Diverse Human Motion Prediction via Gumbel-Softmax Sampling from an Auxiliary Space — Supplementary Material —

Lingwei Dang
South China University of Technology
Guangzhou, Guangdong, China
csdanglw@mail.scut.edu.cn

Yongwei Nie*
South China University of Technology
Guangzhou, Guangdong, China
nieyongwei@scut.edu.cn

Chengjiang Long
Meta Reality Lab
Burlingame, CA, USA
clong1@fb.com

Qing Zhang
Sun Yat-sen University
Guangzhou, Guangdong, China
zhangqing.whu.cs@gmail.com

Guiqing Li
South China University of Technology
Guangzhou, Guangdong, China
ligq@scut.edu.cn

ABSTRACT

In this supplementary material, we provide more information that cannot be included in the paper due to the space limit. We first introduce network architectures of our method in detail. Then, we provide more quantitative and qualitative comparisons. Finally, we give some failure cases and the reasons for these failures. Please refer to our provided video demo to review the results more intuitively.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Activity recognition and understanding**.

KEYWORDS

Human motion prediction, stochastic prediction, diverse prediction

ACM Reference Format:
Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. 2022. Diverse Human Motion Prediction via Gumbel-Softmax Sampling from an Auxiliary Space — Supplementary Material — . In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3547956

1 AUXILIARY-SPACE-BASED SAMPLING NETWORK ARCHITECTURE

Our auxiliary-space-based sampling network is illustrated in Figure 1. The input is $\mathbf{x} \in \mathbb{R}^{[J\times C,H]}$, where H is the number of input poses, J is the number of joints of a pose, and C is the dimension size of each joint. For Human3.6M [2], $J = 17$, $C = 3$, $H = 25$, while for HumanEva-I [5], $C = 3$, $J = 15$, and $H = 15$. Following [4], we repeat the last pose of $\mathbf{x}$, T times, and append them to $\mathbf{x}$. Now, the input is of size $[J \times C, H + T]$, where T is the number of poses to be predicted. For Human3.6M, $T = 100$, and for HumanEva-I, $T = 60$. Then following [4] again, we apply a Discrete Cosine Transform (DCT) operator to transform the temporal information along the $H + T$ dimension of the input data into the frequency space. By keeping only the coefficients of low frequency components and discarding those of high frequency components, we obtain data of $[J\times C, n_{dct}]$ which becomes $[J, C \times n_{dct}]$ after reshaping, where $n_{dct} = 10$ is the number of the remained coefficients. Following, a network $\mathcal{N}_\beta$ made up of a GCN and an MLP learns a base matrix $\mathbf{B} \in \mathbb{R}^{[M\times n_b]}$ from the DCT coefficients, where $M = 40$ and $n_b = 128$. The GCN, which will be described later, extracts hidden features of shape $J \times F$ where $F = 256$ is the feature dimension size. Then, the MLP composed of a linear transformation layer, a Batch Normalization (BN) layer and a Tanh activation function, maps the hidden features into $\mathbf{B}$. The network structures of the GCN and MLP are shown in Table 1. Next, we sample a random coefficient matrix $\mathbf{W} \in \mathbb{R}^{[K\times M]}$ by the Gumbel-Softmax sampling technique, where $K = 50$ is the sampling number. Then the multiplication of $\mathbf{W}$ and $\mathbf{B}$ outputs a point matrix of shape $[K \times n_b]$. Next, the second network $\mathcal{N}_\gamma$

Table 1: The network structure of $\mathcal{N}_\beta$ and $\mathcal{N}_\gamma$. For Human3.6M, $J = 17$. For HumanEva-I, $J = 15$. For both datasets, $M = 40$, $K = 50$, $C = 3$, $F = 256$, $n_{dct} = 10$, $n_b = 128$, $n_h = 64$, $n_z = 64$.

