

StickMotion: Generating 3D Human Motions by Drawing a Stickman

Supplementary Materials

Anonymous CVPR submission

Paper ID 6931

1. Method Details

1.1. Stickman Generation Algorithm

We propose the Stickman Generation Algorithm (SGA) to overcome the limited drawing style of annotators and reduce manual labeling costs. The generation process in detail is as follows:

- 1) Pose Normalization: Firstly, all 3D poses are standardized to a uniform scale across different datasets by utilizing the mean of arm length and leg length.
- 2) Limb Adjustment: Stickmen lack strokes for shoulders or pelvis. However, users would typically consider the length of shoulders and pelvis when drawing a stickman. Thus, the joint at the root of each limb is randomly sampled between this root joint itself and the central point of either the shoulder or pelvis.
- 3) Limb Stroke Generation: Each limb consists of three points. To closely resemble human strokes, we first interpolate the polyline formed by these points, then add Gaussian noise to it before applying a Gaussian filter for smoothing. The combination of noise addition and smoothing process is then used again to get closer to the real hand-drawn strokes. The stroke generation process for spine follows similar steps.
- 4) Head Stroke Generation: A standard circular coordinates array is initially generated for head lines. This array undergoes rotation and scaling in one direction followed by random stretching at both ends of the circle. Finally, noise addition and smoothing processes similar to limbs are applied.
- 5) Body Assembly: Now we get six strokes for one head, one spine, and four limbs. Potential pen placement errors may cause global position deviations in these body parts. Therefore, the root of arm and leg strokes are placed with jitter at start and end positions of the spine stroke. The head stroke is also positioned at the start position of spine stroke based on neck direction. Moreover, the scale of each part is also adjusted randomly get the results closer to human habits.

More visualization can be found in Fig. 1.

1.2. Self-Attention for Multi-Condition

Compared with our proposed Condition Fusion, conventional approaches [1, 5] with self-attention modules introduce unnecessary computation when calculating the attention of the masked token. Specifically, we assume that both conditions occur with a probability of P , and the encoding shapes of input X , condition T , and condition S are $[B, L^X, E^I]$, $[B, L^T, E^I]$, and $[B, L^S, E^I]$, respectively. Here, L and E^I are the tokens' number and length in the encoding. After the projection before the self-attention operation, these encodings are projected to $[B, H, L^X, E^P]$, $[B, H, L^T, E^P]$, and $[B, H, L^S, E^P]$, where H is the head number of this self-attention module. The FLOPs of this projection is $B \cdot (L^X + L^T + L^S) \cdot E^I \cdot H \cdot E^P$. Conventional approaches generate condition masks to remove the attention of networks on part condition inputs along the batch dimension. Then, they take the projected encoding of X and (X, T, S) as the query and key & value of the self-attention module. The FLOPs of the self-attention operation for conventional approaches can be denoted as $B \cdot H \cdot L^X \cdot (L^X + L^T + L^S) \cdot E^P$. Thus the final FLOPs for the projection and self-attention is $B \cdot H \cdot E^P \cdot (E^I + L^X) \cdot 2 \cdot (L^X + L^T + L^S)$. Similarly, the FLOPs of our proposed Condition Fusion can be denoted as $P \cdot B \cdot (L^X + L^T + L^S) \cdot E^I \cdot H \cdot E^P + P \cdot B \cdot H \cdot L^X \cdot (2 \cdot L^X + L^T + L^S) \cdot E^P = P \cdot B \cdot H \cdot E^P \cdot (E^I + L^X) \cdot 2 \cdot (3/2 \cdot L^X + L^T + L^S)$ according to Fig.2 of the main paper. Finally, the FLOPs ratio of Condition Fusion and conventional approaches is approximately P , which is 0.5 during the inference.

2. Experiment Details

2.1. Dataset Details

We conduct experiments on two prominent datasets of human motion generation, namely the KIT-ML dataset [4] and the HumanML3D dataset [2]. The KIT dataset comprises 3,911 motions with a total duration of 11.23 hours and includes 6,278 natural language descriptions composed by 5,371 distinct words, with an average sentence length

