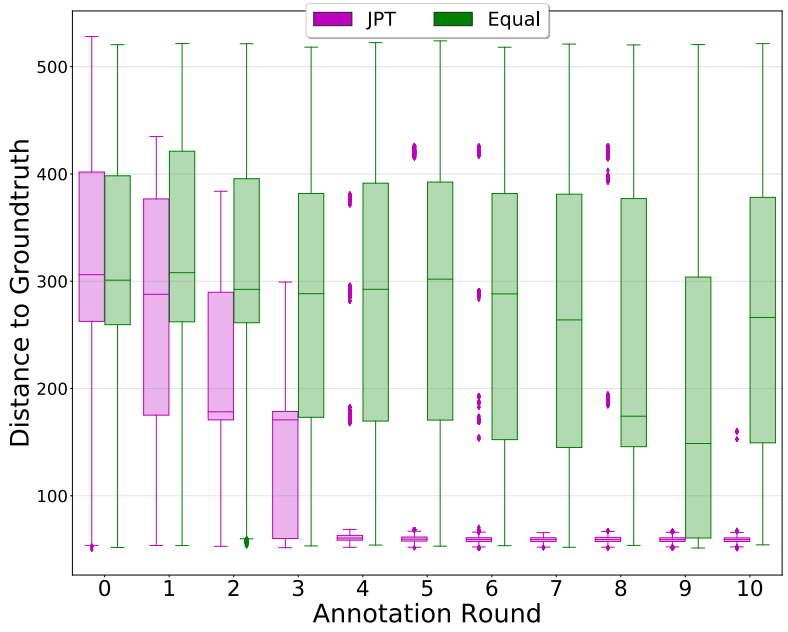


Figure 4-13: Improvement in JPT trajectory estimates from scheduled annotations. JPT (magenta) asks informative questions that rapidly improve posterior trajectory estimates. In contrast, annotations scheduled with a poor representation of uncertainty (green) fail to improve trajectory quality.

rior uncertainty modeling via JPT allows *informative* annotations to be readily identified yielding quality trajectories with very few annotations.

Let there be L annotations $a = \{a_l\}_{l=1}^{L}$ where each indicates whether two observations $y_{t_1 n_1}, y_{t_2 n_2}$ belong to the same or different objects: $a_l(y_{t_1 n_1}, y_{t_2 n_2}) = 1$ or 0. Assume each annotation is correct with probability $p_a = 0.99$. Intuitively, *informative* annotations are related to observations that are *outside* an ambiguous region as these lead to mode elimination. We use Sequential Bayesian Experimental Design (SBED) for automated selection of informative observation pairs over L annotation rounds where utility is quantified by the mutual information (MI) [24] between a prospective annotation and the latent trajectories. Further details are provided in Chapter 4.7.

We perform 5 replicated experiments on the K33 dataset, each with 10 rounds of annotation. The first round starts with no annotations; successive rounds add an annotation. We compare using JPT's uncertainty representation (JPT, magenta) to a baseline (Equal, green) where all annotations are equally informative (i.e., planning with a poor model of uncertainty or none at all). Figure 4-12 compares reduction in posterior trajectory uncertainty between the methods, computed as trajectory variance summed over all times and samples. JPT yields informative annotations that rapidly reduce uncertainty until it reaches a floor by Round 5. Similarly, Figure 4-13 plots trajectory distance to the groundtruth as a function of annotation round. These results show that a small number of targeted annotations enable rapid improvements in track quality as a direct consequence JPT's accurate uncertainty representation.

4.10 Conclusion

We propose JPT, a Bayesian solution to the general (i.e., batch) multi-object tracking problem (Chapter 4.5). We construct efficient inference to reason over permutations of associations (Chapter 4.6) and empirically demonstrate that JPT more effectively represents posterior uncertainty than baselines (Chapter 4.9.4) while outperforming them on standard tracking metrics (Chapter 4.9.5). We then show that JPT's accurate representation of uncertainty enables automatic scheduling of informative disambiguations which rapidly drive down posterior uncertainty while improving trajectory quality (Chapters 4.9.6, 4.7).

We believe that pairing accurate uncertainty quantification with automatic error recovery will enable multi-object tracking to be deployed in scientific domains such as animal behavior where common identity switching errors may otherwise corrupt conclusions drawn from hypothesis testing. Other approaches to reducing identity switching exist: namely, to manually inspect and refine track estimates as in [134] or to use uniquely-colored tags for each tracked target. Manual inspection and refinement of tracks is laborious. We use colored tags

for animal tracking in Chapter 5.3.1. Tags may be colored vests, painted dyes, or surgically implanted markers. Tags have limitations, however: not all animals can be outfitted with vests, dyes wear off over time, and observations may involve the addition of new subjects, as from birth. Experiments should be designed to reduce uncertainty where it is possible to do so. Where uncertainty cannot be reduced, the Joint Posterior Tracker provides an approach that minimizes human labor.

Data association uncertainty could be reduced by incorporating parts modeling such as from Chapter 3.4 and by incorporating visual appearance models, both learned and pre-specified (as when tags are available). Learning a decision rule for scheduling disambiguations that is supervised by the more expensive, quadratically-scaling mutual information estimator (Equation 4.36) could accelerate the time it takes to compute successive annotation questions. Finally, scene modeling could improve reasoning under long-term occlusions, and could be incorporated into the model by making the arrival, departure, detection, and clutter probabilities dependent on scene location.

Chapter 5

Primate Behavior Analysis

Ethologists study the behavior of animals, including modeling relationships with respect to genetics, neural activity, and lifetime development [194]. For the purposes of this work, behavior can be low-level kinematics (e.g., position, pose) or high-level activities (e.g., groom, chase). Figure 5-15 lists high-level behaviors analyzed in this work while low-level behaviors are modeled by multi-object tracking (Chapters 4.5, 5.3.1). In observational settings, scientists discover new behaviors or document relationships between known behaviors and the environment, such as their frequency and composition. In experimental regimes, scientists seek support for or against a hypothesis by assigning participants to treatments that correspond to levels of control exerted on the environment. Both observational studies and experimental designs require substantial human labor to conduct.

The study of behavior often requires scientists to observe hundreds of hours of data (often video), manually annotating timestamped events. The scope and scalability of these studies is limited because annotations are time-consuming (542 hours in a recent study [130]) and cannot be crowdsourced due to a need for privacy and expertise. Analysis is limited to behaviors and phenomena that scientists can directly and reliably perceive. Furthermore, analysis cannot be done in realtime or at all times, reducing many investigations to observational studies or else limited experimental designs, where interventions are manually performed at discrete points.

Automating aspects of behavioral analysis enable larger-scale and perhaps novel experiment designs. Calls for video tracking of multiple objects and classification of high-level behavior have been issued [7, 48] and are beginning to be answered but many of these approaches use representations that are difficult to interpret and relate to output estimates. They also tend to produce a single inference result, such as a single set of motion trajectories over time or a single behavior classification result [134, 77, 160, 133]. Multi-object tracking is prone to identity switching errors [136] and classification based on deep representations is prone to miscalibration [80]. We do not expect models to achieve error-free performance in most settings. Yet, relying on single

estimates that are likely to be wrong at least some of time is equally unsatisfying.

Uncertainty measures confidence in an estimate and can be used to draw attention to ambiguities in the data. Ambiguities can be exploited to identify and recover from potential mistakes (Chapter 4.7). As we will show in Chapters 5.5.3, 5.5.4, data ambiguity as represented by model uncertainty can also be exploited to improve high-level behavior classification performance.

Probabilistic analysis of behavior can assist in studying and identifying animal models of disease and have proven beneficial to the development of human treatments and cures [186]. As an example considered in Chapter 5.3, mutation in the SHANK3 gene is associated with autism spectrum disorder and Phelan-McDermid syndrome in humans [138, 161]. Abnormal behaviors including reduced social interaction, motor impairment, repetitive grooming, and self-injury have been observed in mice with the SHANK3 mutation [101]. Yet, mice are not ideal animal models for human social disorders [18]. Evidence that links SHANK3 mutations in primates to autism-like behaviors would equip scientists with a primate animal model for autism that is more socially and genetically relevant to humans [18]. Consequently, as outlined in Chapter 5.1, this work examines observational and experimental behavior data collected in collaboration with biologists and spans two types of primates: macaques and marmosets.

5.1 Approach

Chapter 5.3 develops the Nonparametric Extents Model (NPE), a Bayesian nonparametric multi-object tracker that infers the positions and per-observation associations over time of multiple objects. NPE combines multi-object tracking and a simple form of parts modeling that later motivated the development of the Nonparametric Parts Model (Chapter 3.4) and Joint Posterior Tracker (Chapter 4.5).

NPE models the motion of multiple objects in a state space with linear dynamics. It samples per-observation object associations by combining an explicit background model with an object likelihood based on a Dirichlet Process Gaussian Mixture [60, 12] observation model. Mixture components correspond to object parts, or extents, which NPE uses to infer object centroids over time. Rauch-Tung-Striebel [170] smoothing provides final track estimates for each set of sampled object associations. The Augmented Nonparametric Extents Model (Chapter 5.3.2) additionally models per-object appearance.

We apply NPE tracking to more than 100 hours of video data in an experimental setting. The experiment asks whether SHANK3 gene mutations in macaque primates cause autism-like behaviors such as social and motor impairment. Tracked videos are repeated trials of pairwise social interaction between primates with and without the SHANK3 gene mutation. We quantitatively evaluate performance on

a subset of data manually labeled for object trajectories and provide tracking estimates to collaborators for further analysis.

Chapter 5.4 describes Marmoset100, a 100-hour observational dataset of primate behaviors. Marmoset100 consists of 48 RGB, depth, and audio recordings that collectively contain more than 9 million video frames. Collaborators record this data using software I developed; they additionally label a 31-hour subset for 25 high-level behaviors (Chapter 5.4.4).

We find that NPE tracking is not robust to background variation or primate occlusions in Marmoset100. In response, Chapter 5.4.2 trains a pixel-accurate marmoset detector for use as input to JPT tracking (Chapter 4.5). We compute marmoset detections and perform JPT tracking on all Marmoset100 data.

Chapter 5.4.3 directly compares JPT tracking performance on Marmoset100 to Multi-Animal DeepLabCut (DLC) [117, 134], a popular approach used in animal tracking. JPT and DLC are both tracking-by-detection approaches. JPT supports arbitrary object representations whereas DLC exclusively works with an object pose representation. We hold the training data available to JPT's object detector and DLC's pose estimator constant for equal comparison and show that JPT significantly outperforms DLC.

Chapter 5.5 performs supervised, multilabel behavior classification of the Marmoset100 behaviors described in Chapter 5.4.4. We perform multi-object tracking on input videos using varying representation (centroid point, object contour, skeletal pose), and presence (JPT) or absence (DLC) of uncertainty in tracking estimates. Behavior classification takes multi-object tracks as input and independently estimates each behavior. We quantitatively demonstrate that uncertainty representation improves average behavior classification performance and that simple, point-based object representations outperform complex, pose-based object representations when the point trajectories are more accurate motion estimates.

5.2 Contributions

Our automated Nonparametric Extents tracking (Chapter 5.3) saved scientists from labeling the positions of primates in over 100 hours of video in an experimental setting, a task that reportedly takes 133% video time *per object* for point trajectories, even when using an efficient annotation interface (cf. Section 5, Figure 6 and Supplemental Section 1 of [132]).[30]

Analysis by collaborators of Nonparametric Extents tracking (NPE) estimates on pairwise macaque interactions contributed to the first experimental evidence that SHANK3 gene mutations cause autism-like behaviors in macaques (Chapter 5.3.4) and was published in [233].

Per-pixel marmoset detections, JPT multi-object tracking results, and labeled behaviors combine to form a novel, terabyte-scale RGB-

[30] The total time saved for point trajectory annotation is 266 hours. The Nonparametric Extents Model provides per-pixel object associations, however. [132] shows significantly longer annotation for bounding boxes over point trajectories but does not consider per-pixel associations, which presumably take even longer.

Depth dataset of marmoset behavior in home cage settings (Chapter 5.4). Our automated detection and tracking facilitate per-pixel labeling of more than 9 million video frames (Chapter 5.4.2).

Quantitative evaluation of JPT (Chapter 4.5) and DLC tracking [134] on a subset of Marmoset100 data demonstrates that JPT produces superior performance when its underlying object detections are trained on the same data as DLC pose estimates (Chapter 5.4.3).

Chapter 5.5 studies how multi-object tracking representation and the presence of uncertainty in tracking estimates affect follow-on behavior classification performance. In Chapter 5.5.4, we quantitatively demonstrate that:

- Behavior classification is improved by uncertainty representation in tracking estimates, regardless of object representation,
- Behavior classification based on JPT tracking outperforms behavior classification based on DLC tracking, even when JPT only uses less informative point-based object representations with no uncertainty (as compared to DLC's pose representation),
- Behavior classification based on a simple, point-based object representation with uncertainty on average outperforms behavior classification with a complex, contour-based object representation without uncertainty,
- Behavior classification based on complex, contour-based object representation outperforms behavior classification based on simple, point-based object representation when both approaches either do or do not have access to uncertainty.

JPT's superior multi-object tracking performance and follow-on behavior classification compared to DLC on Marmoset100 suggest that the animal behavior community is likely to benefit by adopting approaches, such as JPT, that represent uncertainty in tracking estimates.

5.3 Autism in Macaques

We develop the Nonparametric Extents Model, a generative, nonparametric model for multi-object tracking that directly uses 2D pixels or 3D RGB+Depth points to model object and background associations (Chapter 5.3.1). We then specialize it to the Augmented Nonparametric Extents Model (Chapter 5.3.2), which can incorporate target-specific appearance. Chapter 5.3.3 describes inference in both models. Finally, in Chapter 5.3.3, we deploy the Augmented Nonparametric Extents Model in a large-scale, controlled experiment on pairwise behavior in primates and provide quantitative and qualitative performance evaluation on experimental data.

5.3.1 Nonparametric Extents Model

The Nonparametric Extents Model (NPE) tracks the motion and shape of a known number of objects over time. Objects are naively modeled as a body centroid with linear Gaussian dynamics and a collection of Gaussian components that are independent across time, conditioned on object centroid. Within a frame, components may correspond to object parts, but they do not capture part persistence or part dynamics over time. We call these components, "extents." Table 5.1 summarizes NPE notation.

NPE allows objects to claim more than one observation at each time $t = 1, \ldots, T$. It nonparametrically estimates the body centroids and extents of a known number of objects over at each time. The likelihood is,

$$p(y_{tn} \mid z_{tn}, b_n, \theta_t) = \tag{5.1}$$
$$\mathrm{N}\left(y_{tn} \mid b_n, \Sigma_B\right)^{\mathbb{I}(z_{tn}=0)} \prod_{p=1}^{\infty} \mathrm{N}\left(y_{tn} \mid \mu_{tp}, \Sigma_{tp}\right)^{\mathbb{I}(z_{tn}=p)}$$

which forms an infinite mixture model for each observation n if associations z_n are marginalized out. Object components $\theta_t = \{\theta_{tp}\}_{p=1}^{\infty}$ are independent across time, conditioned on object centroid trajectories x_t

for each observation n, is an infinite mixture model over components $\theta_t = \{\theta_{tp}\}_{p=1}^{\infty}$ and a per-observation static background model b_n with shared covariance Σ_B. Components are indexed by p and interpreted as object extents that collectively form a nonparametric density estimate for each object. Components are generated as,

$$p(\mu_{tp} \mid \delta_{tp}, x_{tk}) = \tag{5.2}$$
$$\mathrm{Unif}\left(\mu_{tp} \mid U_0\right)^{\mathbb{I}(\delta_{tp}=0)} \prod_{k=1}^{K} \mathrm{N}\left(\mu_{tp} \mid H\, x_{tk}, \Sigma_X\right)^{\mathbb{I}(\delta_{tp}=k)}$$
$$p(\Sigma_{tp} \mid \delta_{tp}, x_{tk}) = \tag{5.3}$$
$$\mathrm{IW}\left(\Sigma_{tp} \mid \upsilon_0, S_0\right)^{\mathbb{I}(\delta_{tp}=0)} \prod_{k=1}^{K} \mathrm{IW}\left(\Sigma_{tp} \mid \upsilon, S_t\right)^{\mathbb{I}(\delta_{tp}=k)}$$

where x_{tk} is the position and velocity of object k at time t, Σ_X is the shared covariance about which objects generate components, U_0 is a uniform region over the observation space that explains "clutter" components, and H projects object position from the the latent space to component space. Clutter components have broad, uncertain Inverse-Wishart priors with parameters (υ_0, S_0) while object components have weak data-dependent Inverse-Wishart priors with parameters (υ, S_t) estimated at each time over non-background observations:

$$S_t = \sum_{n=1}^{N} y_{tn}\, y_{tn}^{\top}\, \mathbb{I}(z_{tn} > 0) \tag{5.4}$$

Variable	Description
$x_{tk} \in \mathbb{R}^{d_x}$	Trajectory
$y_{tn} \in \mathbb{R}^{d_y}$	Observation
z_{tn}	Observation association
θ_{tp}	Component/part
μ_{tp}	Component mean
Σ_{tp}	Component covariance
δ_{tn}	Component association
b_n	Background model
$t \geq 1$	index for time
$k \geq 1$	index for objects
$p \geq 1$	index for components
$n \geq 1$	index for observations

Table 5.1: Nonparametric Extents Model notation.

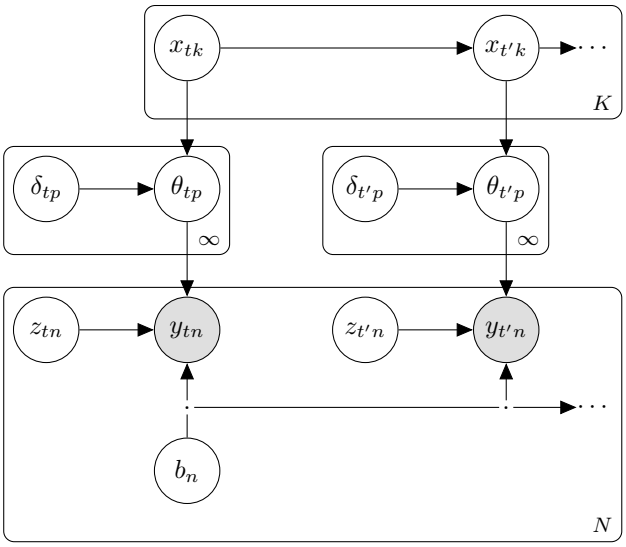


Figure 5-1: The Nonparametric Extents graphical model where $t' = t + 1$. Body centroids x_{tk} are tracked for $k = 1, \ldots, K$ objects over times $t = 1, \ldots, T$. Object components θ_{tp} are independently and nonparametrically estimated at each time t and associated to objects or clutter according to component associations δ_{tp}. Observations y_{tn} for $n = 1, \ldots, N$ are independently associated to background or to components at each time t according to observation associations z_{tn}.

NPE has two sets of associations at each time: $z_{tn} = p$ associates observation y_{tn} to background if $p = 0$ or to an object component $p > 0$. $\delta_{tp} = k$ associate component θ_{tp} to clutter if $k = 0$ or object k if $k > 0$. We emphasize that there is a separate background model for each observation n and a single clutter model for all components p. Associations are generated as,

$$p(\delta_{tp}) = \text{Cat}\left(\delta_{tp} \mid (K+1)^{-1}, \ldots, (K+1)^{-1}\right) \tag{5.5}$$

$$p(z_{tn} \mid \pi_t) = \text{Cat}(z_{tn} \mid \pi_t) \tag{5.6}$$

$$p(\pi_t \mid \alpha) = \text{GEM}(\pi \mid \alpha) \tag{5.7}$$

$$\tag{5.8}$$

for stick-breaking weights π and Dirichlet Process concentration parameter α (Chapter 2.4). Observe that the associations of parts to objects δ_{tp} have uniform mixture weights over the K objects and one clutter class (Equation 5.5) while the associations of observations to components at time t have an infinite set of mixture weights, π_t (Equation 5.7. This was a design decision based on the assumption that tracked objects would contain approximately the same number of parts but that within-object parts would vary in appearance, shape, and visibility at any given time t. We emphasize that components are estimated independently at each time; they have no dynamics nor do they persist over time. Finally, objects evolve according to linear Gaussian dynamics,

$$p(x_{tk} \mid x_{(t-1)k} = \text{N}\left(Fx_{(t-1)k}, Q\right) \tag{5.9}$$

where F is a linear dynamics function represented a random acceleration model and Q is a covariance set to reflect expected object motion.

