Appendix of "HumanMAC: Masked Motion Completion for Human Motion Prediction"

A. Noise Prediction Network TransLinear

The proposed noise prediction network, *i.e.*, TransLinear, is shown in Figure 9. The input of TransLinear is the DCT spectrum at the step t, noted as $\mathbf{y}_t \in \mathbb{R}^{L\times(H+F)}$. TransLinear has two linear layers for both input and output to map the joint's dimension. Besides, N TransLinear blocks are stacked with skip connections [60] in the TransLinear. Motivated by [53], we add two linear-based FiLM modules in the transformer encoder in each TransLinear block. To obtain temporal relationships, the FiLM module is modulated by the first K-frame modulating motion's DCT spectrum and the diffusion time embedding. Since the length of the first K-frame modulating motions is not equal to the length of full motions, we simply to pad the last frame of the modulating motion to the full length and obtain a spectrum of the padding motion. In summary, the TransLinear block is composed of a **Trans**former encoder and some **Linear** operations, which is a simple architecture.

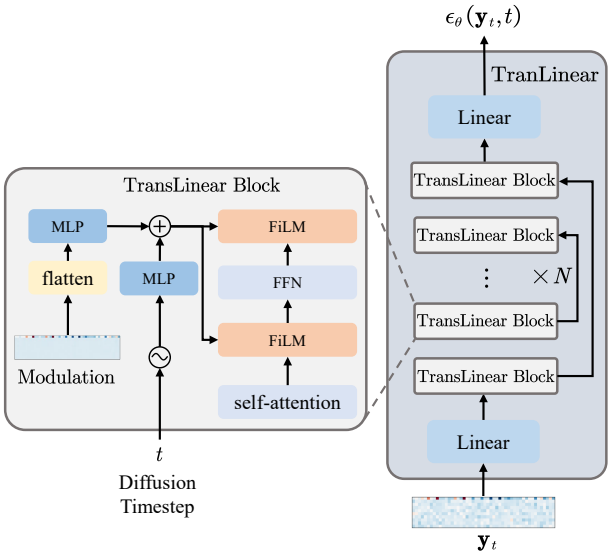


Figure 9: The architecture of the noise prediction network TransLinear, which takes the DCT spectrum $\mathbf{y}_t$ at the diffusion timestep t as input. TransLinear is composed of N blocks with skip connections and a linear layer.

B. Implementation Details

To ensure reproducibility, we report the implementation details of HumanMAC. Source codes are public at https://github.com/LinghaoChan/HumanMAC.

We evaluate our model on two popularly used human motion datasets, *i.e.*, Human3.6M [27] and HumanEva-I [62]. Human3.6M consists of 7 subjects performing 15 different motions, and 5 subjects (S1, S5, S6, S7, and S8) are utilized for training, while the remaining two (S9 and S11) are utilized for evaluation. We apply the original frame rate (50 Hz) and a 17-joint skeleton removing the root joint to build human motions. Our model predicts 100 frames (2s) via 25 observation frames (0.5s). HumanEva-I comprises 3 subjects each performing 5 actions. We apply the original frame rate (60 Hz) and a 15-joint skeleton removing the root joint to build human motions. We predict 60 frames (1s) via 15 (0.25s) frames.