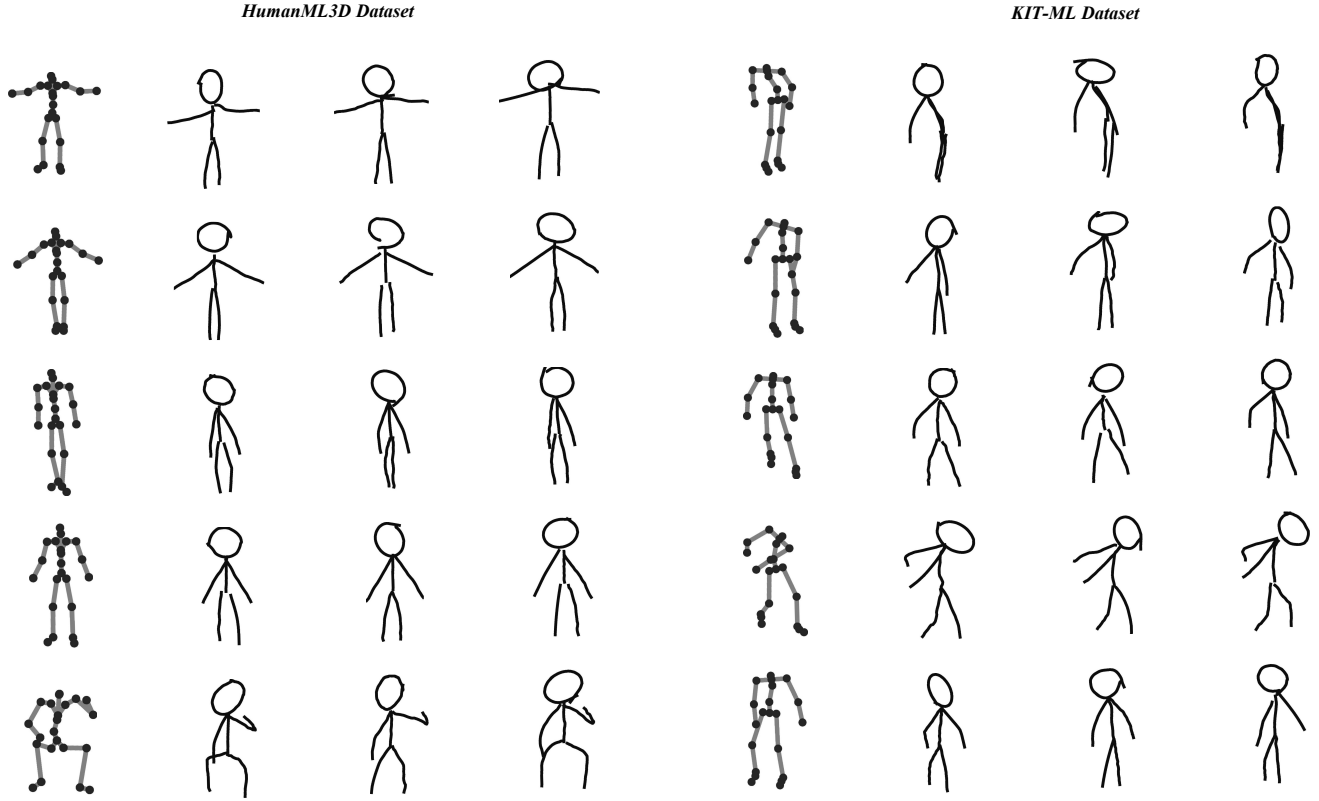


Figure 1. Stickman Visualization.

of 8.43 words. The HumanML3D dataset encompasses a larger scale, comprising 14,616 motions with a total duration of 28.59 hours. It includes 44,970 descriptions composed by 5,371 distinct words, with an average sentence length of 12 words. All experiments without additional instructions were performed on the KIT-ML dataset, with the setting of $p(\hat{w} = w) = 20\%$ and $(w_1 = 1, w_2 = 0, w_3 = 0, w_4 = 0)$ to ensure a balanced generation of motions between stickman and text conditions.

2.2. Stickman for Evaluation

The stickmen used for evaluation are also generated from the test set. While there may be concerns about potential leakage of the test set to the model, it is crucial to note that only the semantic information of the generated motions is considered during evaluation, rather than the spatial information of human poses. Moreover, including additional input stickmen may hinder the generation process from aligning more closely with its corresponding textual description, resulting in less impressive evaluation outcomes as shown in Tab. 4 of the main paper.