NPE is used as a nonparametric baseline to the Nonparametric Parts Model in Chapter 3.6. For 2D video inputs, observations y_{tn} are image coordinates and latent trajectories x_{tk} are describe object location and velocity in image coordinates. For 3D point cloud inputs, observations y_{tn} are world coordinates (assuming the sensor and the world coordinate frames align) and latent trajectories x_{tk} describe object location and velocity in world coordinates. An augmented version of the Nonparametric Parts Model (ANPE) is used to track macaque primates for more than 100 hours in a controlled experiment on behavior discussed in Chapter 5.3.4. The augmented model is defined next.

5.3.2 Augmented Nonparametric Extents Model

The Nonparametric Extents Model is augmented so that it is better suited to specific tracking environments. In the case of macaque tracking in Chapter 5.3.4, one primate always wears a colored bandana around its neck. The bandana is not visible in all frames due to occlusions, but provides strong evidence of object identity when available. Define the logistic regression model for object k, observation n at time t as,

$$p(c_{tkn} \mid w_k, y_{tn}) = \text{Bern}\left(c_{tkn} \mid \sigma(w_k^\top y_{tn})\right) \tag{5.10}$$

where $w_k \in \mathbb{R}^{D_y}$ are a set of object-specific identification weights (e.g., trained to capture the color of the bandana) and c_{tkn} is a per-observation, per-object boolean that indicates whether observation y_{tn} is likely to have been generated by object k. Let,

$$N_{tkp} = \sum_{n=1}^{N} c_{tkn} \tag{5.11}$$

be the count of observations associated to component p that are identified with object k. observation count sN_{tkp} in Equation 5.11 associated to the target-k appearance model in Equation 5.10 should increase the probability that component p is associated to target k. Modify sampling of associations of components to parts so that for $k > 0$:

$$p(\delta_{tp} = k \mid \theta_{tp}, x_t, N_{tkp}) \propto \begin{cases} \text{N}(\mu_{tp} \mid H\, x_{tk}, \Sigma_X) & \text{if } N_{tkp} < \tau_1 \\ 1 - e^{\lambda N_{tkp}} & \text{o.w.} \end{cases} \tag{5.12}$$

The first case is the conditional implied by the Nonparametric Extents generative model in Chapter 5.3.1 whenever the counts N_{tkp} fall below threshold τ_1. The second case is the CDF of the exponential distribution with parameter λ. When $N_{tkp} \geq \tau_1$ then the probability that component p belongs to object k rapidly becomes proportional to 1 as N_{tkp} increases. Figure 5-2 shows examples of the CDF for different λ. The two cases are separated by a threshold so that *not* observing high

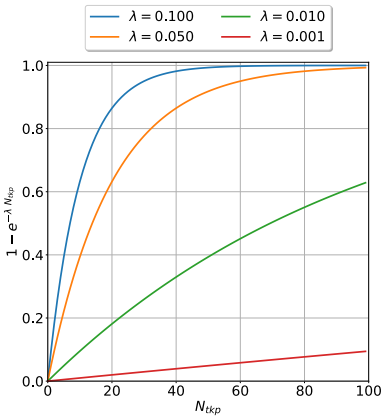


Figure 5-2: Exponential CDF association model in Augmented NPE. Higher observation counts N_{tkp} (Equation 5.11) that are classified as belonging to target k (Equation 5.10) rapidly increase the probability that component θ_{tp} is associated to object k (Equation 5.12).

counts for a given component does not penalize association of that component to object k. This is chosen because the bandana will not be visible in all parts that belong to a macaque, nor will the bandana always be visible to the part(s) that contain it due to occlusions.

The probability of associating components to clutter is also augmented to reduce the occurrence of large, sparsely-populated components being associated to objects. Let $y_t^p = \{y_{tn} : z_{tn} = p, p > 0\}$ be the set of observations associated to component p at time t and $M_{tp} = |y_t^p|$ be the number of observations associated to component p at time t. For each element in y_t^p find its nearest neighbor and compute their L2 norm distance. Then, take the average over all such nearest neighbor distances in y_t^p. Specifically, define the average nearest-neighbor distance of observations in component p as,

$$\gamma_{tp} = \frac{1}{M_{tp}} \sum_{y_{tn} \in y_t^p} \min_{y_{tn'} \in y_t^p \setminus y_{tn}} ||y_{tn} - y_{tn'}|| \tag{5.13}$$

and augment the probability of assigning component p to clutter as,

$$p(\delta_{tp} = 0 \mid \theta_{tp}, \gamma_{tp}) \propto \tag{5.14}$$

$$\begin{cases} \text{Unif}\left(\mu_{tp} \mid U_0\right) \text{IW}\left(\Sigma_{tp} \mid \upsilon_0, S_0\right) & \text{if } \gamma_{tp} < \tau_2 \\ \frac{1}{\beta} e^{-\gamma_{tp}/\beta} & \text{o.w.} \end{cases} \tag{5.15}$$

where the first case corresponds to the standard Nonparametric Extents Model of Chapter 5.3.1 and the second case is the exponential distribution with scale parameter β. Equation 5.15 rapidly increases the probability that a component will be associated to clutter as the average nearest neighbor distance between observations in that component increases.

The augmentations from Equations 5.12 and 5.15 do not conform to the Nonparametric Extents generative model as stated in Figure 5-1 but substantially improve performance for long-term Macaque tracking. Incorporating per-observation classifier decisions can be modeled generatively by adding c_{tkn} as observations that depend on y_{tn}, object appearance weights w_k, and associations z_{tn}. Exact sampling of z, δ and θ would become expensive, however. The nearest-neighbor condition of Equation 5.15 would be more challenging to model in a generative framework because NPE assumes that observations are iid draws from components. Modeling an average distance generatively would require the use of a stochastic process as the conditional probability distribution of observations, and would complicate the model definition and inference without necessarily improving tracking performance. Following, we detail inference for the general Nonparametric Extents Model, along with details relevant for the Augmented Nonparametric Extents Model.

Algorithm 8: Randomized Median Finding Algorithm

Input : $y^n = \{y_{tn}\}_{t=1}^{T}, L$
Output: median $\left(\{y_{tn}\}_{t=1}^{T}\right)$
Sample without replacement $y^q \subset y^n$ s.t. $|y^q| = L \log T$.
Sort y^q
return median$(y^q) = y^q_{(1/2)L \log T}$

5.3.3 Inference in the Nonparametric Extents Model

NPE inference proceeds by estimating the latent variables:

$$\{z_{tn}\}_{t=1,n=1}^{T,N} \qquad \{\theta_{tp} = (\mu_{tp}, \Sigma_{tp})\}_{t=1,p=1}^{T,\infty} \tag{5.16}$$

$$\{\delta_{tp}\}_{t=1,p=1}^{T,\infty} \qquad \{x_{tk}\}_{t=1,k=1}^{T,K} \tag{5.17}$$

Joint sampling from the posterior on (z, θ, δ, x) conditioned on all observations y is well-defined but has no analytic form. Exact sampling can be accomplished by Gibbs iterations where a separate Dirichlet Process is sampled among $N \approx 1e6$ observations, the number of pixels in an HD camera. To save computation, we make three approximations that allow components to be estimated in parallel across time, leaving estimation of latent object states x_{tk} as the only serial process:

1. Foreground associations are estimated by sampling an auxiliary indicator variable f_{tn} for each observation so that background is associated as in the first term in Equation 5.1 ($f_{tn} = 0$) and foreground (the space of all components) is associated according to a Uniform distribution over the observation space ($f_{tn} = 1$),

$$\begin{aligned} p(f_{tn} = 1 \mid y_{tn}, b_n) &\propto \text{Unif}(y_{tn} \mid U_A) \\ p(f_{tn} = 0 \mid y_{tn}, b_n) &\propto \text{N}(y_{tn} \mid b_n, \Sigma_B) \end{aligned} \tag{5.18}$$

2. We use a variational sampler to approximately sample components θ_{tp} in parallel over all time t. Components are only fitted to associations that have been associated to the foreground f_{tn} (Equation 5.18). Background and component associations z_{tn} are then resampled according to Equation 5.1.

3. We estimate marginal states x_{tk} using RTS smoothing [170].

Additionally, we fit the background one time to each collection of observations so that the background model for observation n is,

$$b_n = \text{median}\left(\{y_{tn}\}_{t=1}^{T}\right) \tag{5.19}$$

which can be estimated in $O(\log T \ \log\log T)$ time with probability of error less than $\frac{1}{T^2}$ according to randomized Algorithm 8. See [41] for additional analysis.

Algorithm 9: Nonparametric Extents Inference.

Input : y
Output: z, δ, θ, x, P

1 **for** $n \in 1, \ldots, N$ *in parallel* **do**
2 | Compute b_n according to Algorithm 8;
3 **for** $t \in 1, \ldots, T$ *in parallel* **do**
4 | **for** $n \in 1, \ldots, N$ *in parallel* **do**
5 | | Sample foreground indicator f_{tn} according to Equation 5.18
6 | Let $y_t^f = \{y_{tn} : f_{tn} = 1\}$
7 | Sample $(\theta_t, \pi_t, z_t) \mid y_t^f$ according to Variational DPMM [28].
8 | **for** $n \in 1, \ldots, N$ *in parallel* **do**
9 | | Resample $z_{tn} \mid \pi_t, y_{tn}, \theta_t$ proportional to the product of Equations 5.1, 5.6.
10 **for** $k \in 1, \ldots, K$ *in parallel* **do**
11 | **for** $t \in 1, \ldots, T$ **do**
12 | | Predict $\hat{x}_{tk}, \hat{P}_{tk}$
13 | | Sample $\delta_{tp} \mid \hat{x}_t, \hat{P}_t, \theta_t$ proportional to the product of Equations 5.3, 5.2, 5.5
14 | | Compute filtered estimates $\tilde{x}_{tk}, \tilde{P}_{tk} \mid \hat{x}_{tk}, \hat{P}_{tk}, \delta_t, \theta_t$
15 | **for** $t \in T, \ldots, 1$ **do**
16 | | Compute smoothed estimates $x_{tk}, P_{tk} \mid \theta, x_{t+1}, P_{t+1}, \tilde{x}_{tk}, \tilde{P}_{tk}$

The inference procedure for the Nonparametric Extents Model is specified in Algorithm 9. Lines 1–9 are computed offline and in parallel over all observation indices n for the background model b_n (Line 2) and in parallel over all times t for components (Line 7) and their associations (Line 9). Lines 10–14 are standard linear Gaussian filtering equations where component means act as "observations" for the filter and Lines 15–16 are standard linear Gaussian smoothing equations. The latent state outputs $x, P = \{x_{tk}, P_{tk}\}_{t=1,k=1}^{T,K}$ are the approximate marginal distribution parameters of each object at each time.

The Augmented Nonparametric Extents Model (Chapter 5.3.1) assumes pre-trained logistic regression weights w_k for each object k that has an identifying marker and modifies Line 5.5 so that Equations 5.2 and 5.3, are replaced with Equations 5.12 and 5.15.

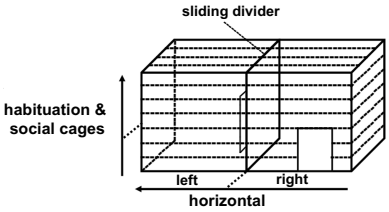


Figure 5-3: Macaque environment schematic, measuring $3m \times 1.5m \times 1.5m$. A divider in the middle is used to physically separate subjects in the first and last ten minutes of each trial.

5.3.4 Evaluation

Collaborators design an experiment that asks: do SHANK3 gene mutations cause autism-like behaviors in macaque primates? If so, the experiment would contribute the first evidence for using primate animal models in autism research. Animal models for autism can aid the development of novel drugs and treatments for humans with autism.

Autism-like behaviors include reduced motor function, environment exploration, and social interaction. Motor function and environment exploration is estimated by collaborators from ANPE video tracking of macaque primates while social interaction is manually scored by

collaborators who are experts in primate behavior.

Macaque subjects belong to one of three groups in this experiment: "Mutants" are genetically edited by collaborators using CRISPR-Cas9 [46, 151] so that they have the SHANK3 gene mutation. "Controls" and "Probes" receive no genetic editing. All other factors about environment and rearing prior to the experiment are held constant.

The experiment proceeds by repeated trials of two macaque subjects being placed into a cage with dimensions $3m \times 1.5m \times 1.5m$ (Figure 5-3) and monitored for movement and social interaction. Each trial lasts 50 minutes and consist of either one mutant and one probe being paired, or one control and one probe being paired. A physical divider is placed in the middle of the cage for the first and last 10 minutes of each trial, separating the subjects. Trials proceed in random order so that each mutant eventually interacts with all 10 probes, and each control also interacts with all 10 probes. Prior to the experiment, each subject is individually habituated to the cage for 30 minutes on two separate days. In total, there are five mutants, six controls, and 10 probes. Figure 5-4 visualizes the 110 social interaction trials that comprise each mutant interacting with all probes and each control interacting with all probes. Social interaction data spans 91.7 hours and habituation data spans 21 hours. ANPE video tracking is performed on social interaction and habituation data.

We are interested in statistical comparisons of behavior between mutant/probe and control/probe social interactions. Behaviors of interest include low-level motion over time and high-level activities such as groom or chase. It is common for biologists to manually label both low-level and high-level behaviors in behavior experiments but the volume of data in this experiment makes that time-consuming. PathTrack, a recent work [132] that augments multi-object tracking with dense annotations collected in an efficient interface, states that it takes $1.33x$ video time *per track* to densely annotate object centroid trajectories in video. By that calculation, it would take 300 hours to densely track Macaque data in a state-of-the-art interface. Unlike the PathTrack approach, annotations cannot be crowdsourced due to the sensitive nature of the data, so all annotations would have to be manually performed by persons within the lab.

Instead, we use the Augmented Nonparametric Extents Model to infer low-level motion behavior over time, both for habituation trials and social interaction trials. We manually annotate a 10000-frame subset of experiment data to use as groundtruth to estimate overall tracking quality. We generate multiple track estimates by running Algorithm 9 and sampling 100 set of trajectories from the RTS smoothed [170] marginal distributions on latent states.

Figure 5-6 provides the CLEAR MOT [23] Multi-Object Tracking Accuracy (MOTA, Equation 4.37) and IDF1 tracking metrics where

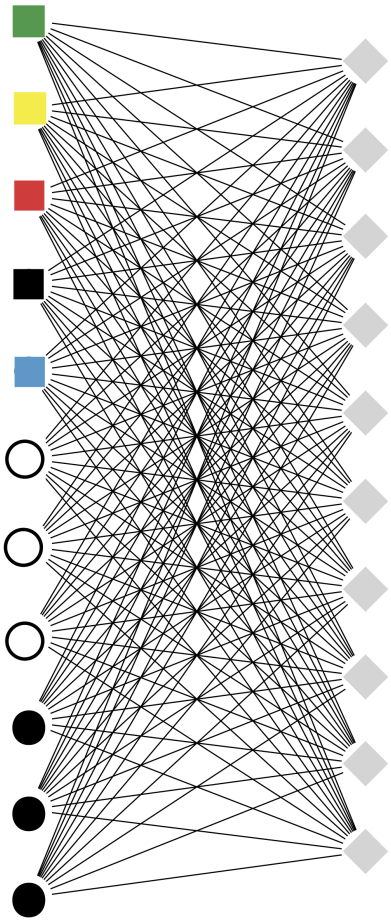


Figure 5-4: Macaque experiment schematic. Each trial consists of a single mutant (square) and probe (diamond) in the same cage, or a single control (circle) and probe in the same cage. There are a total of 110 trials, indicated by edges, one for each mutant/probe and control/probe pairing. All mutants except the black square are male. Controls are male (white circle) or female (black circle). Half of the probes are male, and half are female.

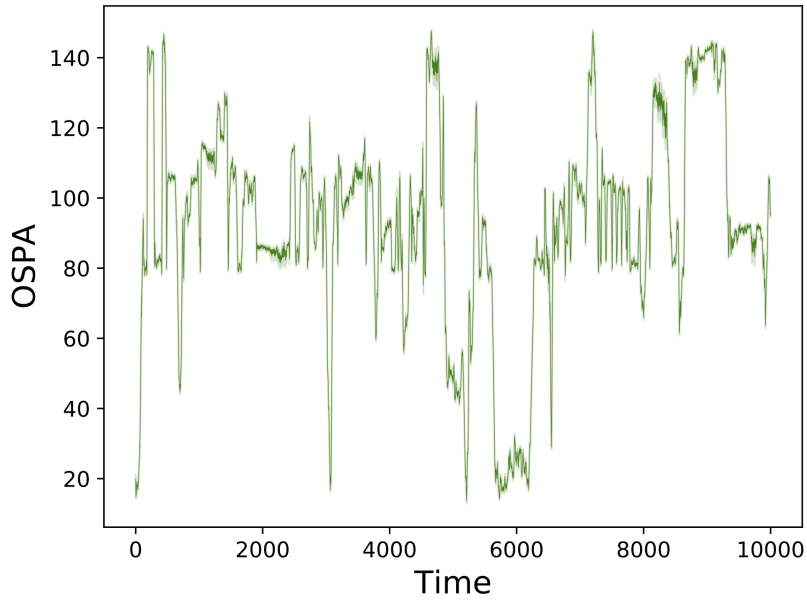


Figure 5-5: Nonparametric Extents OSPA(2) Tracking Metrics. OSPA(2) combines errors in the number of objects estimated at each time and the localization of each object at each time. NPE never incurs cardinality errors for this experiment because the number of objects are specified in advance. Localization errors are summed across targets so that the mean localization error in each frame, summed over both macaques, is 80 pixels, less than 4% of video width for each object. Localization errors tend to increase when macaques are completely occluded.

IDF1 is the analog of the F1 score for multi-object tracking:

$$\mathrm{IDF}_1 = \frac{2\ \mathrm{IDTP}}{2\ \mathrm{IDTP} + \mathrm{IDFP} + \mathrm{IDFN}} \tag{5.20}$$

IDTP, IDFP, and IDFN are true positives, false positives, and false negatives, respectively, but are computed as part of a bipartite matching problem that rewards consistent identification of inferred trajectories and groundtruth trajectories. We see that Augmented NPE has very high IDF1 and MOTA scores (both IDF1 and MOTA have a maximum of 1.0) on macaque experimental data. Further inspection shows that more than 50% of samples have 0 identity switches, with the average number of identity switches being 0.86. We attribute this to explicit modeling of the colored bandana worn by one primate in each social interaction.

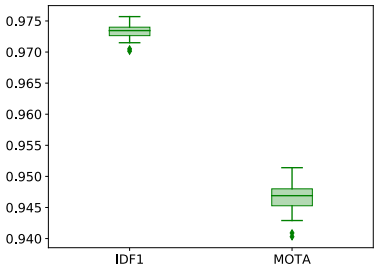


Figure 5-6: Nonparametric Extents IDF1 and MOTA tracking metrics on a representative subset of Macaque data. Tracking performance is very high, with both IDF1 and MOTA in the range of 0.94–0.98 (of a maximum 1.0).

The factors that most degrades performance are track misses (532.3± 19.3) and track fragmentations (298.3 ± 13.1). Misses and fragmentations primarily occur when either macaque is at the center of the cage, where long-term occlusion becomes possible. NPE typically maintains the correct mean location but has increasing uncertainty due to lack of observations; samples from the marginals penalize tracking performance compared to the mean location estimate because the object location uncertainty has no awareness that the occlusion can only oc-

cur in the middle of the cage. Figure 5-7 shows an example where there is high uncertainty but accurate mean location estimation. Observe that the mean estimated location of the blue macaque remains near the groundtruth despite the large uncertainty from occlusion between frames 919 to 963. Figure 5-8 shows another qualitative example where NPE accurately tracks macaques when they pass by one another. Tracking errors occur in less than 5% of frames, yielding generally high tracking performance.

