

# FineMotion: A Dataset and Benchmark with both Spatial and Temporal Annotation for Fine-grained Motion Generation and Editing

## Supplementary Material

### 6. License

The license for human motion sequences in this dataset follows the term specified at [HumanML3D](#) and [AMASS](#). The textual descriptions in our FineMotion dataset are under the CC BY 4.0 International license. For detailed license information, please refer to <https://creativecommons.org/licenses/by/4.0/legalcode>

### 7. Discussion on Selecting of optimal $T_s$

To determine the optimal snippet duration  $T_s$ , we propose two guiding principles to help researchers tailor this value to their own datasets. As shown in Fig. 4, we randomly sample 1,000 snippets with varying durations from all motion sequences in our dataset. Then, we calculate the cosine similarity between the PoseScript [2] semantic features of the start and end poses for each snippet. A higher cosine similarity indicates that the start and end poses are more similar, suggesting that the motion progresses slowly; conversely, a lower similarity indicates faster progression.

Our results show that the motions in our dataset generally progress slowly, prompting the selection of a larger interval to avoid redundancy. Here, we also display the statistical results of a rapidly changing motion (1,000 random samples of start and end points) in Fig. 4. The results indicate that the similarity of the pose semantic features first decreases and then increases as the temporal interval grows. From this, we derive the first principle for selecting the optimal value of  $T_s$ : **Choose the value of  $T_s$  that minimizes the similarity between the start and the end poses.** Meanwhile, PoseFix [3] suggests that larger time differences between two poses allow for a wide range of plausible in-between motions. Therefore, the second principle is that **the value of  $T_s$  should not exceed 0.5s**, which is the maximum time difference for pose pair selection specified by PoseFix [3]. Following these principles, we set  $T_s$  to 0.5s. Notably, any remaining segment of a motion sequence shorter than  $T_s$  is also treated as an individual snippet.

### 8. Data Format Examples

The data format example for all the detailed human body part snippet descriptions (BPMSDs) in a whole human motion sequence is shown below:

```
{
  "000314":      # name of motion sequence
  [
    "",          # 0.0s-0.5s' BPMSD
    "Bend your elbows and raise your hands up to your head.",
    "",          # 1.0s-1.5s' BPMSD
    "",          # 1.5s-2.0s' BPMSD
    "Turn your upper body to the right slightly.",
    "",          # 2.5s-3.0s' BPMSD
    "Straighten your elbows and lower your hands to your thighs.",
    "Straighten your elbows completely and move your hands back to your sides.",
  ],
}
```

The data format example for three different detailed human body part paragraphs (BPMPs) for the same human motion sequence is shown below:

```
{
  "000314":      # name of motion sequence
  [
    "Initially, the person bends his elbows and raises his hands to his head. Then, he slightly turns his upper body to the right. Afterward, he straightens his elbows and lowers his hands to his thighs. Finally, he straightens his elbows completely and moves his hands back to his sides.",
    "First, the person bends the elbows and raises his hands above his head. Then, he slightly rotates his upper body to the right. Subsequently, he straightens the elbows and lowers his hands to rest on his thighs. Finally, he fully extends his elbows and returns his hands to their positions at his sides.",
    "The person begins by bending the elbows and raising the hands toward the head. Subsequently, he slightly twists his upper body to the right. Afterward, he extends the elbows and lowers the hands toward the thighs, then fully straightening the elbows and moving the hands back to the sides."
  ],
}
```

## 9. More Dataset Examples

We display more examples of body part movement descriptions for motion snippet (*i.e.*, BPMSD) and for whole motion sequence (*i.e.*, BPMP) of our FineMotion dataset in Fig. 9 and 10.

