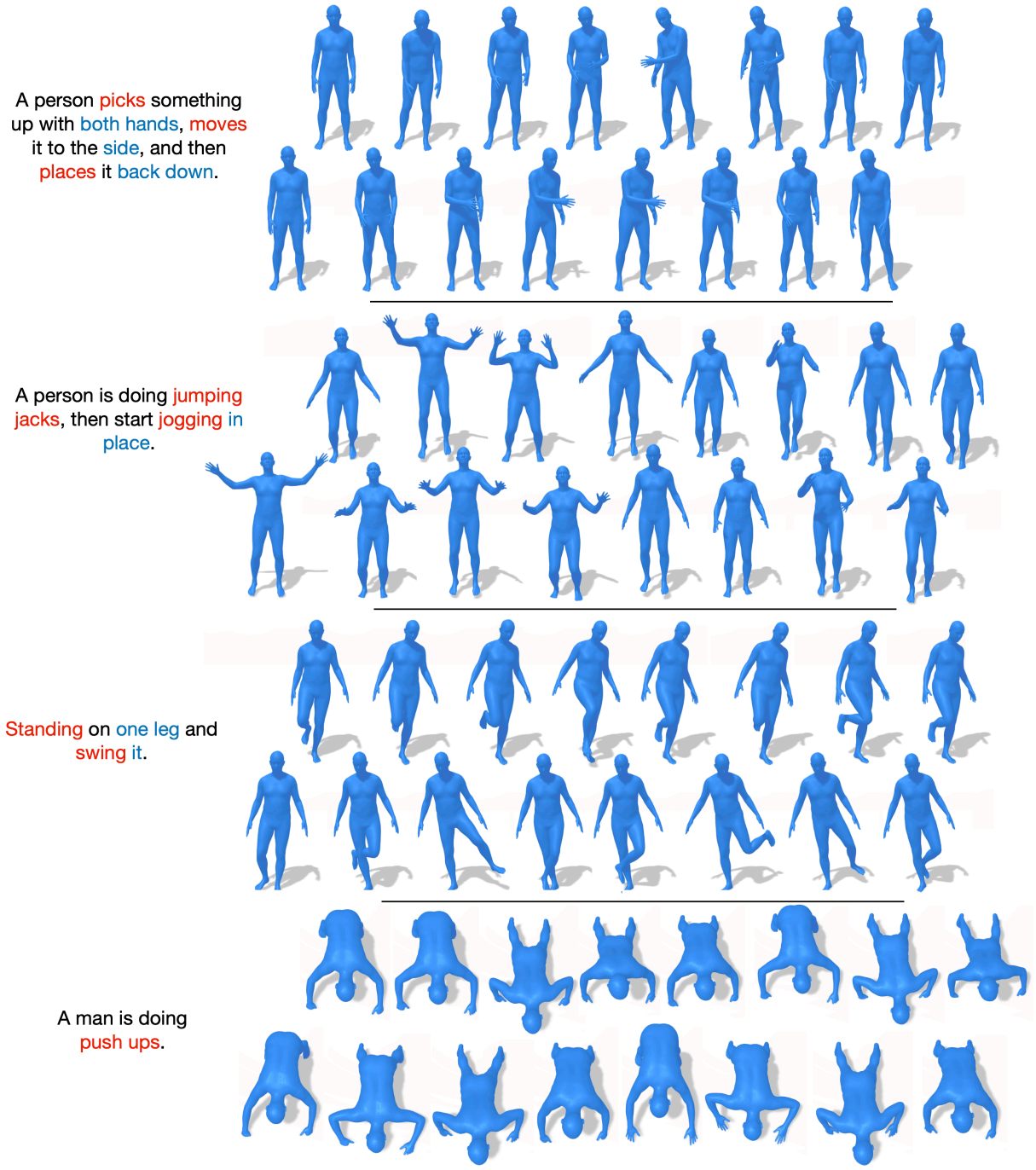


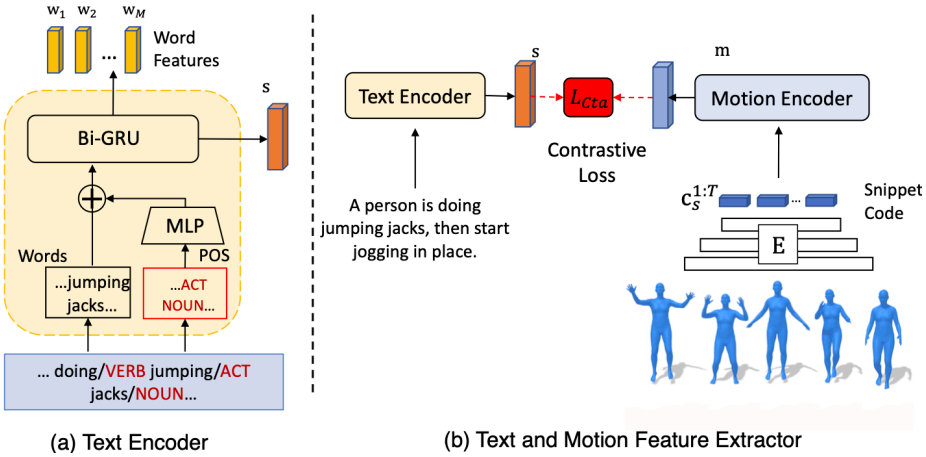
Figure 1. **Additional examples generated from our method.** For each description (left), we show two distinct synthetic motions (right). Key frames are displayed for each sequence. Refer to supplementary video for dynamic animations.

APPENDIX

Abstract

In this supplementary material, we provide more details on implementations, evaluation metrics, baseline implementation, data annotation and user study, dataset results, text2length results, text2motion results, out-of-dataset generation, network architectures and table of notations.

- ***Implementation Details.*** *Here, we provide details on practical implementations.*
- ***Evaluation Metrics.*** *We provide the details of extracting motion and text features for the purpose of quantitative evaluation, as well as mathematical explana-*



(a) Text Encoder (b) Text and Motion Feature Extractor

Figure 2. (a) Architecture of our text encoder. (b) The pipeline of training text and motion extractors.

tions of three evaluation metrics: FID, Diversity and Multimodality.

- ***Baseline Implementation.*** *We provide the details of implementing comparison baseline methods.*
- ***Data Annotation and User Study.*** *We elaborate the process of data annotation and user study survey.*
- ***Dataset Results.*** *We showcase several paired text and motion data in our HumanML3D dataset.*
- ***Text2Length Results.*** *We provide qualitative results of our text2length module.*
- ***Text2Motion Results.*** *We provide more qualitative results of our text2motion module.*
- ***Out-of-the-Dataset Generation, Failure Cases and Limitations.*** *We provide discussions about out-of-the-dataset generation, failure cases, and limitations of our method.*