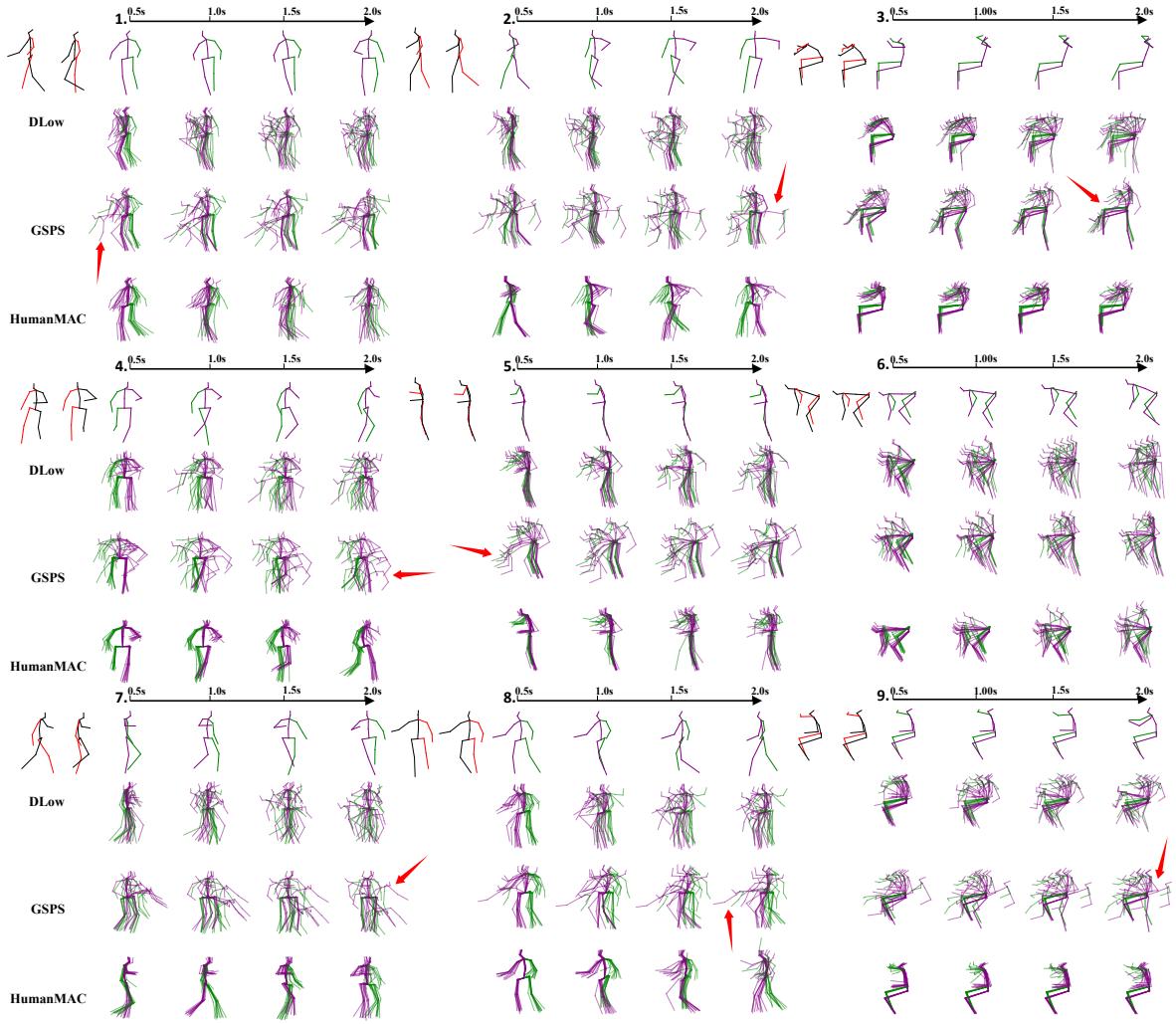
For both datasets, the batch size is set to 64. The model is trained for 500 epochs. The optimizer is set as Adam [31]. The learning rate is 3×10^{-4} with a multi-step learning rate scheduler ($\gamma = 0.9$). The dropout rate is 0.2. In the DCT/iDCT operation, the number of L is set to be 20 and 10 for Human3.6M and HumanEva-I respectively. For the denoising diffusion model, the variance scheduler is the Cosine scheduler [48] with 1000 noising steps. The DDIM sampler is set to 100 steps in the sampling stage. For the network architecture, the number of the self-attention [73, 94] head is set as 8. The latent dimension is 512. In the inference stage, the modulation ratio in the HumanEva-I dataset is set as 0.5 and 1.0 for the Human3.6M dataset. For the motion switch ability, since the content of the observation and target have been mostly recovered in the final denoising steps of the diffusion model, we replace the final 20 steps of DCT-Completion with the vanilla denoising steps, which simplifies computation.

For the experiments of zero-shot motion prediction, we first retarget the skeleton in the AMASS dataset to the skeleton in the Human3.6M dataset by a widely used human motion retargeting tool[1]. After skeleton retargeting, we inference the AMASS motion with the model trained on the Human3.6M dataset directly.

