Learning to Reconstruct 3D Human Motion from Bayesian Mixtures of Experts. A Probabilistic Discriminative Approach

Cristian Sminchisescu[1] **Atul Kanaujia**[2] **Zhiguo Li**[2] **Dimitris Metaxas**[2]
[1]Department of Computer Science, University of Toronto, Canada
crismin@cs.toronto.edu, http://www.cs.toronto.edu/~crismin
[2]Department of Computer Science, Rutgers University, USA
{*kanaujia,zhli,dnm*}*@cs.rutgers.edu, http://www.cs.rutgers.edu/~kanaujia,zhli,dnm*

Abstract

We describe a mixture density propagation algorithm to estimate 3D human motion in monocular video sequences, based on observations encoding the appearance of image silhouettes. Our approach is discriminative *rather than generative, therefore it does not require the probabilistic inversion of a predictive observation model. Instead, it uses a large human motion capture data-base and a 3D computer graphics human model, to synthesize training pairs of typical human configurations, together with their realistically rendered 2D silhouettes. These are used to directly* learn *the conditional state distributions required for 3D body pose tracking, and thus avoid using the 3D model for* inference *(the learned distributions obtained using a discriminative approach can also be used, complementary, as importance samplers, in order to improve mixing or initialize generative inference algorithms). We aim for probabilistically motivated tracking algorithms and for models that can estimate complex multivalued mappings common in inverse, uncertain perception inferences. Our paper has three contributions: (1) we clarify the assumptions and derive the density propagation rules for* discriminative inference *in continuous, temporal chain models; (2) we propose flexible representations and algorithms for* learning *multimodal conditional state distributions, based on compact Bayesian mixture of experts models; and (3) we demonstrate our algorithms by presenting empirical results on real and motion capture-based test sequences and by comparing against nearest-neighbor and regression methods.*

Keywords: *density propagation, mixture modeling, hierarchical mixture of experts, 3D human tracking, Bayesian methods, sparse regression.*

1 Introduction and Motivation

We consider the problem of tracking and reconstructing (inferring) 3D articulated human motion in monocular video sequences. This is a challenging research topic with a broad set of applications for scene understanding, but our argument applies generally to temporal estimation problems. Approaches to tracking and modeling can be classified as *generative* and *discriminative*. They are similar in that both require a state representation, here a 3D human model with kinematics (*e.g.* joint angles) and/or shape (*e.g.* surfaces or joint positions) and they both use a set of image features as observations for state inference. Their computational goal is also common: the conditional distribution (or a point estimate) for the model state, given image observations.

Generative algorithms require a constructive form of the the observer (the observation likelihood or cost function) and explicitly use the 3D model for inference. This process is complex and searches the state space in order to locate the peaks of the likelihood (*e.g.* using non-linear optimization or sampling). Then Bayes' rule is used to compute the model state conditional from the observation conditional and the state prior. Learning in these frameworks can be both unsupervised and supervised. This includes priors on the state [14, 21, 47], dimensionality reduction [10, 61, 63, 50] or learning the hyperparameters of the observation model (*e.g.* texture and color, ridge or edge distributions using problem-dependent, natural image statistics, *etc.*) [20, 46, 43]. Temporal inference (tracking) is framed in a clear probabilistic and computational framework, *e.g.* mixture or particle filters and beyond [24, 47, 14, 54, 23, 56].

It has been argued that generative models can flexibly reconstruct complex unknown motions and can naturally handle problem constraints. It has been counter-argued that both flexibility and modeling difficulties lead to expensive, highly-uncertain inferences [14, 47, 54, 51], and that a constructive form of the observer is somewhat indirect with respect to the problem at hand, which requires conditional state estimation and not conditional observation modeling.

These arguments motivate the complementary study of **discriminative algorithms** [9, 35, 45, 42, 3, 2, 16], that aim to estimate the state conditional directly in order to simplify inference. For this purpose, they work supervised and use a set of examples (samples), $\mathcal{T} = \{(\mathbf{r}_i, \mathbf{x}_i) \mid i = 1 \dots N\}$, from the *joint distribution* of typical 3D human configurations $\mathbf{x}$ paired with their 2D image appearance (*i.e.* observations) $\mathbf{r}$, focusing on modeling only this 'relevant' data distribution. Inference, on the other hand, involves missing data, unlike learning that is supervised. But learning is also difficult, because modeling perceptual data often produces highly multimodal distributions. [47, 51, 50].[1] While this implies that, strictly, the inverse mapping from observations to states is multi-valued and cannot be functionally approximated, several methods aimed to do so [45, 5, 35, 60, 3, 2]. Some authors constructed data structures for fast nearest-neighbor retrieval [45, 5, 60, 35] or learned regression parameters [3, 2, 16]. Inference involved either indexing for the nearest-neighbors of the observation and using their state for locally weighted predictions, direct prediction using the learned regressor parameters [3, 2, 16], or affine reconstruction from joint centers [33, 57, 35].

Among discriminative methods, a notable exception is [42], who clustered their dataset into soft partitions and learned functional approximations (*e.g.* perceptrons or regressors) within each. However, clusterwise functional approximation [40, 13, 42] is only going halfway towards a multivalued inversion, because inference is not straightforward. The problem is that the model represents the joint distribution and not the conditional. Therefore, for new inputs, cluster / perceptron membership probabilities cannot be computed as during (supervised) learning, because the state is missing. The learned mixture coefficients are not useful either because they are (the fixed) averages over the training set. Therefore it is not clear what approximator or set of approximators to use for any new observation. Various post-hoc strategies based on finding input cluster neighbors may be used, but these fall out of the estimated model that is not optimized to consistently compute such queries. On the other hand averaging across different cluster predictors can give poor results (see fig. 2 for a discussion). Nevertheless, clusterwise regression [40, 13, 42] is useful as a proposal mechanism, *e.g.* during generative inference based on quadrature-style Monte-Carlo approximations and indeed this is how it has been primarily used [42]. A related method has been proposed by [31], where a mixture of probabilistic PCA is fitted to the joint distribution represented as silhouette features in multiple views paired with their 3D pose. Reconstruction is based on MAP estimates. In this

[1] This reflects the structure of the problem and not a particular modeling. *E.g.* think of conversations observed from a side, where gestures pointing towards or away from the camera are common. Humans can initiate a large variety of motions starting from passive (*e.g.* stand-up) positions. Many state trajectories will intersect and produce ambiguity in such regions.

imaging setting the state conditional could be unimodal, but missing data makes inference (*i.e.* conditional computation) non-trivial, demanding in principle, an application of Bayes' rule and marginalization (see our §2.2.3).

To summarize, it has been argued that discriminative models can provide fast inference and can interpolate flexibly in the trained region. But they can fail on novel inputs, especially if trained using small datasets. Increasing the training set or the complexity of motion inevitably leads to multimodal state conditionals (see also our §3). But learning such distributions is difficult and most exiting methods [45, 60, 31, 3, 2, 16] are unimodal. Finally, discriminative methods lack a clear probabilistic temporal estimation framework that has been so fruitful with generative models [24, 14, 46, 54]. Existing tracking algorithms [41, 60, 3, 2, 16] involve per-frame state inference, possibly using estimates at previous timesteps [60, 3, 2, 16], but do not rely on a proved set of independence assumptions or propagation rules. What distributions should be modeled and how should they be combined for optimal solutions? This problem is non-trivial and the answer has important implications for the correctness of tracking results.

The work we present has **three contributions:**

1. We propose a probabilistic framework and derive the density propagation rules in discriminative, continuous, temporally chained models. There exist, of course, belief propagation algorithms [39, 23, 56] that in principle apply to any graphical model. However, they haven't been used in a discriminative tracking framework and we are not aware of prior work that derived conditionals that are relevant for this problem. Also, differently from [23, 56], we work parametrically to estimate and propagate mixtures.

2. We describe Bayesian conditional mixture of experts[2] representations that allows flexible discriminative modeling. Our algorithms are based on hierarchical [25, 29, 64, 7] and joint mixture of experts [66, 62], which are elaborated versions of clusterwise or switching regression [40, 13, 42], where the expert mixture proportions (called gates) are themselves observation-sensitive predictors, synchronized across experts to give properly normalized state distributions for any input observation. Inference is simple and produces multimodal state conditionals. Learning is different from [64] in that we use sparse greedy approximations, and differs from [7, 62] in that we use type-II maximum likelihood Bayesian approximations [34, 59] and not structured variational ones. The conditional state distributions we learn can also be used as proposals, *e.g.* importance samplers (to initialize) in generative inference algorithms [14, 47, 54, 51].

3. We demonstrate our methods for real and motion capture-based test sequences and give comparisons with nearest neighbor and regression methods.

2 Formulation

We work with discriminative graphical models with a chain structure, as shown in fig. 1, These have continuous temporal states $\mathbf{x}_t$, $t = 1 \dots T$, prior $p(\mathbf{x}_1)$, observations $\mathbf{r}_t$. For notational compactness, we also consider joint states $\mathbf{X}_t = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t)$ or joint observations $\mathbf{R}_t = (\mathbf{r}_1, \dots, \mathbf{r}_t)$. Learning and inference is based on local conditionals: $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, $p(\mathbf{x}_t|\mathbf{r}_t)$ and $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$.

[2] An expert is any function approximator, *e.g.* a perceptron or regressor.

2.1 Discriminative Density Propagation

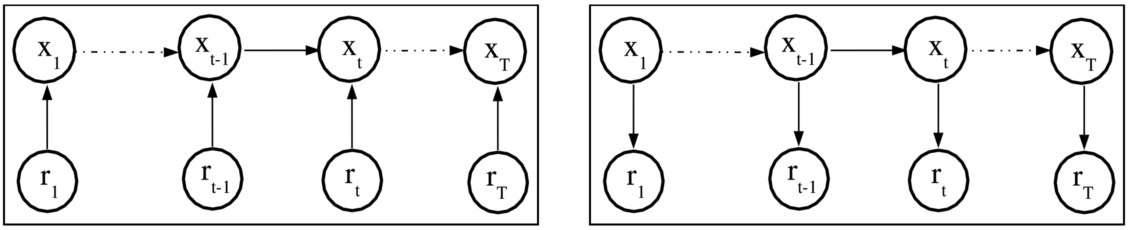

Figure 1: A discriminative chain model *(a, left)* reverses the direction of the arrows that link the state and the observation, compared with a generative one *(b, right)*. The state conditionals $p(\mathbf{x}_t|\mathbf{r}_t)$ or $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ can be learned using training pairs and directly predicted during inference. Instead, a generative approach *(b)* will model and learn $p(\mathbf{r}_t|\mathbf{x}_t)$ and do a more complex probabilistic inversion to compute $p(\mathbf{x}_t|\mathbf{r}_t)$ via Bayes' rule.

For filtering, we wish to compute the optimal distribution $p(\mathbf{x}_t|\mathbf{R}_t)$ for the state $\mathbf{x}_t$, conditioned by observations $\mathbf{R}_t$ up to time t. The filtered density can be derived as:

$$p(\mathbf{x}_t|\mathbf{R}_t) = \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t) p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}) \tag{1}$$

Proof

The following properties can be verified visually in fig. 1a, using a Bayes ball algorithm:

$$\mathbf{x}_t \perp\!\!\!\perp \mathbf{X}_{t-2}|\mathbf{x}_{t-1} \tag{2}$$

$$\mathbf{r}_t \perp\!\!\!\perp \mathbf{R}_{t-1} \tag{3}$$

$$\mathbf{x}_t \perp\!\!\!\perp \mathbf{R}_{t-1}|\mathbf{x}_{t-1}, \mathbf{r}_t \tag{4}$$

$$\mathbf{X}_{t-1} \perp\!\!\!\perp \mathbf{r}_t \tag{5}$$

In this model the observations are *marginally independent*. Notice how this is different from a generative chain model (fig. 1b), where the observations are *conditionally independent*, given the states.

$$p(\mathbf{x}_t|\mathbf{R}_t) = \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{R}_{t-1}, \mathbf{r}_t) = \tag{6}$$

$$= \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{R}_{t-1}, \mathbf{r}_t) p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}, \mathbf{r}_t) = \tag{7}$$

$$= \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t) p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}) \tag{8}$$

where in the last line we used:

$$(4) \Rightarrow p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{R}_{t-1}, \mathbf{r}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$$
$$(5) \Rightarrow p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}, \mathbf{r}_t) = p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})$$

In practice, we do estimation using the mixture conditionals for $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ (a Bayesian mixture of experts *c.f.* §2.2) and the prior $p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1})$, each having, say M components. We first integrate M^2 pairwise products of Gaussians analytically. This requires the linearization of our (generally) non-linear, but parametric (*i.e.* easily differentiable) state conditionals. The means of this expanded posterior are clustered and the centers are used to initialize a reduced M-component approximation that is refined using variational optimization [51].

A filtering propagation rule like (1) can also be derived for a *generative chain model*[3] (fig. 1b) [49], where we directly learn observation-based state conditionals $p(\mathbf{x}_t|\mathbf{r}_t)$, to simplify inference:

$$p(\mathbf{x}_t|\mathbf{R}_t) \propto \frac{p(\mathbf{x}_t|\mathbf{r}_t)}{p(\mathbf{x}_t)} \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{R}_{t-1}) \tag{9}$$

with normalization factor $p(\mathbf{r}_t)/p(\mathbf{r}_t|\mathbf{R}_{t-1})$ and $p(\mathbf{x}_t) = \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1})$.

