Supplementary Material

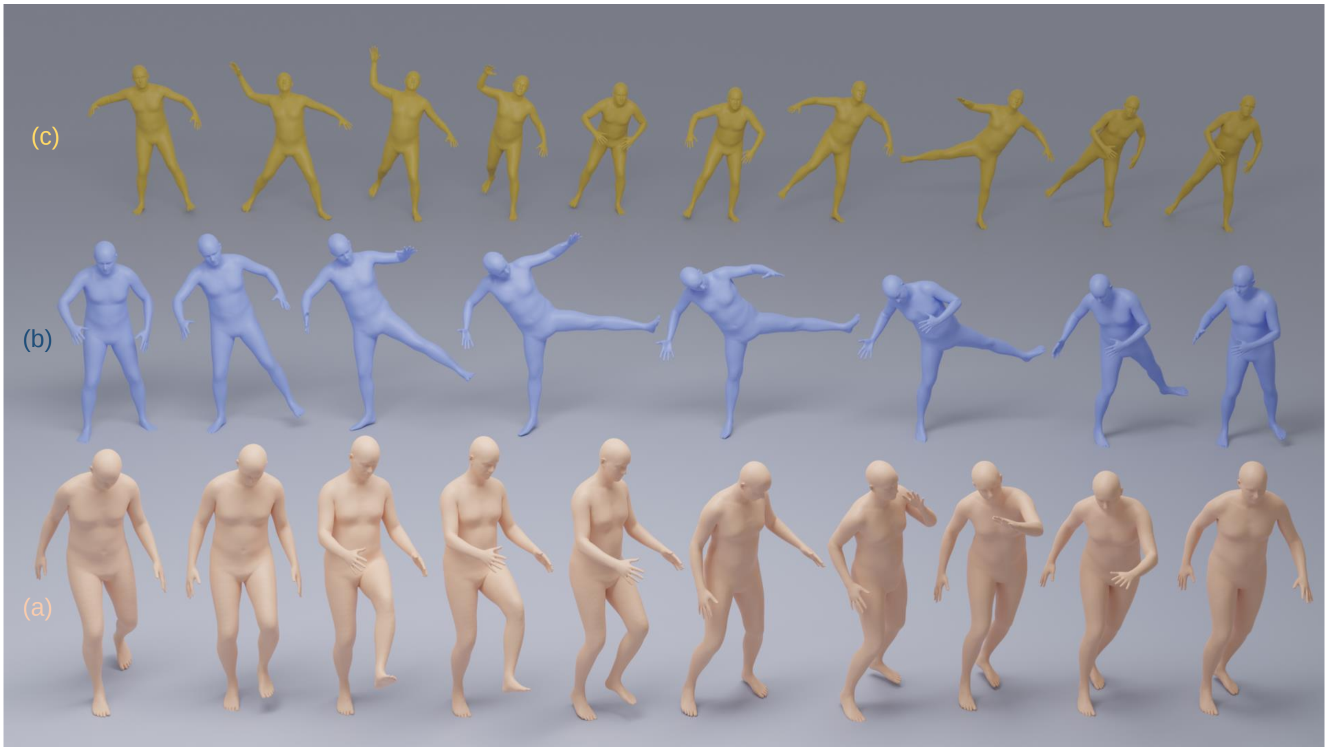

Figure 1: Qualitative results of changing some fine-grained words: (a) A person kicks something with his left leg then throws an object with his left hand. (b) A person kicks something with his left leg while throwing an object with his left hand. (c) A person kicks something with his right leg after throwing an object with his right hand. Motion frames are ordered from left to right.

We have introduced a fine-grained text-driven method for generating human motion sequences that conformed with the text prompt in the main paper. This supplementary material presents more details, including:

- Additional fine-grained visual results in section 1;
- More detailed network architectures in section 2.

1. More Fine-Grained Results

We have shown the qualitative results compared with two state-of-the-art methods in the main paper. Our method compares with the Temporal VAE [1] and MotionDiffuse [2], outperforming these models, especially in terms of grasping the fine-grained details.

In this section, we further apply several subtle modifications to the text prompt, enabling us to assess the quality of the generated motion and demonstrate the robustness of our approach instead of being limited to definite words. Specifically, we implement some changes, such as revising subtle spatial orientation words or temporal prepositions, to the sentences “A person kicks something with his right leg then throws an object with his right hand” and “A person walks forward two steps then backward one step” in the original experiment, as shown in Figure 1 and Figure 2. It can be observed that when replacing some key adjectives or prepositions, our method still yields decent results faithful to the text prompt.

