

Sequential Texts Driven Cohesive Motions Synthesis with Natural Transitions

Supplementary Material

Shuai Li^{1,3}, Sisi Zhuang¹, Wenfeng Song^{2*}, Xinyu Zhang², Hejia Chen¹, Aimin Hao¹

¹State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, P.R. China

²Computer School, Beijing Information Science and Technology University, P.R. China

³Zhongguancun Laboratory, Beijing, P.R. China

{lishuai, sisizhuang}@buaa.edu.cn, songwenfenga@gmail.com

zhangxinyu1@bistu.edu.cn, {chenhj2000, ham}@buaa.edu.cn

A. Overview

More concrete details are provided in this supplementary material. In Sec. B, our model architecture is depicted for intuitional understanding. In Sec. C, our user study procedure and framework are described. Sec. D introduces the evaluation metrics of our task, and Sec. E compares two segmented training strategies. More generated results are illustrated in Sec. F for better visualization of the advances and limitations of our work.

B. Neural Network Architecture

Tab. 9 illustrates the architecture of our whole pipeline.

C. User Study

We invite 45 users to evaluate our results further. 50 texts are randomly selected from the testing data, and 3-5 texts are formed into sequential texts to synthesize motion using different methods. Users are asked to score the motion using semantic matching degree, transition fluency, and realism. The higher the score, the better the performance of the current sequence on this indicator. Fig. 1 shows our questionnaire interface. The black text indicates the completed motion, and the red text indicates the ongoing motion.

D. Evaluation Metrics

In the evaluation stage, to evaluate the synthesis quality of the motion sequence of the sequential texts, we synthesize two adjacent texts into one text with commas, spliced two adjacent motions into one motion, and the text/motion mentioned below represents the assembled text/motion. In addition to our proposed transition fluency metric, we also use the same five metrics as Guo et al. [2]. Here is a detailed description of these five metrics:

*Corresponding author

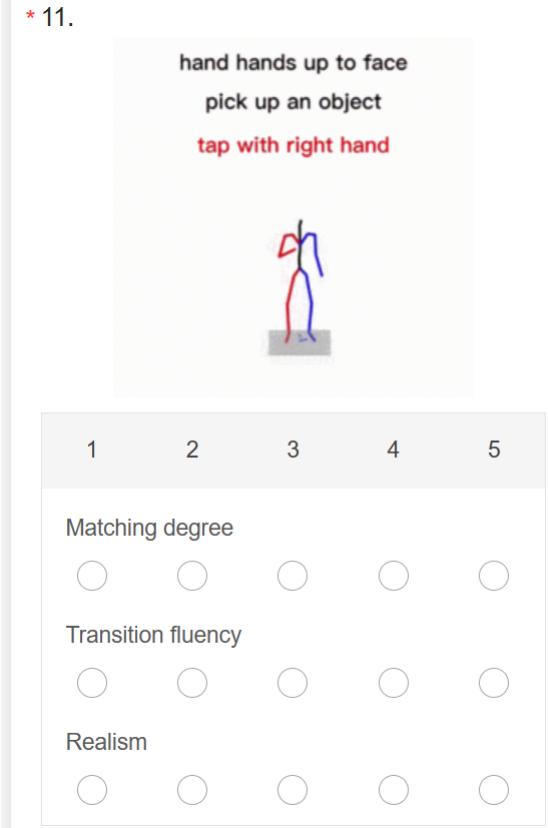


Figure 1: **The user interface of our user study:** Participants evaluate the motion in three aspects (Matching degree, Transition fluency and Realism), from 1 (“No Satisfaction”) to 5 (“High Satisfaction”). The black text indicates the completed motion, and the red text indicates the ongoing motion.

R Precision: For each generated motion, compute the feature euclidean distances between the motion se-

Methods	R Precision↑			FID↓	MultiModal Dist↓	Transition Fluency↓
	Top 1	Top 2	Top 3			
Real motions	0.560 \pm 0.002	0.750 \pm 0.003	0.834 \pm 0.002	0.001 \pm 0.000	3.448 \pm 0.001	-
T2M [2]	0.376 \pm 0.003	0.543 \pm 0.002	0.647 \pm 0.003	6.890 \pm 0.081	5.218 \pm 0.010	1.108 \pm 0.004
T2M-Gus	0.371 \pm 0.002	0.538 \pm 0.003	0.643 \pm 0.003	7.203 \pm 0.075	5.264 \pm 0.014	0.364 \pm 0.001
Ours	0.542 \pm 0.003	0.728 \pm 0.003	0.818 \pm 0.003	1.628 \pm 0.031	3.662 \pm 0.010	0.177 \pm 0.001

Table 1: **Quantitative evaluation on testing data of BABEL-TEACH:** The results show that the motion synthesized by our model outperforms other baselines in terms of semantic matching, transition fluency, and realism.

Methods	R Precision↑			FID↓	MultiModal Dist↓	Transition Fluency↓
	Top 1	Top 2	Top 3			
Real motions	0.322 \pm 0.004	0.505 \pm 0.005	0.625 \pm 0.006	0.014 \pm 0.002	3.461 \pm 0.005	0.454 \pm 0.000
T2M [2]	0.322 \pm 0.006	0.493 \pm 0.005	0.614 \pm 0.005	1.389 \pm 0.049	3.500 \pm 0.009	0.639 \pm 0.003
T2M-Gus	0.314 \pm 0.004	0.485 \pm 0.005	0.607 \pm 0.005	1.599 \pm 0.046	3.541 \pm 0.010	0.226 \pm 0.001
Ours	0.328 \pm 0.006	0.510 \pm 0.006	0.633 \pm 0.005	1.085 \pm 0.063	3.441 \pm 0.017	0.109 \pm 0.001

Table 2: **Quantitative evaluation on the testing data of STDM:** The results show that the motion synthesized by our model outperforms other baselines in terms of semantic matching, transition fluency, and realism.