The form (9) shows a striking analogy with the generative propagation one [19, 24] (based on which it is derived), except from the unpleasant state-dependent division (weighting) by $p(\mathbf{x}_t)$ that results from the application of Bayes' rule to invert the generative conditional $p(\mathbf{r}_t|\mathbf{x}_t)$. Over long time series, $p(\mathbf{x}_t)$ approaches the state equilibrium distribution, under conditional dynamics $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and an approximation could be precomputed. But (9) remains more complex to implement. It requires recursively propagating $p(\mathbf{x}_t|\mathbf{R}_t)$ and computing $p(\mathbf{x}_t)$, two mixture simplification levels (inside the integrand and outside it through the multiplication by $p(\mathbf{x}_t|\mathbf{r}_t)$) and a division (weighting) by $p(\mathbf{x}_t)$. On the other hand $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ requires more training data because of higher input dimensionality[4]. It would be interesting to study such practical trade-offs, as well as the subtle difference between estimates based on discriminative and generative chain models (with different independence properties [49]), but we will not pursue this here.

2.2 Bayesian Mixture of Experts (BME)

This section describes our methodology for learning multimodal conditional distributions for discriminative tracking (*e.g.* $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, $p(\mathbf{x}_t|\mathbf{r}_t)$ or $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ in §2.1). Our proposal is motivated by the fact that many perception problems like reconstruction or tracking involve the recovery of inverse, intrinsically multivalued mappings. Static or dynamic state estimation ambiguities translate into multimodal conditional distributions in fig. 1. To represent them, we use several 'experts' that are simple function approximators. The experts transform their inputs[5] into output predictions that are combined in a probabilistic mixture model based on Gaussians centered around them. The

[3] Notice that 'explaining away' [28] prevents a simple factorization of $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ in (1), corresponding to the discriminative chain model in fig. 1a, based on $p(\mathbf{x}_t|\mathbf{r}_t)$ and $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. While $\mathbf{x}_{t-1}$ and $\mathbf{r}_t$ are marginally independent, they become conditionally dependent when observing $\mathbf{x}_t$.

[4] In fact, (1) can be derived even more generally, based on a predictive conditional that depends on a larger window of observations up to time t [49].

[5] The 'inputs' can be either observations $\mathbf{r}_t$, when modeling $p(\mathbf{x}_t|\mathbf{r}_t)$, states for $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, or observation-state pairs $(\mathbf{x}_{t-1}, \mathbf{r}_t)$ for $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. The 'output' is the state throughout.

model is consistent across experts and inputs, *i.e.* the mixing proportions of the experts reflect the distribution of the outputs in the training set and they sum to 1 for every input. Some domains can be predicted competitively by multiple experts and will have multimodal conditionals. Other 'unambiguous' inputs may be predicted by a single expert, with the others effectively switched-off, having negligible probability (see fig. 2). This is the rationale behind a Bayesian mixture of experts and provides a powerful mechanism for a contextual modeling of complex multimodal distributions. Formally this is described by:

$$p(\mathbf{x}|\mathbf{r}, \mathbf{W}, \boldsymbol{\Omega}, \boldsymbol{\delta}) = \sum_{i=1}^{M} g(\mathbf{r}|\boldsymbol{\delta}_i) p(\mathbf{x}|\mathbf{r}, \mathbf{W}_i, \boldsymbol{\Omega}_i^{-1}) \tag{10}$$

where:

$$g(\mathbf{r}|\boldsymbol{\delta}_i) = \frac{f(\mathbf{r}|\boldsymbol{\delta}_i)}{\sum_{k=1}^{M} f(\mathbf{r}|\boldsymbol{\delta}_k)} \tag{11}$$

$$p(\mathbf{x}|\mathbf{r}, \mathbf{W}_i, \boldsymbol{\Omega}_i) = \mathcal{N}(\mathbf{x}|\mathbf{W}_i \boldsymbol{\Phi}(\mathbf{r}), \boldsymbol{\Omega}_i^{-1}) \tag{12}$$

Here $\mathbf{r}$ are input or predictor variables, $\mathbf{x}$ are outputs or responses, g are *input dependent* positive gates, computed in terms of functions $f(\mathbf{r}|\boldsymbol{\delta}_i)$, parameterized by $\boldsymbol{\delta}_i$ (f should produce gates g within $[0, 1]$, functional choices are given in §2.2.2 and §2.2.3). Notice how g are normalized to sum to 1 for consistency, by construction, for any given input $\mathbf{r}$. Also p are Gaussian distributions (12) with covariances $\boldsymbol{\Omega}_i^{-1}$, centered at different 'expert' predictions, here kernel ($\boldsymbol{\Phi}$) regressors with weights $\mathbf{W}_i$. The parameters of the model including experts and gates are collectively stored in $\boldsymbol{\theta} = \{(\boldsymbol{\alpha}_i, \mathbf{W}_i, \boldsymbol{\Omega}_i, \boldsymbol{\delta}_i) \mid i = 1 \ldots M\}$. As in many Bayesian settings [34, 59, 7], the weights $\mathbf{W}_i$ (and gates $\boldsymbol{\delta}_i$), are controlled by hierarchical priors, typically Gaussians with 0 mean, and having inverse variance hyperparameters $\boldsymbol{\alpha}_i$ controlled by a second level of Gamma distributions. This gives an automatic relevance determination mechanism [34, 59] that avoids overfitting and encourages compact models with fewer non-zero weights for efficient prediction.

Learning the mixture of experts is somewhat complex, various models and algorithms are given in the next sections. As in many prediction problems we optimize the parameters $\boldsymbol{\theta}$ to maximize the log-likelihood of a data set, $\mathcal{T} = \{(\mathbf{r}_i, \mathbf{x}_i) \mid i = 1 \ldots N\}$, *i.e.* the accuracy of predicting $\mathbf{x}$ given $\mathbf{r}$, averaged over the data distribution. For learning, a full Bayesian treatment requires computing posterior distributions over parameters and hyperparameters. Because exact computations are intractable, we rely on approximations and design iterative Bayesian EM algorithms, based on type-II maximum likelihood [34, 59]. These use Laplace approximation for the hyperparameters and analytical integrate the weights, which in this setting become Gaussian [34, 59].

Our algorithms proceed as follows. In the E-step we estimate the posterior:

$$h(\mathbf{x}, \mathbf{r}|\mathbf{W}_i, \boldsymbol{\Omega}_i, \boldsymbol{\delta}_i) = \frac{g(\mathbf{r}|\boldsymbol{\delta}_i) p(\mathbf{x}|\mathbf{r}, \mathbf{W}_i, \boldsymbol{\Omega}_i^{-1})}{\sum_{j=1}^{M} g(\mathbf{r}|\boldsymbol{\delta}_j) p(\mathbf{x}|\mathbf{r}, \mathbf{W}_j, \boldsymbol{\Omega}_j^{-1})} \tag{13}$$

This gives the probability that the expert i has generated the data, and requires knowledge of both inputs and outputs (there is one h for each expert-training pair). In the M-step we solve two optimization problems, one for each expert and one for its gate. The first learns the expert parameters $(\mathbf{W}_i, \boldsymbol{\Omega}_i)$, based on training data $\mathcal{T}$, weighted according to the current membership estimates h (the covariances $\boldsymbol{\Omega}_i$ are estimated from expert prediction errors [64]). The second

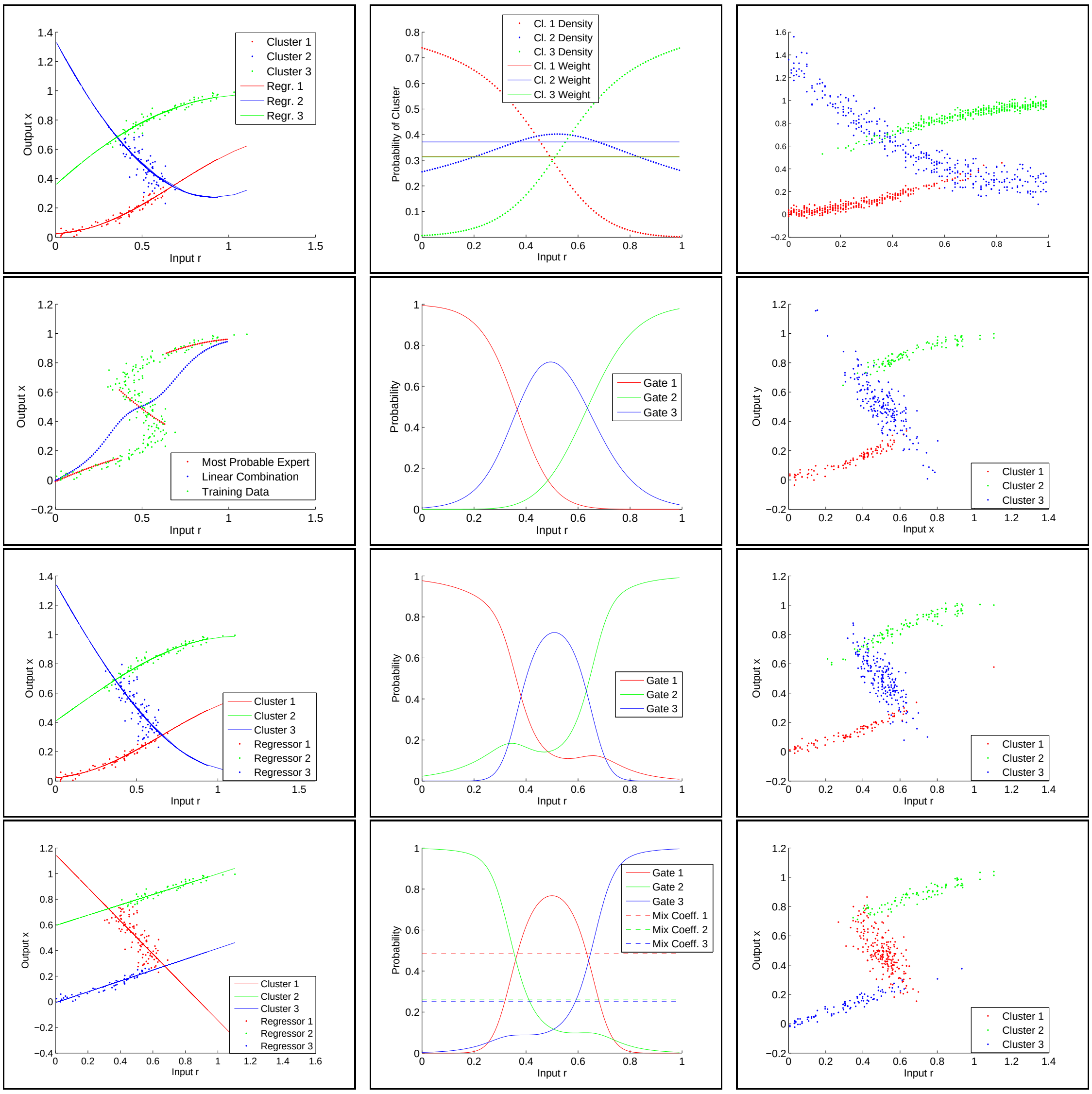

Figure 2: Experts and gates fitted using different models (see text). First two rows show results using the model in §2.2.2. The bottom two rows show results based on models in §2.2.3. Notice that the estimates for the experts are similar, but the gates are somewhat different. However, despite minor inaccuracies, all methods produce well-fitted models, with output distributions close to the original training set.

optimization teaches the gates g how to predict h.[6] The solutions are based on ML-II, with greedy (regressor weight) subset selection. This strategy aggressively sparsifies the experts by eliminating inputs with small weights after each iteration [59, 32].

Inference is straightforward using (10). The result is a conditional mixture distribution with components and mixing probabilities that are input-dependent.

In fig. 2 we explain the Bayesian mixture of experts modeling through an illustrative toy ex-

[6] Prediction based on the input *only* is essential for inference, where membership probabilities (13) cannot be computed because the output is missing.

ample. We show different models that use linear and Gaussian kernel experts, as well as different gating functions, as presented in §2.2.2 and §2.2.3. Our dataset, shown also in [7], consists of about 250 values of x generated uniformly in $(0, 1)$ and then evaluated as $r = x + 0.3\sin(2\pi x) + \epsilon$, with ϵ drawn from a zero mean Gaussian with standard deviation 0.05. Notice that $p(x|r)$ is multimodal. The first two rows show a model fitted as in §2.2.2 wheres the last uses a model described in §2.2.3. *(a) First row, left* shows the data colored by the posterior membership probability h (13) of three expert kernel regressors. *(b) First row, middle* shows the gates g (11), as a function of the input (first iteration), but also the three uniform probabilities that would be computed by a clusterwise regressor [40, 13, 42]. *(c) First row, right* shows samples from a generative conditional model that has not yet converged (second gate iteration corresponding to *(b)*. Notice the overestimated variance of the middle gate with excess contributions away from its true, central, input operating range. *(d) Second row, left* gives the most probable expert as a function of the input as well as their weighted average using the uniform mixing coefficients of the joint. *(e) Second row, middle* show the gates at convergence *(f) Second row, right* shows 400 samples from the estimated model. *(g,j) Third / Fourth row, left* shows the data fitting using a mixture of Gaussian kernel / linear regressors in §2.2.3. *(h,k) middle and (i,j) right* show the gates and data generated from the model. In *(k)* we also show the mixing proportions of the joint, see §2.2.3.