Figure 5-7: Macaque tracking during occlusion events. Top-half of images only show smoothed state estimate of object locations (blue, red filled circle) and covariance (blue, red open circle) along with groundtruth (orange star). Bottom-half of images also show component covariances and observations colored according to the object they are associated to (white components are clutter). The uncertainty in blue's location grows during the occlusion from images 919–931 before diminishing when it is observed again at 963. State estimates remain effective despite high uncertainty.

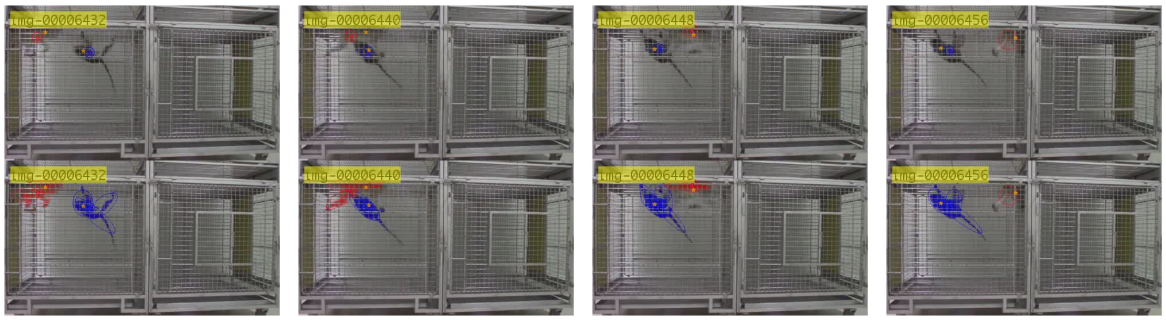


Figure 5-8: Macaque tracking when objects pass by. Top-half of images only show smoothed state estimate of object locations (blue, red filled circle) and covariance (blue, red open circle) along with groundtruth (orange star). Bottom-half of images also show component covariances and observations colored according to the object they are associated to (white components are clutter). Both blue and red remain tracked despite passing near one another.

Figure 5-5 visualizes the OSPA(2) tracking metric [178] over time with 95% shading over 100 samples. OSPA(2) accounts for cardinality errors (incorrect number of tracked targets compared to groundtruth) and localization errors (minimizing sum of L2 norms truncated at distance c for bipartite matchings between estimates and groundtruth). The number of objects is known and always correctly estimated in this experiment; Typical localization errors are 40 pixels per object assuming even distribution of errors across both object. 40 pixels is less than 4% of video width, with minimal variance between samples.

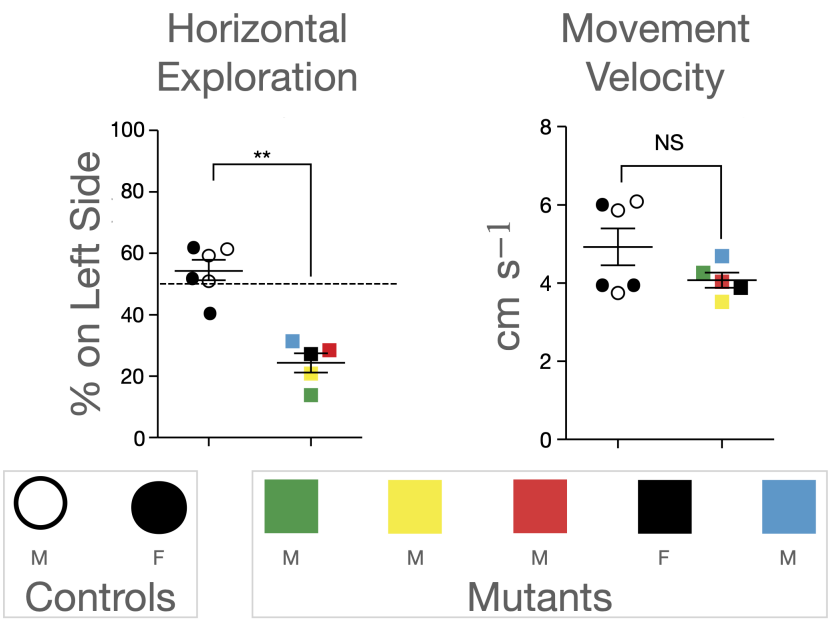


Figure 5-9: Statistical analysis of macaque tracking estimates. Using ANPE tracking estimates, collaborators found significant differences in environment exploration (left) as measured by in-plane motion from smoothed ANPE track estimates ($p = 0.0043$) whereas they found no significant differences in movement velocity. Squares represent mutants and circles represent controls. See Figure 5-4 for additional details on notation.

NPE tracking was performed on habituation sessions and repeated social interaction trials given acceptable qualitative and quantitative performance. Smoothed track estimates were provided to collaborators who conducted Two-Tailed Mann-Whitney U-Tests [58] to compare whether environment exploration between mutants paired with controls is greater than environment exploration between controls paired with probes (Figure 5-9).

Collaborators found significant differences in environment exploration (left) as measured by in-plane motion from smoothed ANPE track estimates ($p = 0.0043$) whereas they found no significant differences in movement velocity, suggesting that impaired motor function did not explain the reduced exploration found in tracking results or reduced social interaction found by manual scoring of higher-level behaviors. Additional details of this experiment and its conclusions can be found in [233].

5.4 Marmoset100 Behavior Dataset

Tracking with uncertainty quantification reduces the human annotation burden when conducting behavior experiments. Yet, tracking only quantifies motion over time. In the paired social interaction experiment with macaques (Chapter 5.3.4), high-level behaviors were still manually labeled. A key limitation to building classification pipelines

in behavior science is that collection conditions vary widely and datasets are not typically shared. In the chapter, we develop Marmoset100, a dataset of low-level motion and high-level marmoset behaviors over time.

Marmoset100 was collected by collaborators using software developed as part of this dissertation. Marmoset100 contains:

1. 100 hours of high-resolution RGB+Depth+Audio recordings taken from a top-down perspective in a variety of home-cage settings. Each recording comes with calibration information so that depth images can be registered to RGB images. Chapter 5.4.1 describes the dataset and its collection.

2. A trained, pixel-accurate marmoset detector that is robust to variations in background, occlusion, and lighting (see Figure 5-11 for qualitative results and Figure 5-13 for quantitative analysis). We computes detection on all Marmoset100 data.

3. Sampled JPT tracking estimates (Chapter 4.5) on all Marmoset100 data using marmoset detections as input. We quantitatively and qualitatively compare JPT tracking performance to Multi-Animal DeepLabCut [117, 134] on a subset of Marmoset100 data (Chapter 5.4.3) and to additional tracking methods in Chapter 4.9.

4. A 31-hour subset of recordings are labeled and segmented by collaborators for 25 high-level behaviors. We describe each behavior in Chapter 5.4.4 in preparation for experiments on supervised behavior classification in Chapter 5.5.

Chapter 5.4.1 details recording equipment and collection conditions. Chapter 5.4.2 describes the pixel-accurate marmoset detector and sampled JPT tracking estimates. Chapter 5.4.4 discusses the collection of labeled behaviors and their curation for supervised behavior analysis.

Figure 5-10: Marmoset home cage schematic. All Marmoset100 videos were captured in cages of this design, though with variations in the location and availability of shelving, branches, and enrichment. See britzco.com for more information.

5.4.1 Data Collection

Marmoset recordings were collected with a Microsoft Kinect2 camera recording at 1920 × 1080 RGB resolution and 512 × 424 depth resolution. Audio was synchronously recorded from the microphone of a Macbook Pro 2016 laptop. Recordings are compressed in realtime so that RGB is stored in lossy H.264 format with a footprint of 4GB / hour, depth is stored in lossless FFV1 format [2] with a footprint of 7.3GB / hour, and audio is stored in lossy Ogg Vorbis format at 0.06GB / hour, so that recordings require 11.36 GB/hour or 272.64GB/day. FFV1 supports 16-bit integers that are commonly used in depth images. Informal tests found that video compression with FFV1 was slightly better than storing a collection of PNG-encoded images.

RGB and depth frames are not explicitly synchronized; timestamps for each are recorded on a per-frame basis in a log file, and camera-specific calibration parameters are recorded so that depth images can

Figure 5-11: Pixel-accurate marmoset detection examples. Different marmoset detections are colored green or blue for easier visualization. Partial occlusions (bottom row) occur when marmosets are under shelves, branches, or one another yet the detector is capable of detecting associations.

be aligned to RGB. Despite lack of synchronization, depth images tend to be within 50ms of RGB captures. The dataset is 1TB of data compressed and contains more than 9 million RGB+Depth frames. I developed software for realtime data collection and compression [3] and post-hoc image alignment [1] while biology collaborators collected the data in accordance with IRB protocols and under veterinary supervision.

There are 49 recordings collected throughout 2017–2018 with a total length of 100 hours and average per-recording length of 2.04 hours. Data collection was performed in home-cage environments of pairs of marmosets where there was typically access to food, water, shelter, privacy (via a nest box), and enrichment (e.g., ropes, balls, wire mesh, and shelves for playing, climbing, and jumping). The camera was mounted on a bracket atop the cage. The cage configuration, mounting position, external lighting, and background varies between recordings though cages were always of dimensions $78 \times 78 \times 147$ cm. Figure 5-10 shows a schematic of the home cage.

5.4.2 Detections and Sampled Tracking Estimates

Informal tests of NPE tracking (Chapter 5.3.1) on Marmoset100 data failed to track marmosets across strong variations in background, occlusion, and lighting. In response, we train a marmoset detector and used marmoset detections as input to JPT tracking (Chapter 4.5).

We collect a representative set of static Marmoset100 images and hand-label the per-pixel associations of each marmoset. There are a total of 618 marmoset annotations in varied environments, lighting conditions, and occlusion states. We adapt the Mask RCNN [90] object detection network for binary classification of marmosets by re-

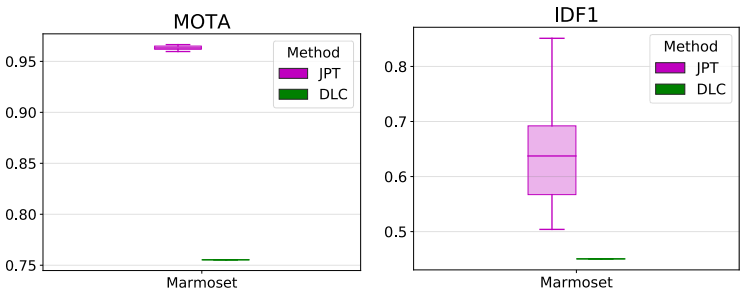


Figure 5-13: Multi-object tracking performance comparison between JPT (magenta) and DeepLabCut (green). DLC performance is represented as a line (green) because it has no uncertainty quantification in track estimation. JPT substantially outperforms DLC on tracking as measured by the composite MOTA (Equation 4.37) and IDF1 (Equation 5.20) measures.

Figure 5-12: Marmoset detection failure cases. Detections are sometimes merged (top) or missed (bottom)when marmosets are proximate.

placing the final sigmoidal layer and training all network weights for 1000 epochs. Network parameters for unmodified layers are initialized with weights that were pre-trained on Microsoft Coco [123]. Data was split 80% training and 20% testing. Training loss fell to 3% of its starting value, ending at 0.032 while testing loss fell to 24% of its starting value, ending at 0.249. As additional validation, the marmoset tracking results in Chapter 4.9.5 and Chapter 5.4.3 are based on marmoset detections but graded using an independently-labeled groundtruth.

Figure 5-11 shows a representative sample of marmoset detections. Visually, detection is robust on Marmoset100 data, including when marmosets are only partially visible. The primary failure mode occurs when marmosets are proximate, in which case their detections are sometimes merged so that only a single detection is generated. Figure 5-12 shows examples of these failure modes, which occur as a consequence of Mask RCNN's use of non-maximum suppression.[31]

JPT tracking (Chapter 4.5) is performed on all frames of Marmoset100 using the centroids of marmoset detections as input observations. JPT associations are used to recover the original, pixel-accurate detections. Figures 4-8, 4-9, and 4-11 provide quantitative and qualitative analysis on 15k-frame subset of this data.

[31] Non-maximum suppression merges overlapping detection regions into a single detection.

5.4.3 JPT and DeepLabCut Tracking Comparison

Multi-Animal DeepLabCut (maDLC, but we refer to it in this work as DLC) [117, 134] is a popular multi-object tracker that has recently been developed for animal behavior research. It uses a ResNet-based deep neural network architecture [89, 93] to independently detect per-frame pose estimates of multiple animals, heuristically combines them into tracklets, then stitches tracklets into trajectories for a pre-specified number of animals using a max-flow graph-based formulation. It's popularity stems from generally effective performance and high ease of use. It is distributed with a graphical interface so that no coding is required for the process of data annotation, pose estimation training and

detection, pose tracking, and manual, post-hoc track refinement.

We directly compare JPT to DLC on a 15k-frame subset of Marmoset100 data. DLC pose estimates are trained on the same images that JPT's input object detections are trained on so that comparison is on level footing at both the detection and the tracking stages. We label the same 618 images used to our pixel-accurate marmoset detector (Chapter 5.4.2) for eight marmoset skeleton points: head, left ear, right ear, neck, body, base of tail, midpoint of tail, and end of tail. Figure 5-17, DLC (Pose) and JPT (Point) visualizes the representations used by DLC and JPT in this comparison.

We train DLC pose estimation for 100k epochs on an NVIDIA GeForce 2080 Ti GPU.[32] We perform DLC tracking and compare to the same 15k-frame subset of Marmoset100 data that was used in Chapter 4.5.

[32] DLC recommends training for at least 50k iterations with a batch size of 8. We train for 100k iterations and find high performance with no evidence of overfitting on held-out data.

Figure 5-13 shows that JPT substantially outperforms DLC in terms of multi-object tracking metrics IDF1 (Equation 5.20) and MOTA (Equation 4.37). In practice, DLC has significantly greater numbers of track misses due to missing pose estimates. In contrast, JPT tracking based on pixel-accurate marmoset detection has far fewer track misses. Even when JPT has missing observations, it maintains a distribution on unknown object locations that informs association sampling. Figures 5-23, 5-24, 5-25 show examples of missing pose estimates in DLC.

Both JPT and DLC occasionally suffer from identity switches, but DLC provides a single tracking estimate with no notion of uncertainty. In contrast, JPT provides a distribution on associations. In Chapter 4.5, we used JPT's uncertainty to identify and correct possible identity switches. In contrast, DLC requires that users manually inspect and refine tracking quality.

5.4.4 Labeled Behaviors

Collaborators label a 31-hour subset of Marmoset100 for 25 high-level behaviors. Figure 5-14 shows each behavior. Behaviors include rapid activities (e.g., Jumping, Scratching, Stretching, Eating) that generally occur over 3–4 second intervals, and longer activities (Nuzzle Body, Request Grooming) that occur over tens of seconds. Collaborators automatically queued video clips for behavior labeling by inspecting pairwise absolute differences in RGB frames so that non-overlapping subsets of video that surpass a threshold are segmented out for labelling. Segments are padded by 15 frames before and after the queued length to provide additional context. Clips are labeled for presence or absence of each behavior for each of two marmosets. There may be 0, 1, or multiple behaviors for each marmoset in each clip. Figure 5-16 shows the behavior label format for a single behavior clip. We emphasize that the exact beginning and end time of each labeled behavior within each clip is not known.

Figure 5-15 displays the total count of annotated behaviors summed over both targets and the distribution of clip durations containing each behavior. 4/25 behaviors frequently occur: Jumping, Peering, Hanging

Figure 5-14: The 25 high-level behaviors of Marmoset100. Some behaviors involve both marmosets, such as Grooming Other, Nuzzle Body, Huddle, and Following. Others occur independently, such as Scratching, Grooming Self, and Hanging on Bars. Several involve interaction with the environment: Eating, Drinking, Inside Nest Box, Carrying Something.

on Bars, Inside Nest Box, Scratching each have more than 1000 labeled instances. 5/25 behaviors are rare: Nuzzle Face, Carrying Something, Rearing/Standing, Head Shake, Nuzzle Anus all have fewer than 50 labeled instances.

5.5 Behavior Classification

It is common for scientists to collect low-level motion behavior data in the form of manual or automatic video tracking. Low-level behaviors can be analyzed for variation between conditions (as in Chapter 5.3, Figures 5-9) and can help explain differences in high-level behavior [233]. Yet, it remains common for high-level behaviors to be manually classified by ethologists [7], as occurred in the study presented in Chapter 5.3. In this section, we investigate how tracked motion can be used to predict high-level behavior.

Tracked motion is commonly represented by simple point trajectories of object centroids. More complex representations include trajectories of pose estimates or object shapes as represented by pixel or point associations. Independent of representation, tracked motion contains frequent ambiguities in the form of potential identity switches. DeepLabCut [134] is currently the most popular tracker in animal behavior yet it contains no representation of association uncertainty. Indeed, most multi-object tracking formulations either do not represent

Figure 5-16: Marmoset100 behavior label format. Video clips are extracted from Marmoset100 based on motion thresholding. For each extracted clip and marmoset, the presence or absence of all 25 behaviors in Figure 5-14 are labeled in a binary vector. Positive labels indicate that the indicated behavior occurred *at some point* in the clip.

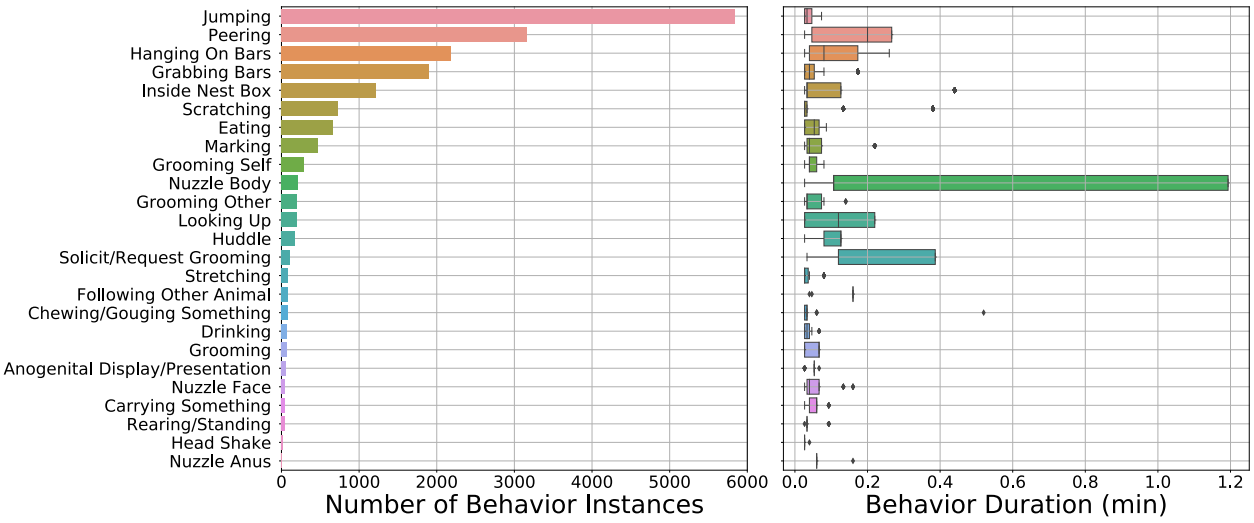


Figure 5-15: Marmoset100 behavior annotation counts and durations. Jumping, Peering, Hanging On Bars, Grabbing Bars, and Inside Nest Box are very short in duration and commonly occur, with more than 1000 labeled instances each. Others, such as Nuzzle Body and Solicit/Request Grooming have longer durations and occur infrequently.

uncertainty in tracking estimates or else they do so poorly as shown in Chapter 4.5. In contrast, we observed that our tracker, JPT, effectively captures uncertainty in tracking estimates (Chapter 4.9.4). We expect that some behaviors contain greater tracking ambiguity and that tacking ambiguity can be used to improve behavior classification.

We compare how tracking representations (point, contour, pose) and presence or absence of uncertainty in tracking estimates affects supervised behavior classification. Figure 5-17 visualizes five separate approaches: JPT tracking with point or contour representations, with and without uncertainty, and DLC tracking, which uses a pose representation.

5.5.1 Tracking Representations

DLC (Pose) directly produces and tracks pose estimates that are represented by a collection of points with consistent order across time. DLC produces a single tracking estimate by solving a graph optimization and thus has no representation of uncertainty.