Component	Block	Layer	Weight Size	Input Size	Output Size
$\mathcal{N}_\beta$	GCN	GCL	A(J, J), W$(C\times n_{dct}, F)$	$(J, C\times n_{dct})$	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
	MLP 1	Linear	W$(J\times F, M\times n_b)$	$(J\times F)$	$(M\times n_b)$
		BN, Tanh	-	$(M\times n_b)$	$(M\times n_b)$
$\mathcal{N}_\gamma$	MLP 2_1	Linear	W(n_b, n_h)	(K, n_b)	(K, n_h)
		BN, Tanh	-	(K, n_h)	(K, n_h)
		Linear	W(n_h, n_z)	(K, n_h)	(K, n_z)
	MLP 2_2	Linear	W(n_b, n_h)	(K, n_b)	(K, n_h)
		BN, Tanh	-	(K, n_h)	(K, n_h)
		Linear	W(n_h, n_z)	(K, n_h)	(K, n_z)
CVAE Decoder	See Table 2			$(K\times J, C\times n_{dct})$, $(K\times n_z)$	$(K\times J, C\times n_{dct})$

---

*Corresponding author.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3547956

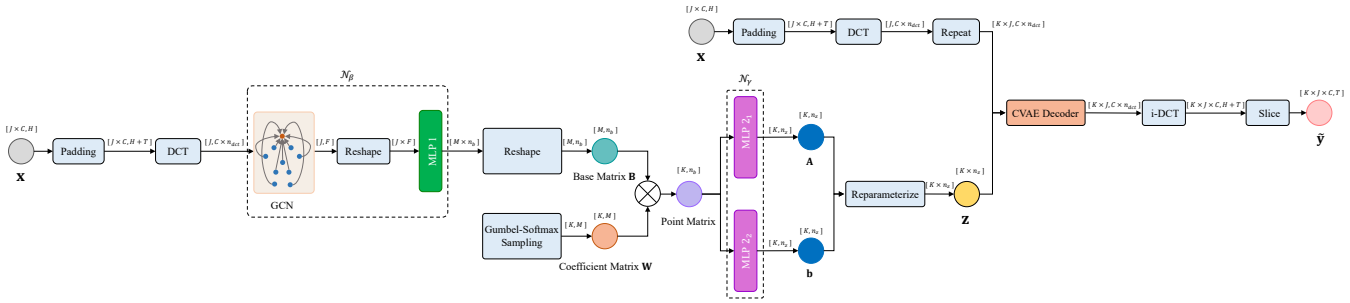


Figure 1: Detailed architecture of our auxiliary-space-based sampling model. A circle represents input, intermediate, or output data. The symbol above a circle indicates the size of the data. For example, $[J \times C, H]$ means the data is a two-dimensional matrix of $J \times C$ rows and H columns. The symbol above an arrow indicates the size of the output of the corresponding previous operator. Please refer to the main text for detailed descriptions of the architecture.

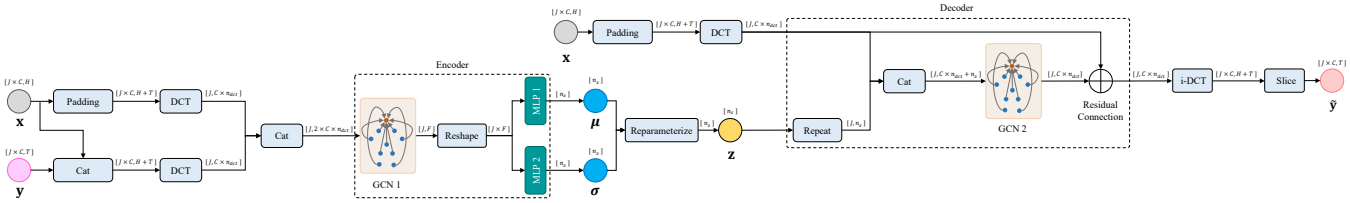


Figure 2: Detailed architecture of the adopted CVAE model. It is composed of an encoder and a decoder that are built on GCNs and MLPs. Please refer to the main text for detailed descriptions of the architecture.

