Bayesian Adversarial Human Motion Synthesis

Rui Zhao*
Amazon
zhaori@amazon.com

Hui Su
RPI and IBM Research
huisuibmres@us.ibm.com

Qiang Ji
RPI
qji@ecse.rpi.edu

Abstract

We propose a generative probabilistic model for human motion synthesis. Our model has a hierarchy of three layers. At the bottom layer, we utilize Hidden semi-Markov Model (HSMM), which explicitly models the spatial pose, temporal transition and speed variations in motion sequences. At the middle layer, HSMM parameters are treated as random variables which are allowed to vary across data instances in order to capture large intra- and inter-class variations. At the top layer, hyperparameters define the prior distributions of parameters, preventing the model from overfitting. By explicitly capturing the distribution of the data and parameters, our model has a more compact parameterization compared to GAN-based generative models. We formulate the data synthesis as an adversarial Bayesian inference problem, in which the distributions of generator and discriminator parameters are obtained for data synthesis. We evaluate our method through a variety of metrics, where we show advantage than other competing methods with better fidelity and diversity. We further evaluate the synthesis quality as a data augmentation method for recognition task. Finally, we demonstrate the benefit of our fully probabilistic approach in data restoration task.

1. Introduction

A model that can describe the dynamic process of pose change is useful in motion analysis tasks such as simulation, restoration and prediction. Recently, generative dynamic models [8, 9, 12, 26, 35, 38, 40, 41, 44, 48] have attracted increasing attention for their capability in generating data that resemble real data distribution. Despite the substantial progress made, the modeling of pose dynamics remains challenging due to several reasons. First, there are significant intra-class and inter-class variations in sequential data. While the inter-class variation is caused by distinct dynamic pattern of different motions, the intra-class variation is mainly caused by different spatial extents and temporal paces of executing the motion. Second, the data may be noisy and can have missing values. Obtaining accurate pose requires expensive motion capture (mocap) system. The alternative is to estimate pose from video data, which is still an open research problem [30, 52]. Furthermore, neither approaches are error-free. Thus, a framework that can handle all these issues in a principled manner is required.

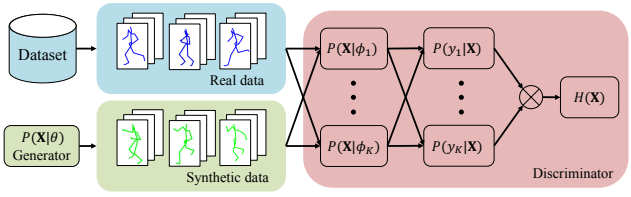

Figure 1. Overview of our framework. The generator is the proposed probabilistic dynamic model. The discriminator consists of K models that jointly determine the class probability of data.

In this work, we propose a generative hierarchical probabilistic dynamic model that explicitly models the spatial and temporal variations in human motion sequences. Compared to conventional state-space model, our method leverages Bayesian framework to improve the model capacity *i.e.* the ability to model variations in data. The model parameters are treated as random variables whose distributions are further governed by prior distributions. The resulting hierarchical dynamic model increases the model capacity for capturing large intra- and inter-class variations. Compared to deterministic models such as RNN/LSTM [8, 35, 50] or extensions of GAN [32, 38, 40], the proposed model is fully probabilistic with a compact parameterization. Therefore, it requires less training compared to purely data-driven deep models and is less susceptible to overfitting.

In the last few years, adversarial learning emerges to be a major framework in synthesizing data of different modalities including image, text, audio, and video [10, 12, 19, 21, 22, 25, 27, 28, 32, 39, 46]. One major issue with adversarial learning is the mode-collapsing, where the generator tends to generate the same data which can fool the discriminator well. To address this issue, we propose to combine Bayesian inference with adversarial learning. Instead of estimating a single set of parameters, Bayesian inference ob-

*This work was performed while at RPI as a student.

tains multiple sets of parameters from the posterior distribution so that the generator can better explore the parameter space and alleviate mode-collapsing. The overall framework is shown in Figure 1. In order to evaluate the model performance, we also adapt several evaluation metrics that can quantitatively measure the fidelity and diversity of the generated data besides qualitative inspection.

Our specific contributions are 1) a generative hierarchical probabilistic dynamic model that explicitly models large spatial and temporal variations in human motion sequences; 2) a unified adversarial Bayesian inference framework on the proposed dynamic model for realistic motion sequence synthesis; 3) a set of quantitative evaluations on the synthetic motion sequence quality.

2. Related Work

Probabilistic graphical models (PGM) have been widely used for modeling human motion sequences, including autoregressive model (AR) [43], hidden Markov model (HMM) [2], dynamic Bayesian networks (DBN) [15], switching linear dynamic system (SLDS) [17], *etc.* In particular, semi-Markov models [6, 20, 51], which relax the Markov dynamics assumption, have shown improved performance on human activity recognition. However, the capacity of modeling dynamics in these models is often limited since the source of variations only comes from the conditional probability distribution of random variables. Once the model is learned, the parameters do not change. More expressive probabilistic models have been proposed such as the conditional restricted Boltzmann machine (CRBM) [36, 37], which exploits a vectorized hidden states to encode the dynamics. However, the increased expressiveness is gained at the cost of requiring condition on proper choice of initial state. The exact learning also becomes intractable. Nonparametric extensions of dynamic PGM have been proposed in [7, 14, 29], which allow the hidden state size to adapt according to data.

More recently, neural networks (NN) based models become a popular alternative for dynamic data synthesis, including RNN-based approaches [8, 18, 19, 26], CNN-based approaches [16, 44, 45] and sigmoid network [9]. Following the success in synthesizing realistic-looking images, variants of generative adversarial networks (GAN) have been proposed to synthesize dynamic data including texts and videos [32, 39, 40, 41, 46]. NN-based models require a large amount of training data as the models do not explicitly consider the cause of variations in dynamic process and rely on a purely data-driven manner. Simply reducing the number of parameters will compromise the capabilities of modeling dynamics.