Methods	Diversity \rightarrow	MultiModality \uparrow
Real motions	10.515 \pm 0.090	-
T2M [2]	9.304 \pm 0.096	2.123 \pm 0.076
T2M-Gus	9.222 \pm 0.078	<u>2.150</u> \pm 0.094
T2M-Joint	9.521 \pm 0.063	2.121 \pm 0.082
Compulsion-Code	9.965 \pm 0.062	3.636 \pm 0.104
TEACH (no Slerp) [1]	<u>10.735</u> \pm 0.080	0.466 \pm 0.011
TEACH (Slerp) [1]	10.790 \pm 0.091	0.496 \pm 0.014
Ours	10.340 \pm 0.072	<u>2.127</u> \pm 0.060
Ours (Slerp)	<u>10.335</u> \pm 0.060	2.127 \pm 0.058

Table 3: **Diversity and multimodality evaluation on testing data of BABEL-TEACH:** Our model performs well on diversity and multimodality metrics.

quence and its ground-truth text, as well as the feature euclidean distances between the motion sequence and 31 non-matching texts randomly selected from the testing set. The 32 feature euclidean distances are sorted, and if the ground-truth text falls in the top-k positions, it is considered that the top-k positions are retrieved successfully. Finally, we calculate the average accuracy of the top-1, top-2, and top-3 retrieval successes.

Frechet Inception Distance (FID): FID is calculated between the feature distribution of generated motions and the real motions. The smaller the value is, the more similar the generated motion sequence is to the real motion sequence.

Methods	Diversity \rightarrow	MultiModality \uparrow
Real motions	9.623 \pm 0.118	-
T2M [2]	9.560 \pm 0.091	1.029 \pm 0.036
T2M-Gus	<u>9.500</u> \pm 0.108	1.020 \pm 0.054
T2M-Joint	<u>9.474</u> \pm 0.090	1.533 \pm 0.058
Compulsion-Code	8.427 \pm 0.068	2.402 \pm 0.074
TEACH (no Slerp) [1]	9.371 \pm 0.084	0.398 \pm 0.015
TEACH (Slerp) [1]	9.347 \pm 0.099	0.394 \pm 0.014
Ours	9.382 \pm 0.105	<u>1.889</u> \pm 0.051
Ours (Slerp)	9.426 \pm 0.113	<u>1.814</u> \pm 0.049

Table 4: **Diversity and multimodality evaluation on testing data of STDM:** Our model performs well on diversity and multimodality metrics.

Methods	Diversity \rightarrow	MultiModality \uparrow
Real motions	10.529 \pm 0.112	-
w/o Transition reasoning	10.188 \pm 0.092	2.105 \pm 0.084
w/o L2G semantic fusion	9.582 \pm 0.099	<u>2.472</u> \pm 0.133
w/o Transition loss	<u>10.207</u> \pm 0.094	2.197 \pm 0.063
w/o Segmented training	10.186 \pm 0.074	2.737 \pm 0.101
Ours	10.257 \pm 0.061	2.105 \pm 0.086

Table 5: **Diversity and multimodality evaluation on testing data of ablation study:** Our model performs well on diversity and multimodality metrics.

Num of snippet code	Diversity→	MultiModality↑
Real motions	10.544 ± 0.085	-
1 snippet code	10.184 ± 0.098	2.108 ± 0.073
2 snippet codes	10.374 ± 0.075	2.116 ± 0.056
3 snippet codes	10.053 ± 0.063	2.208 ± 0.074
5 snippet codes	10.161 ± 0.053	2.248 ± 0.092

Table 6: **Diversity and multimodality evaluation in terms of parameter analysis of transition reasoning module:** The diversity and multimodality of various inference numbers of start snippet code in the transition reasoning module are similar.

MultiModal Distance: For the motion sequence generated by each text, the average euclidean distance between the feature of the text and the feature of the motion sequence generated by the text is calculated. The smaller the average distance, the more the generated motion sequence matches the text.

Diversity: Diversity measures the distance of the motion sequence generated by different texts. For two sets of the texts of size K , generate corresponding motion sequences and extract feature vectors, that is, two sets of motion feature vectors $\mathbf{f} = (\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_K)$ and $\hat{\mathbf{f}} = (\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \dots, \hat{\mathbf{f}}_K)$. Calculate the euclidean distance between the motion feature of the two sets. The higher the value, the higher the motion diversity of different texts. The metric should be as close to the actual value of ground truth as possible when evaluating diversity. The mathematical representation is as:

$$Diversity = \frac{1}{K} \sum_{i=1}^K \|\mathbf{f}_i - \hat{\mathbf{f}}_i\| \quad (1)$$

MultiModality: MultiModality measures the feature distance between the same text. Given C texts, we go through each text iteratively. For c -th text, generate two subsets of motion sequences and extract feature vectors $\mathbf{f}_c = (\mathbf{f}_{c,1}, \mathbf{f}_{c,2}, \dots, \mathbf{f}_{c,Q})$ and $\hat{\mathbf{f}}_c = (\hat{\mathbf{f}}_{c,1}, \hat{\mathbf{f}}_{c,2}, \dots, \hat{\mathbf{f}}_{c,Q})$ with same size Q , and the average feature euclidean distance of the Q motion sequences is calculated. The higher the value, the higher the diversity of the motion sequence generated by the same text. The mathematical representation is as:

$$MultiModality = \frac{1}{C \times Q} \sum_{c=1}^C \sum_{i=1}^Q \|\mathbf{f}_{c,i} - \hat{\mathbf{f}}_{c,i}\| \quad (2)$$

In addition to the several baselines shown in the paper, we also compare the experiments of adding Gaussian smoothing to the T2M model. The experimental results are shown in Tab. 1 and Tab. 2. The results show that the transition smooth motion can not be obtained by simply performing Gaussian smoothing based on T2M.

Training strategy	Diversity→	MultiModality↑
Real motions	10.544 ± 0.085	-
Scheme 1	9.257 ± 0.084	1.802 ± 0.056
Scheme 2	10.377 ± 0.100	2.079 ± 0.056

Table 7: **Diversity and multimodality evaluation in terms of segmented training strategy:** Scheme 2 performs better than scheme 1 on diversity and multimodality metrics.