- ***Network Architectures.*** *We provide details of network structure in our method, including motion autoencoder, text encoder, prior/posterior network, and generator.*
- ***Table of Notations.*** *We provide a table enumerating and explaining all the notations used in our main content, as well as their practical values.*

A. Implementation Details

Our framework is implemented by PyTorch. Dimensions of pose vector for HumanML3D and KIT-ML dataset are 263 and 251 respectively. The motion snippet codes $\mathbf{c}_s$ are 512-dimensional vectors. Word features are 300-dimensional embedding obtained via GloVe [7]. The bi-directional GRU of our text encoder is with hidden size of 512. The triplet of generator, posterior and prior networks in our VAE are 1-layer GRUs with hidden size 1,024. The size of the noise vector $\mathbf{z}$ and the attention vector $\mathbf{w}_{att}$ are 128 and 512, respectively. Values of $\lambda_{spr}, \lambda_{smt}, \lambda_{mot}$ are set to 0.001, 0.001, and 1, respectively. λ_{KL} is set to 0.01 for HumanML3D, and to 0.005 for KIT-ML dataset. Teacher forcing rate p_{tf} is 0.4 throughout all experiments. We use Adam optimizer with learning rate of $2e^{-4}$. In text2motion training, the learning rate of motion decoder D is particularly set to $2e^{-5}$, 10 times smaller than other networks. T_{max} and T_{cur} are 50 and 8 respectively.

Z-score normalization is applied to both training and testing data. And we scale the magnitude of features $(\dot{r}^a, \dot{r}^x, \dot{r}^z, r^y, \mathbf{c}^f)$ by a value of 5 to amplify their importance. In addition, to enhance the robustness of our method, we introduce randomness to training data feed by cutting off the last 4 frames of each input pose sequence with probability of coin flipping.