To model human motion sequences, we develop a hierarchical probabilistic dynamic model which explicitly models the major sources of variations including spatial pose, temporal transition and execution speed. Leveraging on Bayesian framework, the model capacity is enhanced by allowing parameters to vary. We further enhance the adversarial learning framework in two aspects. First, a novel objective of adversarial learning is used to handle multiple classes of data. Second, we formulate the synthesis as a Bayesian inference problem, which allows us to generate data based on the distribution of parameters. Thus, we can reduce overfitting and alleviate mode-collapsing. Finally, the proposed hierarchical model needs only a fraction of the number of parameters compared to NN-based model such as RNN, yet achieves a more competitive performance. Combining adversarial learning with PGM is first proposed in [49], followed by [4]. Our work has several differences compared to [49]. First, we use HSMM rather than HMM as our base generative model. More importantly, we incorporate Bayesian inference into adversarial learning framework. We sample model parameters from the posterior distribution instead of prior distribution as in [49], which is less desirable if the prior distribution is not properly specified or estimated. Finally, we can synthesize multiple classes of actions with one model while [49] needs to learn separate models to synthesize different actions. We differ from Li *et al.* [4] in using Bayesian inference instead of computing point estimate for PGM parameters.

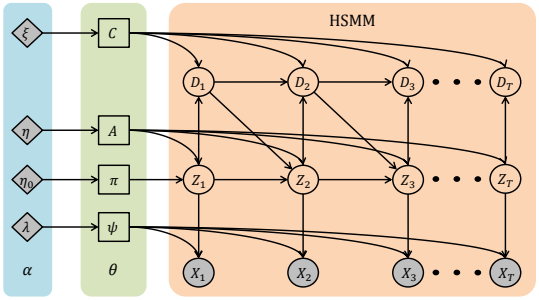

Figure 2. Topology of BH-HSMM. The values of shaded circle nodes are observed and the shaded diamond nodes are fixed.

3. Models

In this section, we describe the proposed Bayesian Hierarchical HSMM (BH-HSMM) model, starting with a description of HSMM and followed by Bayesian extension.

HSMM is an extension to HMM by allowing more flexible modeling of the duration within each state. Figure 2 shows the topology of HSMM. The circle nodes are random variables. Specifically, $\mathbf{X} = \{X_t \in \mathbb{R}^O, t = 1, ..., T\}$ represents a sequence of continuous-valued observations, $\mathbf{Z} = \{Z_t \in \{1, ..., Q\}, t = 1, ..., T\}$ represents a sequence of discrete hidden states associated with observations, and $\mathbf{D} = \{D_t \in \{1, ..., L\}, t = 1, ..., T\}$ represents the corresponding discrete duration of states. O is the dimension of observations. Q is the cardinality of hidden states. L is the

maximum duration of any state. T is the length of sequence. The joint distribution of HSMM can be written as follows.

$$P(\mathbf{X},\mathbf{Z},\mathbf{D}) = P(Z_1,D_1)P(X_1|Z_1)\prod_{t=2}^{T}[P(X_t|Z_t) \quad (1)$$
$$P(D_t|D_{t-1},Z_t)P(Z_t|Z_{t-1},D_{t-1})]$$

Eq. (1) includes four components for parameterizing HSMM, namely initial state distribution, transition distribution, duration distribution, and emission distribution. Different variants of HSMM have been proposed in the literatures. Readers are referred to [47] for a thorough review.

One major limitation of HSMM is the modeling capacity. The number of states needed to encode the dynamics is exponential to the cardinality of variations. To model the dynamics in human motion, we propose a Bayesian hierarchical extension to HSMM, which has a much larger modeling capacity than its non-hierarchical counter-part. Specifically, the model consists of three layers of nodes. As shown in Figure 2 from right to left, the first layer consists of conventional random variables modeling physical or semantic quantities, which are the $\mathbf{X},\mathbf{Z},\mathbf{D}$. The second layer consists of model parameters, which are $\theta = \{\pi, A, C, \psi\}$. Following Bayesian framework, the parameters are also treated as random variables, which can vary across different data samples. The third layer consists of hyperparameters $\alpha = \{\eta_0, \eta, \xi, \lambda\}$, which specify the prior distribution of parameters. We choose conjugate prior for each parameter for the simplicity of computation. The hyperparameters are chosen so that the prior is non-informative. The marginal likelihood of BH-HSMM is as follows with specific parameterization provided in the supplementary materials.

$$P(\mathbf{X}|\alpha) = \int_\theta \sum_{\mathbf{Z},\mathbf{D}} P(\mathbf{X},\mathbf{Z},\mathbf{D}|\theta)P(\theta|\alpha)d\theta \quad (2)$$

Notice that θ is of order $\mathcal{O}(Q(Q+L+O^2))$ and α is of order $\mathcal{O}(Q(Q+L)+O^2)$. The parameterization increase due to α is marginal in our experiment where $O > Q, L$.

4. Methods

4.1. Adversarial Learning

As an alternative of learning generative models, adversarial learning [10] formalizes a mini-max game where two models namely generator G and discriminator D compete against each other. While G tries to generate data as realistic as possible, D tries to differentiate the synthetic data from the real. In conventional adversarial learning the discriminator only makes binary decision on whether the data is real or not. We instead propose to use the following objective for adversarial learning to account for different action types.

$$\min_\theta \max_\phi \; -\mathbb{E}_{P_{data}(\mathbf{X})}[H(\mathbf{X}|\phi)] + \mathbb{E}_{P(\mathbf{X}|\theta)}[H(\mathbf{X}|\phi)] \quad (3)$$

where θ and ϕ are the parameters of generator and discriminator, respectively. $P_{data}(\mathbf{X})$ is the real data distribution, which is represented by the dataset and never explicitly defined. $H(\mathbf{X}|\phi) \triangleq -\sum_y P(y|\mathbf{X},\phi)\log P(y|\mathbf{X},\phi)$ is the Shannon entropy and y is a discrete label of a sequence $\mathbf{X}$. We assume the number of action categories is known. Since the entropy is computed based on class probability, to minimize the entropy, the generator needs to generate data that belongs to exactly one of the action class, yielding realistic motion. A similar objective is also proposed in [34]. We use different generator and discriminator in order to handle sequential data.