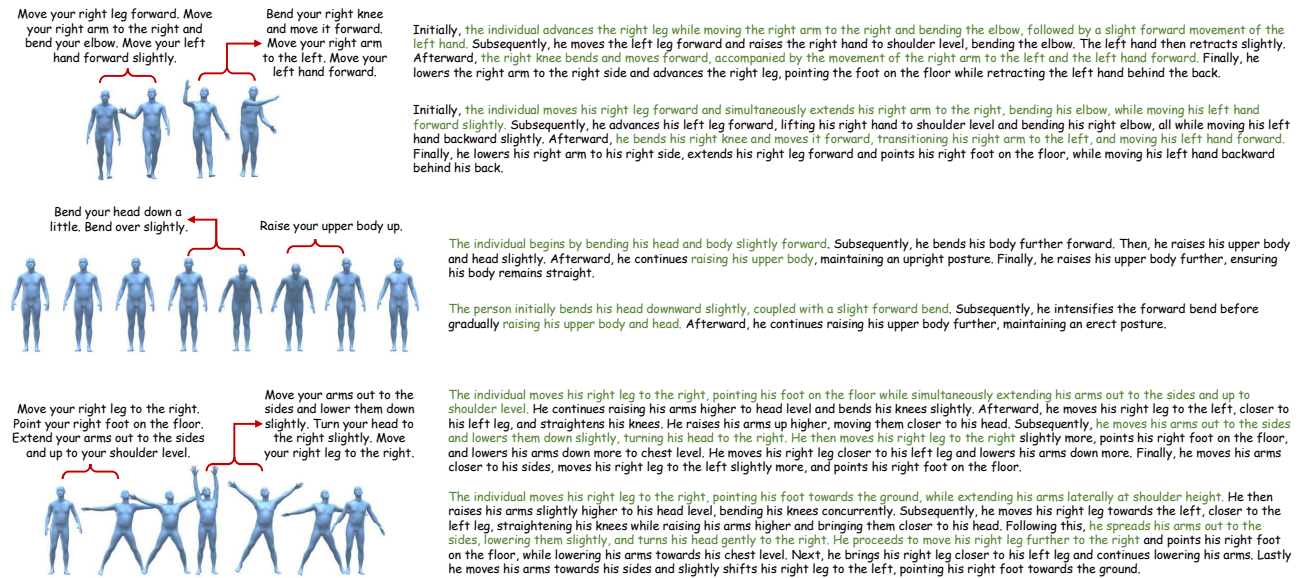


Figure 9. More examples of human-annotated body part movement snippet descriptions (*left*) and paragraphs (*right*). The colored text in paragraphs links to corresponding snippet descriptions.

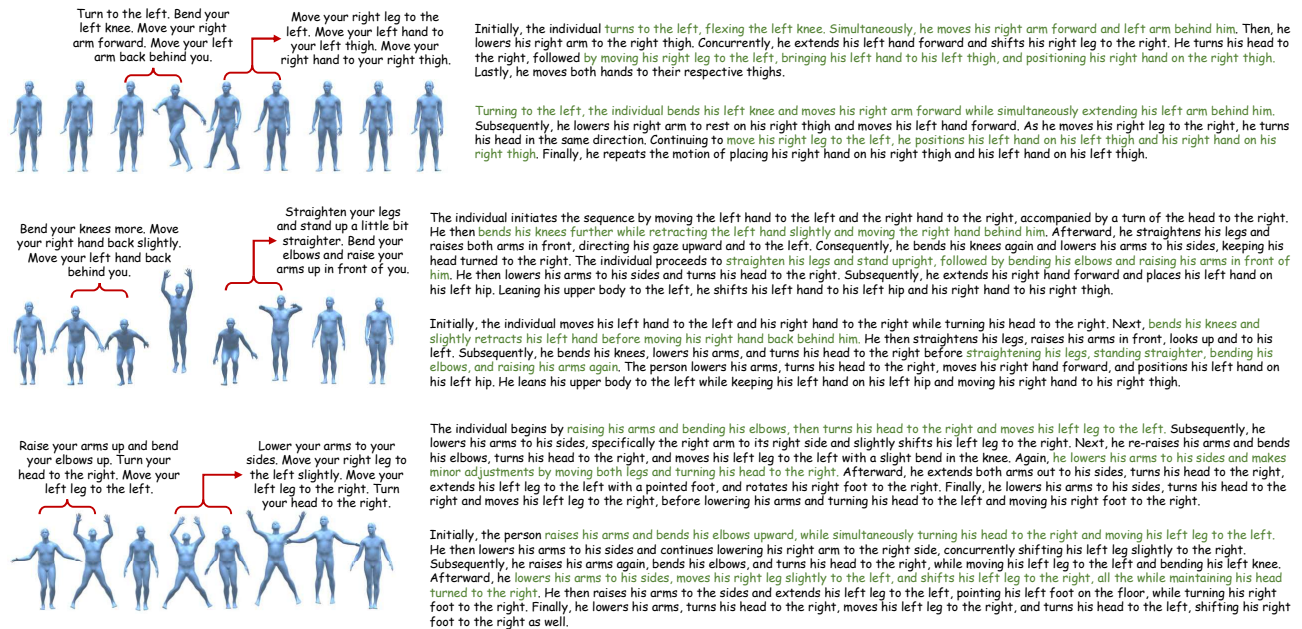


Figure 10. More examples of automatically generated body part movement snippet descriptions (*left*) and paragraphs (*right*). The colored text in paragraphs links to corresponding snippet descriptions.

## 10. Baseline Model Details

This section outlines the network architecture and the implementation of three variants of motion generation methods, including MDM [25], T2M-GPT [30], and MoMask [8] on our dataset, and denoted them as (T&DT)-MDM, (T&DT)2M-GPT, and (T&DT)-MoMask, respectively.

- **(T&DT)-MDM** builds from MDM [25]. It employs a classifier-free, diffusion-based approach for human motion generation using a transformer-based architecture. Unlike standard diffusion models, it directly predicts the sample at each diffusion step. Specifically, the transformer-encoder predicts the final clean motion based on a condition (*i.e.*, a CLIP-based textual embedding), a noising timestep, and random noise. To accommodate detailed textual descriptions, which often contain over ten times the number of tokens compared to coarse captions, we replace the CLIP [21] text encoder with the T5-Base [22] encoder, which uses relative attention for flexible input lengths. We then perform mean pooling along the sequence length dimension of the T5-Base encoder output to obtain a single text embedding for each text. Now, the model’s condition turns out to be the concatenated text embeddings of both the coarse caption and the detailed description.
- **(T&DT)2M-GPT** is derived from T2M-GPT [30] and comprises a Motion VQ-VAE and a GPT model. Motion VQ-VAE learns a mapping between raw motion sequences and discrete token sequences, while the GPT model generates motion tokens conditioned on text embeddings. Likewise, we modified the condition of the GPT model into the concatenated T5 text embeddings of the coarse caption and the detailed text.
- **(T&DT)-MoMask** is based on MoMask [8], featuring a Motion Residual VQ-VAE, a Masked Transformer, and a Residual Transformer. Concretely, the Residual VQ-VAE uses a hierarchical quantization scheme to discretize motions into multiple layers of motion tokens. The Masked Transformer predicts masked motion tokens from the text input, while the Residual Transformer progressively predicts next-layer tokens based on the results from the current layer. The textual embedding is modified similarly to the previous two networks.

