Supplementary Materials for 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

1. Parameterization of BH-HSMM

First, we specify the conditional distributions of $\mathbf{X}, \mathbf{Z}, \mathbf{D}$ nodes. For initial state distribution, we use $P(Z_0 = i) = \pi_i,\ P(D_0 = d) = \delta(d, 1)$, where $\sum_{i=1}^{Q} \pi_i = 1, \pi_i \geq 0, \delta(i, j) = 1$ if $i = j$ and 0 otherwise. We assume D_0 is always 1 for simplicity. For transition distribution,

$$P(Z_t = j | Z_{t-1} = i, D_{t-1} = d) = \begin{cases} A_{ij}, & \text{if } d = 1 \\ \delta(i, j), & \text{otherwise} \end{cases}$$

where $\sum_{j=1}^{Q} A_{ij} = 1, A_{ij} \geq 0, \forall i = 1, ..., Q$. We forbid self-transition *i.e.* $A_{ii} = 0$ to disambiguate the duration count used during inference [2]. For duration distribution,

$$P(D_t = d | D_{t-1} = d', Z_t = i) = \begin{cases} C_{id} & \text{if } d' = 1 \\ \delta(d, d' - 1), & \text{otherwise} \end{cases}$$

where $\sum_{d=1}^{L} C_{id} = 1, C_{id} \geq 0, \forall i = 1, ..., Q$. Here we assumed that the residue D_t decreases deterministically by 1 each time the chain emits an observation until duration expires *i.e.* $d' = 1$. For emission distribution, we use multivariate Gaussian distribution $P(X_t = \mathbf{o} | Z_t = i) = \mathcal{N}(\mathbf{o} | \mu_i, \Sigma_i)$, where $\mu_i \in \mathbb{R}^O, \Sigma_i \in \mathbb{R}^{O \times O}$ are the mean and covariance of state i. In summary, the model parameters are $\theta = \{\pi, A, C, \psi\}$, where $\psi = \{\mu, \Sigma\}$.

Then we specify the prior distribution of each parameter. For initial state π, transition A, and duration C, we place Dirichlet (Dir) prior.

$$P(\pi | \eta_0) = Dir(\pi | \eta_0) \propto \prod_{j=1}^{Q} \pi_j^{\eta_{0j} - 1}$$

$$P(A_{i:} | \eta_i) = Dir(A_{i:} | \eta_i) \propto \prod_{j=1}^{Q} A_{ij}^{\eta_{ij} - 1},\ i = 1, ..., Q$$

$$P(C_{i:} | \xi_i) = Dir(C_{i:} | \xi_i) \propto \prod_{d=1}^{L} C_{id}^{\xi_{id} - 1},\ i = 1, ..., Q$$

where $\eta_{0j} > 0,\ \eta_{ij} > 0$, and $\xi_{ij} > 0$ are hyperparameters for $\pi,\ A$, and C respectively. For emission parameters, we use Normal-inverse-Wishart (NIW) prior.

$$P(\mu_i, \Sigma_i | \lambda) = NIW(\mu_i, \Sigma_i | \mu_0, \kappa_0, \Lambda_0, \nu_0) = N(\mu_i | \mu_0, \frac{1}{\kappa_0} \Sigma_i) IW(\Sigma_i | \Lambda_0, \nu_0),\ i = 1, ..., Q$$

and $\kappa_0 > 0, \mu_0 \in \mathbb{R}^O, \nu_0 > O + 1$ and $\Lambda_0 \in \mathbb{R}^{O \times O}$ is a positive-definite matrix. In summary, the hyperparameters are $\alpha = \{\eta_0, \eta, \xi, \lambda\}$, where $\lambda = \{\mu_0, \kappa_0, \Lambda_0, \nu_0\}$. In our experiment, we use $\eta, \xi = 1$, which yields uniform prior. We use $\mu_0 = \mathbf{0}, \kappa_0 = 1, \nu_0 = O + 2, \Lambda_0 = \mathbf{I}$, which yields standard Normal prior. A better choice or estimation of hyperparameters are left as future work. We consider the current choice of hyperparameters as **non-informative prior**.