2.2.1 Mixture of Experts Modeling Assumptions

Conditional mixture of experts models differ in their assumptions about the input distribution and in the way the (conditional) target is computed. *Direct methods* [25, 29, 64, 7] estimate the conditional directly using EM double loop algorithms. Internal M-step iteration is often required (at least) for estimating the gate parameters, see §2.2.2. Other algorithms used variational approximations in order to bound the conditional likelihood update and avoid iterative M-steps [26]. Direct conditional methods often do not model the distribution over the inputs (fig. 3a), although this needs not be the case. *Indirect methods* [66, 62] model the joint distribution and often represent their input stochastically (fig. 3b). The conditional can in principle be obtained from the joint [40, 13, 42], using Bayes's rule, conditioning and marginalization (see §2.2.3). Earlier methods [42], however, assumed uniform inputs and did not compute a conditional. Instead, they clustered the joint distribution based on the accuracy of expert (regressor, perceptron) predictions and worked with this representation. In the next two sections we describe both direct conditional mixture of experts models and indirect joint methods based on random regression and give their learning algorithms.

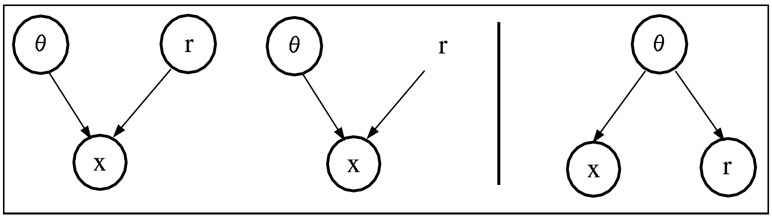

Figure 3: Graphical representations for two conditional models. *(a) Left* shows a direct conditional model, whereas the variation in *(a) middle)* does not include a distribution over the inputs $\mathbf{r}$; The model in *(b) right)* assumes both the inputs and the outputs are stochastic variables. See fig. 4 and fig. 5 for details on possible instantiations.

2.2.2 Bayesian Conditional Mixture of Experts

In this conditional mixture model, the data generation process assumes N datapoints are produced by one of M experts, selected in a stochastic manner. This can be modeled by indicator (hidden) variables $\mathcal{Z} = \{z_i^{(n)} | i = 1 \dots M, n = 1 \dots N\}$ where $z_i^{(n)}$ is 1 if the output datapoint $\mathbf{x}^{(n)}$ has been produced by expert i and zero otherwise. The model has parameters and hyperparameters stored in $\boldsymbol{\theta}$, where $\boldsymbol{\theta} = \{(\mathbf{W}_i, \boldsymbol{\Omega}_i, \boldsymbol{\alpha}_i, \boldsymbol{\delta}_i \equiv (\boldsymbol{\lambda}_i, \boldsymbol{\beta}_i)) \mid i = 1 \dots M\}$, with $\boldsymbol{\lambda}_i$, $\mathbf{W}_i$ individual gate and expert predictor parameters, we omit bias terms for clarity. The conditional probability of output $\mathbf{x}^{(n)}$ (of dimension D) for input $\mathbf{r}^{(n)}$ (of dimension d) is a mixture model with M components:

$$p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \boldsymbol{\theta}) = \sum_{i=1}^{M} p(z_i^{(n)}|\mathbf{r}^{(n)}, \boldsymbol{\lambda}_i) p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\Omega}_i^{-1}) \tag{14}$$

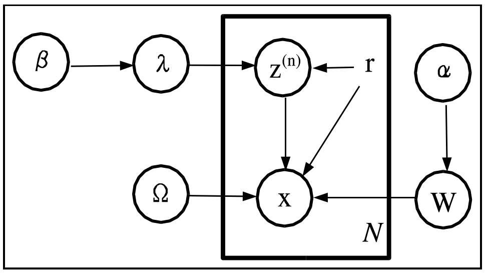

Figure 4: The graphical model for a conditional Bayesian mixture of experts.

The probability of each expert is a Gaussian centered at its prediction $\mathbf{W}_i \boldsymbol{\Phi}(\mathbf{r}^{(n)})$, where $\boldsymbol{\Phi}$ is a vector of kernel functions:

$$\chi_i^{(n)} = p(\mathbf{x}^{(n)}|\mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\Omega}_i^{-1}) = \mathcal{N}(\mathbf{x}^{(n)}|\mathbf{W}_i \boldsymbol{\Phi}(\mathbf{r}^{(n)}), \boldsymbol{\Omega}_i^{-1}) \tag{15}$$

The conditional (prior) probability of selecting expert i, given the input *only*, is implemented using softmax. This ensures that the expert outputs are probabilistically consistent (positive and sum to 1), for any given input:

$$g_i^{(n)} = p(z_i^{(n)} = 1|\mathbf{r}^{(n)}, \boldsymbol{\lambda}_i) = \frac{e^{\boldsymbol{\lambda}_i^\top \boldsymbol{\Phi}(\mathbf{r}^{(n)})}}{\sum_{k=1}^{M} e^{\boldsymbol{\lambda}_k^\top \boldsymbol{\Phi}(\mathbf{r}^{(n)})}} \tag{16}$$

The conditional (posterior) probability $h_i^{(n)}$ of selecting expert i, given *both* the input $\mathbf{r}^{(n)}$ *and* the output $\mathbf{x}^{(n)}$, is:

$$h_i^{(n)} = p(z_i^{(n)} = 1|\mathbf{x}^{(n)}, \mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\lambda}_i, \boldsymbol{\Omega}_i) = \frac{g_i^{(n)} \chi_i^{(n)}}{\sum_{k=1}^{M} g_k^{(n)} \chi_k^{(n)}} \tag{17}$$

The posterior is only available during learning. For inference (prediction) based on (14), the learned prior (16) is used.

The gate and expert weights have Gaussian priors centered at zero, with variance controlled by a second level of Gamma hyperpriors. This avoids overfitting and provides an automatic relevance determination mechanism, encouraging compact models with few non-zero expert and gate weights, for efficient prediction [34, 36, 59, 7]:

$$p(\boldsymbol{\lambda}_i|\boldsymbol{\beta}_i) = \prod_{k=1}^{d} \mathcal{N}(\lambda_i^k|0, \frac{1}{\beta_i^k}) \tag{18}$$

$$p(\mathbf{W}_i|\boldsymbol{\alpha}_i) = \prod_{j=1}^{D}\prod_{k=1}^{d} \mathcal{N}(w_i^{jk}|0, \frac{1}{\alpha_i^k}) \tag{19}$$

$$p(\boldsymbol{\alpha}_i) = \prod_{k=1}^{d} \text{Gamma}(\alpha_i^k|a, b) \tag{20}$$

$$p(\boldsymbol{\beta}_i) = \prod_{k=1}^{d} \text{Gamma}(\beta_i^k|a, b) \tag{21}$$

$$\text{Gamma}(v|a, b) = \frac{b^a v^{(a-1)} e^{-bv}}{\Gamma(a)} \tag{22}$$

The parameters (a, b) are set to $a = 10^{-2}$ and $b = 10^{-4}$ to give broad hyperpriors [7, 34, 36, 59].

We train our BME model in a maximum likelihood framework using EM. We work with a complete data set $\{\mathcal{T}, \mathcal{Z}\}$, including the observed training data $\mathcal{T}$ and the hidden variables $\mathcal{Z}$. Given the current values of the parameters $\boldsymbol{\theta}$, the E-step computes the distribution over the hidden variables $p(\mathcal{Z}|\mathcal{T}, \boldsymbol{\theta})$. This is done using (17). The M step maximizes the expected value of the complete data likelihood $p(\mathcal{T}, \mathcal{Z}|\boldsymbol{\theta})$. This EM scheme can be cast in a variational framework where we optimize the $KL(Q||p)$ divergence that involves the intractable joint $p(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{Z}|\mathcal{T})$ and an approximate separable factorization $Q(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{Z})$ (dependency on input $\mathbf{r}$ is omitted):

$$Q(\boldsymbol{\theta}, \mathcal{Z}) = Q(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{Z}) = Q(\mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\Omega})Q(\boldsymbol{\lambda}, \boldsymbol{\beta})Q(\mathcal{Z}) \tag{23}$$

This is equivalent to minimizing the variational free energy:

$$\mathcal{F}(Q) = \int Q(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{Z}) \log \frac{Q(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{Z})}{p(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{Z}, \mathcal{T})} d\mathbf{W} d\boldsymbol{\lambda} d\boldsymbol{\Omega} d\boldsymbol{\alpha} d\boldsymbol{\beta} \tag{24}$$

where:

$$p(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{Z}, \mathcal{T}) = \prod_{i=1}^{M} p(\boldsymbol{\alpha}_i)p(\boldsymbol{\beta}_i)p(\mathbf{W}_i|\boldsymbol{\alpha}_i)p(\boldsymbol{\lambda}_i|\boldsymbol{\beta}_i)\times \tag{25}$$

$$\times \prod_{i=1}^{N} p(z_i^{(n)}|\mathbf{r}^{(n)}, \boldsymbol{\lambda}_i)p(\mathbf{x}^{(n)}|\mathbf{W}_i\mathbf{r}^{(n)}, \boldsymbol{\Omega}_i^{-1})^{z_i^{(n)}} \tag{26}$$

Optimizing $Q(\mathbf{W}, \boldsymbol{\alpha}, \boldsymbol{\Omega})$ involves the computation of M Gaussian distributions for the weights of each expert. One possibility, followed in [64] is to use a weight decay prior (corresponding to ridge regression) and not an ARD mechanism. In this case, estimating the expert parameters leads to convex least-squares problems whereas estimating the gates requires Laplace approximations. Instead, we use sparse priors and this makes both problems non-convex. For the experts, we use Laplace approximation for the hyperparameters $(\boldsymbol{\alpha})$ and analytically integrate the weights $(\mathbf{W})$, that in this setting become Gaussian [34, 36, 59, 58]. For the gates, we compute a set of M Gaussian distributions $Q(\boldsymbol{\lambda}, \boldsymbol{\beta})$ using Laplace approximation, by maximizing the cross-entropy between the posterior probability g and the posterior probability h (17) [29]. This is based on an iterative procedure (the equations for the gates are coupled) that uses a second-order damped trust region optimization method [18, 15] and not IRLS [29]. This double-loop algorithm is summarized below [29, 64]:

---

1. **E-step:** For each data pair $\{(\mathbf{r}^{(n)}, \mathbf{x}^{(n)}) \mid n = 1 \ldots N\}$ compute posteriors $h_i^{(n)}$ for each expert $i = 1 \ldots M$, using the current value of parameters $(\mathbf{W}_i, \boldsymbol{\lambda}_i, \boldsymbol{\Omega}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$.
2. **M-step:** For each expert, solve weighted regression problem with data $\{(\mathbf{r}^{(n)}, \mathbf{x}^{(n)}) \mid n = 1 \ldots N\}$ and weights $h_i^{(n)}$ to update $(\mathbf{W}_i, \boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i)$. This involves Laplace approximation for the hyperparameters and analytical integration for the weights and optimization with greedy weight subset selection [59, 32].
3. **M-step:** For each gating network i, solve regression problem with data $(\mathbf{r}^{(n)}, h_i^{(n)})$ to update $(\boldsymbol{\lambda}_i, \boldsymbol{\beta}_i)$. This involves maximizing the cross-entropy between g and h, with sparse priors on the gate weights and greedy subset selection [59, 32]. We use Laplace approximation for the hyperparameters and for the weights.
4. Iterate using the updated parameter values $\boldsymbol{\theta} = \{(\mathbf{W}_i, \boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i, \boldsymbol{\lambda}_i, \boldsymbol{\beta}_i) \mid i = 1 \ldots M\}$.