B. Evaluation Metrics

Quantitatively evaluating the performance of a stochastic generative model has been investigated in pixel generation, but less touched in motion synthesis. It usually necessitates extracting abstract features from raw data (e.g., VGG [8]). Since there is no standard motion feature extractor, we train a simple framework that engages a motion extractor and text extractor under a contrastive learning assumption.

Text and Motion Feature Extractor. As shown in Fig. 2(b), a text extractor maps the raw text into a semantic feature vector $\mathbf{s}$; while pose sequences are firstly transformed into snippet codes by the pre-trained motion encoder E, and subsequently into a feature vector $\mathbf{m}$ by motion extractor. Here, we enforce matched text-motion fea-

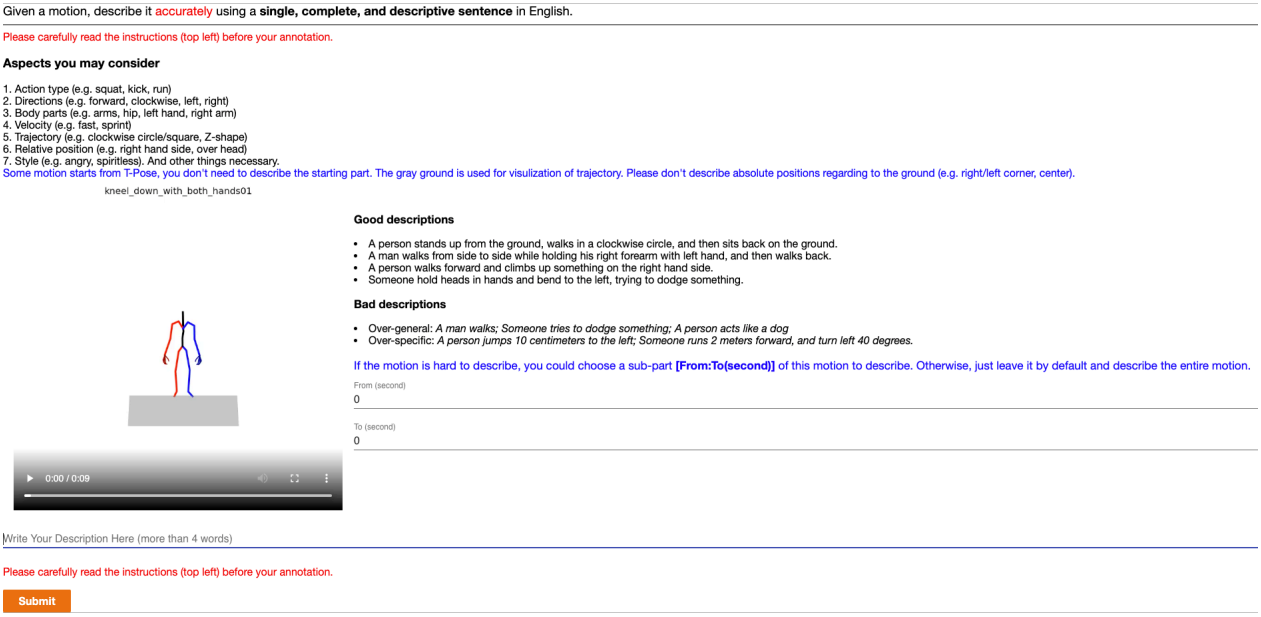


Figure 3. HumanML3D annotation interface on Amazon Mechanical Turk.

Figure 4. User study interface on Amazon Mechanical Turk.

ture pairs to be as close as possible, and mismatched text-motion features pairs to be dispersed with a margin of at least m. This is approached by employing a contrastive loss [4], which mathematically is:

$$D_{\mathbf{s},\mathbf{m}} = \|\mathbf{s} - \mathbf{m}\|_2, \tag{1}$$
$$\mathcal{L}_{Cta} = (1-y)(D_{\mathbf{s},\mathbf{m}})^2 + (y)\{\max(0, m - D_{\mathbf{s},\mathbf{m}})\}^2,$$

where y is a binary label that $y = 0$ if $\mathbf{s}$ and $\mathbf{m}$ come from matched text-motion pairs, and vice versa. $m > 0$ is a margin for mismatched pairs and set to 10 in our experiments. In practice, text extractor adopts the same architecture as our text encoder (Fig. 2(a)); motion extractor is a bidirectional GRU with hidden unit size of 1,024. Please note that test data is untouched in this process.