Due to the limitation of the length of the paper’s main text, we omitted the display of the two metrics (Diversity and MultiModality) in the main text. Tab. 3, Tab. 4, Tab. 5 and Tab. 6 are supplementary experimental results for evaluating diversity and multimodality. Our method takes into account the smoothness of the transition between adjacent motions by eliminating any unreasonable motion during the transition. Our results show that although our model may not be the best in terms of diversity and multimodality, it still performs well.

E. Segmented Training Strategy

In the segmented training strategy, we propose two training schemes. The results are shown in Tab. 8 and Tab. 7.

Scheme 1: Each motion item in our dataset comprises two consecutive motion sequences. To ensure consistency, we sort all the data in ascending order based on the shorter sequence length. Then, we define a standard length for each stage and discard any data that is shorter than this length. For the remaining data, we crop them to the standard length. Importantly, during cropping, we ensure that the first half of the later motion sequence and the second half of the previous motion sequence are retained. This guarantees that the transition between the two sequences remains complete after cropping.

Scheme 2: Each motion item in our dataset consists of two adjacent motion sequences, arranged in ascending order of the length of the longer of the two. To standardize the length of the data, we set a standard length for each stage and discard any data that is longer than the standard length. The remaining data is then padded with zeros to reach the standard length. When padding, we ensure that the previous motion sequence in each data point retains the second half, and the subsequent motion sequence retains the first half. This ensures that the transition between the two sequences remains complete after padding.

The standard length of both schemes starts from 16 frames, and each stage increases by 4 frames to 196 frames. The experimental results show that scheme 2 is better than scheme 1.

Training strategy	R Precision↑			FID↓	MultiModal Dist↓	Transition Fluency↓
	Top 1	Top 2	Top 3			
Real motions	0.562 ± 0.003	0.748 ± 0.003	0.833 ± 0.002	0.002 ± 0.000	3.450 ± 0.002	-
Scheme 1	0.404 ± 0.003	0.576 ± 0.004	0.680 ± 0.003	4.550 ± 0.047	4.850 ± 0.006	0.304 ± 0.001
Scheme 2	0.540 ± 0.004	0.721 ± 0.003	0.812 ± 0.003	1.635 ± 0.036	3.674 ± 0.015	0.178 ± 0.002

Table 8: **Segmented training strategy:** Two segmented training strategies are tested, and the results show that scheme 2 is more appropriate.

F. More Results

Our method synthesizes semantic human motion with natural transition using free-form sequential texts, we show more results synthesized from our method (shown in Fig. 2 and Fig. 3). We show more visualization results compared to the other methods (shown in Fig. 4, Fig. 5, Fig. 6 and Fig. 7). The motion synthesized by our method is the best in terms of semantic matching, transition fluency, and realism.

Failure cases and limitations. The first row of Fig. 8 shows that when only synthesizing head motions, such as “nodding” and “shaking the head,” the digital person has almost no change, there are too few joint nodes involved in the head motion, it is difficult for our model to learn such subtle changes from the overall situation. The second row of Fig. 8 shows that when we switch from “sitting in a chair” to “moving your left foot,” the digital man lie down. The possible reason is the lack of similar data in the training data, and the understanding of the model is biased. The third line in Fig. 8 shows that the model does not understand the abstract word “clean” properly, the digital man only raising his right hand, but does not reflect the semantics of “clean,” indicating that our model still has room for further exploration of abstract semantics.

References

- [1] Nikos Athanasiou, Mathis Petrovich, Michael J. Black, et al. TEACH: Temporal action composition for 3d humans. *2022 International Conference on 3D Vision (3DV)*, pages 414–423, 2022. [2](#)
- [2] Chuan Guo, Shihao Zou, Xinxin Zuo, et al. Generating diverse and natural 3d human motions from text. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. [1](#), [2](#)