2.2.3 Mixture of Experts based on Random Regression and Joint Density

A different approach to estimate a conditional distribution is to model the joint distribution over inputs and outputs and then obtain the conditional using Bayes' rule. While this model is somewhat indirect, potentially wasteful of resources, *i.e.* more difficult to estimate due to higher dimensionality, working with a Gaussian mixture eases some of the computations, which in this case can be performed analytically. Assume for generality, a full covariance mixture model of the joint distribution over input-output pairs $(\mathbf{x}, \mathbf{y})$, given by (33) and (35):

$$p\left(\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix} \middle| \boldsymbol{\theta}\right) = \sum_{i=1}^{M} \rho_i \mathcal{N}\left(\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix} \middle| \begin{pmatrix}\boldsymbol{\mu}_i^r\\ \boldsymbol{\mu}_i^x\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}_i^{rr} & \boldsymbol{\Sigma}_i^{rx}\\ \boldsymbol{\Sigma}_i^{xr} & \boldsymbol{\Sigma}_i^{xx}\end{pmatrix}\right) \tag{27}$$

The conditional $p(\mathbf{x}|\mathbf{r}, \boldsymbol{\theta})$ can be obtained from (27) using Bayes' rule:

$$p(\mathbf{x}|\mathbf{r}, \boldsymbol{\theta}) = \frac{p(\mathbf{r}, \mathbf{x}|\boldsymbol{\theta})}{\int p(\mathbf{r}, \mathbf{x}|\boldsymbol{\theta}) d\mathbf{x}} \tag{28}$$

The Gaussian family is closed under marginalization. This removes lines and columns for the variables that are integrated. The numerator is obtained by Gaussian conditioning:

$$p(\mathbf{x}|\mathbf{r}, \boldsymbol{\theta}) = \frac{\sum_{i=1}^{M} \rho_i \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i^r, \boldsymbol{\Sigma}_i^{rr}) \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i^x + \boldsymbol{\Sigma}_i^{xr}(\boldsymbol{\Sigma}_i^{rr})^{-1}(\mathbf{r} - \boldsymbol{\mu}_i^r), \boldsymbol{\Sigma}_i^{xx} - \boldsymbol{\Sigma}_i^{xr}(\boldsymbol{\Sigma}_i^{rr})^{-1}\boldsymbol{\Sigma}_i^{rx})}{\sum_{i=1}^{M} \rho_i \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i^r, \boldsymbol{\Sigma}_i^{rr})} = \tag{29}$$

$$= \sum_{i=1}^{M} \frac{\rho_i \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i^r, \boldsymbol{\Sigma}_i^{rr})}{\sum_{i=1}^{M} \rho_i \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i^r, \boldsymbol{\Sigma}_i^{rr})} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i^x + \boldsymbol{\Sigma}_i^{xr}(\boldsymbol{\Sigma}_i^{rr})^{-1}(\mathbf{r} - \boldsymbol{\mu}_i^r), \boldsymbol{\Sigma}_i^{xx} - \boldsymbol{\Sigma}_i^{xr}(\boldsymbol{\Sigma}_i^{rr})^{-1}\boldsymbol{\Sigma}_i^{rx}) \tag{30}$$

Working with a mixture mixture of regressors, further constrains the general form above. Assuming a distribution over both inputs and outputs, the mixture of random regressions [44, 62] is given by the graphical model in fig. 5. It is a constrained joint mixture model with component proportions ρ_i, input means and covariance matrices $(\boldsymbol{\mu}_i, \mathbf{\Sigma}_i)$ and expert parameters $(\boldsymbol{\alpha}_i, \mathbf{W}_i, \mathbf{\Omega}_i)$ (as in the conditional model of §2.2.2).

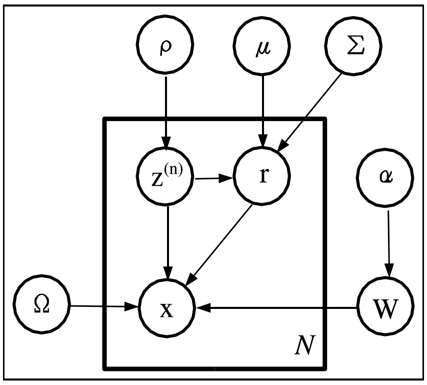

Figure 5: The graphical model of a joint mixture based on random regression.

For a *mixture of random linear regressions* model, with parameters $\boldsymbol{\theta} = \{(\mathbf{W}_i, \boldsymbol{\alpha}_i, \mathbf{\Omega}_i, \boldsymbol{\delta}_i \equiv (\rho_i, \boldsymbol{\mu}_i, \mathbf{\Sigma}_i)) \mid i = 1 \dots M\}$, the *joint distribution* is:

$$p\left(\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix} \middle| \boldsymbol{\theta}\right) = \sum_{i=1}^{M} \rho_i \mathcal{N}\left(\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix} \middle| \begin{pmatrix}\boldsymbol{\mu}_i\\ \mathbf{W}_i\boldsymbol{\mu}_i\end{pmatrix}, \begin{pmatrix}\mathbf{\Sigma}_i + \mathbf{W}_i^{\top}\mathbf{\Omega}_i\mathbf{W}_i & -\mathbf{W}_i^{\top}\mathbf{\Omega}_i\\ -\mathbf{\Omega}_i\mathbf{W}_i & \mathbf{\Omega}_i\end{pmatrix}^{-1}\right) \quad (31)$$

The *conditional distribution* over the responses $\mathbf{x}$, given the covariates $\mathbf{r}$, in the mixture of linear regressions model is:

$$p(\mathbf{x}|\mathbf{r}, \boldsymbol{\theta}) = \sum_{i=1}^{M} \frac{\rho_i \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i, \mathbf{\Sigma}_i^{-1})}{\sum_{j=1}^{M} \rho_j \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_j, \mathbf{\Sigma}_j^{-1})} \mathcal{N}(\mathbf{x}|\mathbf{W}_i\mathbf{r}, \mathbf{\Omega}_i^{-1}) = \sum_{i=1}^{M} g(\mathbf{r}|\boldsymbol{\delta}_i)\mathcal{N}(\mathbf{x}|\mathbf{W}_i\mathbf{r}, \mathbf{\Omega}_i^{-1}) \quad (32)$$

Proof: We write the joint distribution as a mixture model, with Gaussian input and output marginal components:

$$p\left(\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix} \middle| \boldsymbol{\theta}\right) = \sum_{i=1}^{M} \rho_i \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i, \mathbf{\Sigma}_i^{-1})\mathcal{N}(\mathbf{x}|\mathbf{W}_i\mathbf{r}, \mathbf{\Omega}_i^{-1}) = \quad (33)$$

$$= \sum_{i=1}^{M} (2\pi)^{\frac{d+D}{2}} |\mathbf{\Sigma}_i|^{1/2} |\mathbf{\Omega}_i|^{1/2} \exp\Big[-\frac{1}{2}(\mathbf{r} - \boldsymbol{\mu}_i)^{\top}\mathbf{\Sigma}_i(\mathbf{r} - \boldsymbol{\mu}_i) + (\mathbf{x} - \mathbf{W}_i\mathbf{r})^{\top}\mathbf{\Omega}_i(\mathbf{x} - \mathbf{W}_i\mathbf{r})\Big] \quad (34)$$

Denote the quadratic form in the exponent of (33) as J, and rewrite it as:

$$J = \begin{pmatrix}\mathbf{r} - \boldsymbol{\mu}_i\\ \mathbf{x} - \mathbf{W}_i\mathbf{r}\end{pmatrix}^{\top} \begin{pmatrix}\mathbf{\Sigma}_i & \mathbf{O}_{d\mathrm{x}D}\\ \mathbf{0}_{D\mathrm{x}d} & \mathbf{\Omega}_i\end{pmatrix} \begin{pmatrix}\mathbf{r} - \boldsymbol{\mu}_i\\ \mathbf{x} - \mathbf{W}_i\mathbf{r}\end{pmatrix} \quad (35)$$

$$\begin{pmatrix} \mathbf{r}-\boldsymbol{\mu}_i \\ \mathbf{x}-\mathbf{W}_i\mathbf{r} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_d & \mathbf{0}_{d\mathrm{x}D} \\ -\mathbf{W}_i & \mathbf{I}_D \end{pmatrix} \begin{pmatrix} \mathbf{r}-\boldsymbol{\mu}_i \\ \mathbf{x}-\mathbf{W}_i\boldsymbol{\mu}_i \end{pmatrix} \tag{36}$$

$$J = \begin{pmatrix} \mathbf{r}-\boldsymbol{\mu}_i \\ \mathbf{x}-\mathbf{W}_i\boldsymbol{\mu}_i \end{pmatrix}^{\top} \begin{pmatrix} \mathbf{I}_d & -\mathbf{W}_i^{\top} \\ \mathbf{0}_{D\mathrm{x}d} & \mathbf{I}_D \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma}_i & \mathbf{O}_{d\mathrm{x}D} \\ \mathbf{0}_{D\mathrm{x}d} & \boldsymbol{\Omega}_i \end{pmatrix} \begin{pmatrix} \mathbf{I}_d & \mathbf{0}_{d\mathrm{x}D} \\ -\mathbf{W}_i & \mathbf{I}_D \end{pmatrix} \begin{pmatrix} \mathbf{r}-\boldsymbol{\mu}_i \\ \mathbf{x}-\mathbf{W}_i\boldsymbol{\mu}_i \end{pmatrix} \tag{37}$$

$$= \begin{pmatrix} \mathbf{r}-\boldsymbol{\mu}_i \\ \mathbf{x}-\mathbf{W}_i\boldsymbol{\mu}_i \end{pmatrix}^{\top} \begin{pmatrix} \boldsymbol{\Sigma}_i + \mathbf{W}_i^{\top}\boldsymbol{\Omega}_i\mathbf{W}_i & -\mathbf{W}_i^{\top}\boldsymbol{\Omega}_i \\ -\boldsymbol{\Omega}_i\mathbf{W}_i & \boldsymbol{\Omega}_i \end{pmatrix} \begin{pmatrix} \mathbf{r}-\boldsymbol{\mu}_i \\ \mathbf{x}-\mathbf{W}_i\boldsymbol{\mu}_i \end{pmatrix} \tag{38}$$

The joint covariance matrix for component i, $\boldsymbol{\Lambda}_i$ is:

$$\boldsymbol{\Lambda}_i = \begin{pmatrix} \boldsymbol{\Sigma}_i + \mathbf{W}_i^{\top}\boldsymbol{\Omega}_i\mathbf{W}_i & -\mathbf{W}_i^{\top}\boldsymbol{\Omega}_i \\ -\boldsymbol{\Omega}_i\mathbf{W}_i & \boldsymbol{\Omega}_i \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{\Sigma}_i & \boldsymbol{\Sigma}_i\mathbf{W}_i^{\top} \\ \mathbf{W}_i\boldsymbol{\Sigma}_i & \mathbf{W}_i\boldsymbol{\Sigma}_i\mathbf{W}_i^{\top} + \boldsymbol{\Omega}_i \end{pmatrix} \tag{39}$$

The joint distribution (33) can thus be shown to give (31), as claimed. However, at first glance, it is not obvious why the conditional should have the form in (32). It indeed qualifies as a gate function with mixing proportions that are positive and sum to 1. The mixing proportions ρ_i of the joint also appear inside the formula for the gates. Authors working with this form (32), *e.g.* [66, 62] introduced it as as one convenient parametric choice of gate function, motivated by simplified estimation and improved input modeling, now measured with error (see fig. 4, fig. 5). Moreover, the conditional (32) is precisely the distribution obtained from the joint (31) using Bayes' rule. By replacing the means and covariances of the mixture of linear regressions in (31) and (39), into (29), we obtain (32). Therefore, estimating the joint model in (33) gives the necessary parameters for computing the conditional using (32). To estimate the joint model, we introduce hidden variables with similar interpretation as for the conditional in §2.2.2. Then the joint distribution over parameters, hyperparameters and complete data $\{\mathcal{T}, \mathcal{Z}\}$ can be written, similarly with (25), using (33) as:

$$p(\mathbf{W}, \boldsymbol{\lambda}, \boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathcal{Z}, \mathcal{T}) = \prod_{i=1}^{M} p(\boldsymbol{\alpha}_i) p(\mathbf{W}_i|\boldsymbol{\alpha}_i) \times \tag{40}$$

$$\times \prod_{i=1}^{N} \{\rho_i p(\mathbf{r}^{(n)}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i^{-1}) p(\mathbf{x}^{(n)}|\mathbf{W}_i\mathbf{r}^{(n)}, \boldsymbol{\Omega}_i^{-1})\}^{z_i^{(n)}} \tag{41}$$

The gate distribution is:

$$g_i^{(n)} = p(z_1^{(n)} = 1|\mathbf{r}^{(n)}, \boldsymbol{\lambda}_i = (\rho_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)) = \frac{\rho_i \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i^{-1})}{\sum_{k=1}^{M} \rho_k \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1})} \tag{42}$$

Based on (13) and (42), the posterior distribution over the hidden variables is:

$$h_i^{(n)} = p(z_i^{(n)} = 1|\mathbf{x}^{(n)}, \mathbf{r}^{(n)}, \mathbf{W}_i, \boldsymbol{\lambda}_i, \boldsymbol{\Omega}_i, \rho_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \tag{43}$$

$$= \frac{\rho_i \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i^{-1}) \mathcal{N}(\mathbf{x}|\mathbf{W}_i\mathbf{r}, \boldsymbol{\Omega}_i^{-1})}{\sum_{k=1}^{M} \rho_k \mathcal{N}(\mathbf{r}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k^{-1}) \mathcal{N}(\mathbf{x}|\mathbf{W}_k\mathbf{r}, \boldsymbol{\Omega}_k^{-1})} \tag{44}$$

The mixing proportions, means and covariance update can be obtained by maximizing the cross-entropy between the prior g and the posterior h [29, 66]:

$$\rho_i = \frac{\sum_{n=1}^{N} h_i^{(n)}}{N} \tag{45}$$

$$\boldsymbol{\mu}_i = \frac{\sum_{n=1}^{N} h_i^{(n)} \mathbf{r}^{(n)}}{\sum_{n=1}^{N} h_i^{(n)}} \tag{46}$$

$$\boldsymbol{\Sigma}_i = \frac{\sum_{n=1}^{N} h_i^{(n)} (\mathbf{r}^{(n)} - \boldsymbol{\mu}_i)(\mathbf{r}^{(n)} - \boldsymbol{\mu}_i)^\top}{\sum_{n=1}^{N} h_i^{(n)}} \tag{47}$$

---

1. **E-step:** For each data pair $\{(\mathbf{r}^{(n)}, \mathbf{x}^{(n)}) \,|\, n = 1 \ldots N\}$ compute posteriors $h_i^{(n)}$ (43), for each expert $i = 1 \ldots M$, using the current value of parameters $(\mathbf{W}_i, \boldsymbol{\lambda}_i, \boldsymbol{\Omega}_i, \rho_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$.