projects the K sampled points into the parameters of K Gaussian distributions $\{\mathcal{N}(\mathbf{b}_k, \mathbf{A}_k)\}_{k=1}^{K}$, where $\mathbf{b} \in \mathbb{R}^{[K \times n_z]}$ indicates means of these Gaussian distributions and $\mathbf{A} \in \mathbb{R}^{[K \times n_z]}$ are the diagonal values of their co-variance matrices. In particular, $\mathcal{N}_\gamma$ consists of two sub MLPs for generating $\mathbf{b}$ and $\mathbf{A}$, respectively. The detailed structure of $\mathcal{N}_\gamma$ is shown in Table 1. Each MLP is made up of a Linear-BN-Tanh layer to transform the point features into hidden vectors of shape $[K \times n_h]$, and another Linear layer that maps the hidden vectors into Gaussian parameters of shape $[K \times n_z]$, where n_h and n_z are both set as 64. After that, a set of latent variables $\mathbf{z} \in \mathbb{R}^{[K \times n_z]}$ are drawn from the Gaussian distributions by the reparameterization trick. We repeat the input data after DCT, K times and concatenate each of them with $\mathbf{z}$, and feed them into the pretrained CVAE decoder (which will be described later) to produce future motions in the frequency space of shape $[K \times J, C \times n_{dct}]$. Then we project the frequency features back into the pose space by the inverse DCT (i-DCT) function, obtaining K pose sequences of shape $[K \times J \times C, H + T]$. Finally, a slice operator extracts only the future T frames and outputs results of shape $[K \times J \times C, T]$ which are the K future pose sequences predicted by our network.

Now, we introduce the Graph Convolutional Network (GCN). As shown in Table 1, our GCN is made up of five sequentially stacked GCL-BN-Tanh layers, where GCL stands for Graph Convolutional layer. Let $\mathbf{H}^l \in \mathbb{R}^{J \times F^l}$ be the input to the l^{th} GCL where F^l is the hidden feature dimension size, $\mathbf{A}^l \in \mathbb{R}^{J \times J}$ the adjacency matrix, and $\mathbf{W}^l \in \mathbb{R}^{F^l \times F^{l+1}}$ the trainable parameters, the GCL executes the following computation:

$$\mathbf{H}^{l+1} = \mathbf{A}^l \mathbf{H}^l \mathbf{W}^l, \tag{1}$$

where $\mathbf{H}^{l+1} \in \mathbb{R}^{J \times F^{l+1}}$ is the output of the l^{th} GCL. At the very beginning, $F^0 = C \times n_{dct}$.

2 CVAE NETWORK ARCHITECTURE

Recall that before applying our method to generate diverse results, we need to train a CVAE model beforehand. The CVAE network architecture adopted in this paper is shown in Figure 2. Let $\mathbf{x} \in \mathbb{R}^{[J \times C, H]}$ be an observed pose sequence, and $\mathbf{y} \in \mathbb{R}^{[J \times C, T]}$ be the ground truth future poses. We compute the frequency coefficients of shape $[J, C \times n_{dct}]$ from each of them and then concatenate both the frequency content into data of shape $[J, 2 \times C \times n_{dct}]$. Then, an encoder is used to learn the parameters of the posterior Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ of the latent code $\mathbf{z}$ given $\mathbf{x}$ and $\mathbf{y}$, where $\boldsymbol{\mu} \in \mathbb{R}^{[n_z]}$ is the mean of the posterior distribution, and $\boldsymbol{\sigma} \in \mathbb{R}^{[n_z]}$ is the diagonal values of the co-variance matrix of the posterior distribution. Particularly, as shown in Table 2, the encoder consists of a GCN and two MLPs. The GCN is composed of nine GCL-BN-Tanh layers to extract hidden features of shape $[J, F]$. The two MLPs, each of which just comprises a single Linear layer, map the hidden features into $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, respectively. Then a latent variable $\mathbf{z} \in \mathbb{R}^{[n_z]}$ can be drawn from the posterior Gaussian distribution by the reparameterization trick.

Next, a decoder is used to reconstruct $\mathbf{y}$ from the latent code $\mathbf{z}$. To achieve that, we first repeat $\mathbf{z}$, J times and concatenate them with the DCT coefficients of $\mathbf{x}$, resulting in a feature of shape $[J, C \times n_{dct} + n_z]$. Afterwards, we employ another GCN comprising 9 GCL-BN-Tanh layers to extract hidden features of shape $[J, F]$, and a GCL layer that projects the hidden features back into the frequency

Table 2: The network structure of the employed CVAE. For Human3.6M, $J = 17$. For HumanEva-I, $J = 15$. For both datasets, $C = 3$, $F = 256$, $n_{dct} = 10$, $n_z = 64$.