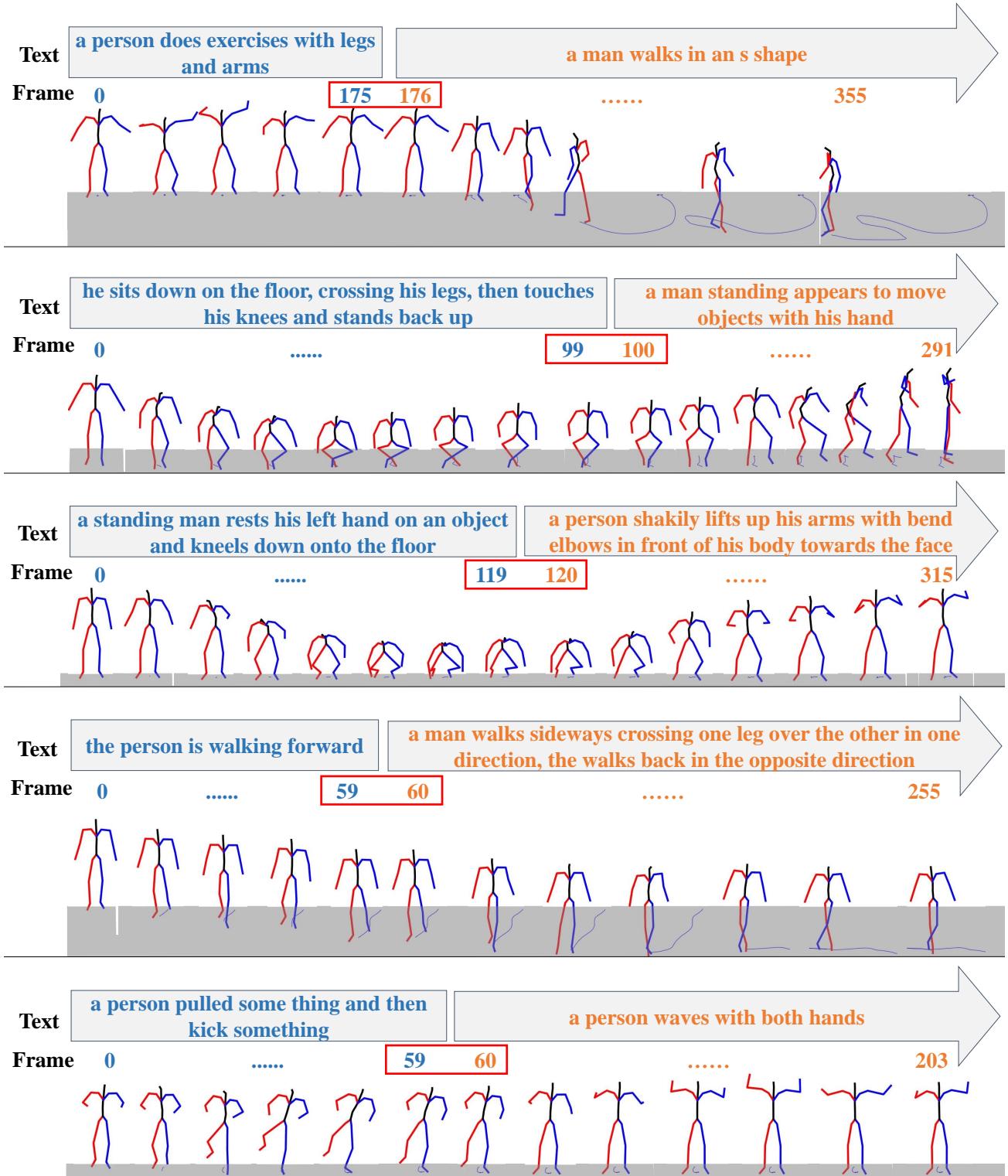


Figure 2: **Additional examples synthesized from our method:** By entering the sequential texts, we show the motion sequence synthesized by our model. Our synthesized motion matches the semantics of sequential texts, and the transitions between adjacent motions are natural and smooth.

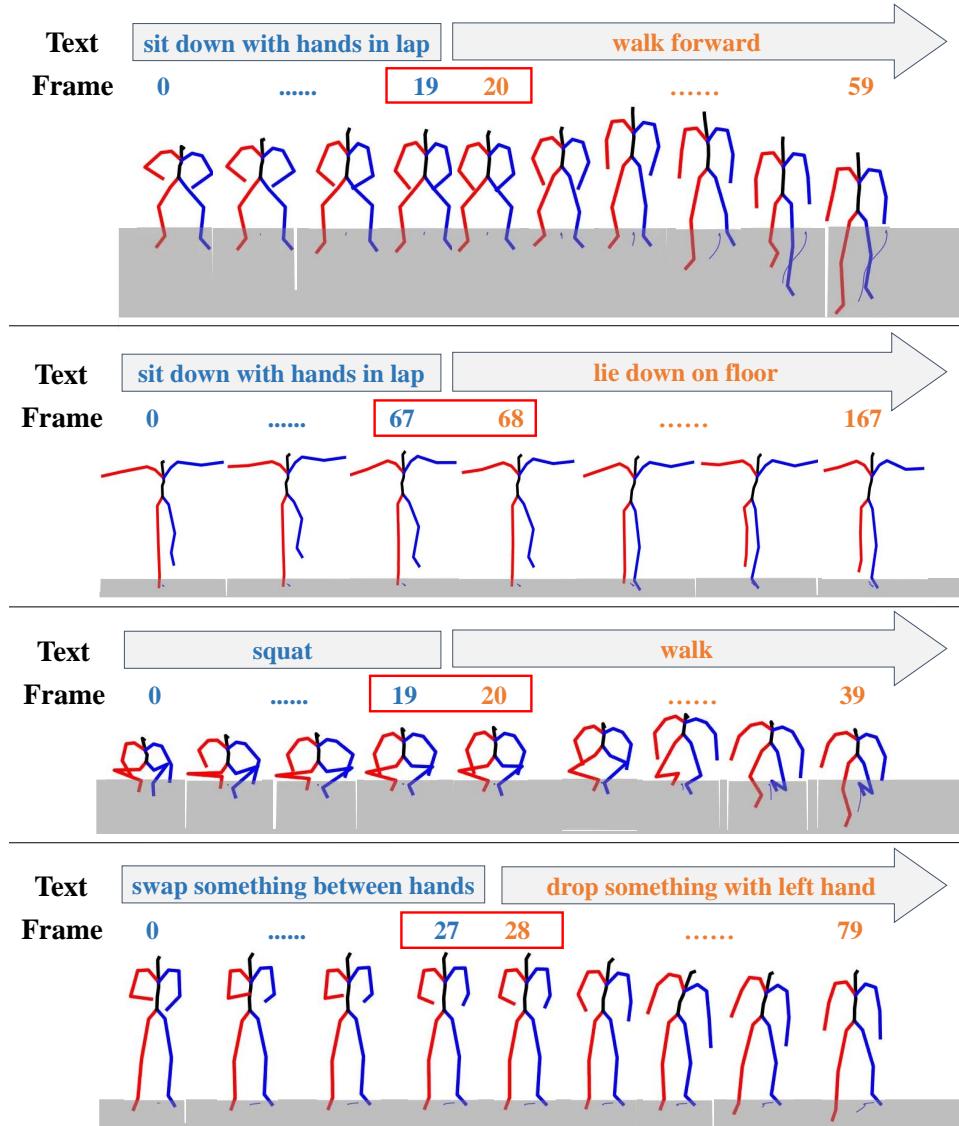


Figure 3: **Additional examples synthesized from our method:** By entering the sequential texts, we show the motion sequence synthesized by our model. Our synthesized motion matches the semantics of sequential texts, and the transitions between adjacent motions are natural and smooth.