All three baseline models are adapted to include our long, detailed body part movement descriptions for the motion sequences. Here, we hold (T&DT)2M-GPT as the example to elaborate on the differences from the original T2M-GPT model. The modifications applied to the other two baseline models, (T&DT)-MDM and (T&DT)-MoMask, follow a similar approach.

(T&DT)2M-GPT mainly contains two parts: Motion VQ-VAE for motion discretization and GPT for generating

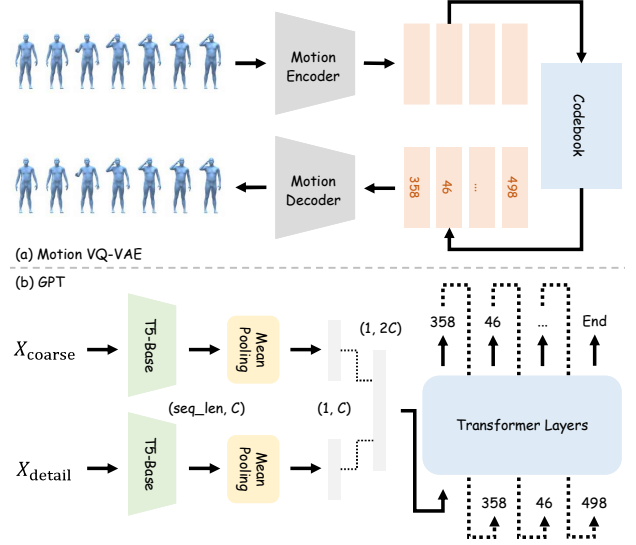


Figure 11. **Overview of the baseline network, (T&DT)2M-GPT.** It generates motions that strictly follow the fine-grained description  $X_{\text{detail}}$  and the coarse-grained caption  $X_{\text{coarse}}$ . It consists of a motion VQ-VAE for discretizing the motion into tokens and a GPT for generating motion tokens.

motion tokens from the coarse caption and detailed text.

**Motion VQ-VAE.** We follow [30] to represent motions in discrete tokens, and vice versa. Specifically, it contains an encoder  $E$ , a decoder  $D$ , and a learnable codebook  $B = \{b_k\}_{k=1}^K$ , where  $K$  is the size of the codebook. Given a  $T$ -frame motion sequence  $M = [m_1, m_2, \dots, m_T]$  with  $m_t \in \mathbb{R}^d$ , the encoder  $E$  maps it into a sequence of latent features  $Z = E(M)$  with  $Z = [z_1, z_2, \dots, z_{\lfloor T/l \rfloor}]$  and  $z_i \in \mathbb{R}^{d_c}$ , where  $l$  represents the temporal downsampling rate of the encoder  $E$ . Then, these latent features are transformed into a sequence of motion codes  $C = [c_1, c_2, \dots, c_{\lfloor T/l \rfloor}]$ , where  $c_i$  is the index of the most similar element to  $z_i$  in  $B$ . With a sequence of motion codes  $C$ , we first project  $C$  back to their corresponding codebook elements  $\tilde{Z} = [\tilde{z}_1, \tilde{z}_2, \dots, \tilde{z}_{\lfloor T/l \rfloor}]$  with  $\tilde{z}_i = b_{c_i}$ . Then, the decoder  $D$  reconstructs  $\tilde{Z}$  into a motion sequence  $\tilde{M} = D(\tilde{Z}) = [\tilde{m}_1, \tilde{m}_2, \dots, \tilde{m}_T]$ . The motion VQ-VAE is optimized by the standard optimization goal [27] that requires the decoded motion  $\tilde{M}$  to be as similar as the input motion  $M$ . The exponential moving average (EMA) and codebook reset (Code Reset) are employed to stabilize the training process. With a learned motion VQ-VAE, a motion sequence can be easily mapped into discrete motion tokens by the encoder  $E$  and the codebook  $B$ . On the other hand, the output of our (T&DT)2M-GPT model, *i.e.*, motion tokens, can be recovered into motion sequences by the decoder  $D$  and the codebook  $B$ .