Component	Block	Layer	Weight Size	Input Size	Output Size
Encoder	GCN 1	GCL	A(J, J), W($2 \times n_{dct}, F$)	$(J, 2 \times C \times n_{dct})$	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
	MLP 1	Linear	W($J \times F, n_z$)	$(J \times F)$	(n_z)
	MLP 2	Linear	W($J \times F, n_z$)	$(J \times F)$	(n_z)
Decoder	GCN 2	GCL	A(J, J), W($C \times n_{dct} + n_z, F$)	$(J, C \times n_{dct}), (n_z)$	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W(F, F)	(J, F)	(J, F)
		BN, Tanh	-	(J, F)	(J, F)
		GCL	A(J, J), W($F, C \times n_{dct}$)	(J, F)	$(J, C \times n_{dct})$

space of shape $[J, C \times n_{dct}]$. The frequency coefficients of $\mathbf{x}$ after DCT is added to this output. Finally, we use an i-DCT function to transform the above output back into the pose space and slice off the future T frames to obtain $\tilde{\mathbf{y}} \in \mathbb{R}^{[J \times C, T]}$.

3 EVALUATION ON FID, ACC, ADE-M, AND FDE-M

We further evaluate our method on four additional metrics: FID, ACC, ADE-median (ADE-m), and FDE-median (FDE-m).

- Recognition Accuracy (**ACC**). A pre-trained action recognition classifier is used to classify the generated poses. ACC is the overall recognition accuracy. We train the action classifier in the way suggested by [1].
- Frechet Inception Distance (**FID**). Features are extracted from the generated and real data by the pre-trained action classifier. FID is then calculated as the Frechet inception distance between the two feature distributions.
- **ADE-m** and **FDE-m** are similar to ADE and FDE except that the median distance are reported.

The results on the four metrics are shown in Table 3. For ADE-m and FDE-m, DLow performs the best. That is because the diversity by DLow (11.741, Human3.6M) is much lower than that by GSPS (14.757) and our method (15.310). The lower the diversity, the lower the median distance. Therefore, it is not surprise that both our method and GSPS have larger ADE-m and FDE-m than DLow. Compared with GSPS, our method has lower ADE-m and FDE-m (ADE-m: 0.924 (our) *v.s.* 1.013 (GSPS), FDE-m: 1.344 (our) *v.s.* 1.372 (GSPS), Human3.6M), even though our method has greater diversity (APD: 15.310 (our) *v.s.* 14.757 (GSPS)).

For FID and ACC, our method is better than DLow and GSPS in terms of ACC. For HumanEva-I, our FID is the best. However, an exception is the FID of Human3.6M, for which DLow performs much better than GSPS and our method. Again, this is because the diversity of the results of DLow (APD=11.741) is much lower than those of GSPS (14.757) and our method (15.310).

4 COMPARISON WITH A VARIANT OF DLOW

We simply add some random noise to the generated Gaussian distributions (i.e., adding noises to the mean and variance of the Gaussian distributions) in DLow [6] and compare with this variant of DLow.

Firstly, we add noises to the Gaussian distributions generated by DLow without re-training the DLow model (DLow-variant w/o retraining). Secondly, we retrain the DLow model, and add noises to the Gaussian distributions at both training and testing phases (DLow-variant w/ retraining). The noises are randomly drawn from a Normal distribution $\mathcal{N}(0, \sigma)$. The mean of the noises is always zero, while for comparison we test noises of different variances.

As shown in Table 4, adding noises can increase the diversity of the generated results (measured by APD). The heavier the noises (produced by larger σ), the more diverse the results. However, the negative effect is that the accuracy of the results is decreased. For the Human3.6M dataset, please see the columns of "DLow-variant w/o retraining $\sigma = 1$" and "DLow-variant w/ retraining $\sigma = 1.7$" that produce similar APD as ours. Their accuracy metrics are much higher than ours. Although adding noises can yield very large APD (20.894 in the column of "DLow-variant w/o retraining $\sigma = 2$"), the generated poses look unrealistically. And those results on the HumanEva-I dataset have the same trend. From this point of view, our method is better than directly adding noises to generated Gaussian distributions.

5 LEARN π INSTEAD OF SETTING A CONSTANT VALUE

In the main paper, we set π (see Eq. 12) to a constant value. With constant π (1/40=0.025), each basis vector has the equal probability to be assigned with the highest weight among all the basis vectors. To treat all the basis vectors equally, we therefore use the same constant π (actually a probability) for each of them.