Choice of generator and discriminator: We use the proposed BH-HSMM as the generator. The discriminator consists of a set of BH-HSMMs. Each BH-HSMM models one type of action. The number of hidden states for each model is the same and the value is chosen such that together, they have about similar modeling capacity to the generator and enough discriminative power. Denote the parameters of k^{th} BH-HSMM as ϕ_k, then we can compute the probability of $\mathbf{X}$ belonging to k^{th} class as $P(y=k|\mathbf{X},\phi) = \frac{P(\mathbf{X}|\phi_k)}{\sum_{j=1}^{K} P(\mathbf{X}|\phi_j)}$. The probabilistic model based discriminator allows better modeling of the randomness in dynamic data.

4.2. Bayesian Adversarial Inference

A common pitfall of adversarial learning is the mode-collapsing [33], where generator only generates similar-looking data that successfully fool the discriminator. We propose to combine Bayesian inference with adversarial learning in order to alleviate this issue. The key idea of Bayesian inference is to treat parameters as random variables, whose prior distributions are specified by the hyperparameters. Instead of finding one single set of parameters as in conventional adversarial learning, we obtain multiple sets of parameters drawn from their posterior distributions. Combining Bayesian inference with adversarial learning was first proposed in [31] for GAN. We extend this framework to the proposed BH-HSMM. To the best of our knowledge, this is the first work to integrate Bayesian inference and adversarial learning for dynamic PGM. This allows sequence synthesis. We also proposed a novel objective Eq. (3) in order to support multi-class data synthesis using one generator. Specifically, we generate samples of generator parameters θ and discriminator parameters ϕ from the posterior distribution.

$$\theta, \phi \sim P(\theta, \phi|\mathcal{D}^+, \alpha_g, \alpha_d) \quad (4)$$

where $\mathcal{D}^+$ is the real data. α_g and α_d are the hyperparameters of generator and discriminator parameters, respectively.

Following the idea of Gibbs sampling, we generate samples of θ, ϕ by sampling alternatively between the conditional posteriors of θ and ϕ. Specifically, the conditional posterior of θ is as follows.

$$\begin{aligned}\theta \sim P(\theta|\mathcal{D}^+, \alpha_g, \alpha_d, \phi) &\propto P(\phi|\theta)P(\theta|\alpha_g) \\ &\propto \prod_i \exp\{-H(\mathbf{X}_i^-|\phi)\}P(\theta|\alpha_g)\end{aligned} \tag{5}$$

where $\mathbf{X}_i^- \sim P(\mathbf{X}|\theta)$. The first line of Eq. (5) results from the posterior being proportional to the product of likelihood and prior. For likelihood we assume θ is marginally independent with $\mathcal{D}^+$ and α_d. This is consistent with Eq. (3), in which θ is only involved in the second term. From the first line to the second line, we assume the likelihood of the generator satisfies $P(\phi|\theta) \propto \prod_i \exp\{-H(\mathbf{X}_i^-|\phi)\}$. Intuitively speaking, more realistic $\mathbf{X}_i^-$ yields a smaller value of H, thus a higher likelihood. Following a similar argument, the conditional posterior of ϕ is as follows.

$$\begin{aligned}\phi \sim P(\phi|\mathcal{D}^+, \alpha_g, \alpha_d, \theta) &\propto P(\mathcal{D}^+, \theta|\phi)P(\phi|\alpha_d) \\ &\propto \prod_i \exp\{-H(\mathbf{X}_i^+|\phi)\} \prod_j \exp\{H(\mathbf{X}_j^-|\phi)\}P(\phi|\alpha_d)\end{aligned} \tag{6}$$

where $\mathbf{X}_i^+ \in \mathcal{D}^+$ and $\mathbf{X}_j^- \sim P(\mathbf{X}|\theta)$. From the first line to the second line of Eq. (6), we assume the likelihood $P(\mathcal{D}^+, \theta|\phi) \propto \prod_i \exp\{-H(\mathbf{X}_i^+|\phi)\} \prod_j \exp\{H(\mathbf{X}_j^-|\phi)\}$. Intuitively, a better ϕ yields a smaller value of $H(\mathbf{X}_i^+|\phi)$ and a larger value of $H(\mathbf{X}_j^-|\phi)$, thus an overall higher likelihood. We depict the assumed conditional dependency using a graphical model as shown in Figure 3.

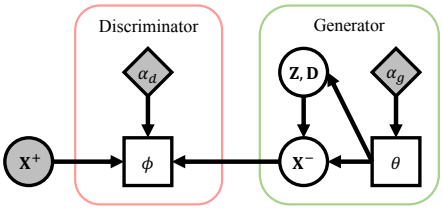

Figure 3. PGM interpretation of Bayesian adversarial model.

The exact inference of posterior distribution is intractable due to the intractable normalization constant. We resort to SGHMC [3] to perform approximate inference. For the generator, based on Eq. (5), we define the negative potential energy function $L_g(\theta) \triangleq -\sum_{j=1}^{n_g} H(\mathbf{X}_j^-|\phi) + \log P(\theta|\alpha_g)$ such that $P(\theta|\alpha_g, \phi) \propto \exp\{L_g(\theta)\}$. Notice that n_g is the number of synthetic data samples in each mini-batch of stochastic gradient update. For the discriminator, based on Eq. (6), we define the negative potential energy function $L_d(\phi) \triangleq -\sum_{i=1}^{n_d} H(\mathbf{X}_i^+|\phi) + \sum_{j=1}^{n_g} H(\mathbf{X}_j^-|\phi) + \log P(\phi|\alpha_d)$ such that $P(\phi|\mathcal{D}^+, \alpha_d, \theta) \propto \exp\{L_d(\phi)\}$. Here n_d and n_g are the number of real and synthetic data samples in each mini-batch, respectively. The derivation of gradient of $L_g(\theta)$ and $L_d(\phi)$ is provided in the supplementary materials. To perform each SGHMC update, we use the momentum-based gradient update. We fix the values of hyperparameters α_g, α_d to yield a non-informative prior. The overall algorithm is summarized in Algorithm 1 with the choice of constants listed in the supplementary materials.