We also show the comparison of some qualitative examples on the ablation study, and present more visual results

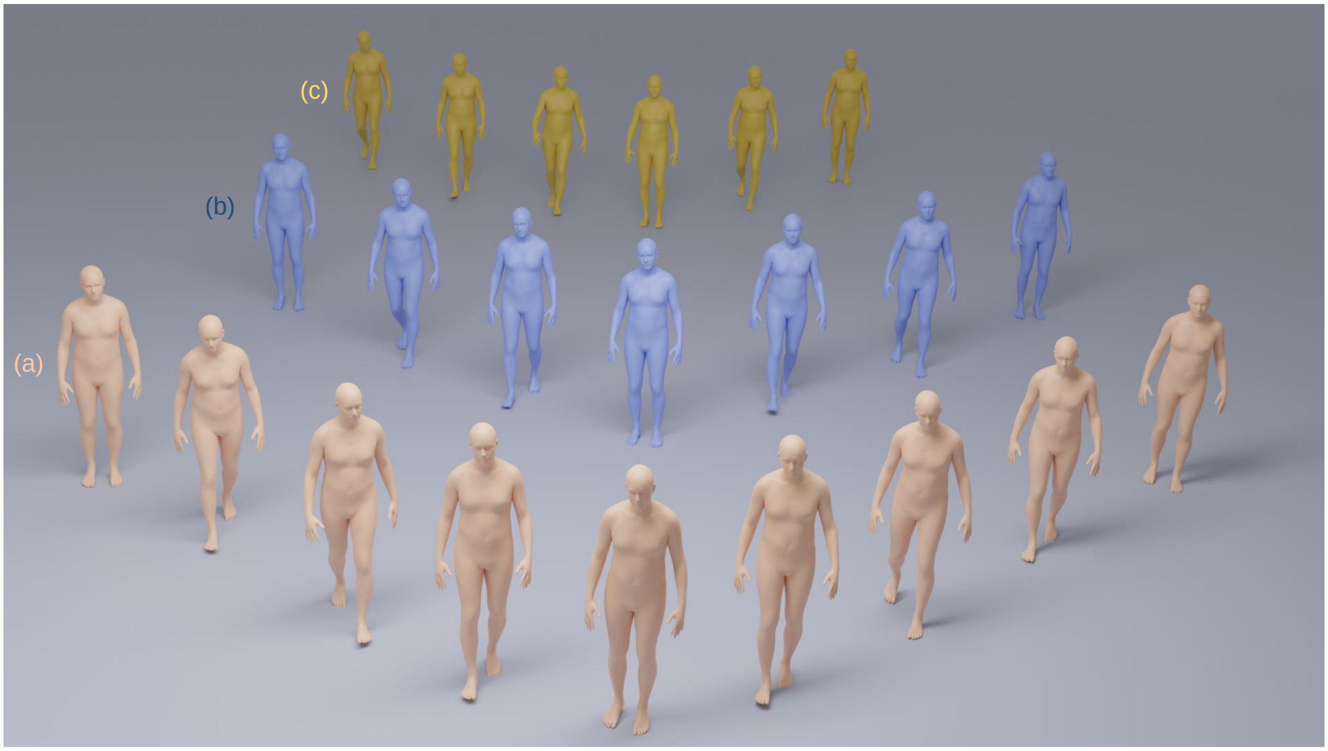

Figure 2: Qualitative results of changing some fine-grained words: (a) A person walks forward three steps then backward three steps. (b) A person walks forward two steps then backward two steps. (c) A person walks forward three steps then backward one step. Motion frames are ordered from left to right.

in Figure 3 and 4.

2. Network Architectures

As mentioned in the main paper, our method consists of two modules, Linguistics-Structure Assisted Module (LSAM) as the text encoder, and Context-Aware Progressive Reasoning Module (CAPR) as the motion decoder. In this section, we further present more detailed network architectures of these two modules, as listed in Table 1. LSAM comprises node embedding, edge embedding, and graph attention convolution, while CAPR incorporates multi-modal sentence-level feature-fusion and multi-head word-level cross-attention.