The aforementioned text and motion feature extractor are then employed in the following evaluation metrics.

- **Frechet Inception Distance** (FID): Features are extracted from real motions in test set and generated motions from corresponding descriptions. Then FID is

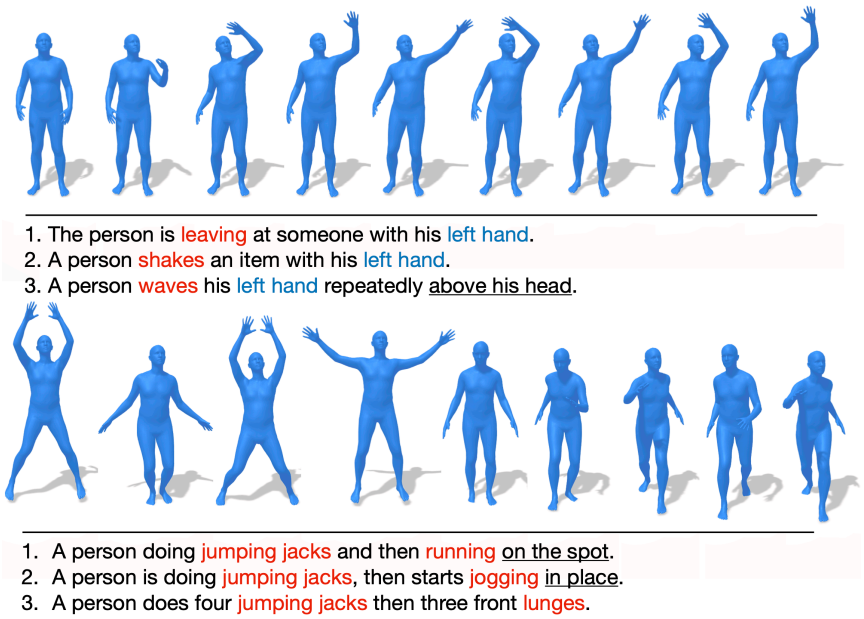


Figure 5. **Examples in HumanML3D dataset**. Two annotation examples in our HumanML3D dataset are shown. Each motion sequence comes with 3 distinct script descriptions.

calculated between the feature distribution of generated motions vs. that of the real motions. FID is an important metric widely used to evaluate the overall quality of generated motions.

- **Diversity**: Diversity measures the variance of the generated motions across all descriptions. From a set of all generated motions from various descriptions, two subsets of the same size S_d are randomly sampled. Their respective sets of motion feature vectors $\{\mathbf{v}_1, ..., \mathbf{v}_{S_d}\}$ and $\{\mathbf{v}'_1, ..., \mathbf{v}'_{S_d}\}$ are extracted. The diversity of this set of motions is defined as

  $\text{Diversity} = \frac{1}{S_d} \sum_{i=1}^{S_d} \|\mathbf{v}_i - \mathbf{v}'_i\|$

  $S_d = 300$ is used in experiments.

- **MultiModality**: Different from diversity, multimodality measures how much the generated motions diversify within each text description. Given a set of motions with C descriptions. For c-th description, we randomly sample two subsets with same size S_m , and then extract two subset of feature vectors $\{\mathbf{v}_{c,1}, ..., \mathbf{v}_{c,S_m}\}$ and $\{\mathbf{v}'_{c,1}, ..., \mathbf{v}'_{c,S_m}\}$. The multimodality of this motion set is formalized as

  $\text{Multimodality} = \frac{1}{C \times S_m} \sum_{c=1}^{C} \sum_{i=1}^{S_m} \|\mathbf{v}_{c,i} - \mathbf{v}'_{c,i}\|$

  $S_m = 10$ is used in experiments.

C. Baseline Implementation

We re-implement Seq2Seq [6] according to the descriptions in the original literature. For Language2Pose [1] and Text2Gesture [2], we utilize their released source code, and make proper modifications to adapt in our scenario for example, skeleton structure. These methods requires initial pose as well as ground truth motion length during inference. We also train these models with curriculum strategy as described in Section 3.3, by gradually increasing the complexity of training examples.