**GPT for generating motion tokens.** First, we extract the text embeddings of the coarse caption  $t_{\text{coarse}}$  and the edited detailed motion script  $\hat{t}_{\text{detail}}$ . Since the number of tokens of our detailed human body part descriptions is usually more than ten times that of the coarse captions, we use the frozen encoder from T5-Base [22] to extract the textual embeddings, considering that its relative attention mechanism allows input with any sequence length. We then perform a mean pooling operation in the  $\text{seq\_len}$  dimension of the output from the T5-Base encoder to obtain a single text embedding for each text.

$$t_{\text{coarse}} = \text{Mean}(\text{T5Encoder}(X_{\text{coarse}})) \in \mathbb{R}^{768}, \quad (1)$$

$$\hat{t}_{\text{detail}} = \text{Mean}(\text{T5Encoder}(\hat{X}_{\text{detail}})) \in \mathbb{R}^{768}. \quad (2)$$

Next, the two text embeddings are utilized as the conditions of the GPT model to autoregressively generate motion tokens. The GPT model is composed of a stack of transformer layers. Besides, casual self-attention is applied to ensure the calculation of the current tokens does not consider the information of the future motion tokens. Since this fine-grained motion generation task can be considered as the next motion token prediction task, which is based on the given coarse textual embedding, the motion script textual embedding, and previous motion tokens, the GPT model is optimized by the cross-entropy loss between the predicted motion tokens and ground-truth ones.

$$L = - \sum_{i=1}^{\lfloor T/U \rfloor} \log(P(c_i | t_{\text{coarse}}, \hat{t}_{\text{detail}}, c_{<i}, \theta_{\text{GPT}})). \quad (3)$$

After sufficient training, the GPT model can generate appropriate motion tokens that can be further decoded into motions by the decoder in Motion VQ-VAE.

## 11. More Implementation Details

The architecture and training hyperparameters of our baseline models strictly follow those in the original paper [8, 25, 30]. Notably, since we replace the text encoder with that of T5, the dimension of output text embedding turns to 768 rather than that of the CLIP text encoder, 512. Therefore, the input of the fully connected layer that projects the CLIP text embedding to the input of the GPT also needs to be changed from 512 to 768. The code is based on PyTorch. The experiments were conducted on the A100-80G GPU, but only about 16G GPU memory was used. Due to replacing the text encoder with a larger model [22] and using it to process longer textual descriptions, the training time for (T&DT)2M-GPT increases to 154 hours, compared to the 78 hours reported in [30] for T2M-GPT. However, the training time can be reduced to the original 78 hours if all text embeddings are pre-extracted and stored before the training begins.

## 12. More Discussion on Motion Generation with Fine-grained Texts Only

We did not evaluate this setting because it will lead to **ambiguity** in motion generation. Fine-grained text captures detailed body part movements and timing, while coarse text supplements global motion semantics, both crucial for precise motion generation. For instance, motions with coarse text ‘*a person is standing still*’ and ‘*a person is sitting*’ share the same fine-grained text ( $\langle \text{Motionless} \rangle$ , *i.e.*, no body part movements). The model cannot distinguish such cases without coarse text, degrading motion generation performance. Given the issue above, we do not train our models using (fine-grained text, motion) pairs. Evaluating such a setting without proper training would lead to unfair or unreliable results.

## 13. More Discussion on Table 2

One may notice that when (T&DT)2M-GPT—*i.e.*, Rows (2)-(5) in Table 2—generates motions using only coarse descriptions (Test Set: T2M), it shows a slight performance drop, compared to our implementation of T2M-GPT trained solely on the T2M task, Row (1). The slight drop in T2M-GPT variants likely stems from their high sensitivity to the shared training budget, as multi-task training with (T&DT) halves the T2M updates compared to the baseline. Additional evaluations on MDM and MoMask variants show that including (T&DT)2M during training actually improves motion generation when only coarse text is available, as shown below.

Train Task	Test Task	MDM		T2M-GPT		MoMask	
		T2M	T2M	R-Top3 $\uparrow$	FID $\downarrow$	R-Top3 $\uparrow$	FID $\downarrow$
✓	-	✓		0.606 $\pm$ .008	3.137 $\pm$ .183	0.781 $\pm$ .003	0.123 $\pm$ .005
✓	(our BPMSD)	✓		0.746 $\pm$ .007	0.760 $\pm$ .064	0.781 $\pm$ .002	0.154 $\pm$ .007
✓	(our BPMP)	✓		0.759 $\pm$ .006	0.436 $\pm$ .043	0.781 $\pm$ .002	0.155 $\pm$ .006
						0.827 $\pm$ .002	0.120 $\pm$ .004
						0.818 $\pm$ .002	0.130 $\pm$ .005

Table 4. Generation performance of all our variants on the T2M test set, *i.e.*, motion generation conditioned on coarse descriptions only.

## 14. Ablation Study on Baseline Model Design

Here, we conduct an ablation study on different strategies for encoding coarse and detailed texts. Specifically, we denote the strategy of connecting the coarse text (T) and detailed text (DT) into a single text and feeding it into the text encoder as ‘TDT’. Meanwhile, ‘T&DT’ refers to encoding T and DT separately and then concatenating their resulting embeddings. Results below show that the ‘TDT’ strategy leads to poorer performance, likely because the model is overwhelmed by the dense information and struggles to capture the global motion semantics. These findings highlight that our baseline designs are carefully considered, rather than naïve implementations.