Algorithm 1 Bayesian adversarial inference of BH-HSMM

Input: $\{\mathbf{X}\}$: real dataset. Q_g: generator hidden state number. Q_d: discriminator hidden state number. M: number of samples per mini-batch. K: class number. a: momentum coefficient. Hyperparameters: α_d, α_g. τ: gap of iterations between different samples. η: learning rate.

Output: Samples of BH-HSMM parameters

```
Initialization of θ, φ, V_g, V_d.
repeat
    Draw M data samples from generator and real dataset.
    for k = 1, ..., K do
        V_d^k ← (1 − a)V_d^k + η ∂L_d(φ)/∂φ_k + ε ∼ N(0, 2aηI)
        φ_k ← φ_k + V_d^k
    end for
    Draw M data samples from generator.
    V_g ← (1 − a)V_g + η ∂L_g(θ)/∂θ + ε ∼ N(0, 2aηI)
    θ ← θ + V_g
    Collect θ every τ iterations after burn-in
until collect enough samples
return {θ}
```

4.3. Data Synthesis

Generating a sequence from our model is a conceptually simple process as we have a directed dynamic model and ancestral sampling can be used to generate a sample. First, we obtain samples of model parameters θ from the posterior as described in the previous section. Second, we sample the hidden state chain $\{\mathbf{Z}, \mathbf{D}\}$ given one θ. Third, we sample the observation sequence $\mathbf{X}$ given $\{\mathbf{Z}, \mathbf{D}\}$ and the corresponding θ. Since θ is drawn from the posterior, by repeating the three steps multiple times, we have $\mathbf{X} \sim P(\mathbf{X}|\mathcal{D}^+, \alpha)$. Notice that for the third step, $\mathbf{X}$ drawn from $P(\mathbf{X}|\mathbf{Z})$ will be too noisy to look realistic in general. We improve the synthesis quality by estimating the most likely observation as follows.

$$\begin{aligned}\mathbf{X}^* &= \arg\max_{\mathbf{X}} \log P(\tilde{\mathbf{X}}|\mathbf{Z}) \\ &= \arg\max_{\mathbf{X}} \sum_t \log N(\tilde{X}_t|\mu_{Z_t}, \Sigma_{Z_t})\end{aligned} \tag{7}$$

where $\tilde{X}_t = [X_t, X_t - X_{t-1}]$ is the augmented observation vector by including the speed of each feature channel. $N(\cdot)$ is Gaussian distribution used in our parameterization. We adopt this formulation as suggested in [2] to improve the smoothness. While the speed is included as part of observations, Eq. (7) remains as a quadratic system of $\mathbf{X}$ and thus can be solved analytically.

5. Experiments

We demonstrate the capability of BH-HSMM and the benefit of Bayesian adversarial inference using four sets of experiments. First, we perform an experiment on synthetic data to show reduced mode-collapsing. Second, we perform data synthesis on real mocap data. Both quantitative and qualitative results are analyzed. Third, we use synthesized data as augmentation for action recognition. Finally, we perform an experiment to restore missing joint angles in mocap data.

5.1. Synthetic Data Experiment

We first demonstrate the benefit of adversarial Bayesian inference using a synthetic dataset. The data is 2D and the distribution is shown in Figure 4 (left), which resides along the edges of a square. Figure 4 (middle) shows the synthesis result obtained by a single set of parameters estimated by adversarial learning, which produces samples that concentrate on the right edge. Figure 4 (right) shows the synthesis result obtained by Bayesian adversarial inference, which improves the coverage of sample distribution without ignoring the minor edges. This result indicates that Bayesian inference alleviates mode-collapsing. While the single model only concentrates on the dominant mode.

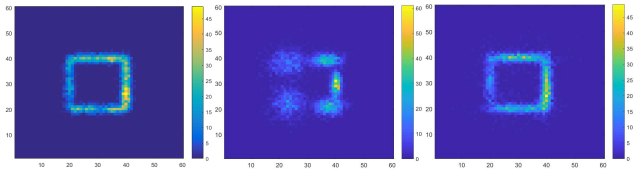

Figure 4. 2D histogram of 1000 synthetic data points. Left: actual data distribution. Middle: synthesized data distribution from adversarially learned model. Right: synthesized data distribution from Bayesian adversarial inference model.

5.2. Mocap Data and Pre-processing

CMU Motion Capture [5]: The original dataset contains 23 categories of motions performed by 144 actors with various variations. We selected a subset of the dataset including *walking, running* and *boxing* actions from 10 subjects with a total number of 166 sequences. **Berkeley MHAD** [23]: The dataset contains 11 locomotions, involving full or partial body motions. Each motion is performed 5 times by 12 subjects, resulting 660 sequences in total. For both datasets, data are provided in the form of joint angles. In CMU dataset, there are 31 joints with 59 angles. In Berkeley dataset, there are 35 joints with 90 angles. We use joint angles instead of joint positions as they provide a body-size and location invariant representation of the motion. We divide each sequence into overlapping subsequences with fixed length. Finally, we normalize the value of each angle by subtracting angle-wise mean and dividing angle-wise standard deviation.

Table 1. Number of parameters and training time in synthesis experiment for different methods on Berkeley: 1. CRBM; 2. TSBN; 3. RRNN; 4. C-RNN-GAN; 5. HHMM; 6. Ours (with burn-in).