As a variant, π can also be learned automatically. We conduct experiments to compare between learning and setting a constant π. The results are shown in Table 5.

In the first experiment, we learn a π for each input sample individually (Ours-Individual-π), the results are slightly worse than those of directly indicating a constant π. In the second experiment, we learn a π shared by all the input samples (Ours-Shared-π), the new results are comparable to those of constant π. We find that the values of the learned shared π fall in the range of [0.021, 0.029], which are nearly equally distributed.

Table 3: Comparison on four additional metrics: ADE-m, FDE-m, FID, and ACC.

	Human3.6M [2]				HumanEva-I [5]			
	ADE-m ↓	FDE-m ↓	FID ↓	ACC ↑	ADE-m ↓	FDE-m ↓	FID ↓	ACC ↑
DLow [6]	**0.896**	**1.284**	**1.566**	0.227	**0.577**	**0.717**	3.472	0.527
GSPS [3]	1.013	1.372	1.915	0.222	0.686	0.794	1.604	0.516
Ours	0.924	1.344	2.060	**0.261**	0.716	0.770	**1.106**	**0.609**

Table 4: Comparison with DLow [6] when adding random noises of different variance to its Gaussian distributions.

	Human3.6M [2]								HumanEva-I [5]							
	Ours	DLow [6]	DLow-variant [6] w/o retraining			DLow-variant [6] w/ retraining			Ours	DLow [6]	DLow-variant [6] w/o retraining			DLow-variant [6] w/ retraining		
			σ = 1	σ = 1.7	σ = 2	σ = 1	σ = 1.7	σ = 2			σ = 1	σ = 5.7	σ = 6	σ = 1	σ = 5.7	σ = 6
APD ↑	15.310	11.741	15.190	19.295	**20.894**	11.100	15.373	18.667	6.109	4.855	4.753	6.147	**6.243**	4.488	6.135	6.205
ADE ↓	**0.370**	0.425	0.560	0.721	0.795	0.526	0.675	0.748	**0.220**	0.251	0.305	0.647	0.654	0.297	0.638	0.649
FDE ↓	**0.485**	0.518	0.674	0.863	0.949	0.627	0.799	0.883	**0.234**	0.268	0.327	0.658	0.664	0.321	0.648	0.657
MMADE ↓	**0.475**	0.495	0.612	0.764	0.835	0.579	0.719	0.789	**0.342**	0.362	0.387	0.659	0.666	0.381	0.652	0.664
MMFDE ↓	**0.516**	0.531	0.682	0.868	0.953	0.634	0.804	0.889	**0.316**	0.339	0.372	0.661	0.667	0.368	0.650	0.661

Table 5: Comparison with learning π (in Eq. 12) automatically.

	Human3.6M [2]					HumanEva-I [5]				
	APD ↑	ADE↓	FDE ↓	MMADE ↓	MMFDE ↓	APD ↑	ADE ↓	FDE ↓	MMADE ↓	MMFDE ↓
DLow [6]	11.741	0.425	0.518	0.495	0.531	4.855	0.251	0.268	0.362	0.339
GSPS [3]	14.757	0.389	0.496	0.476	0.525	5.825	0.233	0.244	0.343	0.331
Ours	**15.310**	0.370	**0.485**	**0.475**	**0.516**	**6.109**	**0.220**	**0.234**	0.342	0.316
Ours-Individual-π	13.530	0.372	0.493	0.481	0.525	5.606	0.229	0.239	0.338	0.317
Ours-Shared-π	14.440	**0.368**	**0.485**	0.477	0.519	5.690	0.228	0.235	**0.324**	**0.295**

6 MORE ABLATION STUDIES ON PARAMETER K

In the main paper, we have demonstrated that K, *i.e.*, the number of predictions for an input, has a large effect on the accuracy but not the diversity of the results produced by our method. Here, we show more ablation studies on K, and perform comparisons between CVAE random sampling, DLow [6], GSPS [3] and our method.

The results are plotted in Figure 3. The first row shows the predicted results' diversity measured by *APD*. As can be seen, for all the compared methods, *APD* does not change much as K increases. We can also see that the diversity of our results is the largest among all the methods, while that of CVAE is the smallest.