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

2. Gradient of Generator and Discriminator

According to Section 4.2 in the main paper, the unnormalized log-posterior of generator parameter θ is as follows.

$$\begin{aligned} L_g(\theta) &\triangleq -\sum_{j=1}^{n_g} H(\mathbf{X}_j^-|\phi) + \log P(\theta|\alpha_g) \\ &\approx -n_g E_{\mathbf{X}\sim P(\mathbf{X}|\theta)}[H(\mathbf{X}|\phi)] + \log P(\theta|\alpha_g) \end{aligned} \tag{1}$$

where n_g is the number of synthetic data samples in each mini-batch of stochastic gradient update. We can compute the gradient of θ as follows.

$$\begin{aligned} \frac{\partial L_g(\theta)}{\partial \theta} &\approx -n_g \frac{\partial}{\partial \theta} \int_{\mathbf{X}} H(\mathbf{X}|\phi) P(\mathbf{X}|\theta) d\mathbf{X} + \frac{\partial \log P(\theta|\alpha_g)}{\partial \theta} \\ &= -n_g \int_{\mathbf{X}} H(\mathbf{X}|\phi) \frac{\partial P(\mathbf{X}|\theta)}{\partial \theta} d\mathbf{X} + \frac{\partial \log P(\theta|\alpha_g)}{\partial \theta} \\ &= -n_g \int_{\mathbf{X}} H(\mathbf{X}|\phi) P(\mathbf{X}|\theta) \frac{\partial \log P(\mathbf{X}|\theta)}{\partial \theta} d\mathbf{X} + \frac{\partial \log P(\theta|\alpha_g)}{\partial \theta} \\ &\approx -\sum_{j=1}^{n_g} H(\mathbf{X}_j^-|\phi) \frac{\partial \log P(\mathbf{X}_j^-|\theta)}{\partial \theta} + \frac{\partial \log P(\theta|\alpha_g)}{\partial \theta}, \ \mathbf{X}_j^- \sim P(\mathbf{X}|\theta) \end{aligned} \tag{2}$$

The gradient of log-likelihood $\frac{\partial \log P(\mathbf{X}_j^-|\theta)}{\partial \theta}$ is derived by extending the method in [1] to HSMM. The gradient of log-prior $\frac{\partial \log P(\theta|\alpha_g)}{\partial \theta}$ has a closed-form based on the prior distribution.

For the discriminator ϕ, the unnormalized log-posterior is defined as follows.

$$\begin{aligned} L_d(\phi) &\triangleq -\sum_{i=1}^{n_d} H(\mathbf{X}_i^+|\phi) + \sum_{j=1}^{n_g} H(\mathbf{X}_j^-|\phi) + \log P(\phi|\alpha_d) \\ &\approx -n_d E_{\mathbf{X}\sim P_{data}(\mathbf{X})}[H(\mathbf{X}|\phi)] + n_g E_{\mathbf{X}\sim P(\mathbf{X}|\theta)}[H(\mathbf{X}|\phi)] + \log P(\phi|\alpha_d) \end{aligned} \tag{3}$$

where n_d and n_g are the number of real and synthetic data samples in each mini-batch, respectively. We can compute the gradient as follows.

$$\frac{\partial L_d(\phi)}{\partial \phi} \approx -\sum_{i=1}^{n_d} \frac{\partial H(\mathbf{X}_i^+|\phi)}{\partial \phi} + \sum_{j=1}^{n_g} \frac{\partial H(\mathbf{X}_j^-|\phi)}{\partial \phi} + \frac{\partial \log P(\phi|\alpha_d)}{\partial \phi} \tag{4}$$

Notice that the discriminator consists of K BH-HSMMs and K is the total number of classes *i.e.* $\phi = \{\phi_1, ..., \phi_K\}$, where ϕ_k is the parameter of k^{th} BH-HSMM. The computation of the first and the second term in Eq. (4) can be further derived as follows.