2. **M-step:** For each expert, solve weighted regression problem with data $\{(\mathbf{r}^{(n)}, \mathbf{x}^{(n)}) \mid n = 1 \ldots N\}$ and weights $h_i^{(n)}$ to update $(\mathbf{W}_i, \boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i)$. This involves Laplace approximation for the hyperparameters and analytical integration for the weights. Optimization uses greedy weight subset selection [59, 32].

3. **M-step:** For each gating network i, compute mixing proportions, means and covariances by maximizing the cross-entropy between g and h. The updates are given by (45),(46) and (47).

4. Iterate using the updated parameter values $\boldsymbol{\theta} = \{(\mathbf{W}_i, \boldsymbol{\alpha}_i, \boldsymbol{\Omega}_i, \rho_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid i = 1 \ldots M\}$.

A note on mixture of experts algorithms. Both algorithms given in §2.2.2 and §2.2.3 are useful for estimating compact conditional mixture of experts models. They are based on different assumptions randomness in the input and they have different computational demands. The conditional model in §2.2.2 requires internal M-step iterations when estimating the parameters of the gates. But computing the gates for prediction has complexity $\mathcal{O}(Ms^2d^2)$ where s controls the sparsity of each expert, *e.g.* $5\% - 25\%$ in our experiments. The random regression model §2.2.3 gives a somewhat simpler M-step when learning the gates although computationally this involves inverting possibly large covariance matrices. Gate computation has $\mathcal{O}(Md^3)$ complexity (both these may be simplified using sparsity priors on the input means and covariances, *e.g.* using Wishart distributions [62], but the factor is cubic in the input dimension vs. quadratic in the direct conditional model). These differences may not be significant for moderate input dimensions, but may be important when training conditionals like $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$, for high-dimensional state and feature spaces.

3 Experiments

This section describes our experiments as well as the training sets and features we use. We show results on real and artificially rendered motion capture-based test sequences and give comparisons with existing methods.

Training Set, Model Representation and Image Features It is difficult to obtain ground truth for human motion and even harder to train using many viewpoints or lighting conditions. Therefore, to gather data, we use as others [42, 45, 3, 2, 60], packages like Maya (Alias Wavefront), with realistically rendered computer graphics human surface models which we animate using human motion capture [1]. Our human representation ($\mathbf{x}$) is based on an articulated skeleton with spherical joints and has 56 d.o.f. including global translation. Our database consists of about 3000 samples that involve a variety of human activities including walking, running, turns, gestures in conversations, quarreling and pantomime.

We have done an empirical analysis on how ambiguous a 2000 sample training subset is. This is shown and discussed in fig. 6.

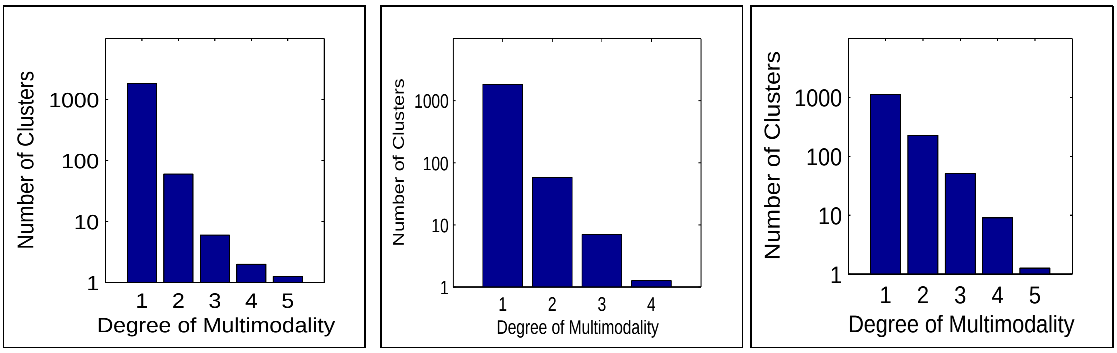

Figure 6: Analysis of 'multimodality' for a training set (the 'number of clusters' axis on logscale): (a) Left: $p(\mathbf{x}_t|\mathbf{r}_t)$ (1912 clusters / 2000 points). (b) Right $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ (1912 clusters / 2000 points). (c) $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$ (1409 clusters / 2000 points). We cluster the features and joint angle vectors, independently, into a large number of clusters. We build histograms for the number of joint angle clusters that fall under the same feature cluster. This quantifies how much ambiguity is there in the database, at the feature and joint angle cluster scale. We select many clusters to simulate the effect of small perturbations in the input. In those cases any feature neighbor (not necessarily the desired one) may be the closest to an input silhouette query. The input neighborhood induces a distribution over clusters of joint angles. We notice that even at this fine scale, the conditionals are multimodal. Decreasing the number of clusters in (c) sharply increases multimodality. Working with the previous state and the current observation (middle and right plot) does not eliminate ambiguity. This is not wild, but severe enough to cause tracking failure or significant errors during initialization (so we observe in various tests). We expect increasing ambiguity for larger training sets.

Our choice of image features is based on previously developed methods for shape and texture modeling [12, 35, 6, 35]. We work with silhouettes and we assume that in real settings these can be obtained using a statistical background subtraction method (we use one based on separately built foreground and background models, using non-parametric density estimation [17] and motion segmentation [8]). Silhouettes are informative for human pose estimation [48, 52], although prone to certain ambiguities (*e.g.* the left / right limb assignment in side views) or occasional lack of observability of some of the d.o.f. (*e.g.* 180^o ambiguities in the global azimuthal orientation for frontal views). These are multiplied by intrinsic forward / backward monocular ambiguities [54] that are common in many human interaction scenarios.[7] As image features, we use shape contexts extracted on the silhouette [6, 35] (5 radial bins, 12 angular bins, with bin size range 1/8 to 3 on log scale).

[7] While no image descriptor set is likely to easily help discriminate them, this further motivates our probabilistic, multiple hypothesis approach.

We have also experimented with pairwise edge angle and distance histograms [4] collected inside the silhouette. The features are computed at a variety of scales and sizes for points sampled on the silhouette. To work in a common coordinate system, we cluster all features in the training set into $K = 40$ clusters (in our experiments). To compute the representation of a new shape feature (a point on the silhouette), we 'project' onto the common basis by (inverse distance) weighted voting into the cluster centers. To obtain the representation $(\mathbf{r})$ for a new silhouette we regularly sample (about 100-200) points on it and add all their feature vectors into a feature histogram. This representation is semi-local, rich and has been effectively demonstrated in many applications, including texture recognition [12] or pose prediction [35, 45, 3, 2].

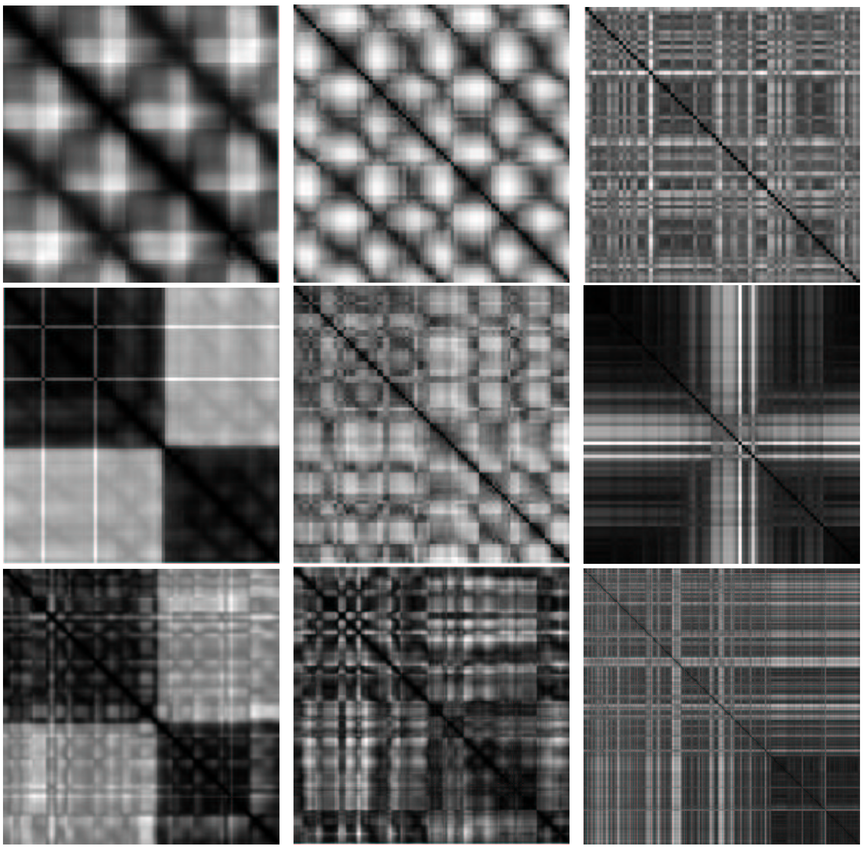

Figure 7: Affinity matrices for (from left to right, on each row) joint angles (JA), external contour shape context (SC) and internal contour pairwise edge (PE) silhouette features: *(a) Top row:* side walk. Notice the periodicity as well as the higher frequencies in the (SC) matrix caused by half-cycle ambiguities for silhouettes; *(b) Middle row:* complex walk; *Bottom row:* conversations. The joint angle and image features correlate far less.

Comparisons We compare our Bayesian mixture of experts (BME) conditional models with other competing methods like weighted nearest neighbor (NN) or the relevance vector machine (RVM) [59]. Our test set consists of a variety of human activities obtained using motion-capture and artificially rendered. This provides ground truth and allows us to concentrate on the algorithms and factor out the variability given by the imperfections of our human model, or the noise in the silhouette extraction in real images. The results are shown and discussed in (the caption of) table 1. In general our BME gives better average estimates and significantly lower maximum errors and the errors can still be reduced. Since the tests do not use any form of temporal coherence, it is clear that a multiple hypothesis tracker and smoother can significantly improve the results, especially for a Bayesian multiple hypothesis method like ours, where we explicitly model uncertainty. It may be the case that occasionally the incorrect mode was selected, but this may be far less often the case in tracking, because of composing with the temporal prior. Various visual results for these tests are shown in: fig. 8 (walking); fig. 9 (running); fig. 10 (complex walk); fig. 12,13 (conversations); fig. 14 (pantomime).

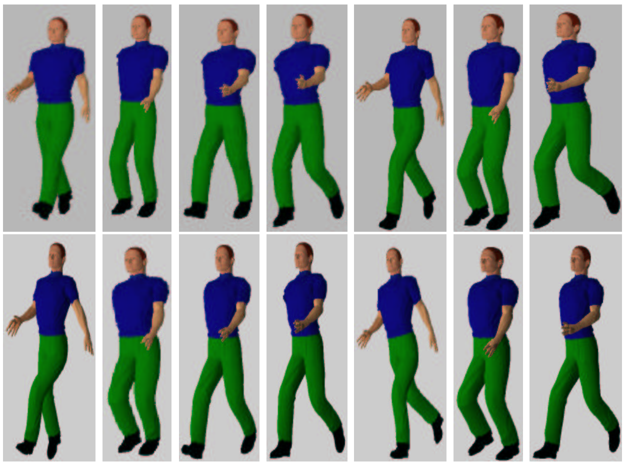

Figure 8: Reconstruction of a walking sequence using $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. *First row:* original images; *Second row:* reconstructed poses seen from the same viewpoint.

Real Image Sequences. Walking, Picking and Dancing We have also run *Bayesian trackers* on different real image sequences with humans doing different activities like walking fig. 15, picking fig. 17 and dancing fig. 19. We track using the propagation rule in (9) and mixture of experts in §2.2.3, with 5 hypotheses and the training set contains about 75 frames of side-viewed walking among examples. Tracking walking is successful, here we show frames from a 3s sequence, 60fps. Occasionally, there are leg assignment ambiguities that may confuse a unimodal tracker as can be seen in the bottom row of fig. 15. Notice also that the affinity matrices for 3D joint angles and for image features correlate quite well (fig. 7), far better that for other motions like conversations or complex walking. This may give an intuition about the difficulty of learning various inverse mappings for these activities.

In fig. 17, we show the result of tracking a real image sequence consisting of 2 seconds of video, 60 fps. Our experiments involve both Bayesian single hypothesis tracking using a single expert, propagated using (1), as well as multiple hypotheses tracking based on a BME model in §2.2.3, learned using 5 experts that are regressors with RBF kernels and degree of sparsity varying between 5%-25%. We initially tested the single hypothesis tracker. This failed to track, as shown in fig. 16, most likely because its input kernels stop firing due to an out-of-range input predicted from the previous timestep. To factor out the effect of imperfect silhouettes or initialization[8] and to make sure that failure is due to motion or feature representation ambiguities, we also tried to

[8] In all cases, we initialize using the conditional $p(\mathbf{x}_t|\mathbf{r}_t)$, learned using BME. For single hypothesis tracking, we select the most probable component.