MoCoGAN [9] and Dance2Music [5] are two non-deterministic methods. Due to the specific discriminator design, they could only generate motions with fixed length. We adopt the implementation of MoCoGAN in [3], and the official implementation of Dance2Music with minor modifications. Specifically, the categorical conditions in MoCoGAN and audio signals in Dance2Music are replaced with text features.

D. Data Annotation and User Study

Fig. 3 presents the interface of annotating our HumanML3D dataset on Amazon Mechanical Turk. Given an animation, users are encouraged to convey the information of *action type*, *direction*, *body parts*, *velocity*, *trajectory*, *relative position* and *style* in text descriptions. If a motion is too complicated to be described, we also provide a channel for users to describe a sub-interval of presented animation, and give the start- and end- time point. Some exemplar good and bad descriptions help workers form a clearer picture. We ask users to avoid **over-general** descriptions (e.g. a man walks) and **over-specific** descriptions (e.g. absolute distance, angles, positions). Unqualified descriptions are manually rejected.

Fig. 4 shows the interface of the conducted user study in

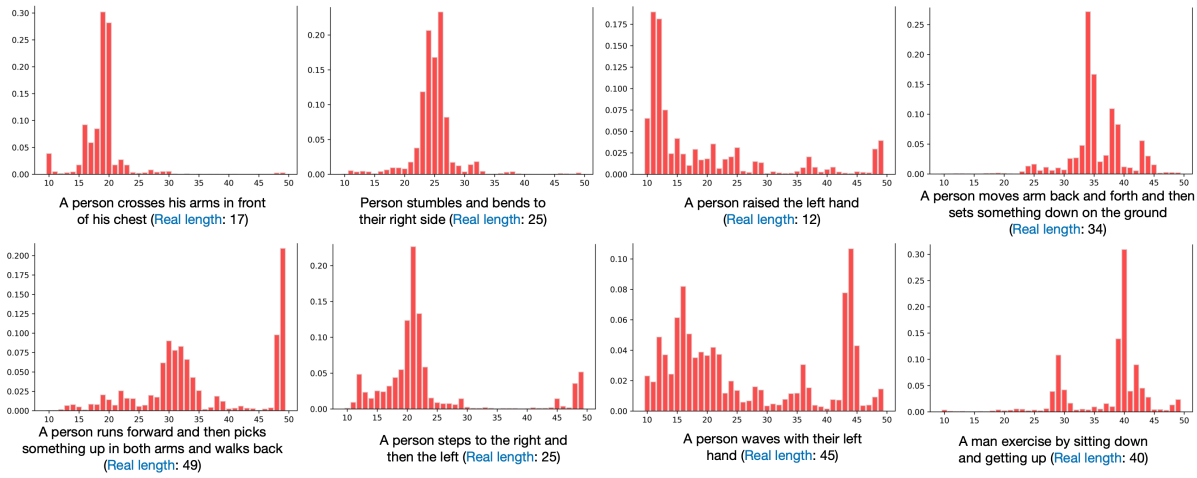


Figure 6. **Exemplar results of text2length.** For each subplot, given one text description (bottom), the estimated probability density of snippet code length is visualized in histogram. The corresponding real lengths are highlighted in blue. Length of motion is 4 times of the snippet code length.