$$\begin{aligned} \frac{\partial H(\mathbf{X}_i^+|\phi)}{\partial \phi_k} \approx & \nabla_{\phi_k} \log P(\mathbf{X}_i^+|\phi_k) \\ & \Big[\sum_{l=1}^{K} P(y=l|\mathbf{X}_i^+,\phi) \log P(y=l|\mathbf{X}_i^+,\phi) - \log P(y=k|\mathbf{X}_i^+,\phi)\Big] P(y=k|\mathbf{X}_i^+,\phi), \ k=1,...K \end{aligned} \tag{5}$$

$$\begin{aligned} \frac{\partial H(\mathbf{X}_j^-|\phi)}{\partial \phi_k} \approx & \nabla_{\phi_k} \log P(\mathbf{X}_j^-|\phi_k) \\ & \Big[\sum_{l=1}^{K} P(y=l|\mathbf{X}_j^-,\phi) \log P(y=l|\mathbf{X}_j^-,\phi) - \log P(y=k|\mathbf{X}_j^-,\phi)\Big] P(y=k|\mathbf{X}_j^-,\phi), \ k=1,...K \end{aligned} \tag{6}$$

where $\mathbf{X}_i^+$ is a sample drawn from real data and $\mathbf{X}_j^-$ is a sample generated from generator. Similar to the case in generator, $\nabla_{\phi_k} \log P(\mathbf{X}|\phi_k)$ is the gradient of HSMM. Finally, the computation of the third term in Eq. (4) is done for each ϕ_k separately, in which $\frac{\partial \log P(\phi_k|\alpha_d)}{\partial \phi_k}$ has a closed-form based on the prior distribution.

3. Details about Perturbation in Analysis of BLEU

We consider three types of perturbation. The first one is random permutation. For this perturbation, we randomly shuffle the frame order of the original sequence. Therefore, each individual frame of the shuffled sequence comes from a real sequence. But the order is completely random. We vary the extent of perturbation by varying the total portion of the frames that are permuted. For example, if 40% of the frames are permuted, then we only randomly permute the first 40% of frames in the real sequence. We vary the portion from 20% to 100%. The second one is adding white noise. Specifically, we add random noise that follows Gaussian distribution with mean 0 and standard deviation σ to all the individual angle in all frames of mocap sequences. We choose $\sigma = 0.1, 0.4, 0.7, 1, 1.3$ to create different extents of perturbation. The third one is occlusion. Specifically, we randomly select a portion of the mocap angles whose values are occluded. And we set the occluded angle to the dataset average. We vary the portion from 10% to 50%. Table 1 shows the results of BLEU under different perturbations.

Table 1. BLEU of Berkeley dataset under different perturbations.

Permutation	20%	40%	60%	80%	100%
BLEU	0.8899	0.7612	0.6339	0.4873	0.3459
Noise σ	0.1	0.4	0.7	1.0	1.3
BLEU	0.8720	0.7780	0.6581	0.5359	0.4240
Occlusion	10%	20%	30%	40%	50%
BLEU	0.8688	0.7819	0.6830	0.5726	0.4807

4. More Synthesis Results

We show more synthesis results of our approach in Figure 1 and 2. Two additional videos are provided in the zip file. 'synthetic.mov' is synthetic data by our method. 'real.mov' is real data from the dataset.

5. Implementation

The choice of constants listed in Algorithm 1 in the main paper are provided as follows.

Table 2. Choice of constants in the input arguments of Algorithm 1 in the main paper for different datasets.

Dataset	CMU	Berkeley
Q_g	12	60
Q_d	4 for each model	6 for each model
M	8	10
K	3	10
a	0.9	0.9
α_d	See Section 1	
α_g	See Section 1	
τ	5	5
η	0.05	0.05
burn-in	10 epochs	10 epochs

We plan to release our code implemented in Matlab here: https://github.com/rort1989/BH-HSMM.