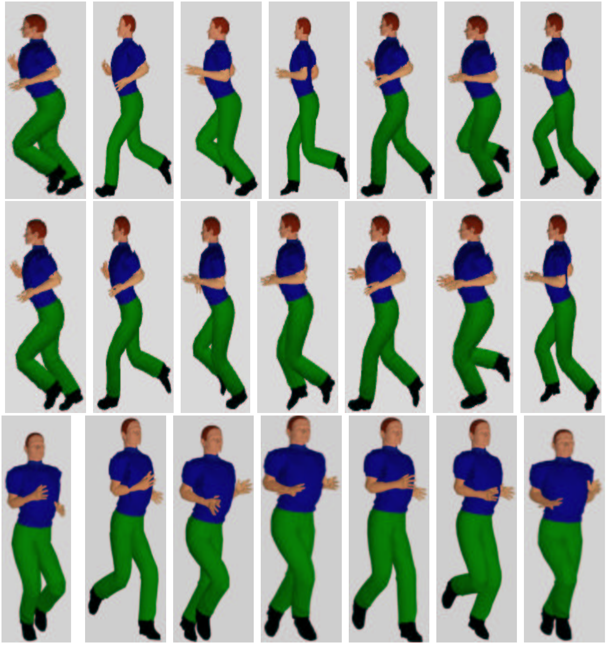

Figure 9: Tracking a ‘running’ sequence using $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. *First row:* Original image sequence. *Second row:* Reconstruction seen from the same viewpoint. *Third row:* Reconstruction seen from a different synthetic viewpoint (notice how the right forearm slightly penetrates the body in the second image on the bottom row).

track a similar sequence using artificially rendered images, generated from a similar motion in our database. Even in that case, the single hypothesis tracker failed. In fig. 17 we show results from a multiple hypothesis BME tracker, that successfully tracks and reconstructs the motion. While the reconstruction is perceptually plausible, there are imperfections – *e.g.* notice that the knee of the model is tilted outward whereas the knee of the human is tilted inward. In fact, we observe persistent multimodality for those joints more actively moving, *e.g.* the right wrist, the right femur and the right shoulder, which have, quite constantly, about 5 modes in their posterior. In general, in the beginning of the sequence there is more ambiguity for almost all the joints, but it tends to fade away during tracking. However, the joints that are occluded or very much project inside the silhouette tend to have persistent ambiguities. Some relevant quantitative results are shown in fig. 18.

We conclude with some experiments where we track and reconstruct using a BME tracker based on the propagation rule (1) and conditional model in §2.2.2. We work with a more challenging dancing sequence and include about 100 dancing examples in the training set. The results are shown in fig. 19. Although the poses we reconstruct are not geometrically perfect and there are some errors at the arms and legs, they give good perceptual results. Quantitative tracking results are shown in fig. 20.

4 Conclusions

We have presented a mixture density propagation framework for temporal inference using discriminative models. We argued that despite their success, existing methods do not offer a formal man-

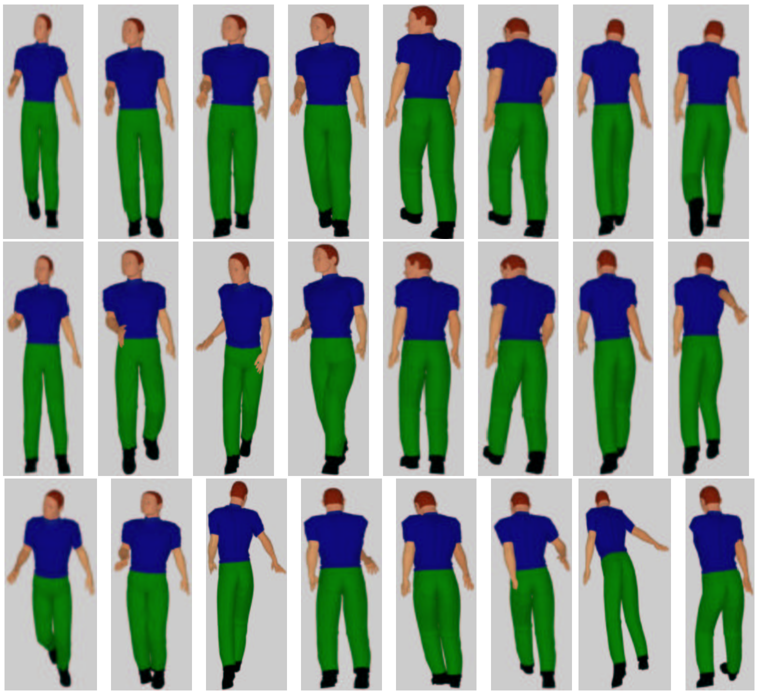

Figure 10: Tracking a complex ‘walking, shake hand and turning’ sequence using $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. *First row:* Original image sequence. *Second row:* Most probable reconstruction (hypothesis) seen from the same viewpoint. *Third row:* the second most probable reconstruction (notice 180^o turn ambiguities) as well as ambiguities of the arms and legs that very much resemble forward-backward flipping ambiguities [54, 55].

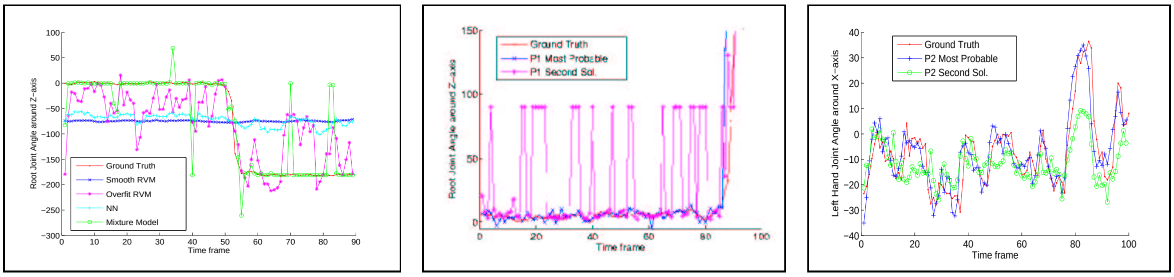

Figure 11: Joint angles for a complex walking sequence: *(a), left* shown in fig. 10 and conversation, *(b) middle and (c) right*, shown in fig. 12,13. Notice that the bimodality in *(a)* cannot be resolved by an RVM or NN estimator. Occasionally there are errors in the multiple hypothesis estimator, but these are quite infrequent, about 6%, reflecting fundamental 180^o orientation ambiguities for very similar input features. In many such cases the correct pose will be either the first or the second most probable mode. Notice also 90^o ambiguities in the conversation sequence *(b)* as apparent in the second most probable mode.

agement of uncertainty and we explained why current representations cannot model multivalued relationships that are pervasive in inverse, perception problems. We contribute by deriving the independence properties and discriminative density propagation rules in continuous, temporal chain models, and by proposing compact Bayesian mixture of experts models capable of learning mul-

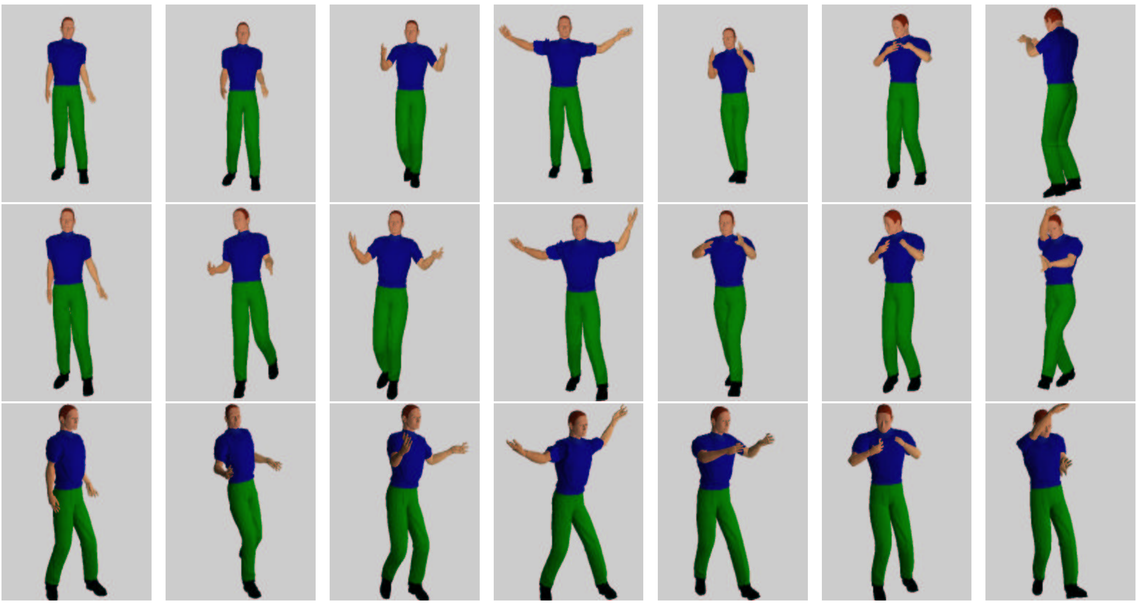

Figure 12: Conversation sequence. Estimates obtained using $p(\mathbf{x}_t|\mathbf{r}_t)$. *First row:* Original image sequence. *Second row:* Most probable reconstruction (hypothesis) seen from the same viewpoint. *Third row:* Most probable reconstruction seen from a different viewpoint. Notice various reconstruction errors. Similar configurations may be also found as local optima when doing inference based on generative models. For silhouette features, many spurious peaks may be eliminated (or significantly downgraded) using more consistent observation likelihoods [48, 52]. This is one advantage of constructive observation modeling in generative approaches.

timodal conditionals. These can be used both as building blocks within genuine discriminative propagation rules, as we show, *e.g.* (1) or (9), and as importance samplers for generative inference (*e.g.* state initialization or recovery from failure). We present results on real and synthetically generated image sequences and give comparisons against nearest neighbor and regression methods. Our study suggests that flexible conditional modeling and uncertainty propagation are both essential for successful tracking. We hope that this work will bring discriminative and generative tracking algorithms closer and help stimulate a fruitful debate on their relative advantages, within a common probabilistic framework.

Future Work We plan to do a detailed sensitivity analysis w.r.t. motions and shapes that deviate from the training set. We will also study alternative, more compact state and feature representations based on dimensionality reduction and investigate scaling aspects for large motion capture databases.

Acknowledgments: Cristian Sminchisescu wants to thank Allan Jepson for many insightful discussions on the topics presented in this paper. The authors acknowledge funding from NSERC, PREA, CITO and NSF.

References

[1] CMU Human Motion Capture DataBase. Available online at http://mocap.cs.cmu.edu/search.html, 2003.

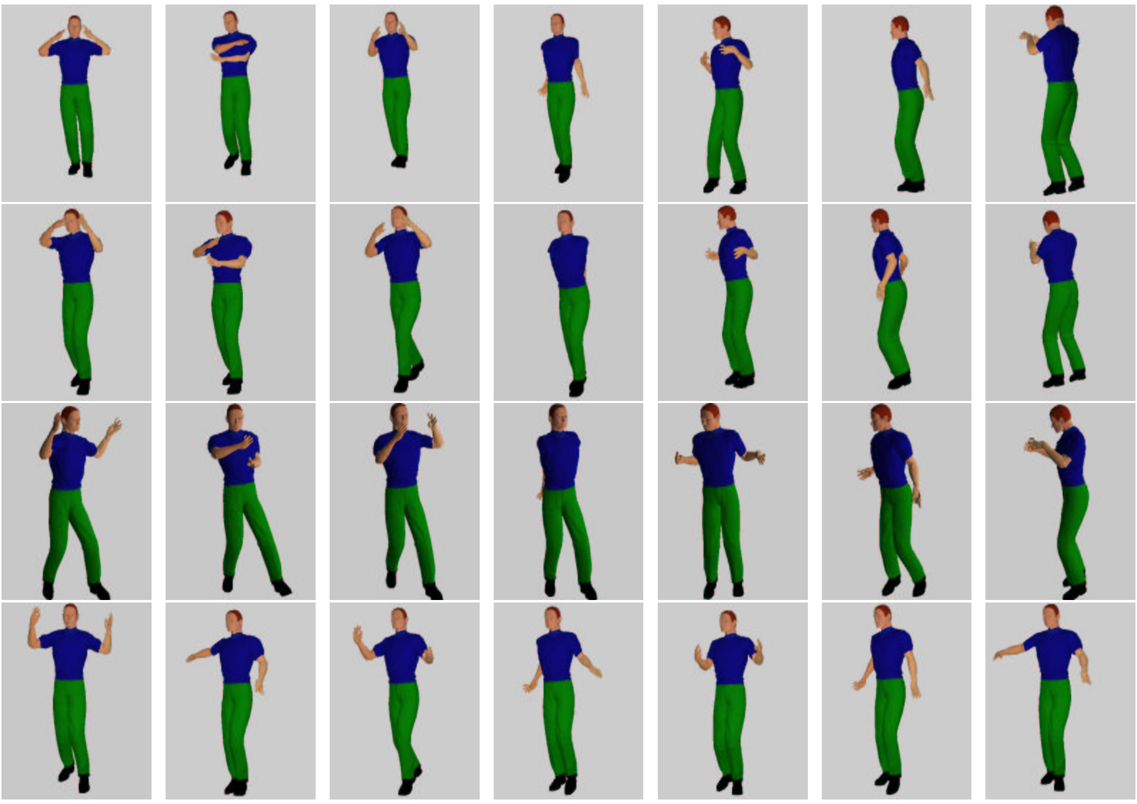

Figure 13: Conversation sequence. Estimates obtained using $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{r}_t)$. *First row:* Original image sequence. *Second row:* Most probable reconstruction (hypothesis) seen from the same viewpoint. *Third row:* Most probable reconstruction seen from a different viewpoint. *Fourth Row:* Less probable reconstruction seen from the same viewpoint as shown also on first and second rows. Notice that the perceptually incorrect estimates are trading-off accuracy of reconstructing the legs with the one of reconstructing arms. It may be the case that no better prediction is available by interpolating based on the training set. Cross-over and transplanting operations between lower and upper body parts may help enriching the training set [22], although generating physically consistent configurations may be non-trivial without a model where constraints can be expressed analytically, *e.g*. [53, 50].