The second and third rows show the predicted results' accuracy measured by *ADE* and *FDE*. As can be seen, for all the compared methods, *ADE* and *FDE* decrease as K increases, indicating that more accurate results are obtained.

The fourth and fifth rows show the predicted results' accuracy measured by *MMADE* and *MMFDE*. For all the compared methods, *MMADE* and *MMFDE* decreases too as K increases.

Observing all the results in Figure 3, we can see that our method outputs more diverse results than DLow and GSPS, and at the same time our results are more accurate than those of DLow and GSPS. Generally, diversity and accuracy are two contradictory objectives. Low diversity usually means high accuracy. That is why CVAE, which generates results of the lowest diversity, produces the most accurate results in the second to fifth rows.

7 MORE QUALITATIVE COMPARISONS

In Figure 4 and Figure 5, we show more qualitative comparisons between CVAE, DLow [6], GSPS [3] and our method on the Human3.6M dataset [2] and the HumanEva-I dataset [5]. For each input sequence, we generate 50 future pose sequences by these methods, and show the end poses of ten of them. In the brackets under the names of different methods, we show the diversity of the corresponding results computed by these methods. For these examples, our method produces more diverse results than the other compared methods.

8 FAILURE CASES

Overall, our method is better than GSPS in term of diversity. Therefore, for most of the test cases our method generates more diverse results than GSPS. Inevitably there are cases for which our method generates less diverse results than GSPS. For example, Figure 6 shows such two cases.

While most of our results look reasonable, there are occasional ones that are implausible. Figure 7 shows some examples. We observe that DLow and GSPS suffer from this problem too, and the implausible poses in their results are highlighted too. One can reduce the number of implausible poses by toning down the requirement for diversity. A more effective way is to collect more data covering more actions of humans to enrich the training dataset.

REFERENCES

[1] Xiaoyu Bie, Wen Guo, Simon Leglaive, Lauren Girin, Francesc Moreno-Noguer, and Xavier Alameda-Pineda. 2022. HiT-DVAE: Human Motion Generation via Hierarchical Transformer Dynamical VAE. *arXiv preprint arXiv:2204.01565* (2022).

[2] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2013. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence* 36, 7 (2013), 1325–1339.

[3] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. 2021. Generating Smooth Pose Sequences for Diverse Human Motion Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13309–13318.

[4] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. 2019. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9489–9497.

[5] Leonid Sigal, Alexandru O Balan, and Michael J Black. 2010. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision* 87, 1 (2010), 4–27.

[6] Ye Yuan and Kris Kitani. 2020. Dlow: Diversifying latent flows for diverse human motion prediction. In *European Conference on Computer Vision*. Springer, 346–364.