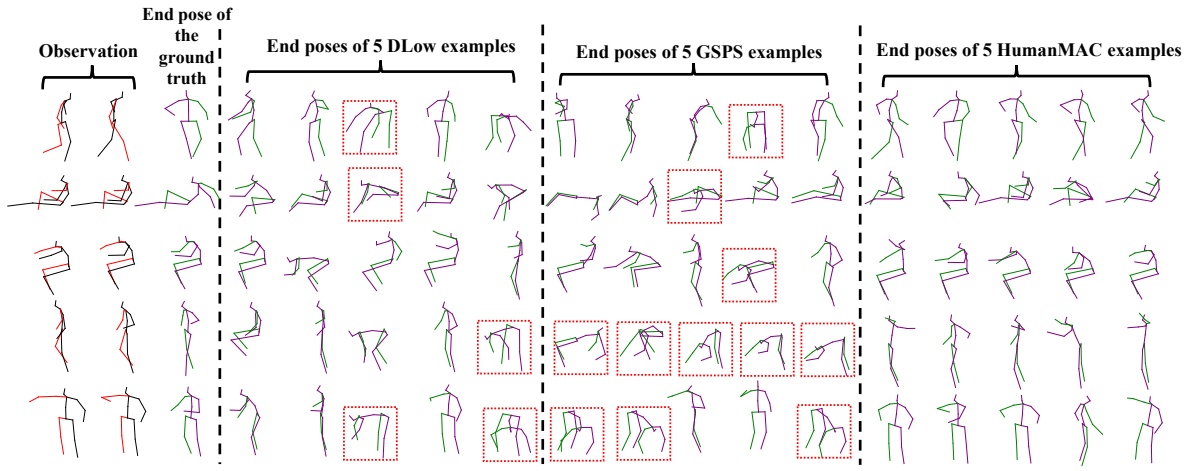
C. Supplementary Visualization Results of Motion Prediction on Human3.6M and HumanEva-I

We provide more empirical evidence of visualization comparison with DLow [89] and GSPS [45] in Figure 10. The visualization results of motion sequences and end poses are shown in Figure 10a and Figure 10b respectively. Cases highlighted with **red arrows** in Figure 10a and end poses with **red dashed boxes** in Figure 10b are failure cases that do not satisfy the physical constraints of the human center of gravity. By contrast, the diversity of predicted motions by our method is more reasonable than baseline methods.

[1] https://theorangeduck.com/page/deep-learning-framework-character-motion-synthesis-and-editing



(a) Comparison on motion sequences visualization.



(b) Comparison on end pose visualization.

Figure 10: Visualization of motion prediction results. We present both motion sequences comparison and end-pose comparison. The *red-black* skeletons and *green-purple* skeletons denote the observed and predicted motions respectively.

D. Motion Switch

In Figure 11, we present more results to show the motion switch ability of our method in Figure 11. We provide some hard cases, *e.g.*, `Sitting-Walking` switch (example *D, E, F, G, I, K*) and `Turning` (example *A, C, D, E, F, I, M, O*). For transferring between two motions with *a large distribution gap*, the motion of the upper and lower bodies changes in a natural way. We provide more animations on the project page.

E. Part-body Controllable Prediction Results

We present more results of our part-body controllable predictions in Figure 12. As shown in Figure 12, different from previous methods, our method supports the controllability of arbitrary body parts, *e.g.*, `Right Leg`, `Left Leg`, `Right Arm`, `Left Arm`, and `Torso`. This ability will facilitate controllable automatic animation synthesis.

F. Ablation Study on Network Architecture

Motivated by the U-Net [60] design, a significant design in our TransLinear network is the skip connection design. As shown in Figure 8, our results show that the skip connection in the network improves the authenticity of motions.

Mechanism	Human3.6M		
	APD↑	ADE↓	FDE↓
w/o skip connection	**6.479**	0.377	0.486
w/ skip connection	6.301	**0.369**	**0.480**
	HumanEva-I		
w/o skip connection	6.207	**0.208**	0.224
w/ skip connection	**6.554**	0.209	**0.223**

Table 8: Ablation study on the skip connection designing.

G. Comparison on Motion Switch and Part-body Controllable Ability

We compare the motion switch and part-body controllable ability with baselines. As shown in Table 9, our method can achieve both motion switch and part-body controllable ability. For the part-body controllable ability, DLow [89] and GSPS [45] need a specific model training stage to achieve it. In more detail, DLow needs to disentangle the human joints into two parts for training. GSPS trains two networks for the upper and lower body respectively. In our implementation, we achieve this only in the inference stage without any specific modeling. Moreover, our method supports any part-body controllable prediction.

H. Supplementary Results of Zero-shot Motion Prediction

We provide more empirical evidence of zero-shot motion prediction results in Figure 13. The visualization results of predicted motion sequences and end poses are shown in Figure 13a and Figure 13b respectively. As the results shown in Figure 13, our method can predict some motions not seen in the Human3.6M dataset, such as *opening arms exaggeratedly* and *kicking sharply*. See more vivid predicted motions in the supplementary video.

Method	Switch Ability	Part-body Controllable
acLSTM [97]	✗	✗
DeLiGAN [20]	✗	✗
MT-VAE [83]	✗	✗
BoM [7]	✗	✗
DSF [90]	✗	✗
DLow⋆ [89]	✗	✓
GSPS⋆ [45]	✗	✓
MOJO [95]	✗	✗
BeLFusion [5]	✓	✗
DivSamp [14]	✗	✗
MotionDiff [75]	✗	✗
HumaMAC	✓	✓

Table 9: Comparison on motion switch and part-body controllable ability. A summary of the motion editing ability of different methods. Methods with ⋆ indicate that they need specific training for achieving the part-body controllable ability.