[2] A. Agarwal and B. Triggs. 3d human pose from silhouettes by Relevance Vector Regression. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[3] A. Agarwal and B. Triggs. Learning to track human motion from silhouettes. In *International Conference on Machine Learning*, Banff, 2004.

[4] F. Aherne, N. Thacker, and P. Rocket. Optimal pairwise geometric histograms. In *British Machine Vision Conference*, 1997.

[5] A. Athistos and S.Sclaroff. Estimating 3d hand pose from a cluttered image. In *ICCV*, 2003.

[6] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 2002.

[7] C. Bishop and M. Svensen. Bayesian mixtures of experts. In *Uncertainty in Artificial Intelligence*, 2003.

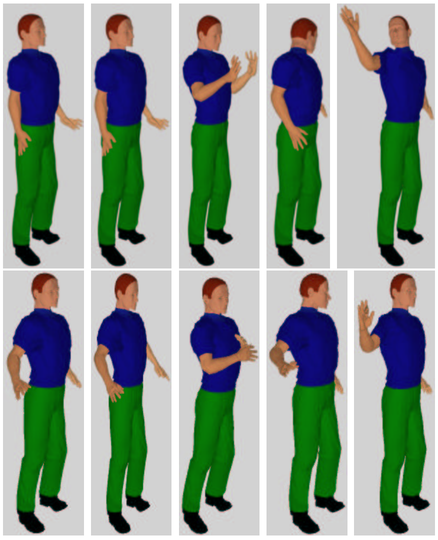

Figure 14: Tracking a pantomime sequence using $p(\mathbf{x}_t|\mathbf{r}_t)$. *First row:* Original image sequence. *Second row:* Most probable reconstruction (hypothesis) seen from the same viewpoint. image on the bottom row.

Sequence	$p(\mathbf{x}_t\|\mathbf{r}_t)$			$p(\mathbf{x}_t\|\mathbf{x}_{t-1},\mathbf{r}_t)$		
	NN	RVM	BME	NN	RVM	BME
Normal walk	4 / 20	2.7 / 12	2 / 10	7 / 25	3.7 / 11.2	2.8 / 8.1
Complex walk	11.3 / 88	9.5 / 60	4.5 / 20	7.5 / 78	5.67 / 20	2.77 / 9
Running	7 / 91	6.5 / 86	5 / 94	5.5 / 91	5.1 / 108	4.5 / 76
Conversation	7.3 / 26	5.5 / 21	4.15 / 9.5	8.14 / 29	4.07 / 16	3 / 9
Pantomime	7 / 36	7.5 / 53	6.5 / 25	7.5 / 49	7.5 / 43	7 / 41

Table 1: Comparative results showing RMS errors per joint angle (average error / maximum joint average error) in degrees for two conditional models, $p(\mathbf{x}_t|\mathbf{r}_t)$ and $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{r}_t)$. We compare three different algorithms on motion-capture, synthetically generated test data (we select the best candidate for each test input, there is no probabilistic tracking, but $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{r}_t)$ has memory). The algorithms are: NN (nearest neighbor with soft state weighing, proportional to the inverse distance to input feature), RVM (relevance vector machine), BME (Bayesian mixture of experts, with most probable mode selected). We use several training sets: walking diagonal w.r.t. to the image plane (train 300, test 56), complex walking towards the camera and turning back (train 900, test 90), running parallel to the image plane (train 150, test 150), conversation involving some hand movement and turning (train 800, test 160), pantomime (1000 train, 100 test). The training has been done separately for each sequence, to limit ambiguity, and we initialize from ground truth. This favors unimodal approaches, especially when using $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{r}_t)$, as they may not recover from an incorrect initialization. Notice that BME has typically smaller average errors and significant smaller maximum errors. The large maximum error for running seems consistent across various methods and corresponds to the right hand joint.

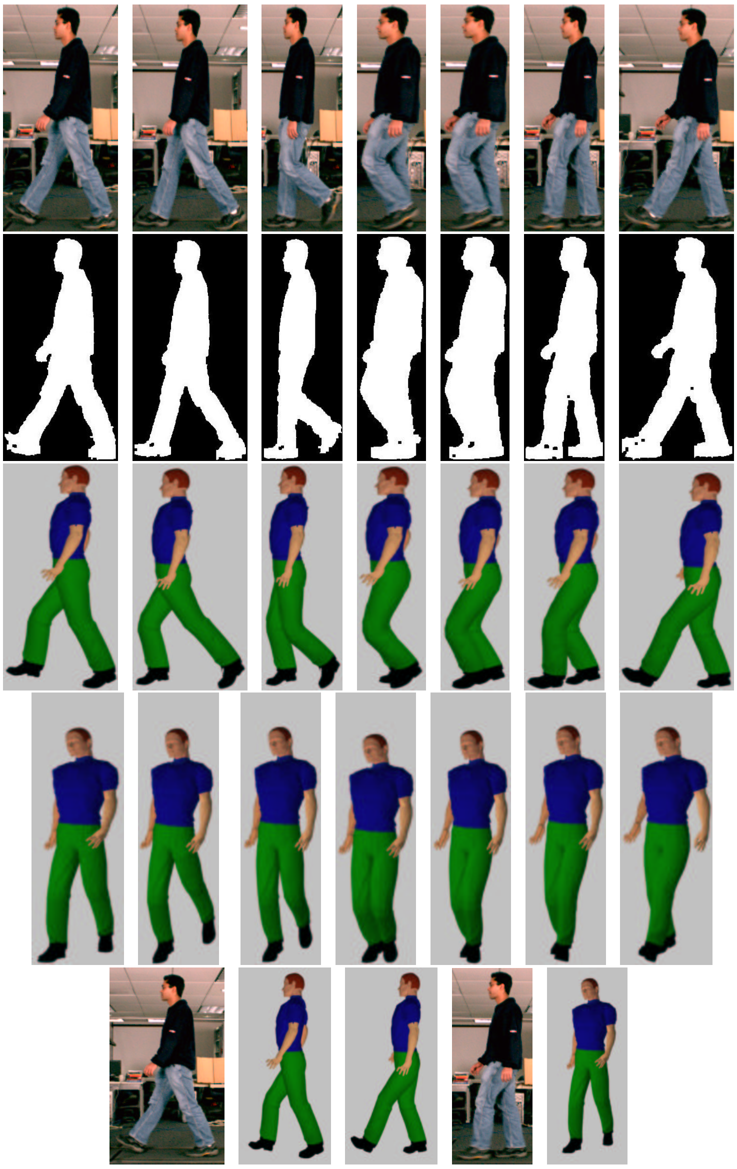

Figure 15: Reconstruction of a side-walk *First row:* original images; *Second row:* extracted silhouettes; *Third row:* reconstruction seen from the same viewpoint used in training; *Fourth row:* reconstruction seen from a synthetic viewpoint. *Fifth row:* first three images show leg assignment ambiguities; last two images show a global rotation ambiguity around vertical axis.

[8] M. Black and P. Anandan. The Robust Estimation of Multiple Motions: Parametric and Piecewise Smooth Flow Fields. *Computer Vision and Image Understanding*, 6(1):57–92, 1996.

[9] M. Brand. Shadow Puppetry. In *IEEE International Conference on Computer Vision*, pages 1237–44, 1999.

[10] C. Bregler and S. Omohundro. Non-linear Manifold Learning for Visual Speech Recogntion. In *IEEE International Conference on Computer Vision*, 1995.

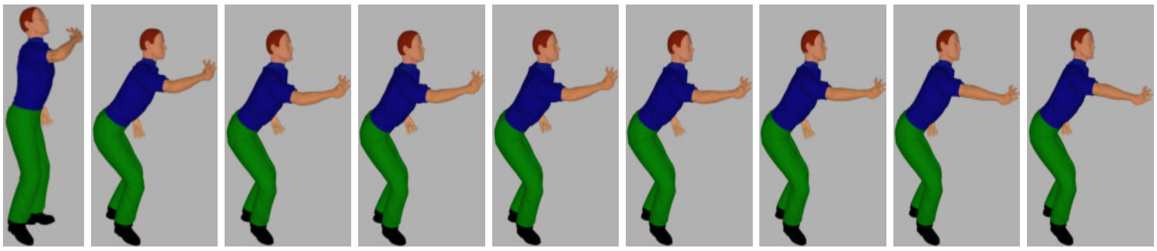

Figure 16: A unimodal Bayesian estimator based on (1) fails to reconstruct the sequence in fig. 17. The expert tracks the initial motion but eventually generates a prediction out of its input kernel firing range. In such cases, depending on its smoothness the regressor may output some mean value.

[11] M. Carreira-Perpinan. Reconstruction of sequential data with probabilistic models and continuity constraints. In *Advances in Neural Information Processing Systems*, pages 414–420, 1999.

[12] O. Cula and K.Dana. 3D texture recognition using bidirectional feature histograms. *International Journal of Computer Vision*, 59(1):33–60, 2004.

[13] W. DeSarbo and W.Cron. A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification*, (5):249–282, 1988.

[14] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2000.

[15] D. Edwards and S. Lauritzen. The TM algorithm for maximising a conditional likelihood function. *Biometrika*, 88(4):961–972, 2001.

[16] A. Elgammal and C. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *IEEE International Conference on Computer Vision and Pattern Recognition*.

[17] A. Elgammal, R.Duraiswami, D.Harwood, and L.Davis. Foreground and background modeling using non-parametric kernel density estimation for visual surveillance. *Proc.IEEE*, 2002.

[18] R. Fletcher. Practical Methods of Optimization. In *John Wiley*, 1987.

[19] N. Gordon, D. Salmond, and A. Smith. Novel Approach to Non-linear/Non-Gaussian State Estimation. *IEE Proc. F*, 1993.

[20] G. Hager and P. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[21] N. Howe, M. Leventon, and W. Freeman. Bayesian Reconstruction of 3D Human Motion from Single-Camera Video. *Advances in Neural Information Processing Systems*, 1999.

[22] L. Ikemoto and D. Forsyth. Enriching a motion collection by transplanting limbs. In *Proc. ACM Symposium on Computer Animation*, 2004.

[23] M. Isard. PAMPAS: Real-valued graphical models for computer vision. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2003.

[24] M. Isard and A. Blake. CONDENSATION – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 1998.

[25] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. *Neural Computation*, (3):79–87, 1991.

[26] T. Jebara and A. Pentland. On reversing Jensen's inequality. In *Advances in Neural Information Processing Systems*, 2000.