Method	1	2	3	4	5	6
Parameters	207k	124k	3374k	2166k	44k	30k
Time (h)	11.9	9.7	19.2	20.7	5.1	3.5

5.3. Mocap Data Synthesis

We demonstrate the benefit of Bayesian adversarial inference in generating motion capture sequences. We compare with five state-of-the-art methods. CRBM [37] represents the conventional PGM-based approach. TSBN [9] represents a combination of NN and PGM. Two RNN/LSTM based approaches are considered. RRNN [18] is trained by minimizing reconstruction loss and C-RNN-GAN [19] is trained using adversarial loss. Finally, HHMM [49] is the most similar method to ours. For baseline, we compare with HSMM and MAP estimate of BH-HSMM, where a single set of parameters is used for synthesis. We compare the number of parameters and training time for different methods in Table 1, which shows the compactness and efficiency of our model. In particular, C-RNN-GAN has about 72 times of parameters and 6 times of training time to ours. For our approach, training refers to posterior inference of parameters. We burn-in the first 10 epochs and each epoch contains 40 gradient updates. We found this sufficient for a stable results. After burn-in, we collect parameter sample every 5 updates.

The evaluation of synthesized data quality remains a challenging issue. One existing practice is through user study [15, 40], which has a large variation in subjective judgment and is time consuming. We propose to utilize a collection of automated evaluation metrics to assess the sequential data quality based on three criteria. First, the fidelity in presenting a realistic-looking motion pattern. Second, the diversity in assembling intra- and inter-class variations. Third, the similarity in overall distribution. For each method of comparison we synthesize 1000 sequences, from which the metrics are computed. We now introduce each metric and the corresponding results, followed by some qualitative results.

BLEU [24] is originally proposed to evaluate the quality of machine translation. It measures the overall consistency in semantics between a translated sentence and a set of reference sentences. It is defined as $\beta \exp(\sum_{n=1}^{N} w_n \log p_n)$, where β is the brevity penalty coefficient, p_n is the modified n-gram precision, w_n is the weight associated with n-gram, and N is the maximum length of n-gram. The motion pattern can be viewed as a sequential composition of poses.

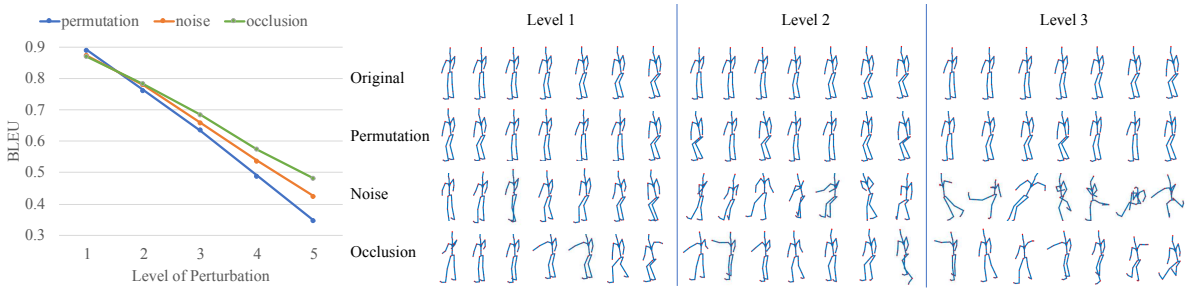

Figure 5. Average BLEU and examples of real mocap sequences under different levels and types of perturbation. (Best view in color)

Thus, the fidelity of synthetic mocap sequences can be evaluated by the similarity to pose sequences of real data. We first quantize real and synthetic data using K-means. Then we compute BLEU score of each synthetic sequence using quantized real sequences as reference. We use NLTK [1] with a maximum of 4-gram and uniform weight. The higher the BLEU between 0 and 1, the better the fidelity.

We conduct several experiments in order to justify the validity of using BLEU to measure the quality of mocap data. We perform the experiment on Berkeley dataset and use $K = 20$. First, we compute BLEU of real mocap sequences, still motion sequences (by replicating one real pose), and randomly generated pose sequences. The results are 0.9922, 0.3602 and 0.1076, respectively. This indicates that real sequences have almost perfect value and random sequences has the worst value. Although the replicated pose is realistic, it is semantically meaningless, resulting a low BLEU. Second, we apply different perturbations to the real data including random permutation, adding Gaussian noise, and random occlusion. Figure 5 shows the results of perturbed data and corresponding BLEU. Detailed discussion on perturbation is provided in the supplementary materials. The results indicate that a high BLEU requires both realistic pose and meaningful sequential order.

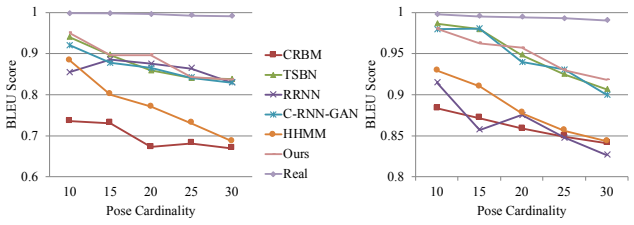

Figure 6. Average BLEU score vs. quantization cardinality in CMU (left) and Berkeley (right) dataset. Results of random sequences are smaller than 0.6 and thus omitted. (Best view in color)

Our empirical evaluation shows that BLEU is a reasonable metric to evaluate the semantic consistency of synthesized mocap sequences. Figure 6 shows the average BLEU scores of all synthetic sequences with varied cardinality K. For all methods, BLEU are decreasing as the cardinality increases, this is expected since the fewer the pose cardinalities, the easier to find n-gram matches. Our method generally achieves higher BLEU score. The average BLEU over 7 different cardinalities on CMU dataset is 0.8830 while the second best is TSBN with a score of 0.8786. On Berkeley, we achieve 0.9491 and the second best is C-RNN-GAN with 0.9443. On both datasets we outperform HHMM by a large margin, which shows the benefit of using semi-Markov dynamics in synthesis task. These results show our model can generate sequences that preserve fidelity well.