2.3. Analysis on Stickman Index

We randomly sample human poses for stickman generation at the start, middle, and end stages within the range

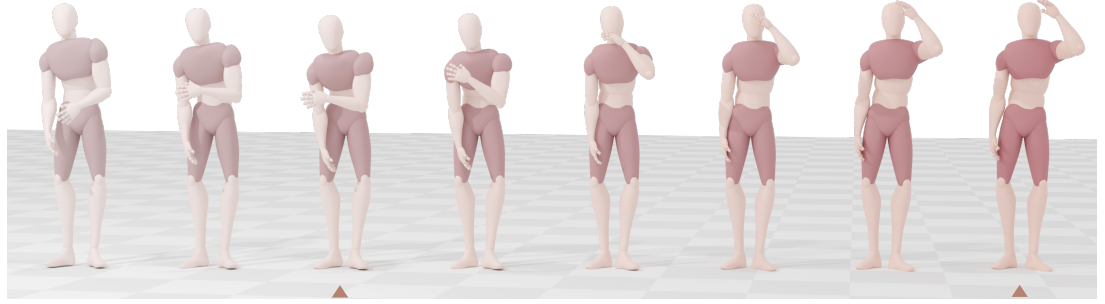
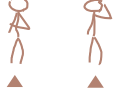
of $[0, 2 \cdot r]$, $[0.5 - r, 0.5 + r]$, and $[0.5 - 2 \cdot r, 1]$ in the motion sequences. Here, r represents the manually setting range for sampling. It should be noted that the predicted index may fall outside the sampled range, resulting in an error percentage known as Out-of-Range Error (ORE). Additionally, we assign a weight W_{index} to \mathcal{L}_{motion} (Equ.5 in the main paper) for adjusting StickMotion’s attention on the input stickmen. As shown in Tab. 1, the 1st line shows that the index of stickman is assigned by user (randomly appointed in the evaluation), indicating that this arbitrary operation will damage the semantic information of the generated result. Then, the 2nd, 3rd lines show that smaller r limits the network’s ability to adjust the stickman’s position. However, r larger than $1/8$ may make excessive adjustment to the stickmen, which leads to incompatibility with the user’s intention and the semantic information of the generations.

2.4. Condition Mixture in Inference

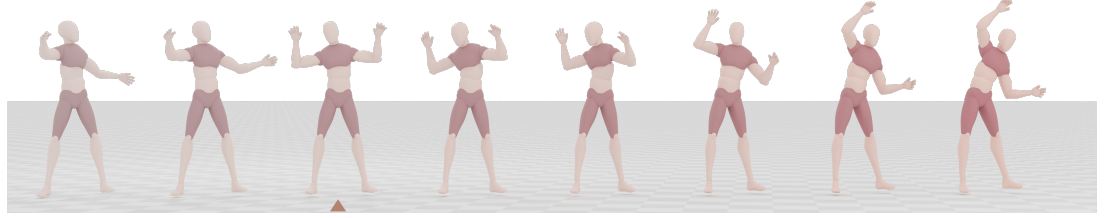
Guidance Strength. The guidance strength w (Sec.3.2 of the main paper) of the classifier-free diffusion [3] is set empirically like the learning rate of the training process. And we set w as 2 and 4 for the HumanML3D dataset and the KIT-ML dataset, respectively. The ablation experiments of w are shown in the Tab. 2.

Beginning Stage. To make it easier to understand, let us

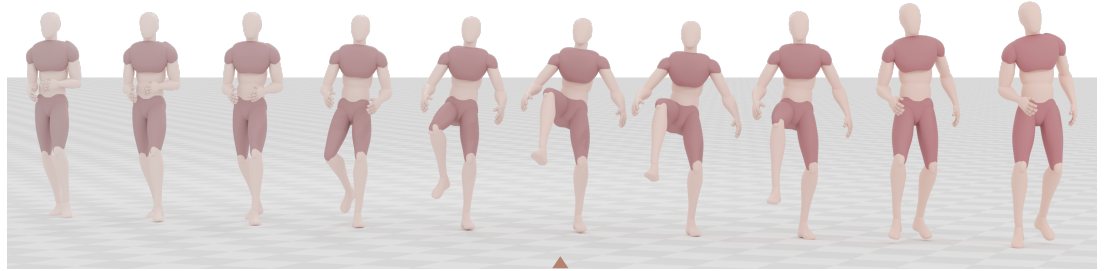
A man held his right arm with his left hand and then scratched his head.



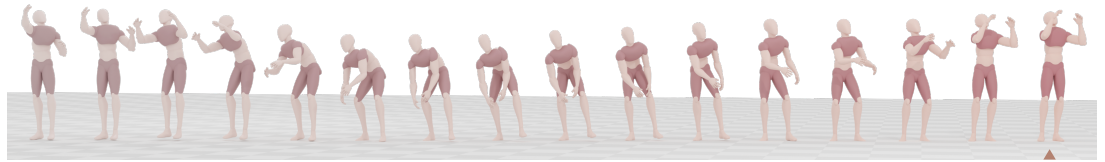
A man raises his hands and bends over.



A person lifted his leg forward and took a step.