Method	R-Precision $\uparrow$			FID $\downarrow$	MM-Dist $\downarrow$	Diversity $\rightarrow$
	Top-1	Top-2	Top-3			
TDT-MoMask (BPMSD)	0.212 $\pm$ .002	0.341 $\pm$ .002	0.434 $\pm$ .002	8.328 $\pm$ .056	5.877 $\pm$ .009	8.899 $\pm$ .009
(T&DT)-MoMask (BPMSD)	0.519 $\pm$ .002	0.715 $\pm$ .002	0.811 $\pm$ .001	0.088 $\pm$ .003	2.946 $\pm$ .005	9.702 $\pm$ .075
TDT-MoMask (BPMP)	0.358 $\pm$ .003	0.528 $\pm$ .002	0.628 $\pm$ .002	0.285 $\pm$ .006	4.145 $\pm$ .008	9.626 $\pm$ .093
(T&DT)-MoMask (BPMP)	0.520 $\pm$ .003	0.717 $\pm$ .002	0.813 $\pm$ .002	0.055 $\pm$ .002	2.935 $\pm$ .009	9.679 $\pm$ .085

Table 5. Ablation study on different strategies for encoding coarse and detailed texts.

## 15. Metrics and Results for Temporal Alignment

Currently, there is no metric that directly evaluates the precision of temporal alignment between detailed texts and generated motion sequences. Given that our detailed texts are strictly aligned with ground-truth motions over time, we reframe this evaluation as measuring the alignment between short clips of generated motions and corresponding ground-truth clips. High similarity between these clips—even at fine temporal granularity—implies accurate alignment with the detailed texts.

To this end, we introduce  $FID_c$ , which computes the similarity between generated and ground-truth motions using overlapping 40-frame clips (the minimum evaluation length), with a stride of 10—matching the minimal temporal interval of our detailed texts. The table below reports  $FID_c$  scores across all clips. As shown, our variants (last two rows) achieve significantly lower  $FID_c$  scores, demonstrating that our generated motions are better temporally aligned with the detailed texts, compared to motions generated by models trained solely on coarse descriptions.

	MDM	T2M-GPT	MoMask
T2M	3.012 $\pm$ .206	1.423 $\pm$ .040	0.293 $\pm$ .011
(T&DT)2M (BPMSD)	1.382 $\pm$ .125	0.398 $\pm$ .011	0.165 $\pm$ .004
(T&DT)2M (BPMP)	0.426 $\pm$ .046	0.624 $\pm$ .015	0.108 $\pm$ .003

Table 6. Comparison of temporal alignment, measured by  $FID_c$ , between baseline text-to-motion models and our fine-grained variants.

## 16. Limitations and Future Work

Since we use temporally augmented data to train the text-to-motion models, editing motions along the temporal dimension becomes more straightforward and accurate compared to spatial editing. Consequently, future work will focus on developing effective methods for spatial human motion editing.

Additionally, obtaining the detailed body part textual descriptions still requires multiple steps. Thus, training an end-to-end model that can directly infer these descriptions from human motion sequences presents a promising research direction.

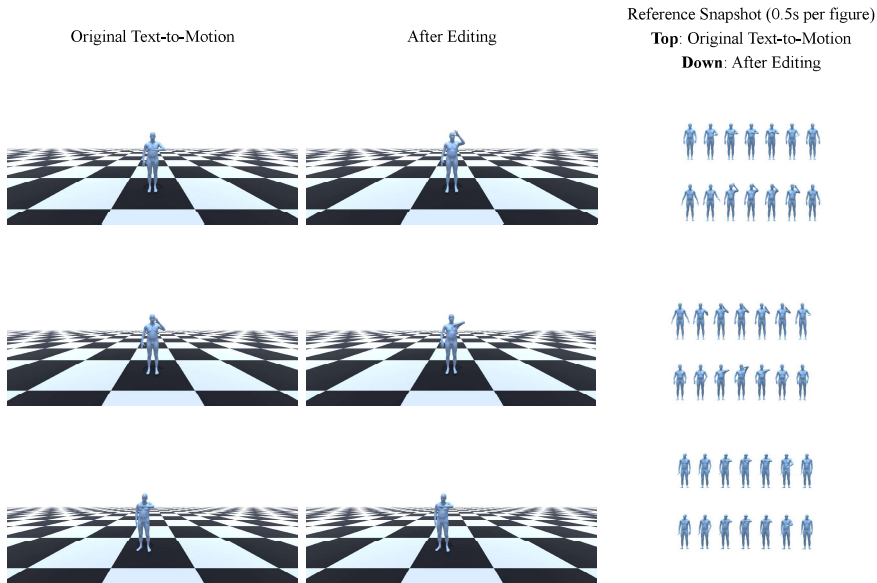
Moreover, the capabilities of large language models (LLMs) could be leveraged to unify text-to-motion and motion-to-text tasks through textual descriptions of varying granularity, potentially enhancing the effectiveness of both tasks.

## 17. User Study

**Case 1: Add** the body part movements **Spatially**.

original text: a person lifts their left wrist towards their face as if to look at a watch

editing requirement: Lift your left hand to the head.