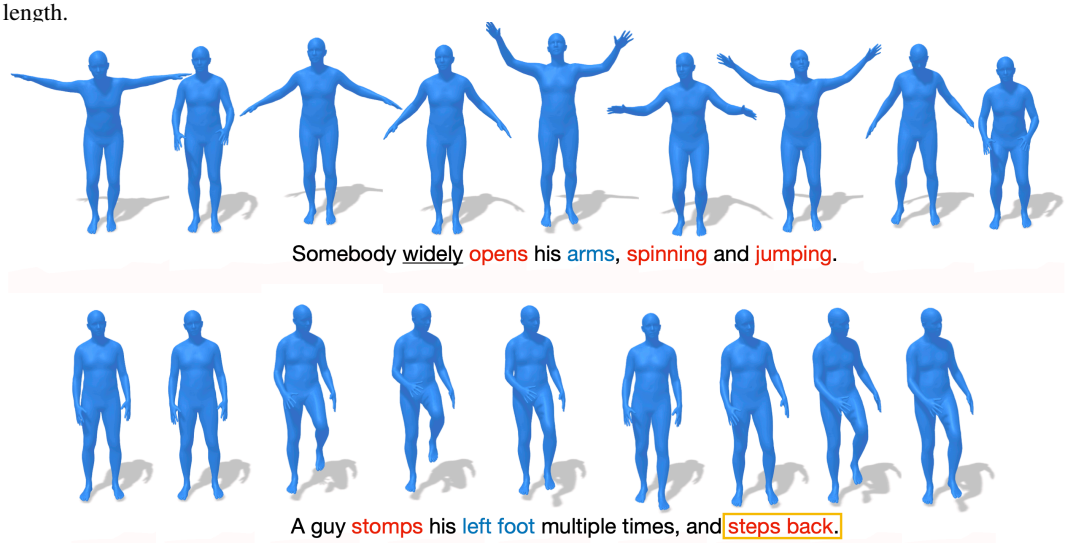


Figure 7. **Generated results from out-of-the-dataset descriptions.** Key frames are displayed. Yellow bound box indicates the parts of description that generated motions fail to present. Refer to supplementary video for more dynamic animations.

our experiments on Amazon Mechanical Turk. For each description, motions are generated from different methods and randomly reordered. Users are asked to rank their preference over these 7 animations based on judgement on *motion naturality*, *multimodal matching degree* and *motion consistency*. Only users with **master** recognition are included.

E. HumanML3D Dataset Results

Fig. 5 presents two scripted motions in our HumanML3D dataset. Each motion is described by 3 distinct worker on Amazon Mechanical Turker, thus resulting in diversified descriptions.

F. Qualitative Results of Text2Length

In Figure 6, we gives some examples of estimated snippet code length distribution from our text2length provided with text descriptions. Note the corresponding motion lengths are 4 times of the presented number. During training, motions with less than 40 frames (2s) are discarded. Therefore, here we produce the distribution over discrete values from 10 to 50. As shown, our method yields probability densities in which the values with high confidence are reasonably close to the corresponding ground truth value. In addition, cyclic motions (e.g., *a person waves with their left hand*) have relatively flatter probability density compared to non-cyclic motions (e.g., *Person stumbles and bends to their right side*). During inference, the target sequence length will be randomly sampled from the estimated prob-

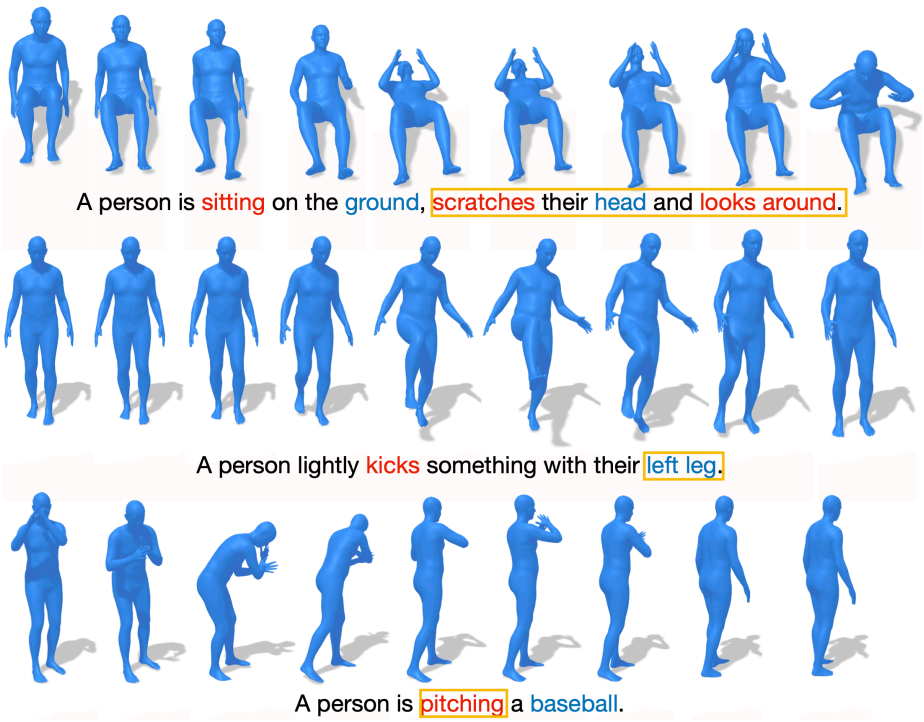


Figure 8. **Examples of failure cases.** Key frames are displayed. Yellow bound box indicates the parts of description that generated motions fail to present.

ability density, which further increases variety of generated results. Refer to supplementary video for generated results with different duration.