References

[1] Olivier Cappé, Vincent Buchoux, and Eric Moulines. Quasi-newton method for maximum likelihood estimation of hidden markov models. In *ICASSP*, 1998. 2

[2] Shun-Zheng Yu and Hisashi Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden markov model. *Signal processing letters*, 2003. 1

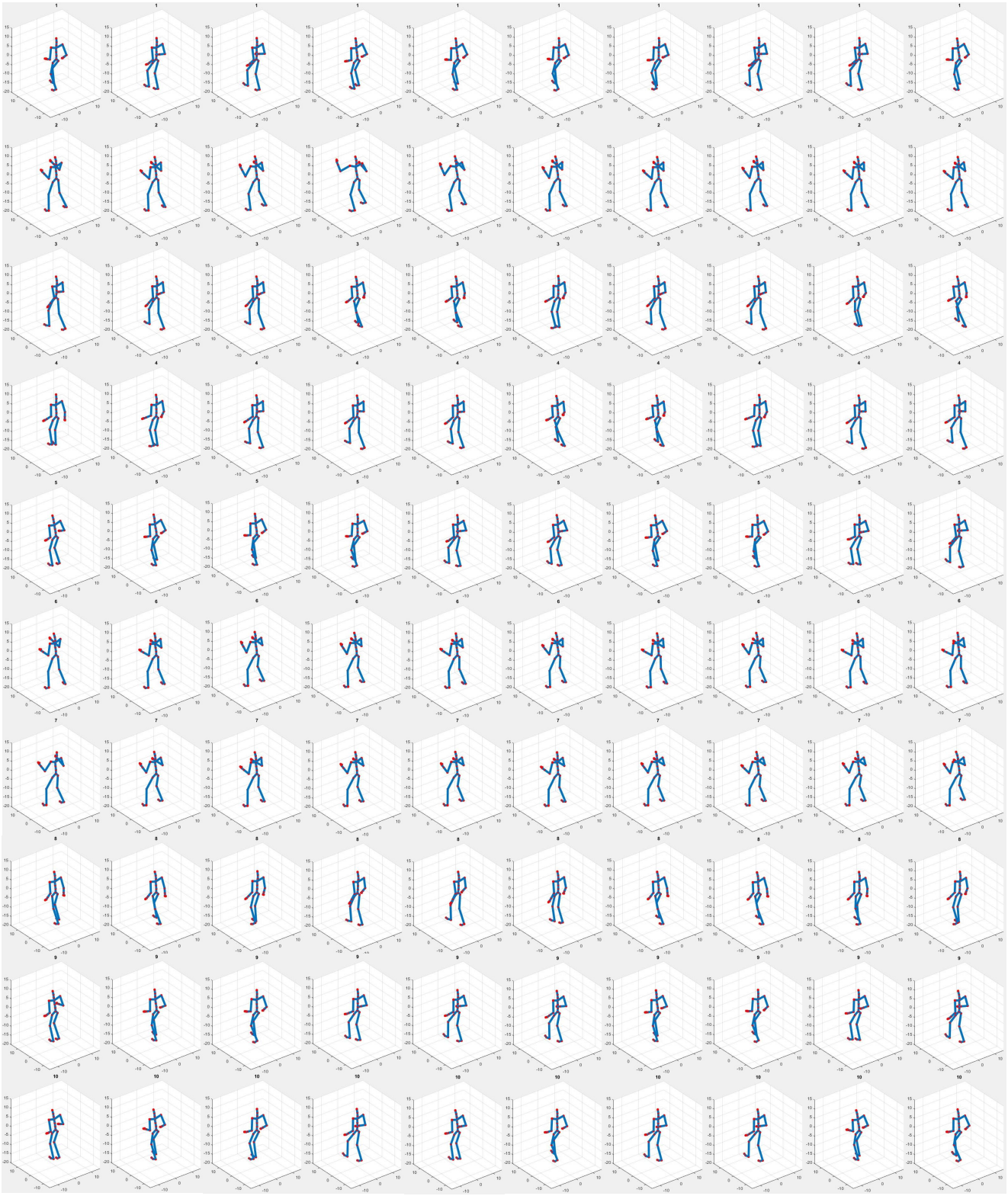

Figure 1. More results on CMU dataset. Each row corresponds to one synthesized sequence.