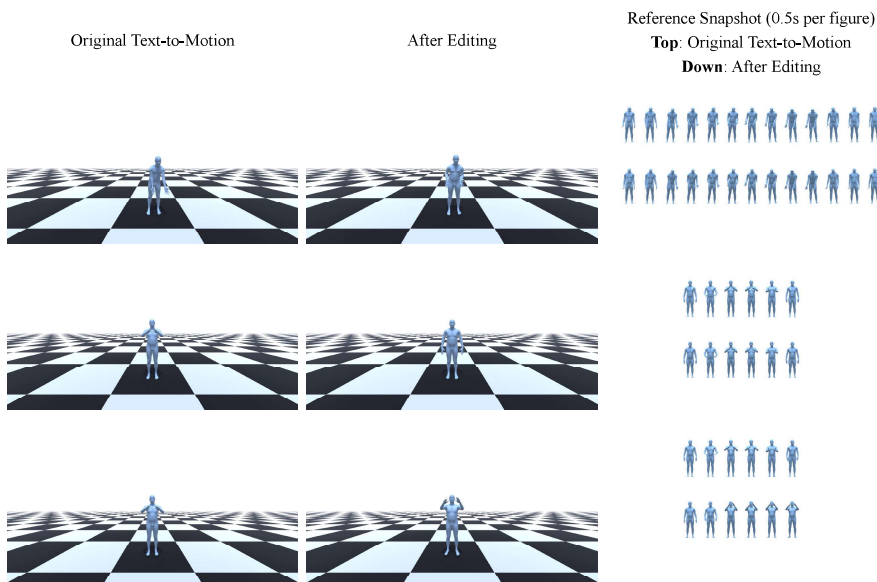


Answer: Row 1: Ours Row 2: T2M-GPT Row 3: FLAME

**Case 2: Delete** the body part movements **Spatially**.

original text: a man bends his arms to touch an object in front of him.

editing requirement: Do not put your hands down to your sides.

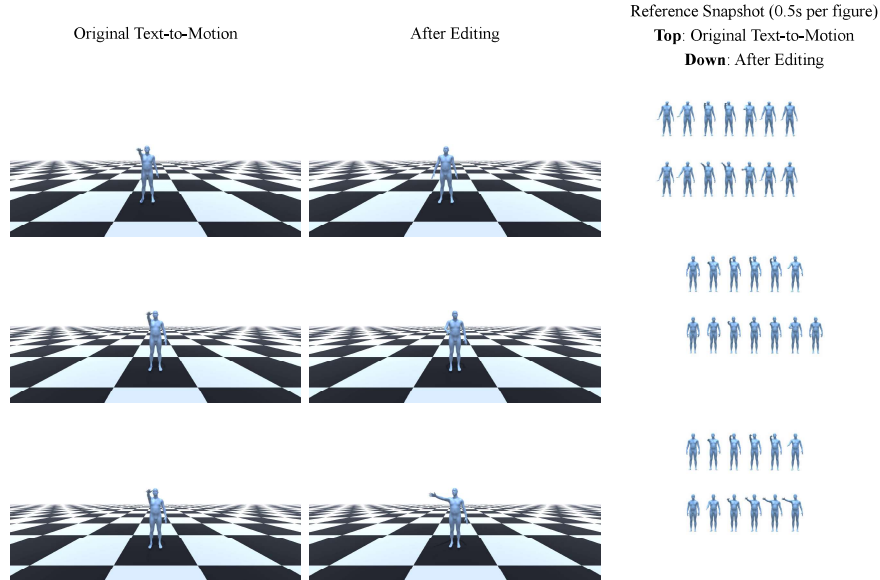


Answer: Row 1: FLAME Row 2: T2M-GPT Row 3: Ours

**Case 3: Modify the body part movements Spatially.**

original text: a person stretches right hand forward

editing requirement: Do not put down your right hand.

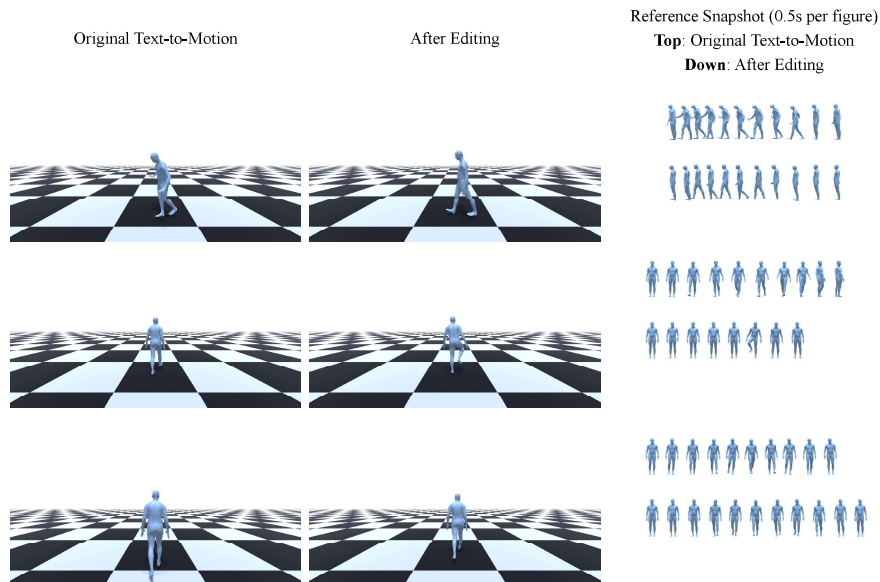


Answer: Row 1: FLAME Row 2: T2M-GPT Row 3: Ours

**Case 4: Extend at the start of the human motion (Temporally).**

original text: a person walks forward while making small adjustments left and right

editing requirement: Stand for one second before start walking.