Inception Score (IS) [33] is a popular metric that evaluates both the diversity and realism of the synthetic data. IS utilizes another pre-trained classifier to classify the synthetic data. The score is defined as $\exp\{\mathbb{E}_{\mathbf{X}}[KL(P(y|\mathbf{X})||P(y))]\}$, where y is the class label associated with data $\mathbf{X}$. $P(y|\mathbf{X})$ is the condition label probability produced by the classifier. $P(y)$ is the marginal label probability over all synthetic data. KL is the KL-divergence. In practice, both the expectation over $\mathbf{X}$ and the integration in KL are approximated using a sample average. The inception score is a positive number. One score is obtained for the entire set of synthetic sequences. A high score indicates that the model can generate more diverse and discernible data across different classes. In our experiment we learn a set of HMMs by maximizing likelihood, one for each action category, as a classifier for handling sequential data. We use the normalized likelihood computed by HMMs as the label probability for a testing sequence.

By repeating the synthesis process 10 times, we report both the mean and standard deviation of inception scores in Table 2. We observe that in both datasets, the real data achieve the highest scores. The Berkeley dataset has a higher score because there are more actions. Our method achieves the best score among all the methods. Besides, using Bayesian inference ('Ours-FB') shows consistent improvement over MAP estimation ('Ours-MAP'), which indicates better diversity of the generated data.

It is helpful for analysis to combine the results of BLEU and IS. Our method achieves the best performance in both scores in both datasets, which indicates a good balance between fidelity and diversity. Although TSBN and C-RNN-

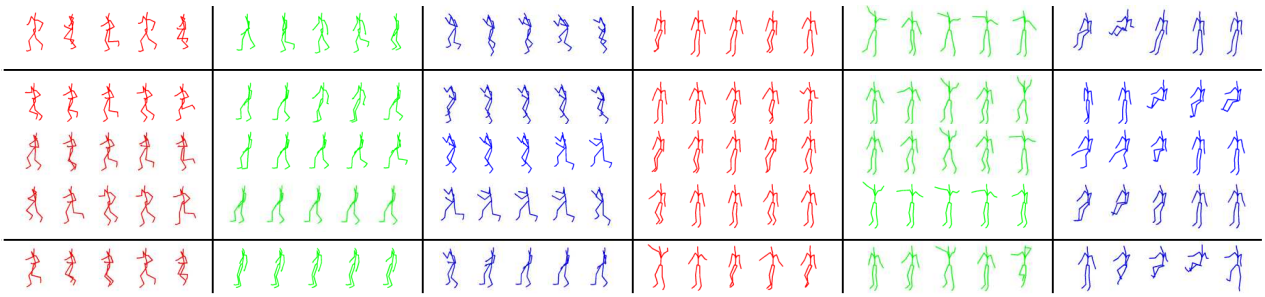

Figure 7. Synthetic motion sequences. Each row in each cell represents a complete motion sequence. The first row contains real sequences. The second to fourth rows contain realistic looking synthetic sequences. The last row contains synthetic sequence with either erroneous pose or incoherent motion pattern. The action categories from left to right are *Running, Walking, Boxing, Jumping, Jumping Jack, Sitting & Standing*. The first three actions are from CMU dataset and the rest are from Berkeley dataset. (Best view in color)

GAN achieve close BLEU to ours, they have much lower IS, which indicate the generated data has high fidelity but low diversity. RRNN achieves a higher IS but lower BLEU than C-RNN-GAN, which indicates that the generatively learned RNN model has an overall better diversity and lower fidelity than adversarially learned model. CRBM achieves the second best IS in Berkeley dataset. But BLEU is much worse than ours, which indicates the fidelity is not good. We outperform HHMM consistently in both metrics, which shows the benefit of using Bayesian inference.

Table 2. IS (higher is better) and MMD (lower is better) mean ± standard deviation. Method ID: 1. HSMM; 2. CRBM; 3. TSBN; 4. RRNN; 5. C-RNN-GAN; 6. HHMM; 7. Ours-MAP; 8. Ours-FB; 9. Real data (for IS) and Gaussian noise (for MMD).

ID	IS		MMD	
	CMU	Berkeley	CMU	Berkeley
1	1.86±0.07	4.99±0.27	5.46±0.62	432.25±0.78
2	2.65±0.09	5.24±0.39	7.43±0.97	55.39±0.75
3	2.58±0.04	2.57±0.14	12.74±0.10	110.55±0.64
4	2.05±0.08	5.01±0.28	30.65±0.85	101.81±0.36
5	1.95±0.03	4.56±0.37	10.58±0.35	83.25±0.96
6	1.94±0.02	4.93±0.18	12.31±0.72	388.76±0.82
7	2.77±0.08	6.19±0.38	3.98±0.23	67.26±0.21
8	**2.86**±0.10	**6.49**±0.23	**2.41**±0.35	**48.70**±0.11
9	*2.96*	*8.79*	*176.27±0.05*	*1089.91±0.10*

Finally, we adopt **Maximum Mean Discrepancy (MMD)** [11] to measure the distribution similarity between synthesized and real data, which has been adopted in [41] to evaluate the synthetic video quality. MMD is defined as $\sup_{f\in\mathcal{F}}\{\mathbb{E}_{X\sim p}[f(X)]-\mathbb{E}_{Y\sim q}[f(Y)]\}$, where $\mathcal{F}$ is a class of functions. MMD determines whether distribution p and q are identical based on samples from the two distributions. We use the implementation by [11] to estimate MMD. We treat each sequence as a sample. One scalar is obtained for the entire synthetic dataset. The smaller the value, the more similar the two distributions. We repeat experiment 10 times and the mean and standard deviation of MMD are shown in Table 2. As a validation, the MMD of Gaussian noise is also reported. We observe that the proposed method achieves the lowest MMD in both datasets, which indicates that the distribution of the synthesized data by our method is closer to real data than other methods.