JPT is a general multi-object tracker and can take as input observations with any vector-valued representation. In this experiment, JPT inputs are the centroids of marmoset detections from the trained marmoset detector in Chapter 5.4.2. The JPT (Point) representation is taken as a single posterior sample of inferred JPT trajectories.

JPT produces a joint distribution over trajectories *and* associations. Associations specify which of the original pixel-accurate marmoset detections are associated to each animal at each time (Figure 5-11). Marmoset detections contain information about the shape and visibility of each marmoset that may be relevant to classification of high-level behaviors. We investigate the utility of shape information in the form of

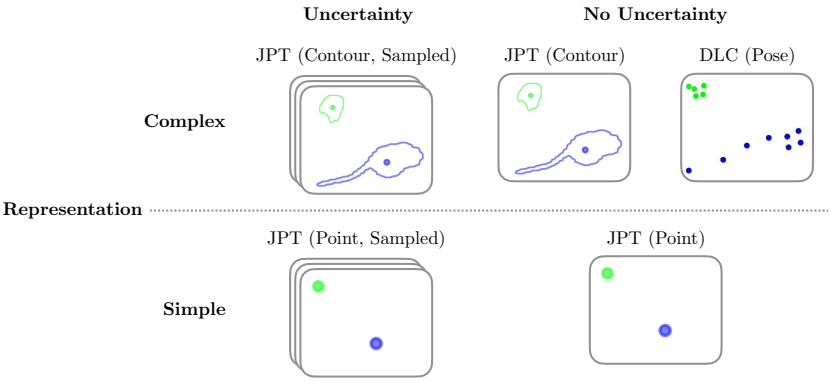


Figure 5-17: Varying multi-object tracking representation and use of uncertainty as inputs to follow-on behavior classification. Top row contains complex tracking representations (contours for JPT, pose for DLC) while bottom row contains simple point trajectories. JPT represents uncertainty (left column) by sampling multiple posterior tracking estimates in either the contour or point representations. For comparison, a single JPT tracking estimate is obtained by drawing one posterior sample. DLC has no uncertainty representation, so it only provides a single tracking estimate. In total, we investigate five conditions: JPT with or without uncertainty in both contour and point representations, and DLC, which uses a pose representation and has no representation of uncertainty. See Figure 5-18 for further visualization of the pose, contour, and point representations.

contours with the JPT (Contour) condition, which draws a single posterior sample of trajectories and associations. Associations are used to recover the original marmoset detection. A contour of the marmoset outline is computed and combined with the inferred centroid to construct contour trajectories. Figure 5-18 visualizes JPT's point and contour representations, and DLC's pose representation.

Finally, JPT samples multiple joint trajectory and association realizations from its posterior. We add two more experiment conditions: JPT (Contour, Sampled) and JPT (Point, Sampled), which are like the JPT (Contour) and JPT (Point) representations, but contain multiple posterior samples. We discuss how multiple samples are used for behavior classification in Chapter 5.5.3.

In summary, the five experiment conditions investigate the effect of tracking representation and presence of uncertainty on supervised behavior classification (Figure 5-17). The five conditions are:

1. JPT (Point) JPT with a point-based tracking representation and no representation of uncertainty.

2. JPT (Contour) JPT with a contour-based tracking representation and no representation of uncertainty.

3. DLC (Pose): Multi-Animal DeepLabCut [117] with a pose-based tracking representation and no representation of uncertainty.

4. JPT (Point, Sample) JPT with a point-based tracking representation and explicit representation of uncertainty.

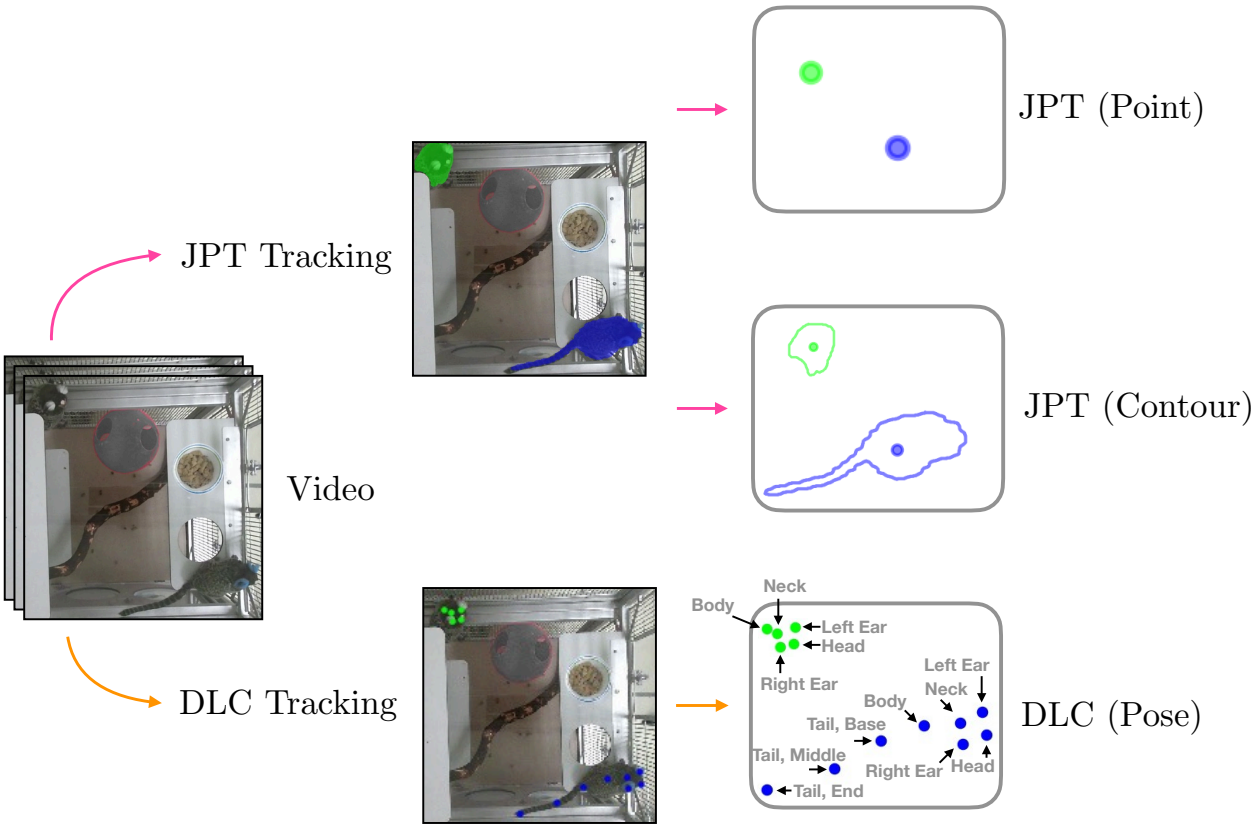


Figure 5-18: Tracking representations used as input to behavior classification. JPT (magenta arrows) reasons over pixel-accurate associations for each tracked object represented as image masks. We construct the JPT (Point) representation as the centroid of each target's mask. We construct the JPT (Contour) representation as the centroid and a collection of evenly-sampled points along the contour, ordered by their angle from the centroid. DLC (orange arrows) reasons over pose estimates represented as a consistent ordering of eight points (Head, Left Ear, Right Ear, Neck, Body, Tail Base, Tail Middle, Tail End). Partial occlusions (green) change the shape of the JPT (Contour) representation and cause missing data for the DLC (Pose) representation. Original images contain a red ball that was desaturated to improve visibility.

5. JPT (Contour, Sample): JPT with contour-based tracking representation and explicit representation of uncertainty.

5.5.2 Experiment Setup

We gather labelled behavior video clips (Figures 5-14, 5-15) from Marmoset100 into training (70%), validation (10%), and test (20%) sets. Each clip contains zero, one, or multiple behaviors so we randomly sample 1000 train/val/test splits and choose the split with least per-behavior deviation from the desired 70/10/20 ratios. Mean variation from the ideal split is less than 0.25% for all behaviors. For behavior video n, behavior label $y_n \in \{0,1\}^{25}$ represents zero, one, or more behaviors that either of two marmosets performed at least once in the clip. In total, there are $N = 6213$ videos, of which 4337 are used for training, 619 for validation, and 1240 for a test set. We investigate behaviors that have at least 50 labelled examples, excluding 5/25 behaviors: Carrying Something, Nuzzle Face, Nuzzle Anus, Head Shake, and Rearing/Standing.

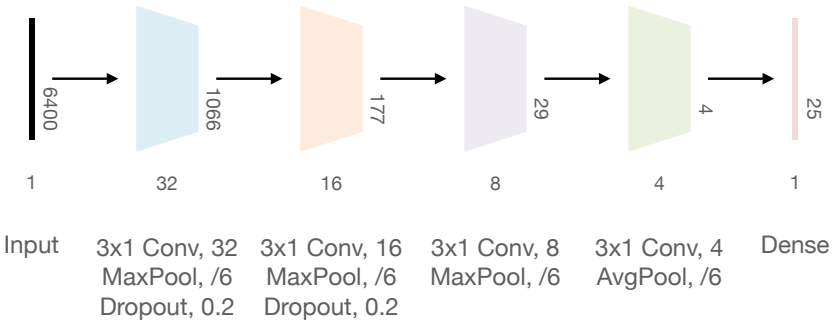


Figure 5-19: Multilabel network used to classify behavior based on tracking representations derived from the Joint Posterior Tracker (Chapter 4.5) and DeepLabCut [134]. Each colored block is a collection of operations labeled with 2D output dimensions. Input is a vector representation of tracked motion in one video. All convolutional layers use RELU activations while the dense layer uses independent sigmoidal activations for each of 25 high-level behaviors.

Tracking for each of the five experiment approaches is performed on all training, validation, and test videos according to Figure 5-17. For each approach, video n is represented by its tracked representation, x_n. Tracking representation x_n is a tensor with dimensions (T_n, K, D) where T_n is the total number of frames in video n, $K = 2$ is the number of marmosets in each video, and D is the representation dimensionality of each tracked object at each time. DLC is trained on an 8-point 2D skeleton of each marmoset (Figure 5-18), represented by consistently-ordered points in $2D$ so that its representation has per-marmoset, per-time dimension of $D = 8 * 2 = 16$. The JPT point representation has a per-marmoset, per-time dimension of $D = 2$ since each object is represented by a single point in image space. The JPT contour representation is the tracked centroid along with the collection of outermost points of each associated marmoset detection. It has variable dimension for each marmoset at each time because the number of pixel associations to each marmoset varies with their visibility, distance to camera, and shape. For fair comparison to DLC, we downsample the JPT (Contour) representation so that it only contains the centroid and 7 contour points.

For each approach, the dataset consists of 6213 pairs (x_n, y_n). Behavior labels y_n are common to each approach but x_n varies with approach. For each approach, behavior prediction is formulated as a multilabel classification problem: one classifier is trained to make 25 independent behavior predictions. All five approaches use the convolutional neural network represented in Figure 5-19. The network is four sets of convolution and pooling layers, two of which employ regularization in the form of dropout. All convolutional layers use RELU activations. The final layer is a dense output of scores $b \in \mathbb{R}^{25}$ with independent sigmoidal activation, one for each of the 25 possible behaviors. In all approaches, we train for 200 epochs with Adam optimization [110] and binary cross-entropy loss.

We transform each tracking representation so that they have con-

sistent input dimensionality (and hence, the same number of behavior classification parameters). We use DLC's 8-point skeleton for reference: JPT's point representation is 0-padded whereas JPT's contour representation is downsampled to a centroid and 7 points on the contour. Track representations are 0-padded or downsampled in the time dimension so that all videos are represented as if they had 200 timesteps. More than 90% of input videos have less than 200 frames. The median video length across the labelled behavior dataset is 71. Each tracking representation has dimension 6400 ($N_t = 200, K = 2, D = 8 * 2$). In all approaches, the multilabel behavior classification network has 2597 trainable parameters, which is less than the number of training samples (4337), to help prevent overfitting.

5.5.3 Exploiting Uncertainty for Behavior Classification

The JPT (Contour, Sampled) and JPT (Point, Sampled) approaches draw $S = 15$ tracking realizations from JPT's posterior over trajectories and associations. For each approach and each video n, we have track realizations $\{x_n^{(s)}\}_{s=1}^{S}$, all sharing the same behavior labels y_n.

Track realizations contain information about ambiguities in the data, such as occur from identity switching: one sample may interpret two objects as crossing while in another, they are interpreted as not crossing. We expect videos that contain behaviors involving objects coming close then diverging to have a higher likelihood of identity switches in tracking estimates than behaviors where objects remain separated. If this is true, a classifier that can train with an awareness of data ambiguities should be able to exploit uncertainty in tracking estimates to perform more effective behavior classification.

In the JPT (Contour, Sampled) and JPT (Point, Sampled) approaches, we provide our classifier with an awareness of uncertainty by using multiple tracking realizations for each video. We augment the original training set,

$$D_{\text{train}} = \{(x_n, y_n)\}_{n=1}^{N_{\text{train}}} \tag{5.21}$$

so that it becomes:

$$D'_{\text{train}} = \{\{(x_n^{(s)}, y_n)\}_{s=1}^{S}\}_{n=1}^{N_{\text{train}}} \tag{5.22}$$

The augmented training set has size $|D'_{\text{train}}| = SN_{\text{train}}$ for S the number of posterior samples we draw from JPT for each video. Training proceeds on the augmented training set as normal.

Inference is similar to training, but has an additional step. Let f represent the neural network in Figure 5-19, trained on D'_{train}. For test video n, independently generate behavior scores for each sample $s = 1, \ldots, S$:

$$b_n^{(1)} = f(x_n^{(1)}) \qquad b_n^{(2)} = f(x_n^{(2)}) \qquad \ldots \qquad b_n^{(S)} = f(x_n^{(S)}) \tag{5.23}$$

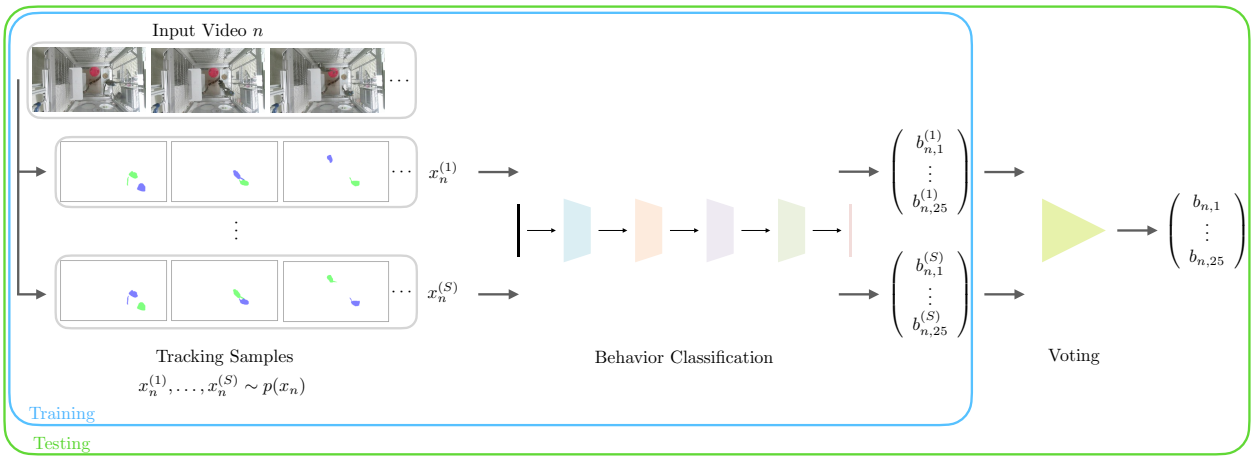


Figure 5-20: Exploiting uncertainty in tracking estimates to improve supervised behavior classification. For video n, multiple tracking samples $x_n^{(1)}, \ldots, x_n^{(S)} \sim p(x_n)$ are drawn from JPT's multi-object tracking posterior (Equation 4.10), which reflects ambiguity in the data. Sample $x_n^{(1)}$ shows that blue and green did not cross whereas sample $x_n^{(S)}$ shows that they did cross. For behavior classification, training and test sets are augmented so that there are S sets of sampled tracking estimates for each video, all sharing the same behavior labels. Training proceeds with the sample-augmented dataset. To classify behaviors at test time, each tracking sample is separately classified and their results are combined by a majority vote.

where,

$$b_n^{(s)} = \left(b_{n,1}^{(s)}, b_{n,2}^{(s)}, \ldots, b_{n,B}^{(s)}\right)^\top \tag{5.24}$$

are the $B = 25$ behavior scores for sample realization s of input video n, where $0 \leq b_{n,i}^{(s)} \leq 1$ for each behavior $i = 1, \ldots, B$. We emphasize our task is multilabel classification; hence, it is possible that,

$$\sum_{i=1}^{B} b_{n,i}^{(s)} > 1 \tag{5.25}$$

since each behavior is independently classified. The S sets of behavior scores for video n are then combined to form a final decision. In this work, we set a common threshold $0 \leq \tau \leq 1$ to generate S binary decisions and combine them by a simple majority vote. Other combination schemes can be used or the S sets of behavior classifications can provide additional uncertainty quantification beyond the probabilistic interpretation of each score $b_{n,i}^{(s)}$.[33] Figure 5-20 visualizes how we use uncertainty as represented by sample realizations in the training and testing phases.

[33] Probabilistic interpretation of neural network classification scores has weak calibration [80]. Perhaps uncertainty quantification using sampled inputs may provide better calibration?

5.5.4 Results

We investigate the effect of track representation and use of uncertainty for multilabel behavior classification as visualized in Figure 5-17. Two approaches quantify uncertainty: JPT (Contour, Sampled) and JPT (Point, Sampled). Three approaches have no representation of uncertainty: DLC (Pose), JPT (Contour), and JPT (Point). We identify key trends in this behavior classification based on representation and show

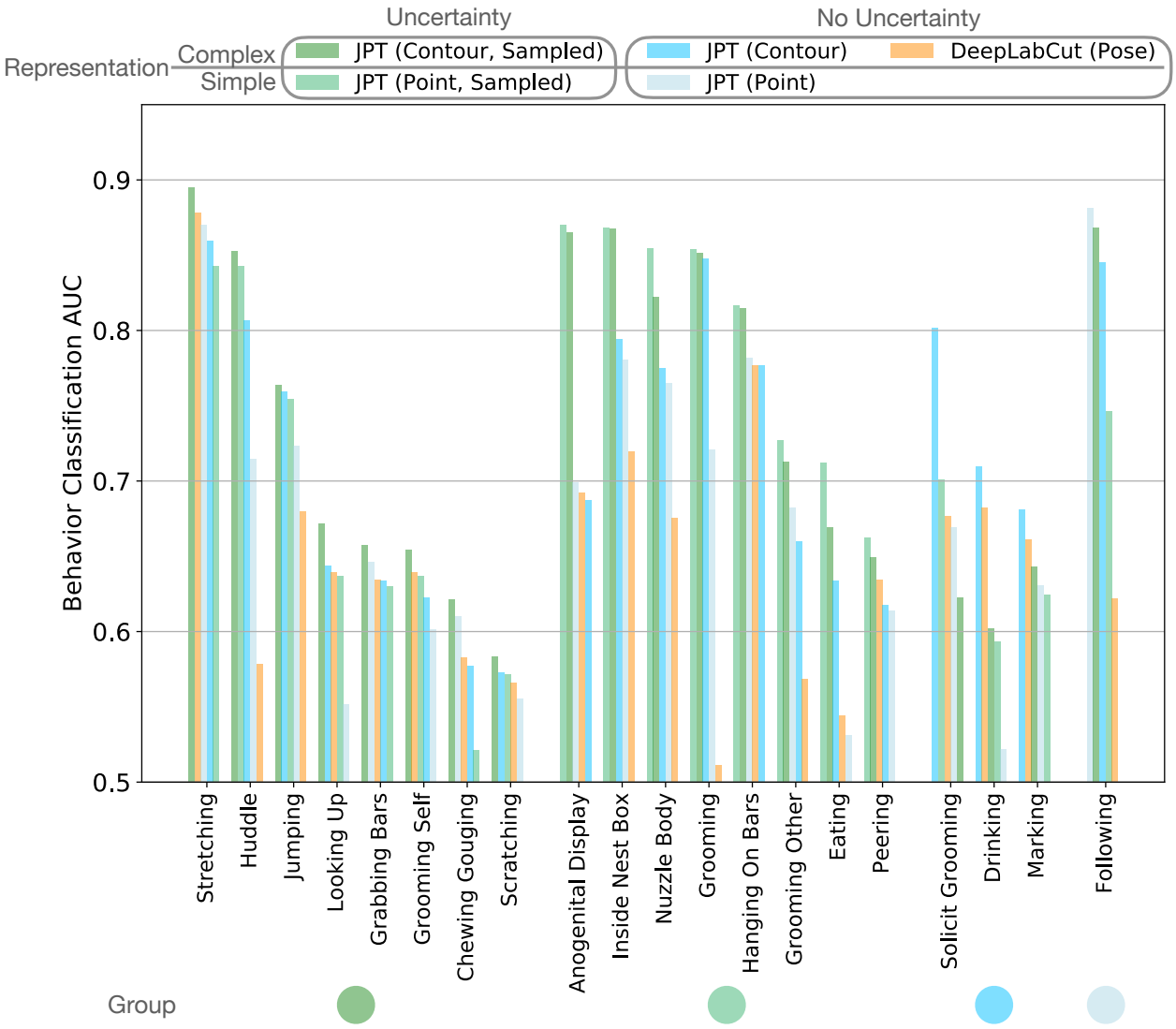


Figure 5-21: Supervised behavior classification performance as measured by AUC based on the presence or absence of uncertainty and use of a complex (pose, contour) or simple (point) representation. JPT samples multiple realizations of tracking estimates to represent uncertainty (green) and uses a single posterior draw (blue) to isolate the benefits of uncertainty. DeepLabCut (orange) only provides a single tracking estimate so it has no representation of uncertainty. JPT with sampled contours (JPT Contour, Sampled) achieves best mean AUC over all behaviors while DeepLabCut yields lowest mean AUC. Behaviors are grouped by approach with best performance. DeepLabCut does not have its own group because it does not have best performance for any behavior. JPT with point representation outperforms DeepLabCut with the more sophisticated pose representation due to superior multi-object tracking. JPT's contour representation typically outperforms JPT's point representation due to the additional information it conveys about object shape. Using uncertainty further increases JPT performance for both point and contour representations.