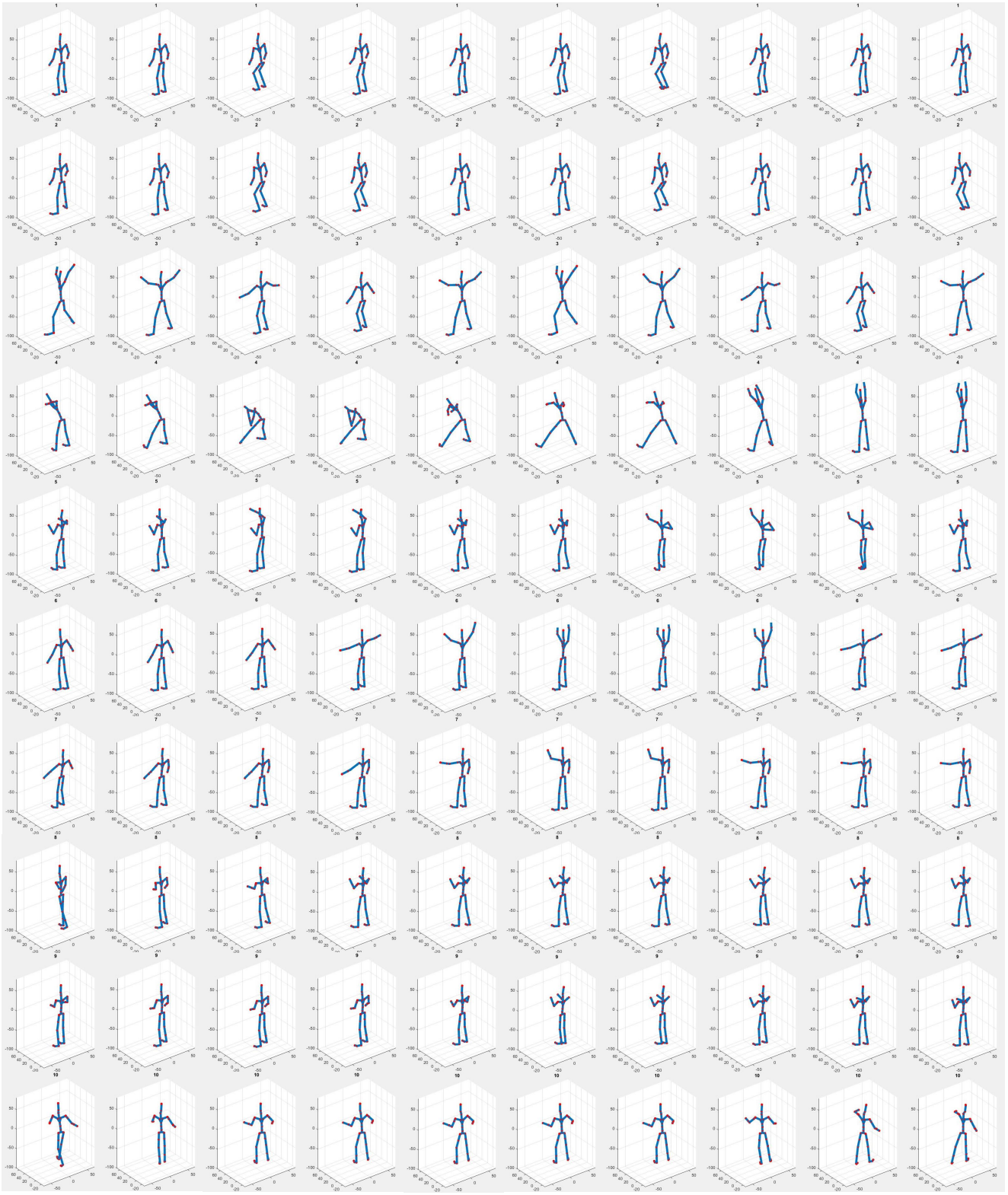

Figure 2. More results on Berkeley dataset. Each row corresponds to one synthesized sequence.