Qualitative results: As shown in Figure 7, we color-code different actions, which are classified by HMMs used for computing inception score. Each sequence is uniformly down-sampled to five frames for visualization purpose. We plot samples of real data, realistic-looking and erroneous synthesized data. From the results, we see our model can generate realistic motion sequences whose motion patterns are clearly discernible. Furthermore, there exist variations in different sequences of the same action, which shows capability of generating diverse motions. More images and videos are shown in the supplementary materials.

5.4. Data Augmentation for Recognition

We further demonstrate the synthesis quality as a way of data augmentation for action recognition task. Specifically, we use the synthesized data as additional training data to train classifier, which we use HMM. We use discriminator output to determine a pseudo-label of the synthesized data. The synthesized and real data are then combined as augmented training data. We perform a four-fold cross-subject classification and the results are shown in Table 4. We observe a consistent improvement as the synthesized portion increases until the performance becomes stable, which shows evidence of the good quality of synthesized data.

5.5. Data Restoration

One of the benefits of probabilistic generative model is handling data with missing values. We perform a data restoration experiment to further demonstrate the benefit of the proposed model as well as Bayesian inference. We train the model using completely observed data. For testing, we omit a subset of joint angles following the same way as [37]. To restore the missing values, we first decode the most prob-

Table 3. Motion restoration results of different methods on different datasets (mean ± standard deviation).

Dataset	CMU (PCC)		Berkeley (PCC)		CMU (MSE)		Berkeley (MSE)	
Joint set	A	B	A	B	A	B	A	B
HSMM	0.26±0.51	0.24±0.50	0.29±0.52	0.34±0.56	0.50±0.48	0.54±0.69	0.40±1.81	0.88±1.51
CRBM	0.42±0.38	0.24±0.49	0.19±0.41	0.34±0.48	0.69±0.62	0.71±1.41	1.28±3.45	1.34±2.74
GPDM	0.65±0.24	0.67±0.24	0.59±0.36	0.61±0.36	0.36±0.35	0.22±0.38	0.41±1.15	0.38±1.87
TSBN	0.84±0.15	0.63±0.32	0.52±0.44	0.64±0.44	0.12±0.10	0.07±0.09	0.29±1.42	0.66±1.25
HHMM	0.82±0.15	0.76±0.19	0.61±0.33	0.64±0.26	0.15±0.15	**0.06**±0.15	0.28±0.87	0.26±0.64
Ours-MAP	0.86±0.12	0.78±0.22	0.53±0.42	0.67±0.36	0.28±0.15	0.24±0.79	0.13±0.38	**0.21**±0.31
Ours-FB	**0.92**±0.06	**0.85**±0.15	**0.68**±0.25	**0.73**±0.16	**0.11**±0.05	0.14±0.03	**0.12**±0.29	0.23±0.37

Table 4. Classification accuracy with different augmentation portions. 0% means only real data are used and 100% means the same amount of synthesized data as the real data is used for training.

Portion	Method	0%	20%	40%	60%	80%	100%
CMU	HHMM	83.0	83.8	84.5	85.7	85.2	87.4
	Ours		85.7	86.0	86.4	87.7	88.9
Berkeley	HHMM	73.2	76.0	76.8	77.3	76.9	77.0
	Ours		76.1	77.6	78.9	78.1	78.3

able hidden state chain by solving

$$\mathbf{Z}_i^*, \mathbf{D}_i^* = \arg\max_{\mathbf{Z},\mathbf{D}} \log P(\bar{\mathbf{X}}, \mathbf{Z}, \mathbf{D}|\theta_i) \tag{8}$$

where $\bar{\mathbf{X}}$ is the partially observed sequence excluding missing channels and θ_i is the i^{th} sample of parameters obtained through Bayesian inference. Eq. (8) can be solved by extending Viterbi algorithm to semi-Markov chain [13] except that the likelihood of each frame is computed by partially observed data. Given decoded state chain, we compute the most likely observations for the missing values by solving Eq. (7), resulting $\mathbf{X}_i^*$. The final restoration is obtained as $\mathbf{X}^* = \frac{1}{M}\sum_{i=1}^{M}\mathbf{X}_i^*$, where M is the total number of parameter samples. To evaluate the performance, we utilize two metrics commonly used in regression task. We compute Pearson correlation coefficient (PCC) and mean square error (MSE) between restored value $\mathbf{X}^*$ and actual values $\mathbf{X}$. PCC and MSE are complementary to each other. PCC is the higher the better and MSE is the lower the better. We use HSMM as baseline. We compare with four state-of-the-art methods that can handle missing inputs: CRBM [37], GPDM [42], TSBN [9] and HHMM [49]. Deterministic models such as RNN cannot handle missing inputs and thus do not apply to this task.

We tried omitting two sets of joint angles. Set A omits 8 angles along the left leg and Set B omits 7 angles along right arms. Both PCC and MSE are computed for each individual angle. We perform a four-fold cross-subject test for all methods. The mean and standard deviation are reported in Table 3. Overall, we achieve the best performance in PCC on both CMU and Berkeley datasets with the average improvement of 9.5% and 8% respectively compared to the second best HHMM. For MSE, the performance on CMU dataset is comparable to both HHMM and TSBN.