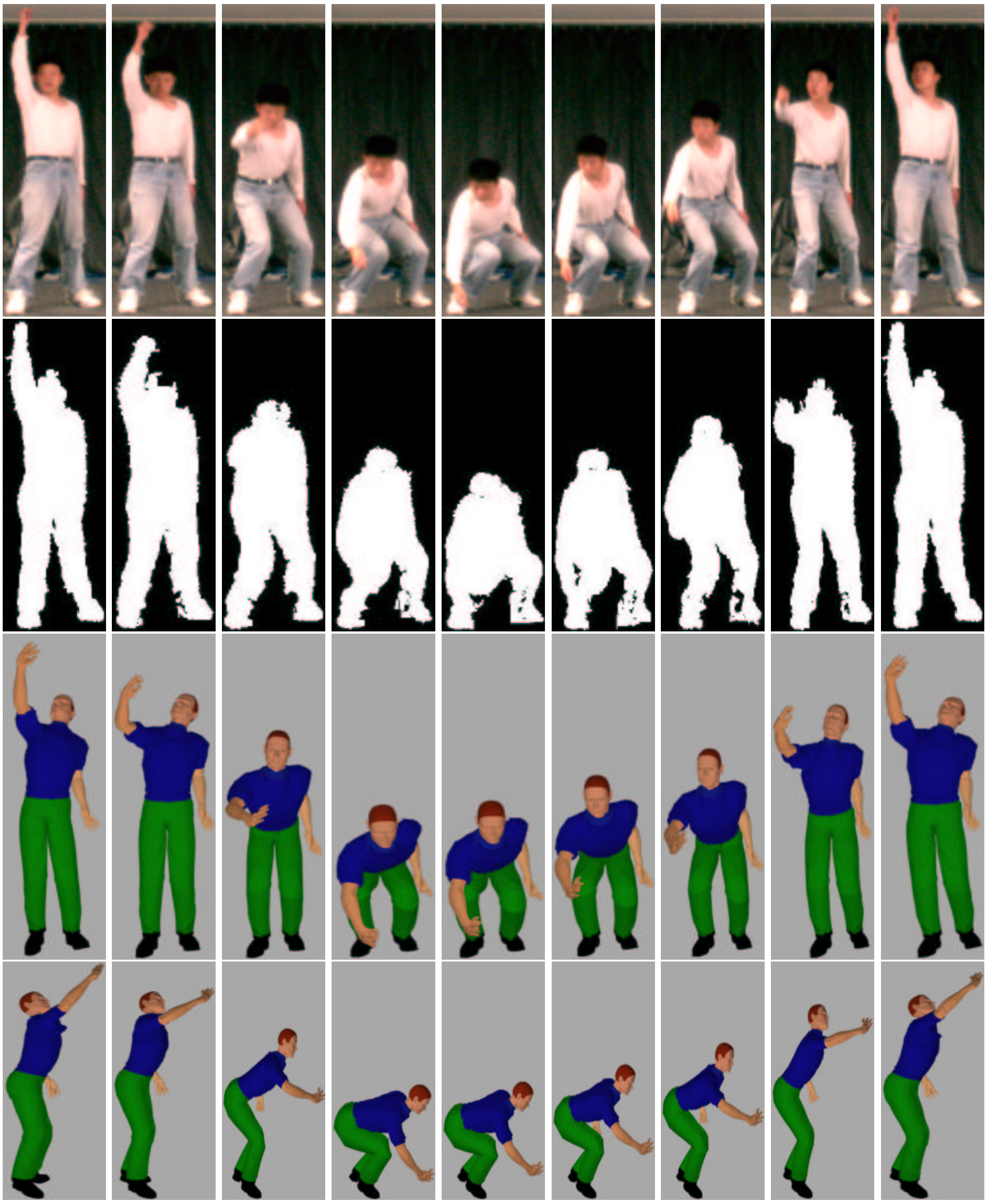

Figure 17: *First row:* Original image sequence. *Second row:* Image silhouettes. *Third row:* Reconstruction seen from the same viewpoint used for training, *Fourth row:* Reconstruction seen from a synthetic viewpoint. Notice that despite noisy silhouettes, our probabilistic tracker based on Bayesian mixture of expert (BME) conditionals can reconstruct the motion with reasonable perceptual accuracy (however, there are imperfections, *e.g*. the right knee of the subject is tilted inward, whereas the one of the model is tilted outward). A single hypothesis Bayesian tracker fails on the same sequence, see fig. 16.

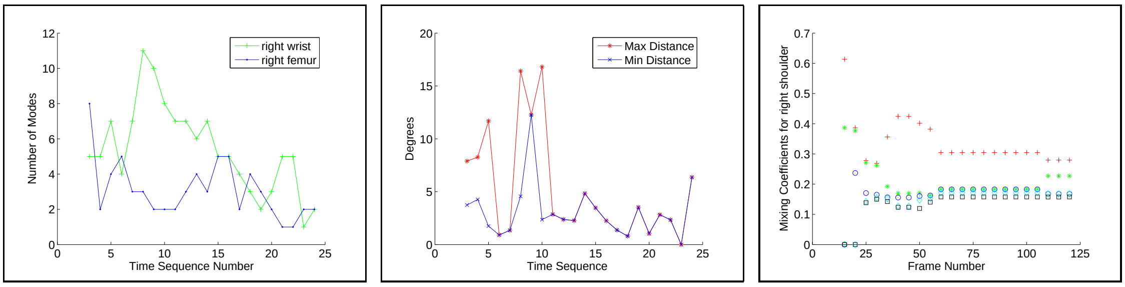

Figure 18: Quantitative results for the sequence shown in fig. 17. *(a) Left* shows the number of modes for the right femur and wrist joint angles (sampled every 6 frames). *(b) Middle* shows the maximum and minimum distance between the modes of the right wrist joint angle (sampled every 6 frames). *(c) Right* shows the mixing proportions of the right shoulder during tracking.

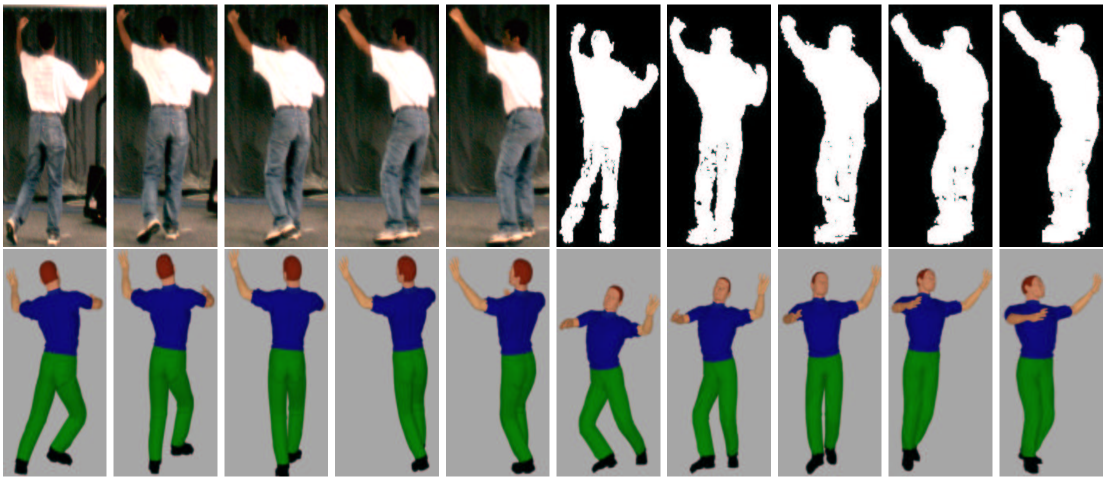

Figure 19: Tracking and reconstruction of dancing. *(a) Top row* shows original images and silhouettes; *(b) Bottom row* shows reconstructions from training (left) and new synthetic viewpoint (right).

[27] A. Jepson, D.Fleet, and T. El-Maraghi. Robust on-line appearance models for visual tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 25(10), pages 1296–1311, 2003.

[28] M. Jordan, editor. *Learning in graphical models*. MIT Press, 2001.

[29] M. Jordan and R.Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, (6):181–214, 1994.

[30] I. Kakadiaris and D. Metaxas. Model-Based Estimation of 3D Human Motion with Occlusion Prediction Based on Active Multi-Viewpoint Selection. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 81–87, 1996.

[31] K.Grauman, G.Shakhnarovich, and T.Darell. Inferring 3D structure with a statistical image-based shape model. In *IEEE International Conference on Computer Vision*, 2003.

[32] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: the informative vector machine. In *Advances in Neural Information Processing Systems*, 2003.

[33] H. J. Lee and Z. Chen. Determination of 3D Human Body Postures from a Single View. *Computer Vision, Graphics and Image Processing*, 30:148–168, 1985.

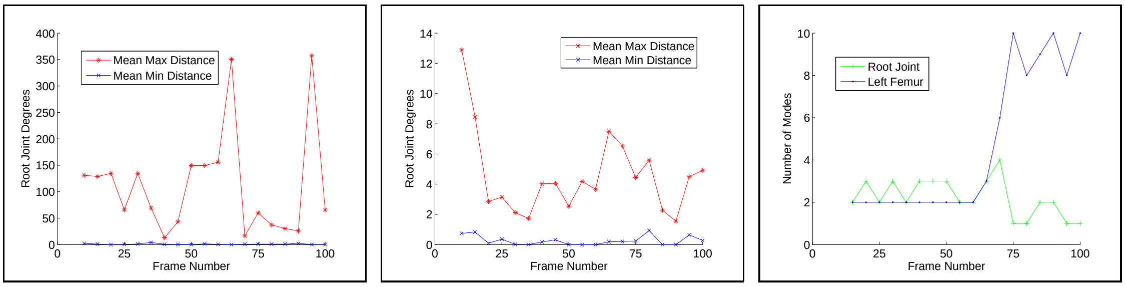

Figure 20: Quantitative tracking results for the dancing sequence. *(a) Left* shows the maximum and minimum distance for the modes of the root joint vertical axis rotation angle. The minimum distance is only informatively shown, it does not necessarily reflect modes that will survive the mixture simplification. Most likely, modes that cluster together will collapse. *(b) Middle* same as *(a)* for the left femur. *(c) Right* gives the number of modes for variables in *(a)* and *(b)* over time.

[34] D. Mackay. Bayesian interpolation. *Neural Computation*, 4(5):720–736, 1992.

[35] G. Mori and J.Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision*, 2002.

[36] R. Neal. *Bayesian learning for neural networks*. Springer-Verlag, 1996.

[37] A. Ng and M. Jordan. On discriminative versus generative classifiers. A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*, 2002.

[38] M. Osborne, B. Presnell, and B. Turlach. On the Lasso and its Dual. *J.Comput.Graphical Statist*, 9:319–337, 2000.

[39] J. Pearl. *Probabilistic Reasoning in Intelligent Systems. Networks of plausible inference*. Morgan-Kaufmann, 1988.

[40] R. Quandt and J. Ramsey. A new approach to estimating switching regressions. *Journal of the American Statistical Society*, 67:306–310, 1972.

[41] R. Rosales and S. Sclaroff. Inferring Body Pose without Tracking Body Parts. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 721–727, 2000.

[42] R. Rosales and S. Sclaroff. Learning Body Pose Via Specialized Maps. In *Advances in Neural Information Processing Systems*, 2002.

[43] S. Roth, L. Sigal, and M. Black. Gibbs Likelihoods for Bayesian Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[44] G. Seber and C. Wild. *Non-linear regression*. Willey, 1989.

[45] G. Shakhnarovich, P. Viola, and T. Darrell. Fast Pose Estimation with Parameter Sensitive Hashing. In *IEEE International Conference on Computer Vision*, 2003.

[46] H. Sidenbladh and M. Black. Learning Image Statistics for Bayesian Tracking. In *IEEE International Conference on Computer Vision*, 2001.

[47] H. Sidenbladh, M. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *European Conference on Computer Vision*, 2002.

[48] C. Sminchisescu. Consistency and Coupling in Human Model Likelihoods. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 27–32, Washington D.C., 2002.

[49] C. Sminchisescu and A. Jepson. Density propagation for continuous temporal chains. Generative and discriminative models. Technical Report CSRG-401, University of Toronto, October 2004.

[50] C. Sminchisescu and A. Jepson. Generative Modeling for Continuous Non-Linearly Embedded Visual Inference. In *International Conference on Machine Learning*, pages 759–766, Banff, 2004.

[51] C. Sminchisescu and A. Jepson. Variational Mixture Smoothing for Non-Linear Dynamical Systems. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 608–615, Washington D.C., 2004.

[52] C. Sminchisescu and A. Telea. Human Pose Estimation from Silhouettes. A Consistent Approach Using Distance Level Sets. In *WSCG International Conference for Computer Graphics, Visualization and Computer Vision*, Czech Republic, 2002.

[53] C. Sminchisescu and B. Triggs. Estimating Articulated Human Motion with Covariance Scaled Sampling. *International Journal of Robotics Research*, 22(6):371–393, 2003.

[54] C. Sminchisescu and B. Triggs. Kinematic Jump Processes for Monocular 3D Human Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 69–76, Madison, 2003.

[55] C. Sminchisescu and B. Triggs. Mapping Minima and Transitions in Visual Models. *International Journal of Computer Vision*, 61(1), 2005.

[56] E. Sudderth, A. Ihler, W. Freeman, and A.Wilsky. Non-parametric belief propagation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2003.

[57] C. J. Taylor. Reconstruction of Articulated Objects from Point Correspondences in a Single Uncalibrated Image. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 677–684, 2000.

[58] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *J. Roy. Statist.Soc*, B58(1):267–288, 1996.

[59] M. Tipping. Sparse Bayesian learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, 2001.

[60] C. Tomasi, S.Petrov, and A.Sastry. 3d tracking = classification + interpolation. In *IEEE International Conference on Computer Vision*, 2003.

[61] K. Toyama and A. Blake. Probabilistic Tracking in a Metric Space. In *IEEE International Conference on Computer Vision*, 2001.

[62] N. Ueda and Z. Ghahramani. Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, 15:1223–1241, 2002.

[63] Q. Wang, G. Xu, and H. Ai. Learning Object Intrinsic Structure for Robust Visual Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2003.

[64] S. Waterhouse, D.Mackay, and T.Robinson. Bayesian methods for mixtures of experts. In *Advances in Neural Information Processing Systems*, 1996.

[65] O. Williams, A. Blake, and R. Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *IEEE International Conference on Computer Vision*, 2003.

[66] L. Xu, M. Jordan, and G. Hinton. An alternative model for mixture of experts. In *Advances in Neural Information Processing Systems*, 1995.