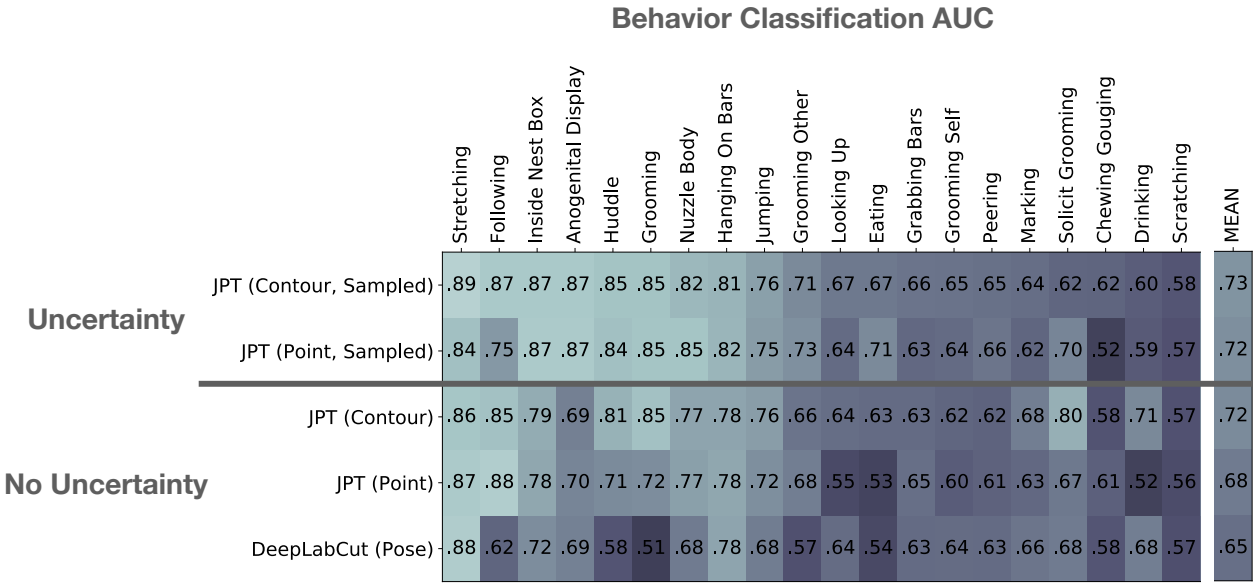


Figure 5-22: Supervised behavior classification performance as measured by AUC based on different tracking methods (JPT, DeepLabCut), representations (DLC: Pose, JPT: Point, Contour), and presence (JPT) or absence (DLC) of uncertainty.

qualitative examples that may help explain why those trends occur.

Figure 5-21 shows multilabel classification performance as measured by AUC for each approach and behavior. Behaviors are grouped by approach with best AUC. Within each group and behavior, approaches are sorted by highest AUC. Figure 5-22 shows the same information as a heatmap where approaches (rows) are sorted by best mean AUC (far-right column) and behaviors (columns) are sorted by JPT (Contour, Sampled) performance.

There should be five groups in Figure 5-21 since there are five approaches but behavior classification using DeepLabCut's tracking estimates fails to achieve best performance for any behavior despite using a sophisticated pose representation. Even JPT point trajectories with no uncertainty representation yield better mean behavior classification AUC (0.68) than DLC pose (0.65). We attribute this to JPT's superior multi-object tracking performance (Figure 5-13), even when their underlying pixel-accurate detections (JPT) or pose estimates (DLC) are trained on the same data. As we investigate behavior performance, we will qualitatively see that DLC has greater numbers of track misses than JPT.

JPT contour representations are more performant on average than JPT point representations when either do or do not represent uncertainty. Figure 5-23 shows JPT (Point) and JPT (Point, Sampled) representations accurately tracking marmosets as they engage in the Huddle behavior. Point representations only capture proximity and so the classifier fails to classify this instance correctly. In contrast, JPT (Contour) and JPT (Contour, Sampled) accurately capture their proximity *and* relative shape configuration; classification based on the contour representations correctly captures this instance of Huddle. DLC (Pose)

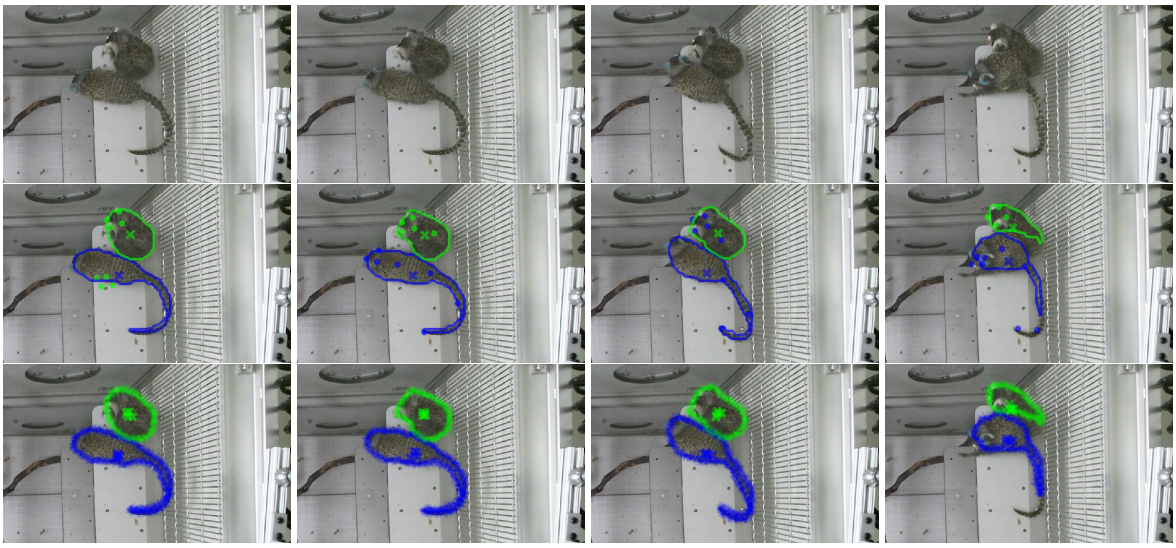


Figure 5-23: Marmosets engaging in the Huddle behavior. JPT contour representations (with or without uncertainty) tend to outperform JPT point representations despite both representations accurately tracking without identity switching. Contour representations capture their relative configuration while point representations only capture proximity. DLC inaccurately splits a single pose estimate across two marmosets (Columns 1,3), confusing a classifier's ability to capture relative configuration. (**Top Row**): Original images. (**Middle Row**): JPT (Point) as an X, JPT (Contour) as an outline, and DLC (Pose) as dots. (**Bottom Row**): Five realizations of JPT (Point, Sampled) and JPT (Contour, Sampled), visualized as in the second row but with Gaussian noise for improved visibility.

misses detections of one or the other marmoset (Columns 1, 3) and incorrectly estimates a single pose that spans the two marmosets. Classification based on DLC (Pose) fails on this instance of Huddle and, in general, underperforms all JPT representations on Huddle.

Uncertainty representation for JPT point or contour representations improves average behavior classification performance. Anogenital Display receives the greatest benefit from uncertainty representation with 0.18 absolute AUC gain as compared to the best approach without uncertainty. These displays often exhibit repeated occurrence of marmosets being in proximity, then pursuing each other, then coming close together again. Figure 5-24 shows this pattern: blue and green marmosets begin close together, then one pursues the other, during which JPT samples become split on whether an identity switch occurred (Columns 2-4). The confusion continues as they become proximate and perform the behavior. Extended confusion in identity occurring from repeated (proximity, chasing, proximity) events may explain why uncertainty improves classification of this behavior.

JPT (Point, Sampled) marginally outperforms JPT (Contour) in terms of average AUC over all behaviors, but significantly in terms of number of behaviors it performs best on: 8 to 3. This suggests that uncertainty representation may provide as much information about behavior as does a more sophisticated tracking representation. Inside Nest Box is one behavior where uncertainty representation of simple, point trajectories outperforms the complex, contour representation without uncertainty. Figure 5-25 shows an example of the Inside Nest Box behav-

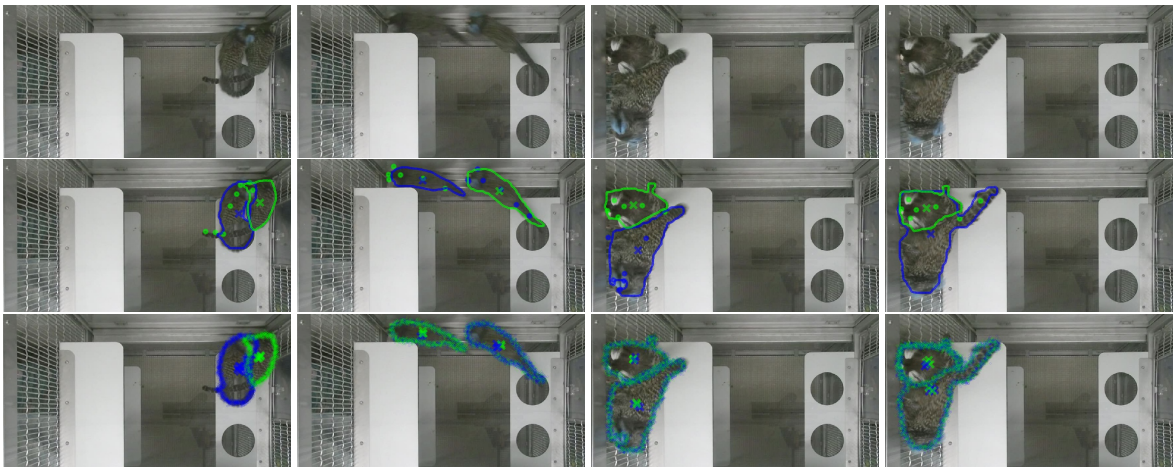


Figure 5-24: Marmosets engaging in the Anogenital Display behavior, which receives the highest benefit from uncertainty representation for JPT point and contour representations. JPT (Point, Sampled) and JPT (Contour, Sampled) determine that marmosets are close enough that they may or may not have switched, as visualized by the conflicting red/green sampled point or contour representations (last row, columns 2–4). (**Top Row**): Original images. (**Middle Row**): JPT (Point) as an X, JPT (Contour) as an outline, and DLC (Pose) as dots. (**Bottom Row**): Five realizations of JPT (Point, Sampled) and JPT (Contour, Sampled), visualized as in the second row but with Gaussian noise for improved visibility.

ior. The JPT (Contour) representation undergoes large and rapid shape variation due to partial occlusion when one marmoset (green) emerges from the nest box. As the green marmoset jumps out, a small percentage of JPT samples capture a potential identity switch (columns 3, 4) due to proximity with the blue marmoset. JPT (Point, Sampled) represents this confusion with simple point trajectories that are not confounded by large shape variations (which also commonly occur when marmosets are proximate, perhaps making it harder for a contour-based classifier to distinguish Inside Nest Box from other proximate social behaviors based on shape alone). In this example, classification based on JPT (Point, Sampled) is correct, but classification based on JPT (Contour) is incorrect. DLC (Pose) fails to estimate any pose for the green marmoset in the first two columns and misses the head, ears, and neck pose estimates (Column 3). Classification based on DLC (Pose) fails on this instance and, in general, underperforms all JPT representations on the Inside Nest Box behavior.

5.6 Related Works

Works related to multi-object tracking and parts representations are surveyed in Chapters 4.4 and 3.7. We highlight additional works that relate to the Nonparametric Extents Model (Chapter 5.3.1) in their use of nonparametric priors for modeling object shape or motion. We then conclude with discussion of animal behavior datasets.

Nonparametric Object Motion and Shape [147] segments foreground into an unknown number of objects with a Dependent Dirich-

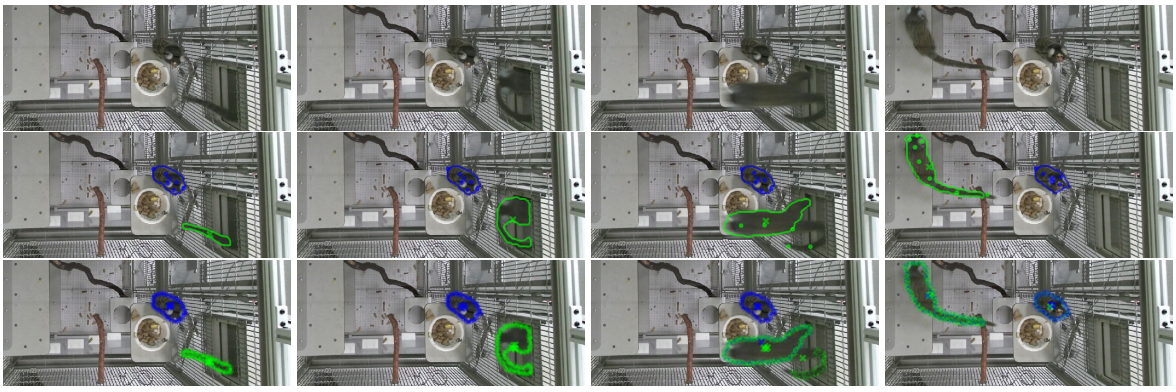


Figure 5-25: DLC (Pose) fails to track the green marmoset (columns 1–2) due to partial occlusion while JPT (Contour) and JPT (Contour, Sampled) experience large variations in shape. JPT (Point) and JPT (Point, Sampled) are robust to these shape changes. One of five JPT samples believes targets crossed (bottom row, columns 3, 4). (**Top Row**): Original images. (**Middle Row**): JPT (Point) as an X, JPT (Contour) as an outline, and DLC (Pose) as dots. (**Bottom Row**): Five realizations of JPT (Point, Sampled) and JPT (Contour, Sampled), visualized as in the second row but with Gaussian noise for improved visibility.

let Process [128] but, unlike NPE, they parametrically model each individual as a single component. Similarly, [81] models each video pixel with a Dirichlet Process mixture to segment multiple foreground objects from background, but does not model multiple components of foreground objects as NPE does. [222] nonparametrically models object shape, but only of a single object in motion. Most relevant is Zanotto et al. [227], a nonparametric model that infers group structure in the motion of multiple objects by modeling input tracking estimates as being generated from an infinite mixture of groups. The groups of Zanotto et al. are similar to the extents of NPE, but crucially differ in that they only generate single observations of individuals. In contrast, NPE generates multiple components per object, and each object component can generate multiple observations.

Animal Datasets We know of no animal behavior datasets containing marmoset behavior. In general, relatively few animal behavior datasets are publicly available. OpenMonkeyStudio [14] uses 62 cameras to infer pose as represented by 13 joint locations on macaque primates. They release a set of 192k synchronized image captures totalling 3.8 GB. 3D ZEF [159] releases eight video sequences containing 1–10 zebrafish. Sequences range from 15–120 seconds in duration. The total dataset contains 86k point or bounding box annotations. CAPTURE [133] releases a 24-hour kinematics dataset that infers the 3D pose of mice using 12 cameras and 20 surgically-implanted retroreflective markers. CAPTURE does not provide raw data, only the results of their analysis. Many animal behavior analyses are surveyed in [7] but few of the referenced works make the data available.

5.7 Conclusion

We develop the the Nonparametric Extents Model (Chapter 5.3), a multi-object tracker that automated motion analysis of free-moving macaques in over 100 hours of experimental macaque data. Tracking results were analyzed by collaborators and contributed to the first experimental evidence for primate animal models of autism (Chapter 5.3.4). In further collaboration, we develop Marmoset100 (Chapter 5.4), a dataset containing 100 hours of marmoset behaviors, including a subset that is labeled by collaborators for 25 high-level behaviors (Chapter 5.4.4). We train a pixel-accurate marmoset detector and perform JPT tracking on all Marmoset100 data (Chapter 5.4.2), facilitating the automatic labeling of more than 9 million frames of video.

We directly compare JPT and DLC tracking on a subset of Marmoset100, and show that JPT significantly outperforms DLC, even when their underlying observation models are trained on the same data (Chapter 5.4.3). We then perform supervised, multilabel behavior classification (Chapter 5.5) based on varying representation (JPT centroid point, JPT object contour, DLC skeletal pose), and presence (JPT) or absence (DLC) of uncertainty in tracking estimates. We show (Chapter 5.5.4) that uncertainty in tracking estimates improves behavior classification performance and that JPT's higher-quality tracking on simpler, point-based representations outperforms behavior classification based on DLC's more complex pose representation. Adding contour information and uncertainty to JPT tracking estimates further improves behavior classification performance.

Chapter 6

Conclusions

"A good scientist has freed himself of concepts and keeps his mind open to what is."

— Laozi

This dissertation contributes Bayesian methods that discover part structure from object motion (Chapter 3), track the motion of multiple objects with explicit uncertainty quantification and reduction (Chapter 4), and conduct behavior analysis in experimental and observational settings at scale (Chapter 5).

Chapter 3 develops the Nonparametric Parts Model (NPP) and demonstrates that NPP's nonparametric representation of kinematic bodies (Chapter 3.4) infers meaningful part decompositions of objects in an unsupervised way by simply observing them in motion (Chapters 3.6.1, 3.6.2). NPP's Lie group representation constrains articulations of moving parts to physically plausible kinematic states without the requirement of object-specific knowledge such as skeletal structures. Part decompositions are learned on short sequences and generalize to other datasets and instances of the same object type (Chapter 3.6.5). In contrast to methods that rely on extensive training data or object-specific 2D/3D models, I demonstrate robust analysis by direct observation of single instances of an object without distinct visual part appearance.

Chapter 4 develops the Joint Posterior Tracker, a Bayesian solution to the batch multi-object tracking problem (Chapter 4.5). I construct efficient inference to reason over permutations of associations (Chapter 4.6) and empirically demonstrate that JPT more effectively represents posterior uncertainty than baselines (Chapter 4.9.4) while outperforming them on standard tracking metrics (Chapter 4.9.5). JPT's accurate representation of uncertainty enables automatic scheduling of informative disambiguations that rapidly drive down posterior uncertainty while improving trajectory quality (Chapters 4.9.6, 4.7).