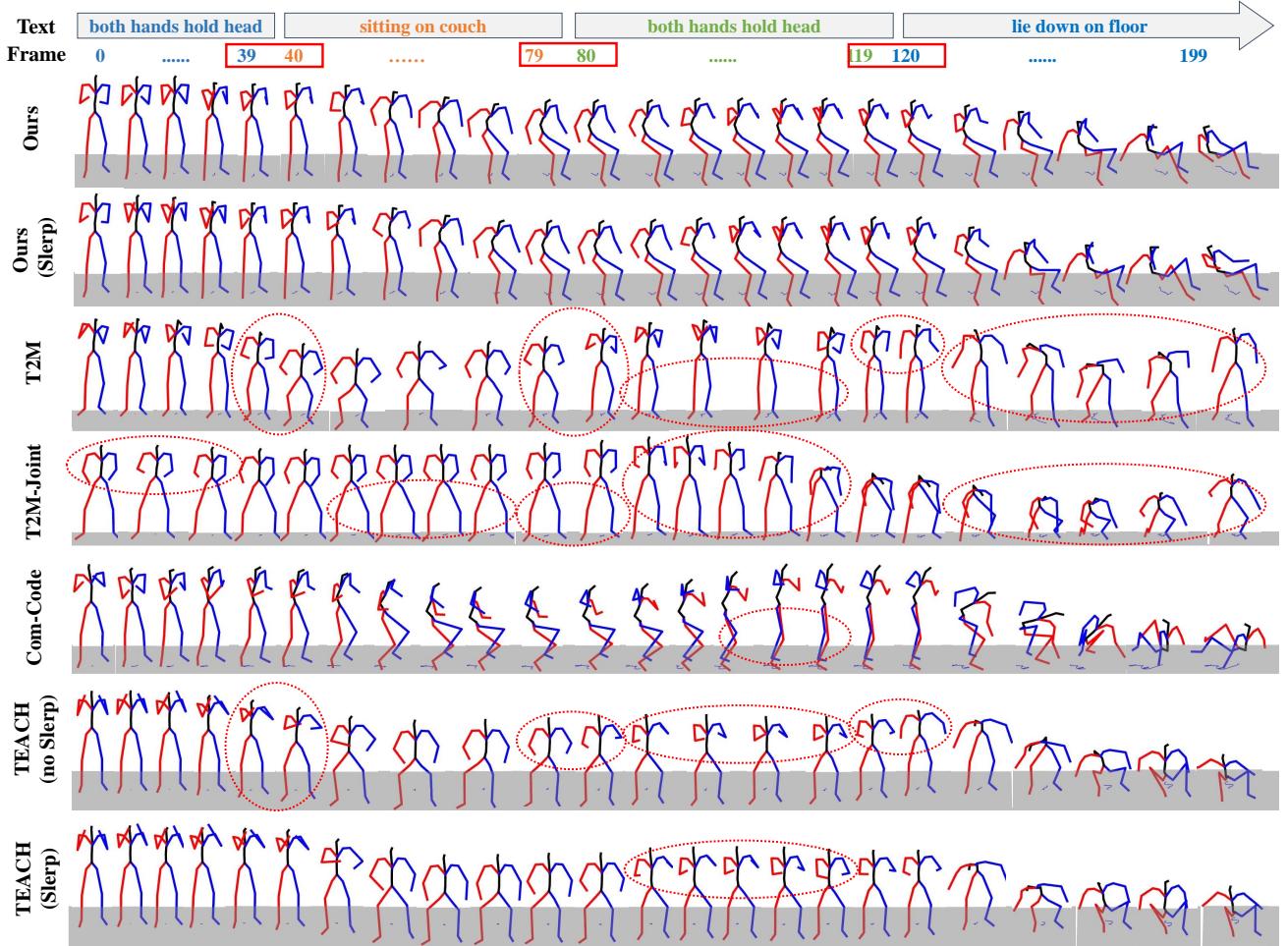


Figure 4: Comparison with other methods: We show more visualization results compared to the other methods. T2M is the most unnatural at the transition and does not reflect the semantics of the last two texts, T2M-Joint does not reflect the semantics of all texts, the semantics of Com-Code are incoherent (stand up in the third text), TEACH (no Slerp) is unnatural at the transition. The motion synthesized by our method is the best in terms of semantic matching, transition fluency, and realism.