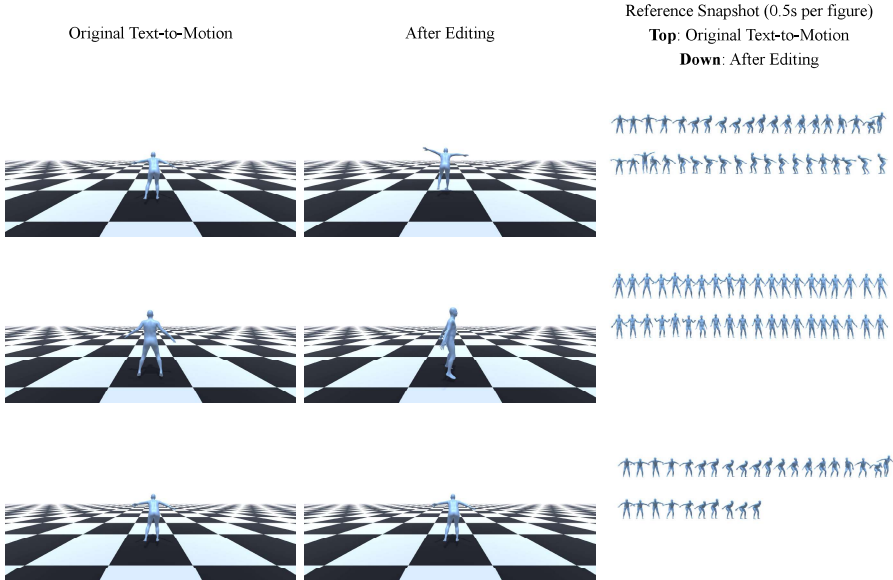


Answer: Row 1: FLAME Row 2: T2M-GPT Row 3: Ours

**Case 5: Delete at the end of the human motion (Temporally).**

original text: a person hops with both feet in a half circle while both arms are positioned backwards.

editing requirement: Delete the motion in the last 4 seconds.

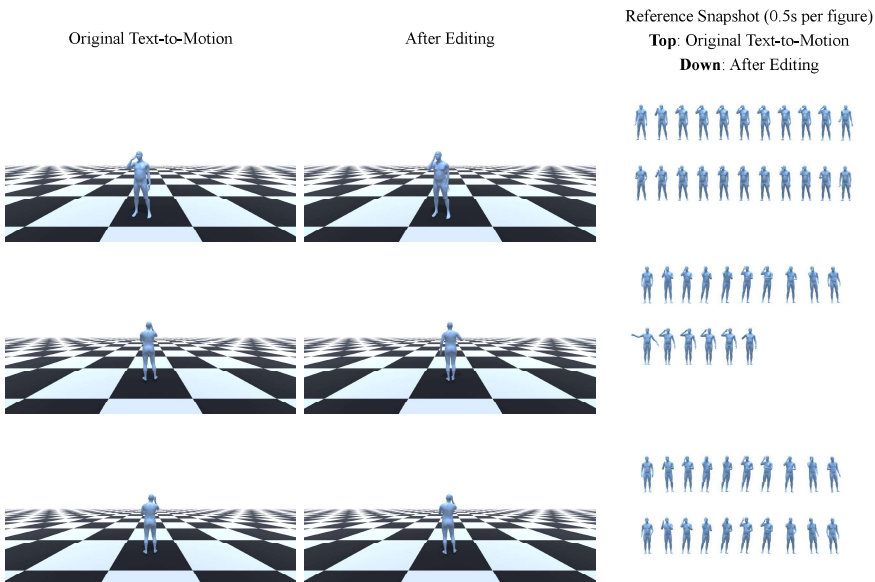


Answer: Row 1: T2M-GPT Row 2: FLAME Row 3: Ours

**Case 6: Delete in the middle of the human motion (Temporally).**

original text: a person leaned something near to face with right hand

editing requirement: Delete the motion from 1.0-2.5s.