Chapter 5 develops reliable tracking of pairwise macaque primate interactions in more than 100 hours of data recorded in an experimental setting. Tracking from my Nonparametric Extents Model (NPE) (Chapter 5.3.1) saved scientists from more than 250 hours of manual labeling effort and contributed to the first evidence for primate animal models in autism research (Chapter 5.3.4).

Additional JPT tracking with uncertainty quantification is performed on 100 hours of pairwise marmoset interactions in an observational

setting as part of the development of Marmoset100, a novel dataset on primate interactions (Chapter 5.4). I show that JPT tracking outperforms Multi-Animal DeepLabCut [117, 134] tracking on Marmoset100 data, even when JPT detections and DLC pose estimates use the same training data (Chapter 5.4.3).

Finally, I show (Chapter 5.5.4) that uncertainty in multi-object tracking estimates improves behavior classification performance and that behavior classification based on JPT's higher-quality tracking on simpler, point-based representations outperforms behavior classification based on DLC's more complex pose representation. Adding contour information and uncertainty to JPT tracking estimates further improves behavior classification performance.

Broader Thoughts

My contributions come at a time when machine learning and computer vision are being integrated into the basic approaches of fields that are far removed from computer science. Scientific disciplines are no exception. Many first-order challenges in scientific workflows are being framed as supervised problems: large, labeled datasets are paired with scalable, optimization-based classifiers that have no uncertainty representation to automate data collection and experiment design as well as to improve simulation. These are valuable first steps but it has been my experience that many datasets contain significant ambiguity.

I argue that uncertainty quantification helps make automation trustworthy in scientific approaches by correctly weighting hypotheses. It is also useful for drawing attention to ambiguities that can be resolved before hypotheses are evaluated by appealing to mechanisms beyond the model. I demonstrate automated scheduling of corrections using a noisy human oracle, but the oracle could also be a more expensive, precise, or targeted algorithm or measurement. Modeling disambiguation in the framework of Bayesian experiment design maintains transparency in representation.

Unsupervised approaches have long been used in machine learning, but they are frequently performed on $\mathbb{R}^D$, which can be difficult to interpret when representations are some function of a set of features or weights. I have combined Bayesian nonparametrics with distributions on well-understood manifolds such as the Lie group SE(D) to enable discovery of interpretable structure in objects with articulated motion. Other Lie groups and Riemannian manifolds, such as $\mathcal{P}(D)$ can be used to describe systems of interest in ways that are interpretable. Nonparametric modeling stands to benefit scientific approaches by facilitating discovery of novel structure in these systems.

Combining BNP with distributions on manifolds requires nontrivial user knowledge of two diverse fields. Probabilistic programming languages can, in principal, ease the inference and implementation of these models, but few support BNP and none support general Lie

groups or Riemannian manifolds. Furthermore, current implementations have significant limitations, including limited support for discrete random variables or variable-dimension latent spaces, arcane Lisp-style syntax, and support for either conjugacy or gradient-based methods, not both. In contrast, optimization-based approaches have many packages that automate inference, handle large-scale data, and visualize results. Is it any wonder they have seen widespread adoption?

It is an irony that Bayesian methods have long been described as having automatically-prescribed inference given that they often require great manual effort to implement. This is a barrier to their adoption in other fields. Bayesian methods offer interpretability, uncertainty quantification and reduction, and discovery of structure, but they await a revolution, not just in software that simplifies model definition and posterior inference, but also in follow-on tasks such as using uncertainty to guide decision making.

Appendix A

Proofs and Derivations

A.1 Non-Ergodicity in Linear Gaussian Random Acceleration Models

Consider the full-conditional $p(b \mid c, a)$ on Markov chain $a \to b \to c$

$$p(b \mid c, a) = \frac{p(a, b, c)}{p(a, c)} = \frac{p(c \mid b)p(b \mid a)}{\int_b p(c \mid b)p(b \mid a)} \tag{A.1}$$

Let the conditionals be linear Gaussian with shared dynamics F and covariance Q as is common in linear Gaussian state space models,

$$p(c \mid b) = \mathrm{N}(c \mid Fb, Q) \tag{A.2}$$

$$p(b \mid a) = \mathrm{N}(b \mid Fa, Q) \tag{A.3}$$

$$\Lambda = Q^{-1} \tag{A.4}$$

Then, we can determine the covariance of $p(b \mid c, a) \propto p(c \mid b)p(b \mid a)$ by observing that a, b, c are jointly Gaussian, enabling us to directly read off the covariance of the marginals from the joint covariance:

$$\log p(b \mid c, a) \propto -\frac{1}{2}\left((c - Fb)^\top \Lambda (c - Fb) + (b - Fa)^\top \Lambda (b - Fa)\right) \tag{A.5}$$

$$= -\frac{1}{2}\Big(c^\top \Lambda c - c^\top \Lambda F b - (Fb)^\top \Lambda c + (Fb)^\top \Lambda F b + b^\top \Lambda b - b^\top \Lambda F a - (Fa)^\top \Lambda b + (Fa)^\top \Lambda F a\Big) \tag{A.6}$$

$$= \begin{pmatrix} b \\ c \\ a \end{pmatrix}^\top \underbrace{\begin{pmatrix} F^\top \Lambda F + \Lambda & -F^\top \Lambda & -\Lambda F \\ -\Lambda F & \Lambda & 0 \\ -F^\top \Lambda & 0 & F^\top \Lambda F \end{pmatrix}}_{M} \begin{pmatrix} b \\ c \\ a \end{pmatrix} \tag{A.7}$$

$$= \begin{pmatrix} b \\ d \end{pmatrix}^\top \begin{pmatrix} e & f \\ g & h \end{pmatrix} \begin{pmatrix} b \\ d \end{pmatrix} \tag{A.8}$$

Eqn A.7 collects all terms quadratic in a, b, c into matrix M, which we know to be the inverse covariance of the joint, and Eqn A.8 relabels

the blocks:

$$d = \begin{pmatrix} c \\ a \end{pmatrix} \qquad e = F^\top \Lambda F + \Lambda \qquad f = \begin{pmatrix} -\Lambda F \\ 0 \end{pmatrix}^\top$$
$$g = f^\top \qquad h = \begin{pmatrix} \Lambda & 0 \\ 0 & F^\top \Lambda F \end{pmatrix} \tag{A.9}$$

Using Schur complements and only concerning ourselves with the upper-left entry (the marginal for b):

$$M^{-1} = \begin{pmatrix} e^{-1} + e^{-1} f (h - g e^{-1} f)^{-1} g e^{-1} & \cdots \\ \cdots & \cdots \end{pmatrix} \tag{A.10}$$

Now consider,

$$e^{-1} = (F^\top \Lambda F + \Lambda)^{-1} \tag{A.11}$$
$$= \Lambda^{-1} - \Lambda^{-1} F^\top (\Lambda^{-1} + F \Lambda^{-1} F^\top)^{-1} F \Lambda^{-1} \tag{A.12}$$
$$= Q - Q F^\top (Q + F Q F^\top)^{-1} F Q \tag{A.13}$$

where Eqn A.12 uses the Woodbury Matrix Identity [220]. Now, suppose the Markov chain follows a random acceleration model. That is:

$$Q = \begin{pmatrix} 0 & 0 \\ 0 & q \end{pmatrix} \qquad F = \begin{pmatrix} I & I \\ 0 & I \end{pmatrix} \tag{A.14}$$

where $0, I, q$ are of appropriate (square) dimensions and the state space is organized as position in each dimension followed by velocity in each dimension. Assume $q = \rho I$ for some $\rho > 0$. Then,

$$(Q + F Q F^\top)^{-1} = \begin{pmatrix} 2q^{-1} & -q^{-1} \\ -q^{-1} & q^{-1} \end{pmatrix} \tag{A.15}$$
$$Q F^\top = (F Q)^\top = \begin{pmatrix} 0 & 0 \\ q & q \end{pmatrix} \tag{A.16}$$

Combining Eqns A.13–A.16,

$$e^{-1} = \begin{pmatrix} 0 & 0 \\ 0 & q \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ q & q \end{pmatrix} \begin{pmatrix} 2q^{-1} & -q^{-1} \\ -q^{-1} & q^{-1} \end{pmatrix} \begin{pmatrix} 0 & q \\ 0 & q \end{pmatrix} \tag{A.17}$$
$$= \begin{pmatrix} 0 & 0 \\ 0 & q \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & q \end{pmatrix} \tag{A.18}$$
$$= 0 \tag{A.19}$$

Hence, the covariance for $p(b \mid a, c) = 0$. Intuitively, knowledge of a determines the position of b and knowledge of c determines the velocity of b. A linear dynamical system would typically have noisy observations of each latent state but, in this case, measurements will never provide information about b to the Gibbs sampler because it is already determined by a and c. Thus, a Gibbs sampler for a linear dynamical system of this (very common) form will fail because it is not ergodic;

specifically, it can reach *no* state other than the state it is currently in (such as the state the model was initialized into).

There are two obvious remedies: permit noise on the position terms of the latent state (violating the physical model), or conduct blocked Gibbs sampling on pairwise (or higher) collections of latent states. Blocked Gibbs should be the preferred because small position noise will cause undue posterior certainty and large position noise will move the posterior even further from the desired physical model.

A.2 Nonparametric Parts Full Conditionals

Consider the multivariate Gaussian

$$\mathrm{N}\left(\begin{pmatrix} Cx_1 + u \\ x_2 \end{pmatrix} \mid \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^\top & \Sigma_{22} \end{pmatrix}\right) \tag{A.20}$$

where $x_1, u, \mu_1 \in \mathbb{R}^{D_1}$, $x_2, \mu_2 \in \mathbb{R}^{D_2}$ and covariance $\Sigma \in \mathbb{R}^{D_1+D_2}$ has blocks $\Sigma_{11} \in \mathbb{R}^{D_1 \times D_1}$, $\Sigma_{12} \in \mathbb{R}^{D_1 \times D_2}$, $\Sigma_{21} \in \mathbb{R}^{D_2 \times D_1}$, $\Sigma_{22} \in \mathbb{R}^{D_2 \times D_2}$. Then, because Gaussian conditionals are Gaussian (see [142], Ch. 4), it follows that the conditional $Cx_1 + u \mid x_2$ is Gaussian:

$$Cx_1 + u \mid x_2 \sim \mathrm{N}\left(Cx_1 + u \mid \mu', \Sigma'\right) \tag{A.21}$$

$$\mu' = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}\left(x_2 - \mu_2\right) \tag{A.22}$$

$$\Sigma' = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{A.23}$$

And, by transformation of random variables, the conditional $x_1 \mid x_2$ is Gaussian with parameters:

$$x_1 \mid x_2 \sim \mathrm{N}\left(x_1 \mid \mu'', \Sigma''\right) \tag{A.24}$$

$$\mu'' = C^{-1}\left(\mu' - u\right) \tag{A.25}$$

$$\Sigma'' = C^{-1}\Sigma' C^{-\top} \tag{A.26}$$

A.2.1 Concentrated Gaussian Priors with Gaussian Likelihoods

In this section, we show that Concentrated Gaussian priors on the Lie Group $\mathrm{SE}(D)$ coupled with multivariate Gaussian observation models have Gaussian conditionals for the translation component.

Let $a, b, c, \mu \in \mathrm{SE}(D)$ where each contain a rotation component R and translation component d with notation,

$$a = \begin{pmatrix} R_a & d_a \\ 0 & 1 \end{pmatrix} \qquad b = \begin{pmatrix} R_b & d_b \\ 0 & 1 \end{pmatrix} \tag{A.27}$$

and similarly for c, μ. These can be viewed as linear operators on homogeneous coordinates. Let $y \in \mathbb{R}^D$ be a point and $E \in \mathbb{R}^{D \times D}$ be a covariance matrix. For vector v, let $\tilde{v}$ be the projection of v into homogeneous coordinates (append 1). For covariance Σ, let $\tilde{\Sigma}$ be the projec-

tion of Σ into homogeneous coordinates (append a 0 row and column). Consider the following distribution, for Σ a covariance in the tangent plane about μ:

$$p(b \mid y, a, b) \propto \mathrm{N}_{\mathrm{L}}(b \mid \mu, \Sigma)\, \mathrm{N}\left(\tilde{y} \mid abc\tilde{0}, (abc)\,\tilde{E}\,(abc)^{\top}\right) \tag{A.28}$$

$$= \mathrm{N}\left(\mathrm{Log}_{\mu} b \mid 0, \Sigma\right) \mathrm{N}\left(a^{-1}\tilde{y} \mid bc\tilde{0}, (bc)\,\tilde{E}\,(bc)^{\top}\right) \tag{A.29}$$

$$= \mathrm{N}\left(\log\left(\mu^{-1}b\right) \mid 0, \Sigma\right) \tag{A.30}$$

$$\mathrm{N}\left(a^{-1}\tilde{y} \mid bc\tilde{0}, (bc)\,\tilde{E}\,(bc)^{\top}\right) \tag{A.31}$$

$$= \mathrm{N}\left(\begin{pmatrix} V_{\mu^{-1}b}^{-1} d_{\mu^{-1}b} \\ \phi_{\mu^{-1}b} \end{pmatrix} \mid 0, \Sigma\right) \tag{A.32}$$

$$\mathrm{N}\left(R_a^{\top}(y - d_a) \mid d_b + R_b d_c\,(R_b R_c)\,E\,(R_b R_c)^{\top}\right) \tag{A.33}$$

$$= \mathrm{N}\left(\begin{pmatrix} V_{\mu^{-1}b}^{-1} \left(R_{\mu}^{\top}\left(d_b - d_{\mu}\right)\right) \\ \phi_{\mu^{-1}b} \end{pmatrix} \mid 0, \Sigma\right) \tag{A.34}$$

$$\mathrm{N}\left(R_a^{\top}(y - d_a) \mid d_b + R_b d_c\,(R_b R_c)\,E\,(R_b R_c)^{\top}\right) \tag{A.35}$$

Homogeneous coordinates are used up to Eqn. (A.31), then dropped in Eqn. (A.33). Observe that Eqn. (A.33) is of the form:

$$\mathrm{N}\left(\begin{pmatrix} Cd_b + u \\ \phi \end{pmatrix} \mid \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}\right) \mathrm{N}(z \mid d_b + g, \Lambda) \tag{A.36}$$

where

$$C = V_{\mu^{-1}b}^{-1} R_{\mu}^{\top} \qquad u = -Cd_{\mu} \qquad z = R_a^{\top}(y - d_a) \tag{A.37}$$

$$g = R_b d_c \qquad \Lambda = (R_b R_c)\,E\,(R_b R_c)^{\top} \qquad \phi = \phi_{\mu^{-1}b} \tag{A.38}$$

The conditional $p(d_b \mid R_b, y, a, b)$ is proportional to Eqn. (A.36), which is of the form of Eqn. (A.20), and $C, u, z, g, \Lambda, \phi$ are all computable given R_b, y, a, b (and C is invertible), hence $p(d_b \mid R_b, a, b) = \mathrm{N}(d_b \mid \mu', \Sigma')$ for some μ', Σ'. Then,

$$p(d_b \mid R_b, y, a, b) \propto \mathrm{N}(d_b \mid \mu', \Sigma')\, \mathrm{N}(z \mid d_b + g, \Lambda) \tag{A.39}$$

$$\propto \mathrm{N}(d_b \mid \mu'', \Sigma'') \tag{A.40}$$

where Eqn. (A.40) follows from Eqn. (A.39) because it is a linear Gaussian system, hence is itself proportional to a Gaussian with some mean and covariance μ'', Σ'' (see [142], Ch. 4).

A.2.2 Translation Full Conditionals

In the following, let $x_{t-1}, x_t, x_{t+1}, \{\omega_k, \theta_{(t-1)k}, \theta_{tk}, \theta_{(t+1)k}\}_{k=1}^{K}, I \in \mathrm{SE}(D)$ with rotation and translation components defined similarly to Eqn. (A.27). Let $\{y_{tn}\}_{n=1}^{N_t} \in \mathbb{R}^D$. Let $\{E_k\}_{k=1}^{K} \in \mathbb{R}^{D\times D}$ be observation covariances in $\mathbb{R}^D$ and $Q, W, \{S_k\}_{k=1}^{K}$ be covariances in the Lie algebra $\mathfrak{se}(D)$. Let

$\{z_{tn}\}_{n=1}^{N_t}$ be assignments of observations to one of K instantiated components. The full conditional body frame translation update is of the form:

$$p(d_{x_t} \mid R_{x_t}, x_{t-1}, x_{t+1}, Q, \{\omega_k, \theta_{tk}\}_{k=1}^{K}, \{y_{tn}, z_{tn}\}_{n=1}^{N_t} \tag{A.41}$$

$$\propto \mathrm{N_L}\left(x_t \mid x_{t-1}, Q\right) \mathrm{N_L}\left(x_{t+1} \mid x_t, Q\right) \prod_{n=1}^{N_t} \tag{A.42}$$

$$\mathrm{N}\left(\tilde{y}_{tn} \mid x_t \omega_k \theta_{tk} \tilde{0}, \left(x_t \omega_k \theta_{tk}\right) \tilde{E}_k \left(x_t \omega_k \theta_{tk}\right)^{\top}\right)^{\mathbb{I}(z_{tn}=k)}$$

The full conditional for the k^{th} canonical part translation update is of the form:

$$p(d_{\omega_k} \mid R_{\omega_k}, W_k, \{x_t, \theta_{tk}, \{y_{tn}\}_{n=1}^{N_t}\}_{t=1}^{T}, E_k) \tag{A.43}$$

$$\propto \mathrm{N_L}\left(\omega_k \mid I, W\right) \prod_{t=1}^{T} \prod_{n=1}^{N_t} \tag{A.44}$$

$$\mathrm{N}\left(\tilde{y}_{tn} \mid x_t \omega_k \theta_{tk} \tilde{0}, \left(x_t \omega_k \theta_{tk}\right) \tilde{E}_k \left(x_t \omega_k \theta_{tk}\right)^{\top}\right)^{\mathbb{I}(z_{tn}=k)}$$

In both of the above cases, the concentrated Gaussians have Gaussian conditionals for the translation component, and combine with a product of Gaussian likelihoods, yielding a Gaussian posterior for translations d_{x_t}, d_{ω_k} (per Appendix A.2.1).

Suppose θ_{tk} has dynamics:

$$\theta_{tk} = \begin{pmatrix} R_{\theta_{tk}} & d_{\theta_{tk}} \\ 0 & 1 \end{pmatrix} \tag{A.45}$$

$$= \begin{pmatrix} \mathrm{Exp}_{R_{\theta_{(t-1)k}}} \phi_{tk} & A\, d_{\theta_{(t-1)k}} + B\, m_{tk} \\ 0 & 1 \end{pmatrix}$$

where

$$(m_{tk}, \phi_{tk}) \sim \mathrm{N}\left(0, S_k\right) \tag{A.46}$$

Then $p(d_{\theta_{tk}} \mid R_{\theta_{tk}}, \theta_{(t-1)k})$ is of the form Eqn. (A.20) with $C = B, u = A\, d_{\theta_{(t-1)k}}$. The conditional $p(d_{\theta_{tk}} \mid R_{\theta_{tk}}, \theta_{(t+1)k})$ has a similar Gaussian form. Hence, full conditional translation updates for $d_{\theta_{tk}}$ have a similar structure to the above object and canonical part translation updates and are themselves Gaussian.