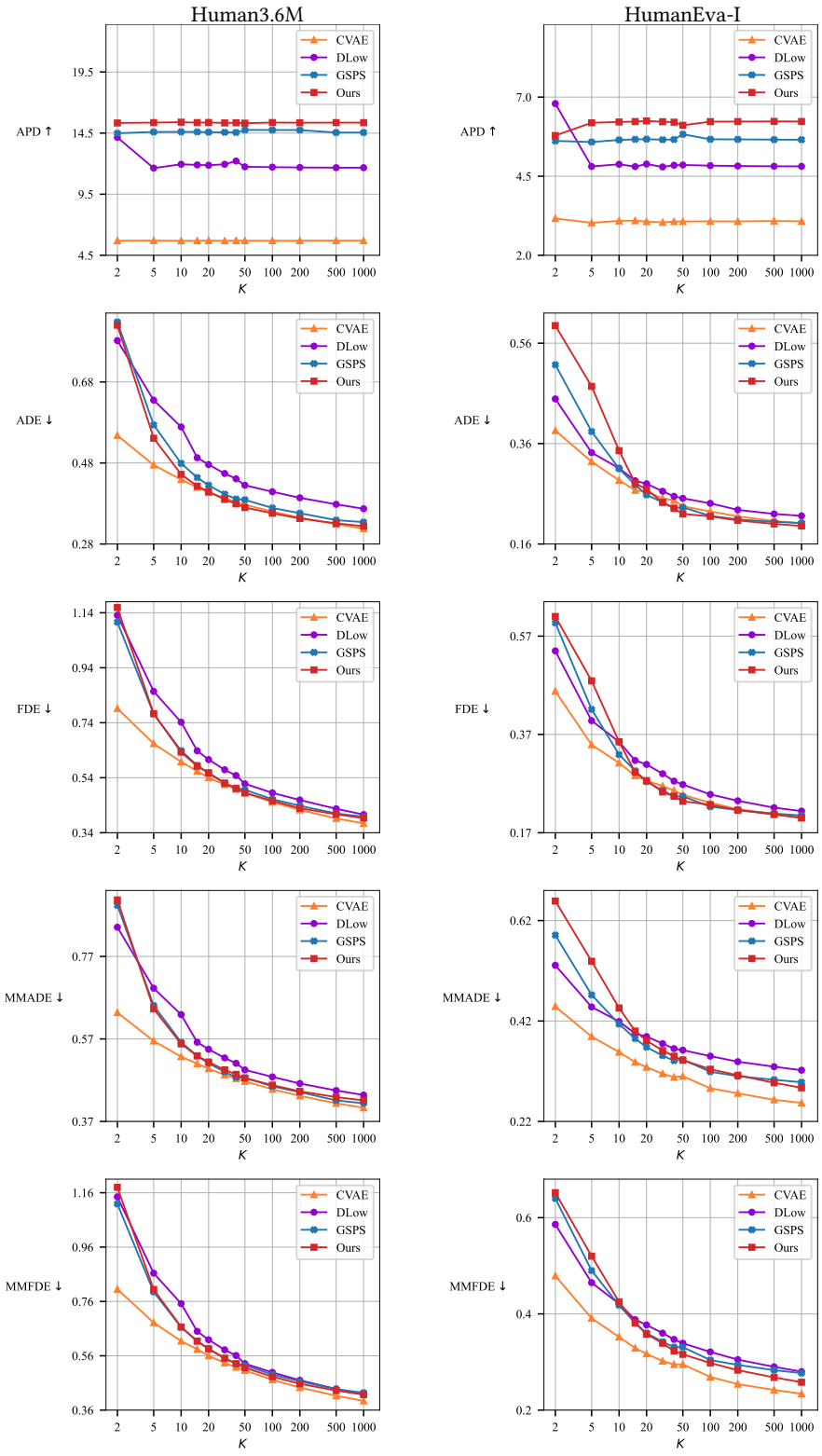
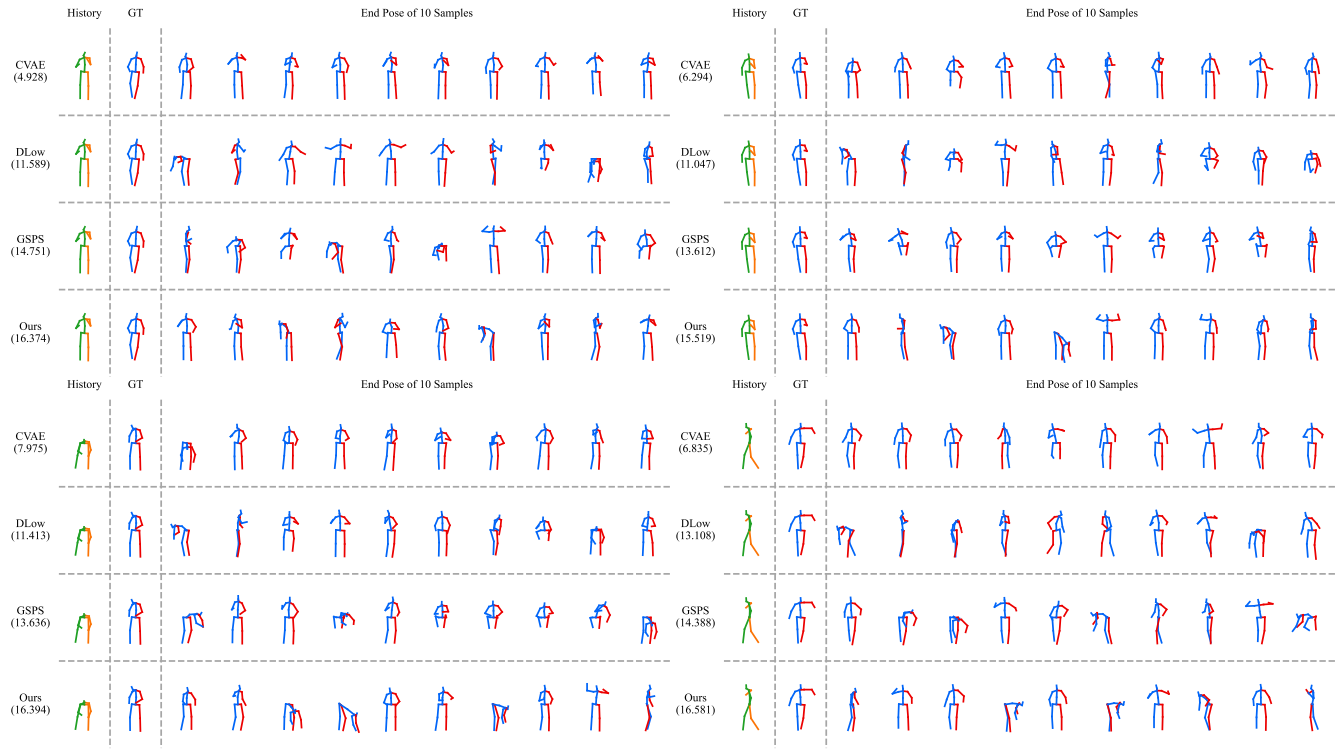