A man catches a basketball and throws it.



A man pumped his fist forward.

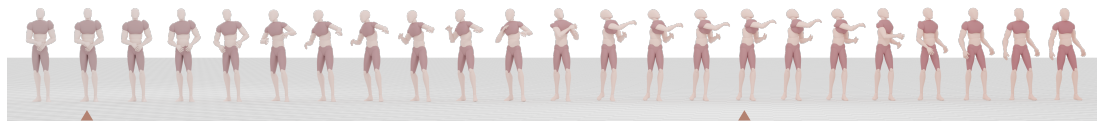
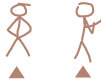


Figure 2. Visualization of stickman motion generation results. These pictures from top to bottom are generated with progressively smaller frame intervals to fully demonstrate stickmotion’s strengths and weaknesses.

Table 1. Ablation study on Stickman Index.

r	W_{index}	StiSim \uparrow	ORE \downarrow	FID \downarrow
-	1	37.3%	-	0.256
1/12	1	41.4%	8.33%	0.185
1/8	1	42.6%	6.32%	0.141
1/6	1	43.2%	5.87%	0.169
1/8	0.5	36.1%	13.64%	0.145
1/8	2	54.7%	7.54%	0.284

with the final stage having a smaller learning rate compared to the beginning stage. The ablation experiments of the final stage have been presented in the main paper, as the final stage makes more important and detailed contribution to the final results. For the beginning stage, there could be a conflict between $\epsilon_{\theta}(stick)$ and $\epsilon_{\theta}(text)$ that would slow down the reverse process. Therefore, $\epsilon_{\theta}(stick, text)$ is always present to reconcile the conflict, and $\epsilon_{\theta}()$ is used for maintaining the constant distribution of the final output. Moreover, we set different strategies for the ablation experiments as shown in Tab 2. Here, $p(\hat{w} = w) + p(\hat{w} = 0) = 1$ as shown in Sec.3.2 in the main paper. We set $p(\hat{w} = w) = 20\%$ following Sec.4.2 in the main paper for achieving a

126
127
128
129
130
131
132
133
134
135
136
137
138

124
125

consider the beginning stage and the final stage in the condition mixture are like the same stages of a training process,

Table 2. Ablation study on the beginning stage in the condition mixture.

w	w_1	w_2	w_3	w_4	FID↓
4	w	\hat{w}	$w - \hat{w}$	$1 - 2 \cdot w$	0.141
4	0	\hat{w}	$w - \hat{w}$	$1 - w$	0.266
4	w	0	0	$1 - w$	0.742
3	w	\hat{w}	\hat{w}	$1 - 2 \cdot w$	0.235
5	0	\hat{w}	\hat{w}	$1 - w$	0.373

balance performance under the two conditions. According to our experiments, there exists an adversarial relationship between the stickman condition (w_2) and the text condition (w_3), which necessitates the combination of both conditions (w_1) for mediation and performance enhancement.

3. Limitations and Future Work

We selected several representative examples, as illustrated in Fig. 2, to demonstrate the excellent performance of StickMotion in generating motion sequences under text and stickman conditions. Nevertheless, StickMotion does exhibit certain limitations: 1) While preserving the semantic information of the generated motions, there is no guarantee that the intended poses for stickmen will be accurately generated. This issue is commonly encountered in condition-based diffusion models, and we aim to mitigate this error by enhancing both the forward and reverse design strategies. 2) Occasionally, poses generated around the stickman pose tend to resemble it due to the influence of supervision $\mathcal{L}_{\text{index}}$ described in Equ. 5 of the main paper. This occurrence was more frequent before we apply the Softmax function to the predicted index scores; thus, we plan on devising more sophisticated supervision strategies to address this concern. In addition to these aforementioned problems requiring further investigation, the generation and application of stickman can also be extended beyond its current scope into other domains, such as interactive motion generation, 3D object generation, and 3D mesh editing, *etc.* Our future works will focus on exploring these areas.

* Main Paper Revision

The Equ.1 in the main paper should be modified as Equ. 1 as follows,

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad (1)$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1 - \alpha_t)\mathbf{I}).$$

The following conclusion “ This formula is equivalent to $\mathbf{x} = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_t$, where $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ ” is correct.

References

- [1] Wenhui Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. *arXiv preprint arXiv:2209.14491*, 2022. 1
- [2] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, pages 5152–5161, 2022. 1
- [3] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2
- [4] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016. 1
- [5] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Re-modiffuse: Retrieval-augmented motion diffusion model. In *ICCV*, pages 364–373, 2023. 1