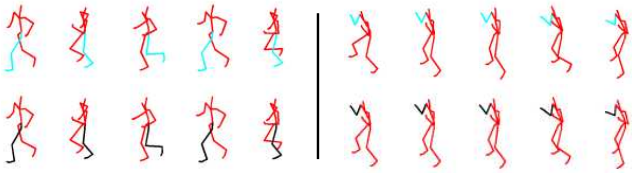

Figure 8. Examples of motion sequence from CMU dataset. Left column misses joint angles in left leg and right column misses joint angles in right arm. Top row is the original data and bottom row is the restored data by BH-HSMM. The missing and restored joints are in cyan and black color respectively. (Best view in color)

On Berkeley dataset, we improve MSE by 0.1 compared to the second best HHMM. In both datasets, we outperform HSMM by a large margin. The high variance reflects the significant variation among different joint angles. Our method achieves both smaller mean and variance, indicating a better and more stable performance. These results show the proposed BH-HSMM captures the variations across motion sequences well. Compared to the MAP estimation result, our Bayesian inference shows improvement in 7 out of 8 experiments in Table 3, which demonstrates that the use of Bayesian inference improves generalization.

6. Conclusion

To summarize, we developed a Bayesian hierarchical dynamic generative model which explicitly models the spatio-temporal dynamics in human motion. We developed an adversarial Bayesian inference framework for the model and demonstrated its benefit in sequential data synthesis and restoration tasks. The use of fully probabilistic framework can better handle the variation in data with a much more compact parameterization cost than deep models. The integration of adversarial learning with Bayesian inference not only retains the benefit of adversarial learning in synthesizing realistic-looking data, but also alleviates the overfitting issue such as mode-collapsing.

Acknowledgment: This work is partially supported by Cognitive Immersive Systems Laboratory (CISL), a collaboration between IBM and RPI, and also a center in IBM's AI Horizon Network.

References

[1] Steven Bird. Nltk: the natural language toolkit. In *ACL*, 2006. 6

[2] Matthew Brand and Aaron Hertzmann. Style machines. In *SIGGRAPH*. ACM, 2000. 2, 4

[3] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *ICML*, 2014. 4

[4] Li Chongxuan, Max Welling, Jun Zhu, and Bo Zhang. Graphical generative adversarial networks. In *NIPS*, 2018. 2

[5] CMU. Cmu mocap database http://mocap.cs.cmu.edu/. 5

[6] Thi V Duong, Hung Hai Bui, Dinh Q Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *CVPR*, 2005. 2

[7] Emily Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *NIPS*, 2009. 2

[8] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *ICCV*, 2015. 1, 2

[9] Zhe Gan, Chunyuan Li, Ricardo Henao, David E Carlson, and Lawrence Carin. Deep temporal sigmoid belief networks for sequence modeling. In *NIPS*, 2015. 1, 2, 5, 8

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 3

[11] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *JMLR*, 2012. 7

[12] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial geometry-aware human motion prediction. In *ECCV*, 2018. 1

[13] Nicholas P Hughes, Lionel Tarassenko, and Stephen J Roberts. Markov models for automated ecg interval analysis. In *NIPS*, 2004. 8

[14] Matthew J Johnson and Alan S Willsky. Bayesian nonparametric hidden semi-markov models. *JMLR*, 2013. 2

[15] Manfred Lau, Ziv Bar-Joseph, and James Kuffner. Modeling spatial and temporal variation in motion data. In *TOG*, 2009. 2, 5

[16] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *CVPR*, 2018. 2

[17] Yan Li, Tianshu Wang, and Heung-Yeung Shum. Motion texture: a two-level statistical model for character motion synthesis. In *ToG*, 2002. 2

[18] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *CVPR*, 2017. 2, 5

[19] Olof Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv*, 2016. 1, 2, 5

[20] Pradeep Natarajan and Ramakant Nevatia. Coupled hidden semi markov models for activity recognition. In *WMVC*, 2007. 2

[21] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *CVPR*, 2017. 1

[22] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016. 1

[23] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *WACV*, 2013. 5

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 5

[25] Santiago Pascual, Antonio Bonafonte, and Joan Serra. Segan: Speech enhancement generative adversarial network. *arXiv*, 2017. 1

[26] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. *BMVC*, 2018. 1, 2

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv*, 2015. 1

[28] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016. 1

[29] Lu Ren, David B Dunson, and Lawrence Carin. The dynamic hierarchical dirichlet process. In *ICML*, 2008. 2

[30] Grégory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *NIPS*, 2016. 1

[31] Yunus Saatci and Andrew G Wilson. Bayesian gan. In *NIPS*, 2017. 3

[32] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *ICCV*, 2017. 1, 2

[33] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016. 3, 6

[34] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv*, 2015. 3

[35] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 1

[36] Graham W Taylor and Geoffrey E Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *ICML*, 2009. 2

[37] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Modeling human motion using binary latent variables. In *NIPS*, 2006. 2, 5, 7, 8

[38] Ngoc-Dung T Tieu, Huy H Nguyen, Hoang-Quoc Nguyen-Son, Junichi Yamagishi, and Isao Echizen. Spatio-temporal generative adversarial network for gait anonymization. *Journal of Information Security and Applications*, 2019. 1

[39] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018. 1, 2

[40] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NIPS*, 2016. 1, 2, 5

[41] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. *ICCV*, 2017. 1, 2, 7

[42] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *PAMI*, 2008. 8

[43] Shihong Xia, Congyi Wang, Jinxiang Chai, and Jessica Hodgins. Realtime style transfer for unlabeled heterogeneous human motion. *TOG*, 2015. 2

[44] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*, 2016. 1, 2

[45] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *ICCV*, 2019. 2

[46] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017. 1, 2

[47] Shun-Zheng Yu. Hidden semi-markov models. *Artificial Intelligence*, 2010. 3

[48] Jason Y Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3d human dynamics from video. *ICCV*, 2019. 1

[49] Rui Zhao and Qiang Ji. An adversarial hierarchical hidden markov model for human pose modeling and generation. In *AAAI*, 2018. 2, 5, 8

[50] Rui Zhao, Kang Wang, Hui Su, and Qiang Ji. Bayesian graph convolution lstm for skeleton based action recognition. In *CVPR*, 2019. 1

[51] Rui Zhao, Wanru Xu, Hui Su, and Qiang Ji. Bayesian hierarchical dynamic model for human action recognition. In *CVPR*, 2019. 2

[52] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *ICCV*, 2017. 1