G. Qualitative Results of Text2Motion

Figure 1 demonstrates a gallery of synthetic results from our approach. Our method is able to handle with complex descriptions (e.g. *a person picks something up with both hands, moves it to the side, and then places it back down.*) as well as complicated actions such as *push up*. Please refer to our supplementary video for more results.

H. Neural Network Architecture

Table 1 illustrates the architecture we used on dataset HumanML3D. For KIT-ML dataset, the dimension of input vector may vary according to the dimension of pose vector.

I. Out-of-the-Dataset Generation, Failure Cases and Limitations

Figure 7 presents two results generated from out-of-the-dataset descriptions. In other words, the descriptions are collected independently to our dataset. Our method are able to demonstrate the overall content in the text descriptions. Nonetheless, the model may fail in descriptions involving rare actions (e.g., *'stomp'*). In the second pose sequence, *'step back'* is unfaithfully missed after *'stomping foot'*.

When further looking into multimodal alignment, our method sometime fail in finer grained descriptions. We showcase some failure results from our method in Figure 8. Actions such as "scratch" and "pitching baseball" are too sophisticated and beyond the capability of our method. We also find our method less sensitive to fine-grained descriptions regarding to body parts, for example, *left/right leg*. There are still space for exploring broader scenarios such as, text descriptions with overlength, environment interactions.

J. Table of Notations

Table 2 lists all the notations being used in our main content, as well as their practical values throughout experiments.

K. Code and Dataset

We upload the raw code as well as the command to reproduce our method and baselines (*readme.txt*) as another supplementary file. We will release the pre-trained model, a neater repository of code and data upon paper acceptance.

Components	Architecture
Convolutional Motion Encoder (E)	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)
Convolutional Motion Decoder (D)	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)
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)
Prior Network (F_ψ)	(z2init): Linear(in_features=1024, out_features=1024, bias=True) (embedding): 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)) (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)
Posterior Network (F_ϕ)	(z2init): Linear(in_features=1024, out_features=1024, bias=True) (embedding): Sequential((0): Linear(in_features=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)
Generator (F_θ)	(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))
Attention layer (F_{att})	(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)

Table 1. Architecture of our networks on dataset HumanML3D.