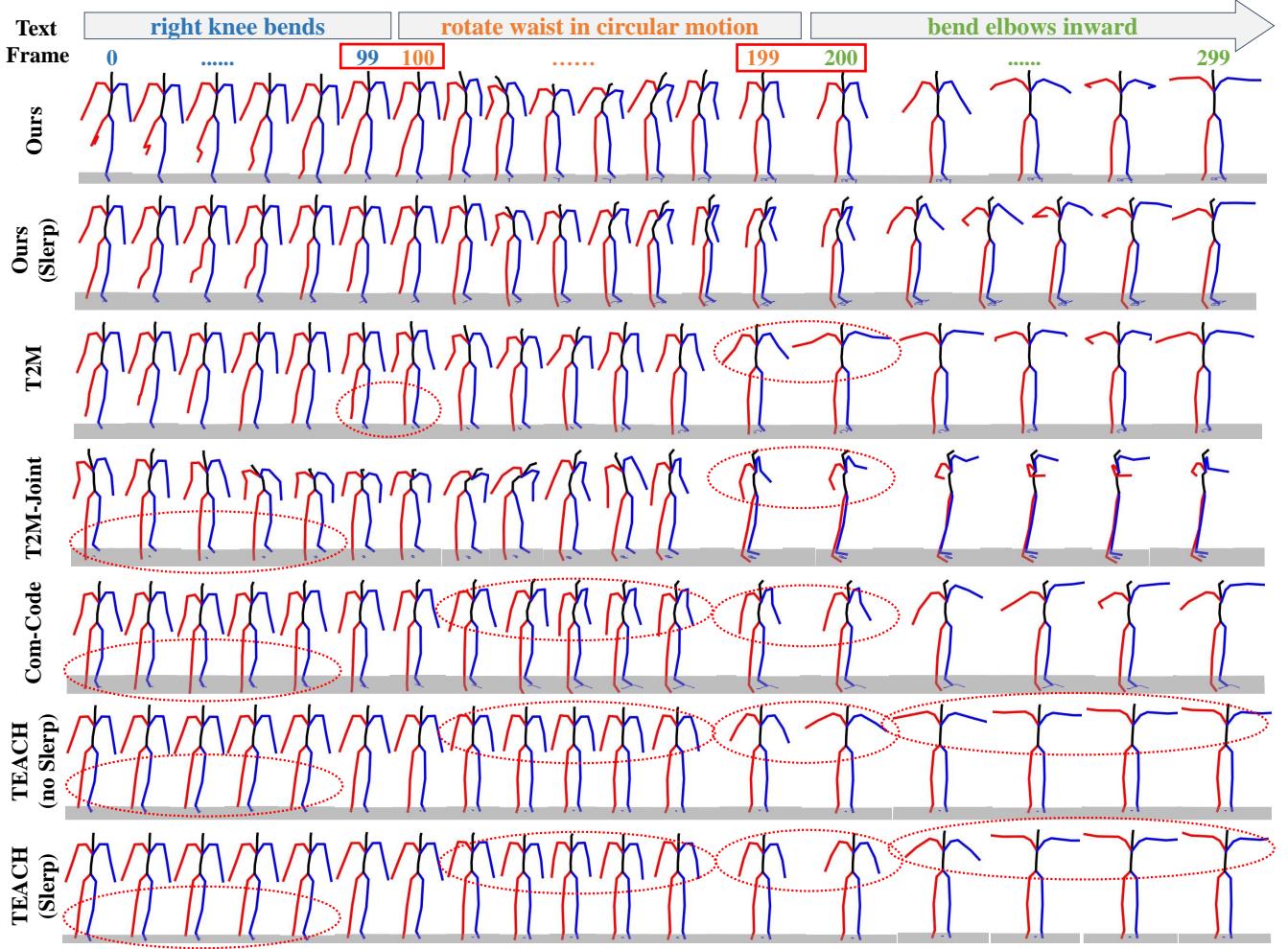


Figure 5: Comparison with other methods: We show more visualization results compared to the other methods. T2M is the most unnatural at the transition, T2M-Joint is unnatural at the transition and does not reflect the semantics of the first text, Com-Code is unnatural at the transition and does not reflect the semantics of the first two texts, TEACH is unnatural at the transition and does not reflect the semantics of all texts. The motion synthesized by our method is the best in terms of semantic matching, transition fluency, and realism.

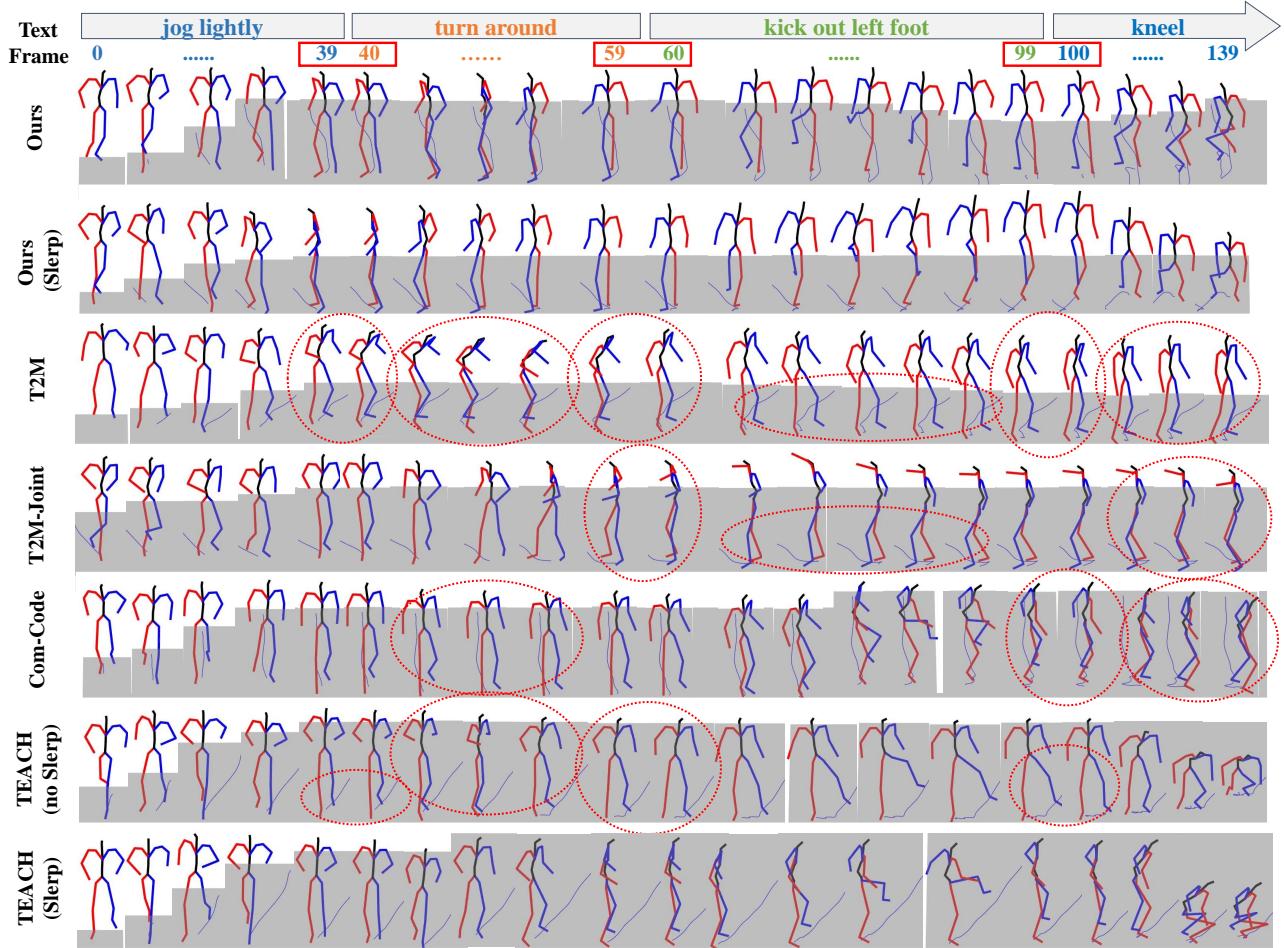


Figure 6: Comparison with other methods: We show more visualization results compared to the other methods. T2M is the most unnatural at the transition and does not reflect the semantics of the last three texts, T2M-Joint is unnatural at the transition and does not reflect the semantics of the last two texts, Com-Code is unnatural at the transition and does not reflect the semantics of the second and the fourth text, TEACH (no Slerp) is unnatural at the transition and does not reflect the semantics of the second text. The motion synthesized by our method is the best in terms of semantic matching, transition fluency, and realism.