A.3 Stabilized Random Walks

The random walk model is one of many models that approximate the dynamics of moving objects in tracking applications [122]. For $x_t \in \mathbb{R}$,

$$x_0 = 0 \tag{A.47}$$

$$x_t = x_{t-1} + q_t \qquad q_t \sim \mathrm{N}(0, \sigma^2) \tag{A.48}$$

Observe that x_t has moments,

$$\begin{aligned}
\mathbb{E}[x_t] &= \mathbb{E}[x_{t-1}] + \mathbb{E}[q_t] && \text{(A.49)}\\
&= 0 && \text{(A.50)}\\
\text{Var}(x_t) &= \text{Var}(x_{t-1}) + \text{Var}(q_t) + 2\text{Cov}(x_{t-1}, q_t) && \text{(A.51)}\\
&= \text{Var}(x_{t-1}) + \sigma^2 + 0 && \text{(A.52)}\\
&= t\sigma^2 && \text{(A.53)}
\end{aligned}$$

so that it evolves about the origin but,

$$\lim_{t\to\infty} \text{Var}(x_t) = \infty \tag{A.54}$$

causing it to wander infinitely far off. We can stabilize this random walk so that it has asymptotically-bounded variance with the following modification, where $a \in \mathbb{R}$,

$$\begin{aligned}
x_0 &= 0 && \text{(A.55)}\\
x_t &= \sqrt{a}\, x_{t-1} + \sqrt{1-a}\, q_t \qquad q_t \sim \text{N}(0, \sigma^2) && \text{(A.56)}
\end{aligned}$$

which has moments:

$$\begin{aligned}
\mathbb{E}[x_t] &= \mathbb{E}[x_{t-1}] + \mathbb{E}[q_t] && \text{(A.57)}\\
&= 0 && \text{(A.58)}\\
\text{Var}(x_t) &= \mathbb{E}[x_t^2] + \mathbb{E}[x_t]^2 && \text{(A.59)}\\
&= \mathbb{E}[x_t^2] + 0 && \text{(A.60)}\\
&= \mathbb{E}[(\sqrt{a}\, x_{t-1} + \sqrt{1-a}\, q_t)^2] && \text{(A.61)}\\
&= a\, \mathbb{E}[x_{t-1}^2] + (1-a)\, \sigma^2 && \text{(A.62)}\\
&= (1-a)\, \sigma^2 \sum_{i=0}^{t-1} a^i && \text{(A.63)}
\end{aligned}$$

Taking the limit, we have,

$$\begin{aligned}
\lim_{t\to\infty} \text{Var}(x_t) &= \lim_{t\to\infty} (1-a)\, \sigma^2 \sum_{i=0}^{t-1} a^i && \text{(A.64)}\\
&= (1-a)\, \sigma^2 \lim_{t\to\infty} \sum_{i=0}^{t-1} a^i && \text{(A.65)}\\
&= (1-a)\, \sigma^2 \frac{1}{1-a} \qquad \text{for } -1 < a < 1 && \text{(A.66)}\\
&= \sigma^2 && \text{(A.67)}
\end{aligned}$$

where Equation A.66 follows because the sum in Equation A.65 is a geometric series, which converges for $-1 < a < 1$. Thus, we can design a stabilized random walk with arbitrary variance by choosing $0 \leq a < 1$. Figure A-1 visualizes unstable and stable random walks for $\sigma = 10.0$.

Stabilized random walks can be extended to arbitrary dimension.

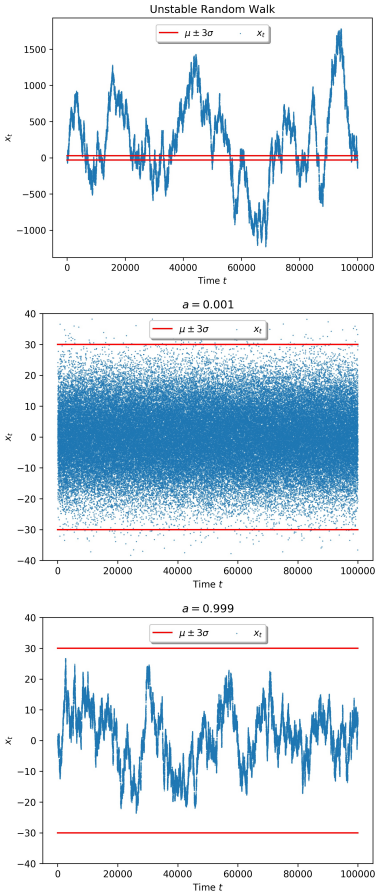


Figure A-1: Random walks. In all cases, $q_t \sim \text{N}(0, \sigma^2)$ for $\sigma = 10.0$.
(**Top**) Unstable
$x_t = x_{t-1} + q_t$.
(**Middle**) Stable ($a = 0.001$)
$x_t = \sqrt{a}\, x_{t-1} + \sqrt{1-a}\, q_t$
(**Bottom**): Stable ($a = 0.999$)
$x_t = \sqrt{a}\, x_{t-1} + \sqrt{1-a}\, q_t$
Observe that the unstable random walk wanders arbitrarily far from the origin whereas the stable random walks generally stay with $\pm 3\sigma$ of the origin. Stabilized random walks with $a \to 0$ increasingly look like the IID draws q_t whereas $a \to 1$ are increasingly smooth.

If $x_t \in \mathbb{R}^D$ then the the above results hold for,

$$x_t = A\, x_{t_1} + B\, q_t \tag{A.68}$$

where $q_t \sim \mathrm{N}(0, \Sigma)$ and:

$$A = \mathrm{diag}\left(\sqrt{a}, \ldots, \sqrt{a}\right) \tag{A.69}$$

$$B = \mathrm{diag}\left(\sqrt{1-a}, \ldots, \sqrt{1-a}\right) \tag{A.70}$$

$$\Sigma = \mathrm{diag}\left(\sigma^2, \ldots, \sigma^2\right) \tag{A.71}$$

which is a special case of the algebraic Ricatti equations and has a solution when A has eigenvalues less than 1 (which the above satisfies).

The parts modeling in Chapter 3.4 adopts a stabilized random walk for parts that orbit about a common body frame of reference so that they do not wander arbitrarily far away. The covariance for their movement is inferred by the model. We found this approach to reduce identity switching in the parts. In particular, early on in inference when parts are not well-fitted, the observation sets would make rapid motions that would be best explained by only small motions of the parts. Without be encouraged to stay proximate to some frame of reference, the model would find it better to explain that the parts frequently "snapped" across one another in fewer large motions so that overall part motion was as small as possible. Stabilizing parts about a frame of reference discouraged this wandering and snapping behavior.

A.4 Switch Inference Generalize Extended HMM Proposals

Switch proposals generalize the Extended HMM (EHMM) proposals of [144] by permitting discretizations that depend on the latent space. In brief, EHMM proposals compose an inference method that helps explore a posterior distribution by proposing a discretization of latent states (called "pool states") over time. A hidden Markov model is then defined over the pool states and a joint sample drawn using forward-filtering, backward sampling. Crucially, the discretization sampled by an EHMM proposal includes the current latent state, but *must not* otherwise depend on it. If it does, detailed balance is lost because calculating the reverse move probability would require a difficult integration over the latent space.

In contrast, Switch proposals sample from a discretization that depends on the current latent state while maintaining detailed balance. In the nomenclature of EHMM proposals, the "pool states" of JPT's Switch proposal are permutations of latent state x, z. JPT then samples from the generative model of an HMM that contains no future information; thus, no backwards pass is required. We note that the Switch proposal always contains the current state as represented by the identity permutation over all times.

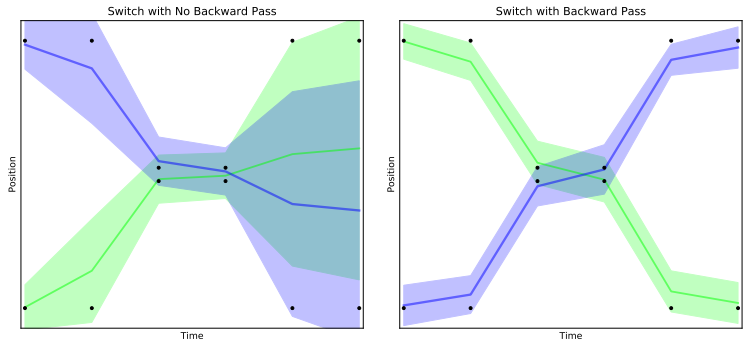


Figure A-2: Observations y (black points) with two modes (blue, green objects crossed or not). Shading indicates marginal posterior trajectory variance. (**Left**): Switch statements that leave no future associations fixed have strong ability to explore modes because future associations do not force an outcome. (**Right**): Switch statements that leave future associations fixed (from the final timestep) will favor the modes supported by those fixed associations.

Switch proposals can easily be constructed to contain future information by restricting switch times τ so that some future associations from the sample set $\mathcal{K}$ remain fixed. Doing so causes a need for future information to propagate backward. We found that doing so without being careful about which associations to leave fixed impairs the ability of Switch proposals to explore different modes as future information encouraged the current sample to remain in the same mode. See Figure A-2 for an example. Backward propagation *is* desirable when future information comes from annotations since they are intended to reduce posterior uncertainty. But, in the absence of annotations, future information in the form of restricted Switch times is not desirable.

A.5 Change of Basis

For basis E, F represented as non-singular matrices, we define $F = E\,{}^{E}T_{F}$ where ${}^{E}T_{F}$ is called the change of basis from E to F. Then, for a point v, we want to find v_E, v_F, the coordinates of v in bases E, F respectively:

$$v = Ev_E = Fv_F = E\,{}^{E}T_{F}v_F$$

Hence, $Ev_E = E\,{}^{E}T_{F}v_F$ and so $v_E = {}^{E}T_{F}v_F$. Note the possibly confusing detail that the matrix representing "the change of basis from E to F" actually takes coordinates from basis F into coordinates in basis E (when the matrix is on the left and the coordinates are on the right, as is common). The manner in which this can be called a change of basis "from E to F" is that we compose basis E with post-multiplication of ${}^{E}T_{F}$ to yield basis F. It changes bases in the manner the wording would suggest, but changes coordinates between bases in the opposite direction than the wording would suggest.

Chapter 3.4 defines a parts model where $x_t, \omega_k, \theta_{tk} \in G = \mathrm{SE}(D)$

for time t, part k. Each can be interpreted as a basis so that,

$$x_t \ \omega_k \ \theta_{tk} \tag{A.72}$$

is a change of basis from world coordinates to part coordinates. For $\epsilon \in \mathbb{R}^D$, the operation,

$$x_t \ \omega_k \ \theta_{tk} \begin{pmatrix} \epsilon \\ 1 \end{pmatrix} \tag{A.73}$$

takes homogeneous vector $\begin{pmatrix} \epsilon \\ 1 \end{pmatrix}$ from part coordinates to world coordinates. Understanding that, "a change of basis from world coordinates to part coordinates," actually takes part coordinates to world coordinates in reverse order significantly aids understanding.

Bibliography

[1] Depth to Alignment Software). https://github.com/dshayden/depth2rgb. Accessed: 2021-05-31.

[2] The FFV1 Video Codec Specification (Development Draft). https://github.com/FFmpeg/FFV1/blob/master/ffv1.md. Accessed: 2021-05-31.

[3] The RGB-Depth-Audio Recorder. https://github.com/dshayden/rgbda_record. Accessed: 2021-05-31.

[4] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[5] Hadi Mohasel Afshar, Justin Domke, et al. Reflection, refraction, and hamiltonian monte carlo. In *NIPS*, pages 3007–3015, 2015.

[6] Ignacio Alvarez, Jarad Niemi, and Matt Simpson. Bayesian inference for a covariance matrix. *arXiv preprint arXiv:1408.4050*, 2014.

[7] David J Anderson and Pietro Perona. Toward a science of computational ethology. *Neuron*, 84(1):18–31, 2014.

[8] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001.

[9] Christophe Andrieu, Gareth O Roberts, et al. The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[10] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72 (3):269–342, 2010.

[11] Anton Andriyenko and Konrad Schindler. Multi-target tracking by continuous energy minimization. In *CVPR*, volume 2, page 7, 2011.

[12] Charles E Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, pages 1152–1174, 1974.

[13] Abdollah Arasteh, Bijan Vosoughi Vahdat, and Reza Salman Yazdi. Multi-target tracking of human spermatozoa in phase-contrast microscopy image sequences using a hybrid dynamic bayesian network. *Scientific reports*, 8(1):1–19, 2018.

[14] Praneet C Bala, Benjamin R Eisenreich, Seng Bum Michael Yoo, Benjamin Y Hayden, Hyun Soo Park, and Jan Zimmermann. Automated markerless pose estimation in freely moving macaques with openmonkeystudio. *Nature communications*, 11(1):1–12, 2020.

[15] Egon Balas and Manfred W Padberg. Set partitioning: A survey. *SIAM review*, 18(4):710–760, 1976.

[16] Robert Stawell Ball. The theory of screws: A study in the dynamics of a rigid body. *Mathematische Annalen*, 9(4):541–553, 1876.

[17] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

[18] Melissa D Bauman and CM Schumann. Advances in nonhuman primate models of autism: Integrating neuroscience and behavior. *Experimental neurology*, 299:252–265, 2018.

[19] Matthew J Beal, Zoubin Ghahramani, and Carl E Rasmussen. The infinite hidden markov model. In *Advances in neural information processing systems*, pages 577–584, 2002.

[20] Juan Carlos Izpisua Belmonte, Edward M Callaway, Sarah J Caddick, Patricia Churchland, Guoping Feng, Gregg E Homanics, Kuo-Fen Lee, David A Leopold, Cory T Miller, Jude F Mitchell, et al. Brains, genes, and primates. *Neuron*, 86(3):617–631, 2015.

[21] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR 2011*, pages 3457–3464. IEEE, 2011.

[22] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1806–1819, 2011.

[23] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[24] José M. Bernardo. Expected information as expected utility. *The Annals of Statistics*, 7(3): 686–690, 1979. ISSN 00905364. URL http://www.jstor.org/stable/2958753.

[25] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[26] Samuel Blackman, Samuel S Blackman, and R Popoli. Design and analysis of modern tracking systems. 1999.

[27] David Blackwell, James B MacQueen, et al. Ferguson distributions via pólya urn schemes. *The annals of statistics*, 1(2):353–355, 1973.

[28] David M Blei, Michael I Jordan, et al. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

[29] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6233–6242, 2017.

[30] Guillaume Bourmaud, Rémi Mégret, Marc Arnaudon, and Audrey Giremus. Continuous-discrete extended kalman filter on matrix lie groups using concentrated gaussian distributions. *Journal of Mathematical Imaging and Vision*, 51(1):209–228, 2015.

[31] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[32] C. Bregler and J. Malik. Tracking people with twists and exponential maps. *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, pages 8–15, 1998. ISSN 1063-6919. doi:10.1109/CVPR.1998.698581. URL http://ieeexplore.ieee.org/document/698581/.

[33] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1820–1833, 2010.

[34] Martin Brossard, Silvere Bonnabel, and Jean-Philippe Condomines. Unscented kalman filtering on lie groups. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2485–2491. IEEE, 2017.

[35] Asad Butt and Robert Collins. Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1846–1853, 2013.

[36] Bradley P Carlin and Siddhartha Chib. Bayesian model choice via markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(3):473–484, 1995.

[37] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, Allen Riddell, et al. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(i01), 2017.

[38] Gilles Celeux, Merrilee Hurn, and Christian P Robert. Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, 95(451):957–970, 2000.

[39] Suman Chakravorty, Weston R Faber, Islam I Hussein, and UR Mishra. A belief space perspective of rfs based multi-target tracking and its relationship to mht. *arXiv preprint arXiv:2001.08803*, 2020.

[40] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

[41] Timothy M Chan. Comparison-based time-space lower bounds for selection. *ACM Transactions on Algorithms (TALG)*, 6(2):1–16, 2010.

[42] Jason Chang and John W Fisher III. Parallel sampling of dp mixture models using sub-cluster splits. In *Advances in Neural Information Processing Systems*, pages 620–628, 2013.

[43] Gregory S Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 1: Classical Results and Geometric Methods*. Springer Science & Business Media, 2009.

[44] Mandar Chitre et al. The multiple hypothesis tracker derived from finite set statistics. In *2017 20th International Conference on Information Fusion (Fusion)*, pages 1–8. IEEE, 2017.

[45] Robert T Collins. Multitarget data association with higher-order motion models. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1744–1751. IEEE, 2012.

[46] Le Cong, F Ann Ran, David Cox, Shuailiang Lin, Robert Barretto, Naomi Habib, Patrick D Hsu, Xuebing Wu, Wenyan Jiang, Luciano A Marraffini, et al. Multiplex genome engineering using crispr/cas systems. *Science*, 339(6121):819–823, 2013.

[47] Georges Darmois. Sur les lois de probabilitéa estimation exhaustive. *CR Acad. Sci. Paris*, 260 (1265):85, 1935.

[48] Sandeep Robert Datta, David J Anderson, Kristin Branson, Pietro Perona, and Andrew Leifer. Computational neuroethology: a call to action. *Neuron*, 104(1):11–24, 2019.

[49] Christopher L Dean, Stephen J Lee, Jason Pacheco, and John W Fisher III. Lightweight data fusion with conjugate mappings. *arXiv preprint arXiv:2011.10607*, 2020.

[50] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[51] T Denoeux. Belief functions for the working scientist, 2015.

[52] Persi Diaconis and David Freedman. Finite exchangeable sequences. *The Annals of Probability*, pages 745–764, 1980.

[53] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[54] David B. Dunson. Bayesian dynamic modeling of latent trait distributions. *Biostatistics*, 7 (4):551–568, 2006. ISSN 14654644. doi:10.1093/biostatistics/kxj025.

[55] Ethan Eade. Lie groups for computer vision. *Cambridge Univ., Cam-bridge, UK, Tech. Rep*, 2014.

[56] Michael David Escobar. *Estimating the means of several normal populations by nonparametric estimation of the distribution of the means*. PhD thesis, Yale University Unpublished dissertation, 1988.

[57] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/.

[58] Michael P Fay and Michael A Proschan. Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics surveys*, 4:1, 2010.

[59] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[60] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[61] James Ferryman. *PETS: Performance Evaluation of Tracking and Surveillance*, 2020. URL http://www.cvg.reading.ac.uk/slides/pets.html.

[62] Martin A Fischler and Robert A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, (1):67–92, 1973.

[63] Nicholas I Fisher. *Statistical analysis of circular data*. cambridge university press, 1995.

[64] Adam Foster, Martin Jankowiak, Eli Bingham, Paul Horsfall, YW Tee, Tom Rainforth, and Noah Goodman. Variational bayesian optimal experimental design. Conference on Neural Information Processing Systems, 2019.

[65] Emily Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, 59(4): 1569–1585, 2011.

[66] Emily B Fox, Erik B Sudderth, Michael I Jordan, Alan S Willsky, et al. A sticky hdp-hmm with application to speaker diarization. *The Annals of Applied Statistics*, 5(2A):1020–1056, 2011.

[67] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015.

[68] Oren Freifeld, Alexander Weiss, Silvia Zuffi, and Michael J Black. Contour people: A parameterized model of 2d articulated human shape. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 639–646. IEEE, 2010.

[69] Jurgen V Gael, Yee W Teh, and Zoubin Ghahramani. The infinite factorial hidden markov model. In *Advances in Neural Information Processing Systems*, pages 1697–1704, 2009.

[70] Dani Gamerman and Hedibert F Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press, 2006.

[71] Hong Ge, Yutian Chen, Moquan Wan, and Zoubin Ghahramani. Distributed inference for dirichlet process mixture models. In *International Conference on Machine Learning*, pages 2276–2284. PMLR, 2015.

[72] Weina Ge and Robert T Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *BMVC*, volume 2. Citeseer, 2008.

[73] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[74] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[75] Samuel Gershman, Matt Hoffman, and David Blei. Nonparametric variational inference. *arXiv preprint arXiv:1206.4665*, 2012.

[76] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[77] Jacob M Graving, Daniel Chae, Hemal Naik, Liang Li, Benjamin Koger, Blair R Costelloe, and Iain D Couzin. Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife*, 8:e47994, 2019.

[78] Peter J Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[79] Peter J Green and Sylvia Richardson. Modelling heterogeneity with and without the dirichlet process. *Scandinavian journal of statistics*, 28(2):355–375, 2001.

[80] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

[81] Tom SF Haines and Tao Xiang. Background subtraction with dirichletprocess mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):670–683, 2013.

[82] Brian Hall. *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 222. Springer, 2015.

[83] Brian C Hall. Lie groups, lie algebras, and representations. In *Quantum Theory for Mathematicians*, pages 333–366. Springer, 2013.

[84] Hamid, Seyed Rezatofighi, Anton Milan, Zhen Zhang, Qinfeng Shi, Anthony Dick, and Ian Reid. Joint probabilistic data association revisited. In *Proceedings of the IEEE international conference on computer vision*, pages 3047–3055, 2015.

[85] WK HASTINGS. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[86] Søren Hauberg, François Lauze, and Kim Steenstrup Pedersen. Unscented kalman filtering on riemannian manifolds. *Journal of Mathematical Imaging and Vision*, 46(1):103–120, 2013. ISSN 09249907. doi:10.1007/s10851-012-0372-9.