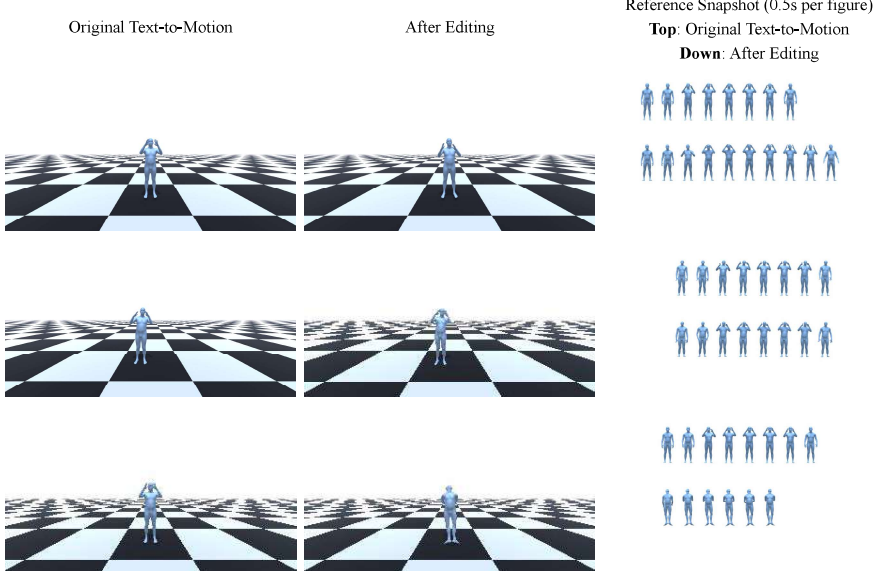


Answer: Row 1: FLAME Row 2: Ours Row 3: T2M-GPT



**Case 7: Insert** in the middle of the human motion (**Temporally**).

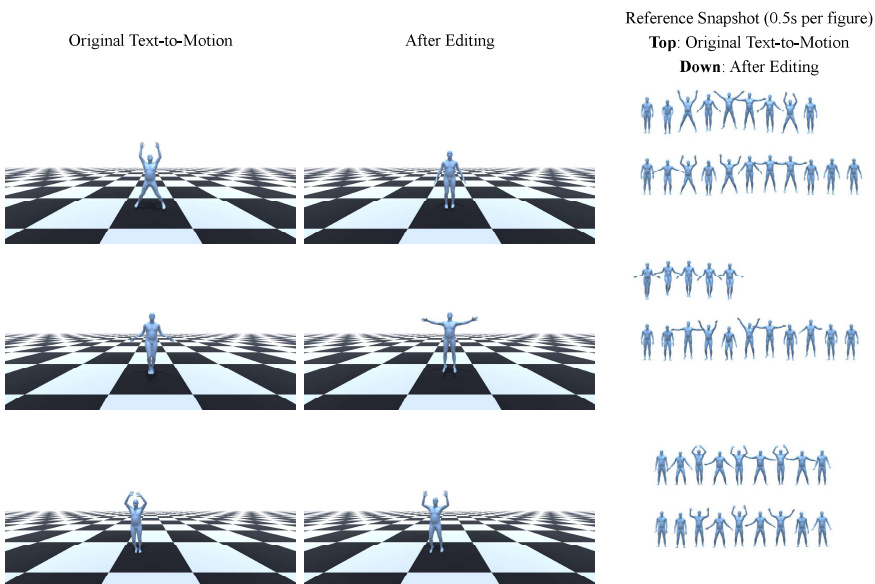
original text: a person lifts both hands toward face and then lowers them to their sides.  
 editing requirement: After lifting your hands, stay for one more second, and then lower them down.



Answer: Row 1: Ours Row 2: FLAME Row 3: T2M-GPT

**Case 8: Extend** at the **end** of the human motion (**Temporally**).

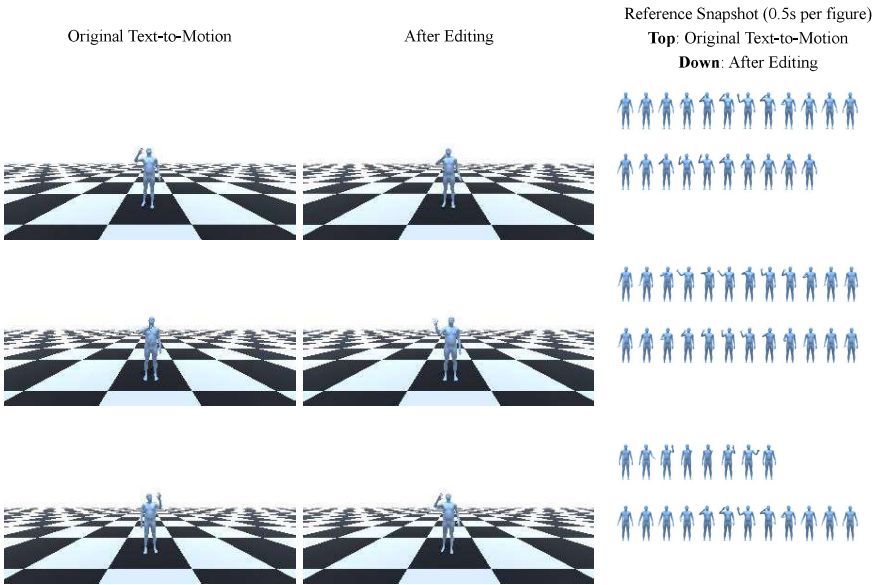
original text: a person does jumping jacks.  
 editing requirement: Stand for one more second after jumping.



Answer: Row 1: Ours Row 2: T2M-GPT Row 3: FLAME

**Case 9: Delete** at the **start** of the human motion (**Temporally**).

original text: **a man waves his right hand.**  
editing requirement: **Delete the standing still segment at the start of the motion.**



Answer:    Row 1: Ours    Row 2: FLAME    Row 3: T2M-GPT