References

[1] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. 1

[2] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 1

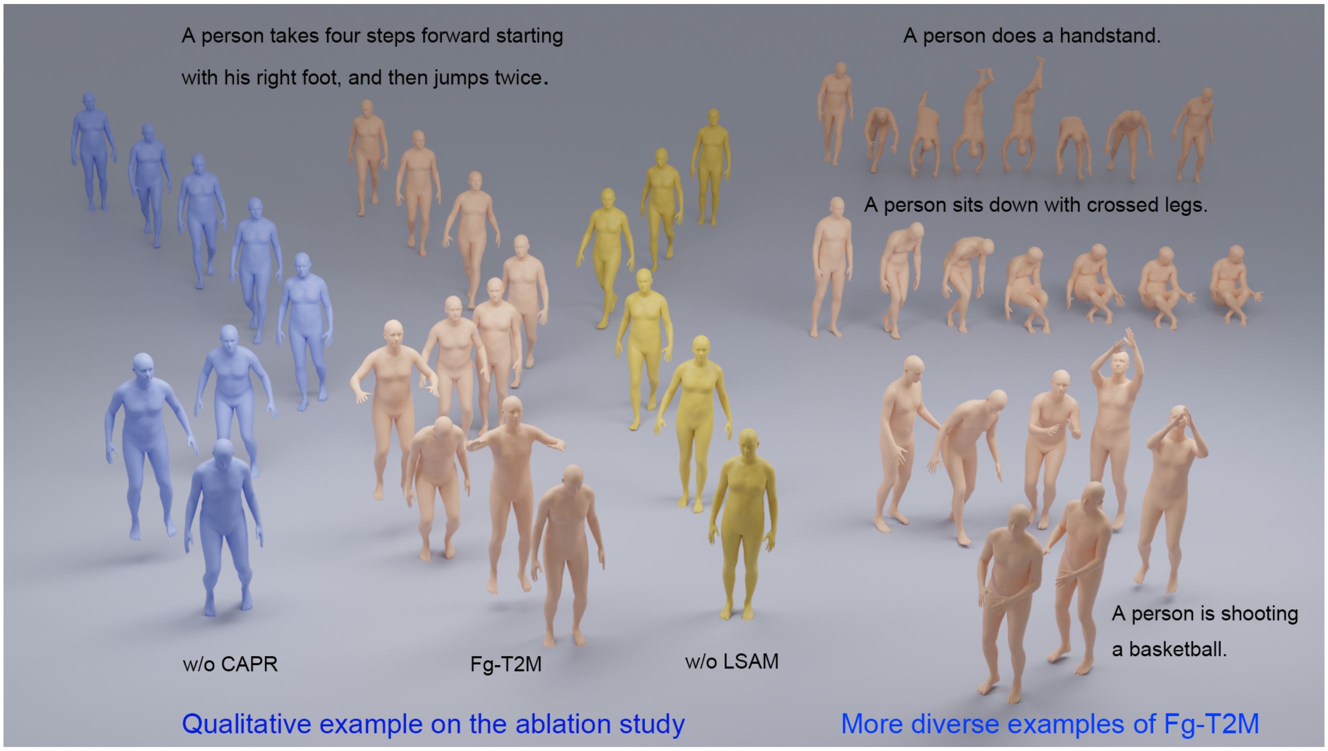
A person takes four steps forward starting with his right foot, and then jumps twice.
A person does a handstand.
A person sits down with crossed legs.
A person is shooting a basketball.
w/o CAPR
Fg-T2M
w/o LSAM
Qualitative example on the ablation study
More diverse examples of Fg-T2M

Figure 3: Qualitative examples on the ablation study and more diverse examples.