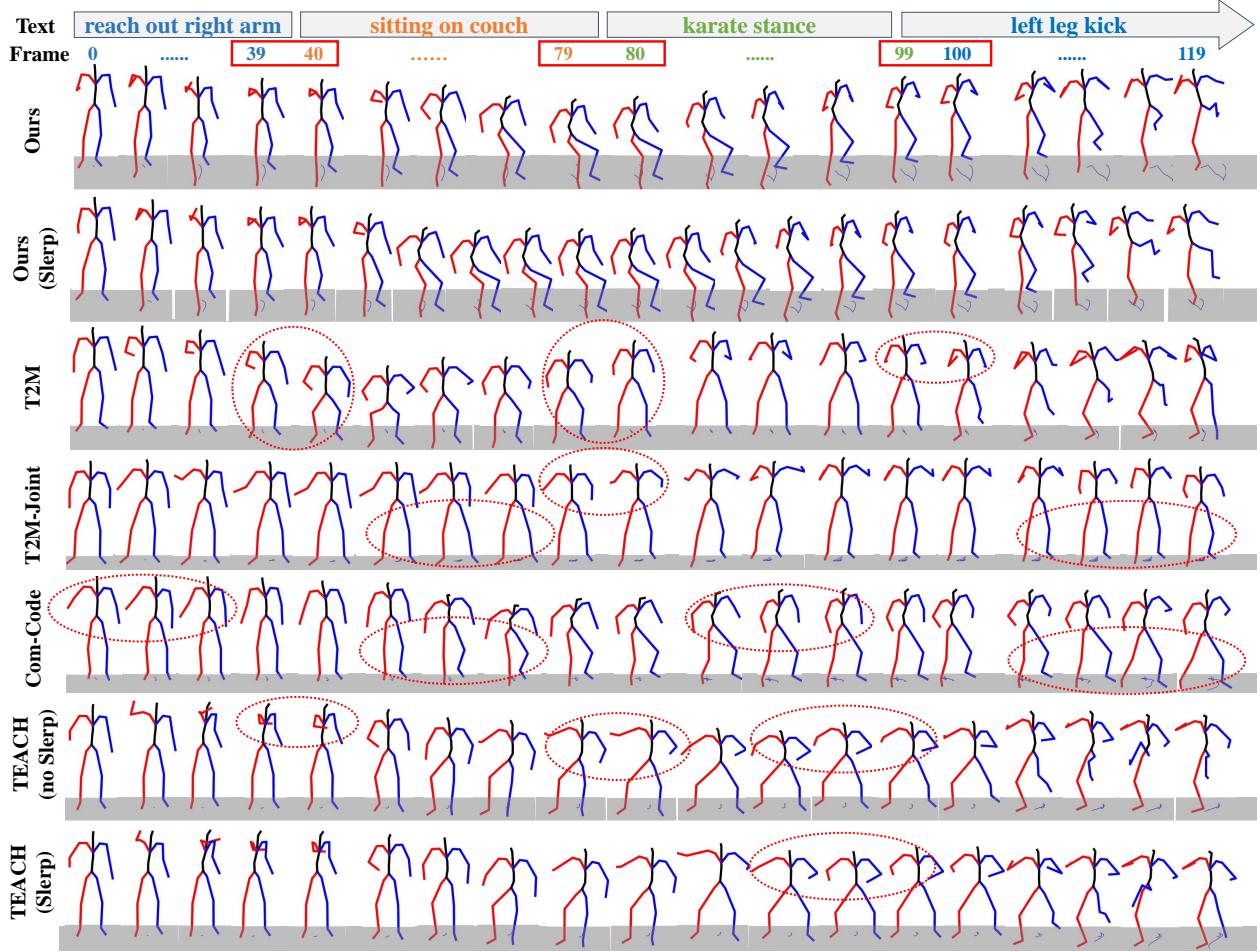


Figure 7: Comparison with other methods: We show more visualization results compared to the other methods. T2M is the most unnatural at the transition, T2M-Joint is unnatural at the transition and does not reflect the semantics of the second and the fourth texts, Com-Code is unnatural at the transition and does not reflect the semantics of all texts, TEACH (no Slep) is unnatural at the transition. The motion synthesized by our method is the best in terms of semantic matching, transition fluency, and realism.