Symbol	Size	Practical Value	Meaning
$\mathbf{c}$	-	-	A notation of overall conditions.
$\mathbf{c}^f$	$\mathbb{R}^4$	-	Ground contact labels of heel and toe joints.
C_s	$\mathbb{R}^{512\times T}$	-	A snippet code sequence with T snippet codes.
$\mathbf{c}_s$	$\mathbb{R}^{512}$	-	A snippet code vector.
$\hat{\mathbf{c}}_s$	$\mathbb{R}^{512}$	-	A reconstructed snippet code
d	$\mathbb{R}^1$	1024	Dimension of input embedding in positional encoding.
d_{att}	$\mathbb{R}^1$	512	Number of channels in attentive vector $\mathbf{w_{att}}$.
d_h	$\mathbb{R}^1$	1024	Number of channels in generator hidden unit $\mathbf{h}_\theta$.
d_w	$\mathbb{R}^1$	1024	Number of channels in word feature $\mathbf{w}$.
$\mathbf{h}_\theta$	$\mathbb{R}^{1024}$	-	Hidden unit of generator F_θ.
j	$\mathbb{R}^1$	22(H), 21(K)	Number of joints in pose
$\mathbf{j}^p$	$\mathbb{R}^{3j}$	-	Local joints positions.
$\mathbf{j}^v$	$\mathbb{R}^{3j}$	-	Local joints velocity.
$\mathbf{j}^r$	$\mathbb{R}^{6j}$	-	Local joints rotations.
M	$\mathbb{R}^1$	-	Number of words in a text description.
$\mathcal{N}(\mu_\phi(t), \sigma_\phi(t))$	-	-	Posterior Gaussian distribution at time t generated by F_ϕ.
$\mathcal{N}(\mu_\psi(t), \sigma_\psi(t))$	-	-	Prior Gaussian distribution at time t generated by F_ψ.
$\mathbf{p}$	$\mathbb{R}^{263}$(H) $\mathbb{R}^{251}$(K)	-	A pose vector.
$\hat{\mathbf{p}}$	$\mathbb{R}^{263}$(H) $\mathbb{R}^{251}$(K)	-	A reconstructed pose vector.
P	$\mathbb{R}^{263\times T'}$(H) $\mathbb{R}^{251\times T'}$(K)	-	Pose sequence with T' poses.
$\hat{P}$	$\mathbb{R}^{263\times T'}$(H) $\mathbb{R}^{251\times T'}$(K)	-	Reconstructed pose sequence with T' poses.
p_{tf}	$\mathbb{R}^1$	0.4	Teacher forcing ratio.
$\dot{r}^a$	$\mathbb{R}^1$	-	Root angular velocity along Y-axis.
$\dot{r}^x$, $\dot{r}^z$	$\mathbb{R}^1$	-	Root linear velocity on XZ-plane
r^y	$\mathbb{R}^1$	-	Root height
$\mathbf{s}$	$\mathbb{R}^{512}$	-	Sentence feature vector extracted by text encoder.
T'	$\mathbb{R}^1$	-	Number of poses in a pose sequence.
T	$\mathbb{R}^1$	-	Number of snippet code in a snippet code sequence.
T_{cur}	$\mathbb{R}^1$	-	Present target sequence generation length in curriculum learning.
T_{max}	$\mathbb{R}^1$	50	Maximum length of snippet code sequences.
$\mathbf{w}_{1:M}$	$\mathbb{R}^{1024}$	-	Word features extracted by text encoder from a sentence of M words.
$\mathbf{w}_{att}$	$\mathbb{R}^{d_{att}}$	-	Local word attention vector.
$\mathbf{W}^K$	$\mathbb{R}^{d_w\times d_{att}}$	-	Trainable weight in local word attention.
$\mathbf{W}^Q$	$\mathbb{R}^{d_h\times d_{att}}$	-	Trainable weight in local word attention.
$\mathbf{W}^V$	$\mathbb{R}^{d_w\times d_{att}}$	-	Trainable weight in local word attention.
X	list of M elements	-	A text description of M words.
x	-	-	A single word.
$\mathbf{z}$	$\mathbb{R}^{128}$	-	Noise vector sampled from posterior/prior distribution.
λ_{spr}	$\mathbb{R}^1$	0.001	Weight of sparsity constraint in autoencoder during training.
λ_{smt}	$\mathbb{R}^1$	0.001	Weight of smoothness constraint in autoencoder during training.
λ_{mot}	$\mathbb{R}^1$	1	Weight of motion Reconstruction constraint during training.
λ_{KL}	$\mathbb{R}^1$	0.01(H), 0.05(K)	Weight of KL Divergence constraint during training.
D	-	-	Decoder of motion autoencoder.
E	-	-	Encoder of motion autoencoder.
F_θ	-	-	Generator of VAE models.
F_ϕ	-	-	Posterior network of VAE models.
F_ψ	-	-	Prior network of VAE models.
F_{att}	-	-	Local word attention function.
PE	-	-	Positional encoding function.

Table 2. Table of Notations. (H) denotes the value in HumanML3D dataset, (K) denotes the value in KIT ML dataset.

References

[1] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *International Conference on 3D Vision (3DV)*, pages 719–728. IEEE, 2019. 4

[2] Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 1–10. IEEE, 2021. 4

[3] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. 4

[4] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742, 2006. 3

[5] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019. 4

[6] Angela S Lin, Lemeng Wu, Rodolfo Corona, Kevin Tai, Qixing Huang, and Raymond J Mooney. Generating animated videos of human activities from natural language descriptions. *Learning*, 2018:1, 2018. 4

[7] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 2

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations*, 2014. 2

[9] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 4