Figure 3: The first row shows how APD of different methods (including CVAE, DLow, GSPS, and our method) varies as K increases. The other rows show the trends of ADE, FDE, MMADE, MMFDE, respectively. The figures on the left are plotted based on the data computed on Human3.6M, while those on the right are plotted based on HumanEva-I.

(a) Human3.6M

(b) HumanEva-I

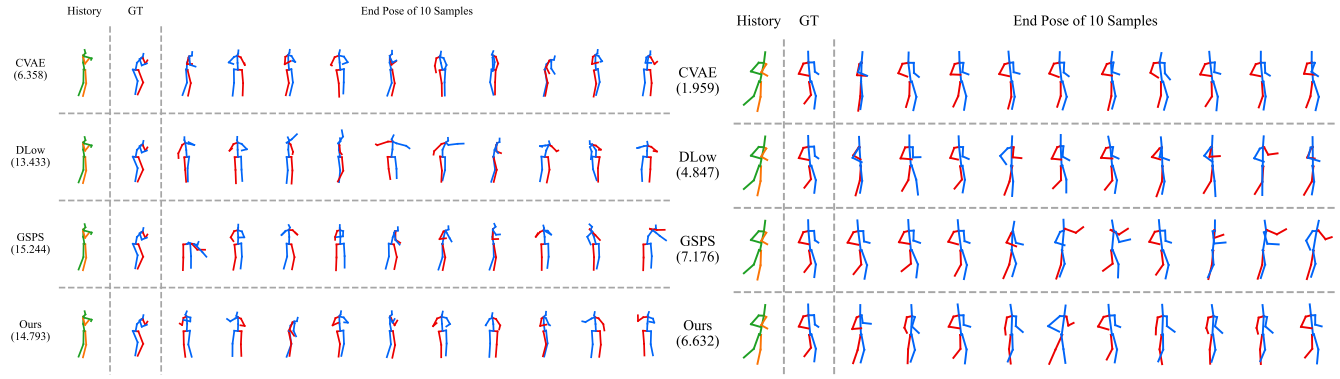
Figure 4: More qualitative results of CVAE, DLow, GSPS, and our method. The numbers in the brackets below the names of different methods show the diversity of the results computed by these methods. In these examples, our results are more diverse than the results of the other methods.



(a) Human3.6M

(b) HumanEva-I

Figure 5: More qualitative results of CVAE, DLow, GSPS, and our method. The numbers in the brackets below the names of different methods show the diversity of the results computed by these methods. In these examples, our results are more diverse than the results of the other methods.



(a) Human3.6M (b) HumanEva-I

Figure 6: Two examples for which our method generates lower diverse results than GSPS.

(a) CVAE (b) DLow

(c) GSPS (d) Ours

Figure 7: Examples of implausible poses in the results of DLow, GSPS, and our method.