[87] David S Hayden, Jason Pacheco, and John W Fisher. Nonparametric object and parts modeling with lie group dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7426–7435, 2020.

[88] David S. Hayden, Sue Zheng, and John W Fisher III. Efficient data association and uncertainty quantification for multi-object tracking. *arXiv preprint*, 2020.

[89] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[90] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[91] Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[92] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5), 2013.

[93] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016.

[94] Hemant Ishwaran and Lancelot F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001. ISSN 1537274X. doi:10.1198/016214501750332758.

[95] Tommi S Jaakkola and Michael I Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25–37, 2000.

[96] Sonia Jain and Radford M Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of computational and Graphical Statistics*, 13(1): 158–182, 2004.

[97] Ajay Jasra, Chris C Holmes, and David A Stephens. Markov chain monte carlo methods and the label switching problem in bayesian mixture modeling. *Statistical Science*, pages 50–67, 2005.

[98] Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.

[99] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007): 453–461, 1946.

[100] Charles G Jennings, Rogier Landman, Yang Zhou, Jitendra Sharma, Julia Hyman, J Anthony Movshon, Zilong Qiu, Angela C Roberts, Anna Wang Roe, Xiaoqin Wang, et al. Opportunities and challenges in modeling human brain disorders in transgenic primates. *Nature neuroscience*, 19(9):1123–1130, 2016.

[101] Yong-hui Jiang and Michael D Ehlers. Modeling autism by shank gene mutations in mice. *Neuron*, 78(1):8–27, 2013.

[102] Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/7d6044e95a16761171b130dcb476a43e-Paper.pdf.

[103] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[104] Shanon X Ju, Michael J Black, and Yaser Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 38–44. IEEE, 1996.

[105] Maria Kalli, Jim E Griffin, and Stephen G Walker. Slice sampling mixture models. *Statistics and computing*, 21(1):93–105, 2011.

[106] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960.

[107] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[108] Zia Khan, Tucker Balch, and Frank Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1805–1819, 2005.

[109] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4696–4704, 2015.

[110] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[111] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[112] Marin Kobilarov, Keenan Crane, and Mathieu Desbrun. Lie group integrators for animation and control of vehicles. *ACM transactions on Graphics (TOG)*, 28(2):1–14, 2009.

[113] Mykel J Kochenderfer and Tim A Wheeler. *Algorithms for optimization*. Mit Press, 2019.

[114] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[115] Bernard Osgood Koopman. On distributions admitting a sufficient statistic. *Transactions of the American Mathematical society*, 39(3):399–409, 1936.

[116] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1): 430–474, 2017.

[117] Jessy Lauer, Mu Zhou, Shaokai Ye, William Menegas, Tanmay Nath, Mohammed Mostafizur Rahman, Valentina Di Santo, Daniel Soberanes, Guoping Feng, Venkatesh N Murthy, et al. Multi-animal pose estimation and tracking with deeplabcut. *bioRxiv*, 2021.

[118] John M Lee. Smooth manifolds. In *Introduction to Smooth Manifolds*, pages 1–31. Springer, 2013.

[119] Bastian Leibe, Konrad Schindler, and Luc Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[120] Daniel A Levitis, William Z Lidicker Jr, and Glenn Freund. Behavioural biologists do not agree on what constitutes behaviour. *Animal behaviour*, 78(1):103–110, 2009.

[121] Chongxuan Li, Max Welling, Jun Zhu, and Bo Zhang. Graphical generative adversarial networks. *arXiv preprint arXiv:1804.03429*, 2018.

[122] X Rong Li and Vesselin P Jilkov. Survey of maneuvering target tracking: dynamic models. In *Signal and Data Processing of Small Targets 2000*, volume 4048, pages 212–235. International Society for Optics and Photonics, 2000.

[123] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[124] Fredrik Lindsten, Michael I Jordan, and Thomas B Schön. Particle gibbs with ancestor sampling. *The Journal of Machine Learning Research*, 15(1):2145–2184, 2014.

[125] Giuseppe Loianno, Michael Watterson, and Vijay Kumar. Visual inertial odometry for quadrotors on SE(3). *Proceedings - IEEE International Conference on Robotics and Automation*, 2016-June(3):1544–1551, 2016. ISSN 10504729. doi:10.1109/ICRA.2016.7487292.

[126] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Bjorn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10955–10964, 2019.

[127] Steven N MacEachern. Estimating normal means with a conjugate style dirichlet process prior. *Communications in Statistics-Simulation and Computation*, 23(3):727–741, 1994.

[128] Steven N MacEachern. Dependent nonparametric processes. In *ASA proceedings of the section on Bayesian statistical science*, volume 1, pages 50–55. Alexandria, Virginia. Virginia: American Statistical Association; 1999, 1999.

[129] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[130] Jesus E Madrid, Tara M Mandalaywala, Sean P Coyne, Jamie Ahloy-Dallaire, Joseph P Garner, Christina S Barr, Dario Maestripieri, and Karen J Parker. Adaptive developmental plasticity in rhesus macaques: the serotonin transporter gene interacts with maternal care to affect juvenile social behaviour. *Proceedings of the Royal Society B: Biological Sciences*, 285 (1881):20180541, 2018.

[131] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5442–5451, 2019.

[132] Santiago Manen, Michael Gygli, Dengxin Dai, and Luc Van Gool. Pathtrack: Fast trajectory annotation with path supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 290–299, 2017.

[133] Jesse D Marshall, Diego E Aldarondo, Timothy W Dunn, William L Wang, Gordon J Berman, and Bence P Ölveczky. Continuous whole-body 3d kinematic recordings across the rodent behavioral repertoire. *Neuron*, 109(3):420–437, 2021.

[134] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289, 2018.

[135] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[136] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[137] Jeffrey W Miller and Matthew T Harrison. Mixture models with a prior on the number of components. *Journal of the American Statistical Association*, 113(521):340–356, 2018.

[138] Rainald Moessner, Christian R Marshall, James S Sutcliffe, Jennifer Skaug, Dalila Pinto, John Vincent, Lonnie Zwaigenbaum, Bridget Fernandez, Wendy Roberts, Peter Szatmari, et al. Contribution of shank3 mutations to autism spectrum disorder. *The American Journal of Human Genetics*, 81(6):1289–1297, 2007.

[139] Pierre Monteiller, Sebastian Claici, Edward Chien, Farzaneh Mirzazadeh, Justin M Solomon, and Mikhail Yurochkin. Alleviating label switching with optimal transport. In *NeurIPS*, 2019.

[140] Charles Morefield. Application of 0-1 integer programming to multitarget tracking problems. *IEEE Transactions on Automatic Control*, 22(3):302–312, 1977.

[141] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[142] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[143] Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

[144] Radford M Neal, Matthew J Beal, and Sam T Roweis. Inferring state sequences for non-linear systems with embedded hidden markov models. In *Advances in neural information processing systems*, pages 401–408, 2004.

[145] Radford M Neal et al. Slice sampling. *The annals of statistics*, 31(3):705–767, 2003.

[146] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[147] Willie Neiswanger, Frank Wood, and Eric Xing. The dependent dirichlet process mixture of objects for detection-free tracking and object modeling. In *Artificial Intelligence and Statistics*, pages 660–668. PMLR, 2014.

[148] Peter Neri, M Concetta Morrone, and David C Burr. Seeing biological motion. *Nature*, 395 (6705):894–896, 1998.

[149] Juan Nieto, Jose Guivant, Eduardo Nebot, and Sebastian Thrun. Real time data association for fastslam. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 1, pages 412–418. IEEE, 2003.

[150] Akihiko Nishimura, David B Dunson, and Jianfeng Lu. Discontinuous hamiltonian monte carlo for discrete parameters and discontinuous likelihoods. *Biometrika*, 107(2):365–380, 2020.

[151] Yuyu Niu, Bin Shen, Yiqiang Cui, Yongchang Chen, Jianying Wang, Lei Wang, Yu Kang, Xiaoyang Zhao, Wei Si, Wei Li, et al. Generation of gene-modified cynomolgus monkey via cas9/rna-mediated gene targeting in one-cell embryos. *Cell*, 156(4):836–843, 2014.

[152] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, pages 3153–3160. IEEE, 2011.

[153] Songhwai Oh, Stuart Russell, and Shankar Sastry. Markov chain monte carlo data association for multi-target tracking. *IEEE Transactions on Automatic Control*, 54(3):481–497, 2009.

[154] Jason Pacheco and John Fisher. Variational information planning for sequential decision making. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2028–2036. PMLR, 2019.

[155] Dimitri J Papageorgiou and Michael R Salpukas. The maximum weight independent set problem for data association in multiple hypothesis tracking. In *Optimization and Cooperative Control Strategies*, pages 235–255. Springer, 2009.

[156] Omiros Papaspiliopoulos and Gareth O Roberts. Retrospective markov chain monte carlo methods for dirichlet process hierarchical models. *Biometrika*, 95(1):169–186, 2008.

[157] Frank C. Park. Distance Metrics on the Rigid-Body Motions with Applications to Mechanism Design. *Journal of Mechanical Design*, 117(1):48–54, 1995.

[158] Eduardo L Pasiliao. Local neighborhoods for the multidimensional assignment problem. In *Dynamics of information systems*, pages 353–371. Springer, 2010.

[159] Malte Pedersen, Joakim Bruslund Haurum, Stefan Hein Bengtson, and Thomas B Moeslund. 3d-zef: A 3d zebrafish tracking benchmark dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2426–2436, 2020.

[160] Talmo D Pereira, Diego E Aldarondo, Lindsay Willmore, Mikhail Kislin, Samuel S-H Wang, Mala Murthy, and Joshua W Shaevitz. Fast animal pose estimation using deep neural networks. *Nature methods*, 16(1):117–125, 2019.

[161] Katy Phelan and HE McDermid. The 22q13. 3 deletion syndrome (phelan-mcdermid syndrome). *Molecular syndromology*, 2(3-5):186–201, 2011.

[162] Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1201–1208, 2011. ISSN 10636919. doi:10.1109/CVPR.2011.5995604.

[163] Edwin James George Pitman. Sufficient statistics and intrinsic accuracy. In *Mathematical Proceedings of the cambridge Philosophical society*, volume 32, pages 567–579. Cambridge University Press, 1936.

[164] Jim Pitman. Some developments of the blackwell-macqueen urn scheme. *Lecture Notes-Monograph Series*, pages 245–267, 1996.

[165] Jim Pitman et al. Combinatorial stochastic processes. Technical report, Technical Report 621, Dept. Statistics, UC Berkeley, 2002. Lecture notes for . . . , 2002.

[166] George Pólya. *Mathematics and plausible reasoning: Induction and analogy in mathematics*, volume 1. Princeton University Press, 1954.

[167] Gerard Pons-Moll, Andreas Baak, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bodo Rosenhahn. Multisensor-fusion for 3d full-body human motion capture. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 663–670. IEEE, 2010.

[168] Aubrey B Poore. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. *Computational Optimization and Applications*, 3(1):27–57, 1994.

[169] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[170] Herbert E Rauch, F Tung, and Charlotte T Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

[171] Lu Ren, David B. Dunson, and Lawrence Carin. The dynamic hierarchical Dirichlet process. *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 824–831, 2008. doi:10.1145/1390156.1390260. URL http://portal.acm.org/citation.cfm?doid=1390156.1390260.

[172] Christian Robert and George Casella. A short history of markov chain monte carlo: Subjective recollections from incomplete data. *Statistical Science*, pages 102–115, 2011.

[173] Lionel Roques and Olivier Bonnefon. Modelling population dynamics in realistic landscapes with linear elements: A mechanistic-statistical reaction-diffusion approach. *PloS one*, 11(3): e0151217, 2016.

[174] David A Ross, Daniel Tarlow, and Richard S Zemel. Unsupervised learning of skeletons from motion. In *European Conference on Computer Vision*, pages 560–573. Springer, 2008.

[175] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[176] John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016.

[177] S Sastry, D Culler, M Howard, T Roosta, B Zhu, J Taneja, Sukun Kim, S Schaffert, J Hui, P Dutta, et al. Instrumenting wireless sensor networks for real-time surveillance. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3128–3133. IEEE, 2006.

[178] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE transactions on signal processing*, 56(8):3447–3457, 2008.

[179] Gideon Schwarz et al. Estimating the dimension of a model. *Annals of statistics*, 6(2):461–464, 1978.

[180] Valentina Sclafani, Laura A Del Rosso, Shannon K Seil, Laura A Calonder, Jesus E Madrid, Kyle J Bone, Elliott H Sherr, Joseph P Garner, John P Capitanio, and Karen J Parker. Early predictors of impaired social functioning in male rhesus macaques (macaca mulatta). *PLoS One*, 11(10):e0165401, 2016.

[181] Aleksandr V Segal and Ian Reid. Latent data association: Bayesian model selection for multi-target tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2904–2911, 2013.

[182] Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.

[183] Glenn Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

[184] Shuo Shang, Lisi Chen, Zhewei Wei, Christian S Jensen, Kai Zheng, and Panos Kalnis. Trajectory similarity join in spatial networks. 2017.

[185] J Shotton, A Fitzgibbon, M Cook, T Sharp, M Finocchio, R Moore, A Kipman, and A Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304. IEEE Computer Society, 2011.

[186] Danielle Simmons. The use of animal models in studying genetic disease: transgenesis and induced mutation. *Nature education*, 1(1):70, 2008.

[187] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[188] Justin Solomon. Optimal transport on discrete domains. *AMS Short Course on Discrete Differential Geometry*, 2018.

[189] Julian Straub, Jason Chang, Oren Freifeld, and John Fisher III. A dirichlet process mixture model for spherical data. In *Artificial Intelligence and Statistics*, pages 930–938, 2015.

[190] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal*, pages 1–30, 2019.

[191] Erik B Sudderth, Antonio Torralba, William T Freeman, and Alan S Willsky. Describing visual scenes using transformed dirichlet processes. In *Advances in neural information processing systems*, pages 1297–1304, 2006.

[192] Yee Whye Teh. Dirichlet process.

[193] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006. ISSN 01621459. doi:10.1198/016214506000000302.

[194] Niko Tinbergen. On aims and methods of ethology. *Zeitschrift für tierpsychologie*, 20(4): 410–433, 1963.

[195] Nikolaas Tinbergen. *The study of instinct*. Pygmalion Press, an imprint of Plunkett Lake Press, 1951.

[196] Dustin Tran, Matthew D. Hoffman, Rif A. Saurous, Eugene Brevdo, Kevin Murphy, and David M. Blei. Deep probabilistic programming. In *International Conference on Learning Representations*, 2017.

[197] Myron Tribus. *Rational Descriptions, Decisions and Designs: Pergamon Unified Engineering Series*. Pergamon Press, 1969.

[198] George Tucker, Andriy Mnih, Chris J Maddison, Dieterich Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *arXiv preprint arXiv:1703.07370*, 2017.

[199] Ryan D Turner, Steven Bottone, and Bhargav Avasarala. A complete variational tracker. In *Advances in Neural Information Processing Systems*, pages 496–504, 2014.

[200] Caroline Uhler, Alex Lenkoski, and Donald Richards. Exact formulas for the normalizing constants of wishart distributions for graphical models. *arXiv preprint arXiv:1406.4901*, 2014.

[201] Jeffrey K Uhlmann. Algorithms for multiple-target tracking. *American Scientist*, 80(2):128–141, 1992.

[202] Isabel Valera, Francisco Ruiz, Lennart Svensson, and Fernando Perez-Cruz. Infinite factorial dynamical model. In *Advances in Neural Information Processing Systems*, pages 1666–1674, 2015.

[203] Jurgen Van Gael, Yunus Saatci, Yee Whye Teh, and Zoubin Ghahramani. Beam sampling for the infinite hidden markov model. In *Proceedings of the 25th international conference on Machine learning*, pages 1088–1095, 2008.

[204] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics (TOG)*, volume 27, page 97. ACM, 2008.

[205] Ba-Ngu Vo and Ba-Tuong Vo. A multi-scan labeled random finite set model for multi-object state estimation. *IEEE Transactions on Signal Processing*, 67(19):4948–4963, 2019.

[206] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7942–7951, 2019.

[207] Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation: A set of best practices for high quality, economical video labeling. *International Journal of Computer Vision*, 101(1):184–204, 2013. ISSN 09205691. doi:10.1007/s11263-012-0564-1.

[208] Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

[209] Stephen G Walker. Sampling the dirichlet mixture model with slices. *Communications in Statistics—Simulation and Computation®*, 36(1):45–54, 2007.

[210] Peter Walley. Statistical reasoning with imprecise probabilities. 1991.

[211] E.a. A Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. *Technology*, v:153–158, 2000. ISSN 15270297. doi:10.1109/ASSPCC.2000.882463. URL `http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=882463`.

[212] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang. Tracklet association with online target-specific metric learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1234–1241, 2014.

[213] Yunfeng Wang and Gregory S Chirikjian. Error propagation on the euclidean group with applications to manipulator kinematics. *IEEE Transactions on Robotics*, 22(4):591–602, 2006.

[214] Mike West and Michael D Escobar. *Hierarchical priors and mixture models, with application in regression and density estimation.* Institute of Statistics and Decision Sciences, Duke University, 1993.

[215] Jason L Williams. Marginal multi-bernoulli filters: Rfs derivation of mht, jipda, and association-based member. *IEEE Transactions on Aerospace and Electronic Systems*, 51(3): 1664–1687, 2015.

[216] Jason L Williams and Roslyn A Lau. Data association by loopy belief propagation. In *2010 13th International Conference on Information Fusion*, pages 1–8. IEEE, 2010.

[217] Alexander B Wiltschko, Matthew J Johnson, Giuliano Iurilli, Ralph E Peterson, Jesse M Katon, Stan L Pashkovski, Victoria E Abraira, Ryan P Adams, and Sandeep Robert Datta. Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 2015.

[218] John Wishart. The generalised product moment distribution in samples from a normal multivariate population. *Biometrika*, pages 32–52, 1928.

[219] Frank Wood, Jan Willem Meent, and Vikash Mansinghka. A new approach to probabilistic programming inference. In *Artificial Intelligence and Statistics*, pages 1024–1032. PMLR, 2014.

[220] Max A Woodbury. *Inverting modified matrices.* Statistical Research Group, 1950.

[221] Max A Woodbury. A missing information principle: theory and applications. Technical report, Duke University Medical Center Durham United States, 1970.

[222] Kevin Wyffels and Mark Campbell. Joint tracking and non-parametric shape estimation of arbitrary extended objects. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3360–3367. IEEE, 2015.

[223] Zhenjia Xu, Zhijian Liu, Chen Sun, Kevin Murphy, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Unsupervised discovery of parts, structure, and dynamics. *International Conference on Learning and Representation*, 2019.

[224] Kexin Yi and Finale Doshi-Velez. Roll-back hamiltonian monte carlo. *arXiv preprint arXiv:1709.02855*, 2017.

[225] Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28, 1978.

[226] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *European Conference on Computer Vision*, pages 343–356. Springer, 2012.

[227] Matteo Zanotto, Loris Bazzani, Marco Cristani, and Vittorio Murino. Online bayesian non-parametrics for group detection. In *Proc. of BMVC*, 2012.

[228] Milos Zefran, Vijay Kumar, and Christopher Croke. Choice of Riemannian Metrics for Rigid Body Kinematics. *ASME Design Engineering Technical Conference and Computers in Engineering Conference*, (3):1–11, 1996.

[229] Jianwen Zhang, Yangqiu Song, Changshui Zhang, and Shixia Liu. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*, page 1079, 2010. ISSN 1015-9770. doi:10.1145/1835804.1835940. URL http://dl.acm.org/citation.cfm?doid=1835804.1835940.

[230] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[231] Sue Zheng, David Hayden, Jason Pacheco, and John W Fisher III. Sequential bayesian experimental design with variable cost structure. *Advances in Neural Information Processing Systems*, 33, 2020.

[232] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[233] Yang Zhou, Jitendra Sharma, Qiong Ke, Rogier Landman, Jingli Yuan, Hong Chen, David S Hayden, John W Fisher, Minqing Jiang, William Menegas, et al. Atypical behaviour and connectivity in shank3-mutant macaques. *Nature*, 570(7761):326–331, 2019.

[234] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6365–6373, 2017.

[235] Silvia Zuffi, Angjoo Kanazawa, and Michael J Black. Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3955–3963, 2018.