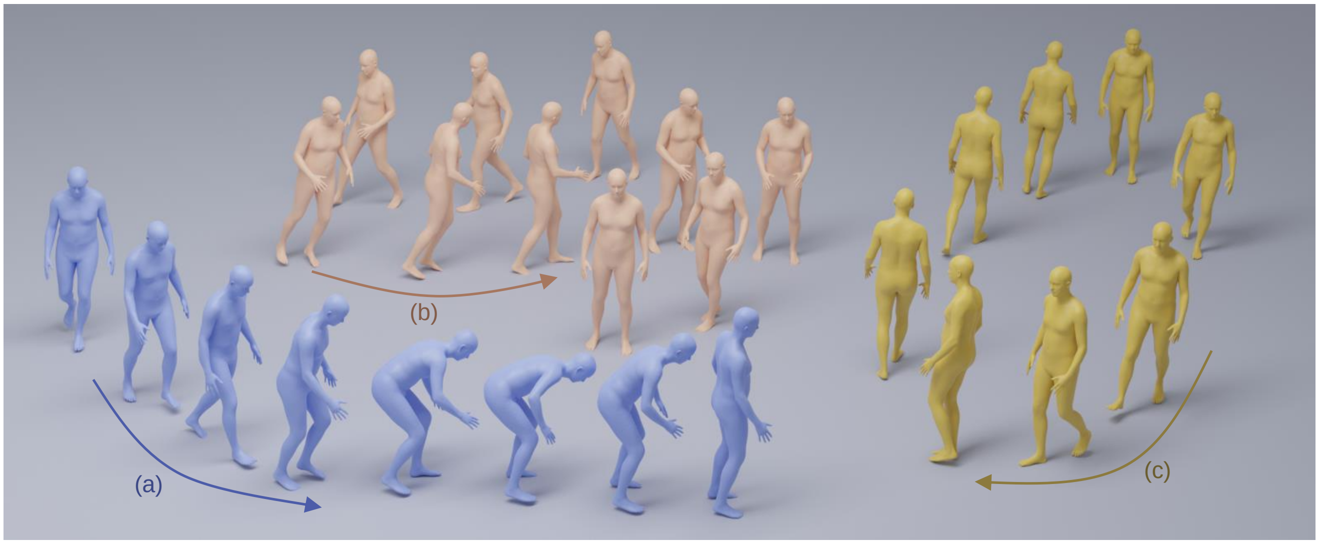

Figure 4: **More Visual results**: (a) A person walks to his left side, then bends down and picks something up. (b) A person runs to his right side first, runs to his left, then back to the middle. (c) A person walks forward two steps, pivots 180 degrees then walks two steps back to where he started.

Components	Architecture
LSAM (Node)	(Clip_Embedding): Clip ((0): clip.tokenize() (1): clip.encode_text()) (Node_Embedding): TransformerEncoder(TransformerEncoderLayer(512, 8, 2048, batch_first=True), num_layers=8) (Ln): LayerNorm(normalized_shape=512)
LSAM (Edge)	(Spacy): spacy.load('en_core_web_sm') (Onehot): OneHotEncoder('ignore') (Edge_Embedding): Sequential((0): Linear(in_features=45, out_features=512) (1): Silu() (2): Linear(in_features=512, out_features=512))
LSAM (Graph)	(Gat_layer 1): Sequential((0): GATConv(512, 512, heads=8, concat=False, edge_dim=512) (1): Silu() (2): Linear(in_features=512, out_features=512) (3): Silu()) (Gat_layer 2): Sequential((0): GATConv(512, 512, heads=8, concat=False, edge_dim=512) (1): Silu() (2): Linear(in_features=512, out_features=512) (3): Silu()) (Gat_layer 3): Sequential((0): GATConv(512, 512, heads=8, concat=False, edge_dim=512) (1): Silu() (2): Linear(in_features=512, out_features=512) (3): Silu()) (layer1_output): Linear(in_features=512, out_features=512) (layer2_output): Linear(in_features=512, out_features=512) (layer3_output): Linear(in_features=512, out_features=512)
CAPR (Sentence-Level)	(Text_Embedding): Sequential((0): Linear(in_features=512, out_features=512) (1): LayerNorm(normalized_shape=512)) (Motion_Embedding): Sequential((0): Linear(in_features=dim_in, out_features=512) (1): LayerNorm(normalized_shape=512)) (Sent_Conv): Conv1d(in_channels=512, out_channels=512, kernel_size=77) (Activation): Sigmoid() (Fusion_Ln): LayerNorm(normalized_shape=512) (Query): Linear(in_features=512, out_features=512) (Key): Linear(in_features=512, out_features=512) (Value): Linear(in_features=512, out_features=512)
CAPR (Word-Level)	(Text_Ln): LayerNorm(normalized_shape=512) (Motion_Ln): LayerNorm(normalized_shape=512) (Query): Linear(in_features=512, out_features=512) (Key): Linear(in_features=512, out_features=512) (Value): Linear(in_features=512, out_features=512) (Mlp): Sequential((0): Linear(in_features=512, out_features=2048) (1): Silu() (2): Linear(in_features=2048, out_features=512)) (3): Silu() (4): Linear(in_features=512, out_features=dim_out))

Table 1: Detail Architecture of our LSAM and CAPR.