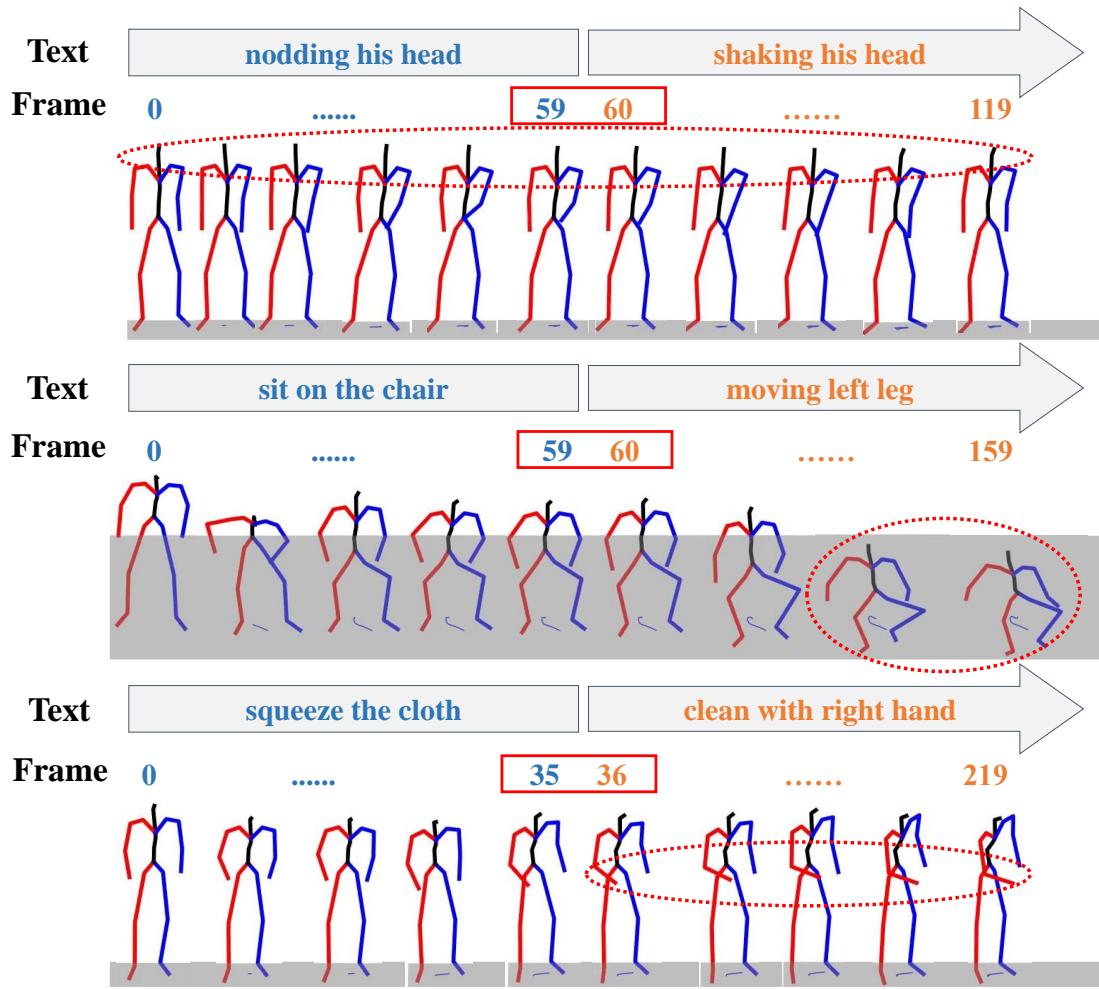


Figure 8: **Examples of failure cases:** Our model does not perform well at synthesizing head motion and still needs to understand some abstract words well.

Components	Architecture
Transition Reasoning Encoder	(sequence_pos_encoder): PositionalEncoding_Transition() (seqTransEncoder): TransformerEncoder(d_model=512, nhead=4, dim_feedforward=1024, dropout=0.1, activation="gelu", num_layers=4)
Previous Text Encoder	(clip_model): CLIP() (input_emb): Linear(in_features=512, out_features=1024, bias=True)
Current Text Encoder	(pos_emb): Linear(in_features=15, out_features=300, bias=True) (input_emb): Linear(in_features=300, out_features=512, bias=True) (gru): GRU(512, 512, batch_first=True, bidirectional=True)
Semantic Fusion Attention	(W_q): Linear(in_features=1024, out_features=512, bias=True) (W_k): Linear(in_features=1024, out_features=512, bias=False) (W_v): Linear(in_features=1024, out_features=512, bias=True) (softmax): Softmax(dim=1)
Motion Encoder	Conv1d(247, 384, kernel_size=(4,), stride=(2,), padding=(1,)) Dropout(p=0.2, inplace=True) LeakyReLU(negative_slope=0.2, inplace=True) Conv1d(384, 512, kernel_size=(4,), stride=(2,), padding=(1,)) Dropout(p=0.2, inplace=True) LeakyReLU(negative_slope=0.2, inplace=True) Linear(in_features=512, out_features=512, bias=True)
Motion Decoder	ConvTranspose1d(512, 384, kernel_size=(4,), stride=(2,), padding=(1,)) LeakyReLU(negative_slope=0.2, inplace=True) ConvTranspose1d(384, 251, kernel_size=(4,), stride=(2,), padding=(1,)) LeakyReLU(negative_slope=0.2, inplace=True) Linear(in_features=251, out_features=251, bias=True)
Prior Network / Posterior Network	(z2init): Linear(in_features=1024, out_features=1024, bias=True) (embedding): Sequential((0): Linear(in_features=1024 / 1536, out_features=1024, bias=True) (1): LayerNorm((1024,), eps=1e-05, elementwise_affine=True) (2): LeakyReLU(negative_slope=0.2, inplace=True)) (gru): ModuleList((0): GRUCell(1024, 1024)) (positional_encoder): PositionalEncoding() (mu_net): Linear(in_features=1024, out_features=128, bias=True) (logvar_net): Linear(in_features=1024, out_features=128, bias=True)
Motion Snippet Code Generator	(z2init): Linear(in_features=1024, out_features=1024, bias=True) (embedding): Sequential((0): Linear(in_features=1152, out_features=1024, bias=True) (1): LayerNorm((1024,), eps=1e-05, elementwise_affine=True) (2): LeakyReLU(negative_slope=0.2, inplace=True)) (gru): ModuleList((0): GRUCell(1024, 1024)) (positional_encoder): PositionalEncoding() (output_net): Sequential((0): Linear(in_features=1024, out_features=1024, bias=True) (1): LayerNorm((1024,), eps=1e-05, elementwise_affine=True) (2): LeakyReLU(negative_slope=0.2, inplace=True) (3): Linear(in_features=1024, out_features=512, bias=True))

Table 9: **Architecture of our networks.**