I. Engineering Optimization for Evaluation

To provide more convenience for the research community, we optimize the evaluation process from an engineering aspect. The optimization consists of two aspects: (1) parallelized model inference; (2) parallelized metric calculation. We implement the evaluation on both parallelized model inference and parallelized metric calculation. For the parallelized model inference, we parallelize the serialized inference process over multiple examples in [89]. We present the simulation results of the parallelized metric calculation in Table 10. The results show that our method has $\sim 6\times$ speed up than the previous implementation[2]. For intuitive comparison, the comparison of the simulation is also shown in Figure 14. After the engineering optimization, the overall (both parallelized model inference and parallelized metric calculation) speedup is $\sim$ 1k $\times$ than the previous implementation. The overall optimization speedup is shown in Table 11. For engineering optimization, we perform the experiment of engineering optimization on a machine with 30 GB memory, 32 CPU cores, and one NVIDIA Tesla A5000 GPU. For more details, please refer to `https://github.com/LinghaoChan/HumanMAC`.

[2] `https://github.com/Khrylx/DLow`

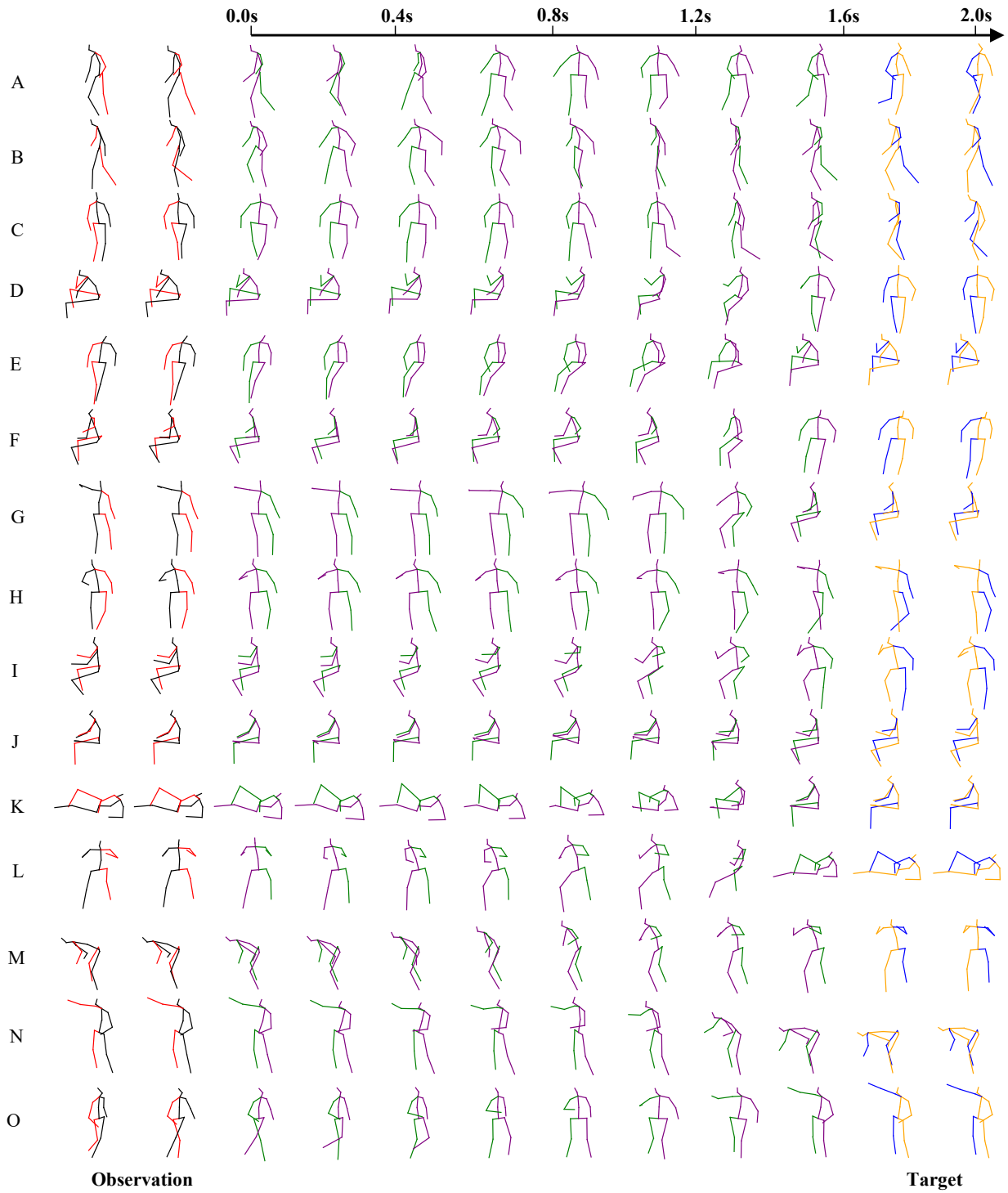


Figure 11: Additional motion switch results. Visualization of motion transfer using DCT-Completion from the Human3.6M dataset.

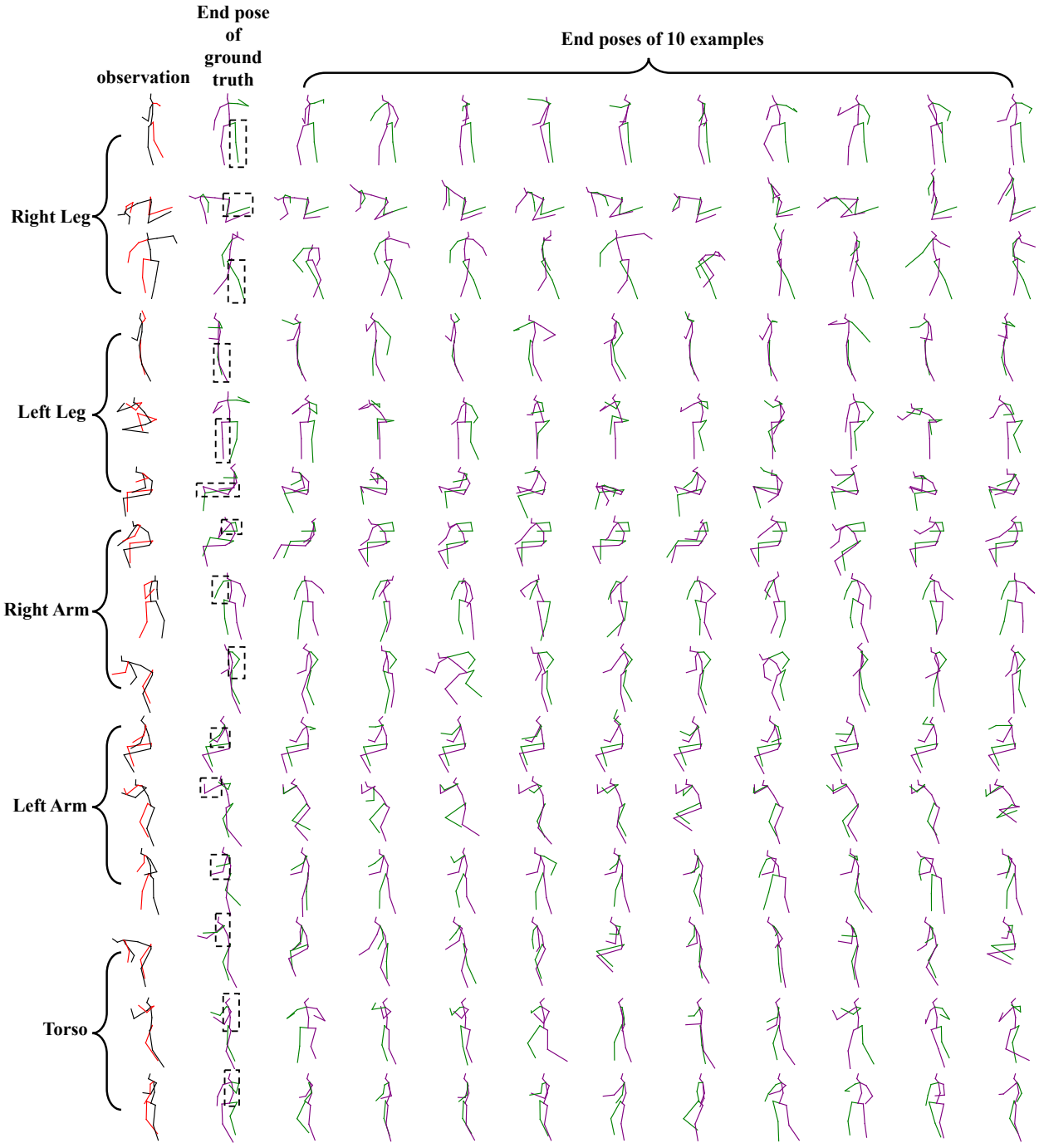
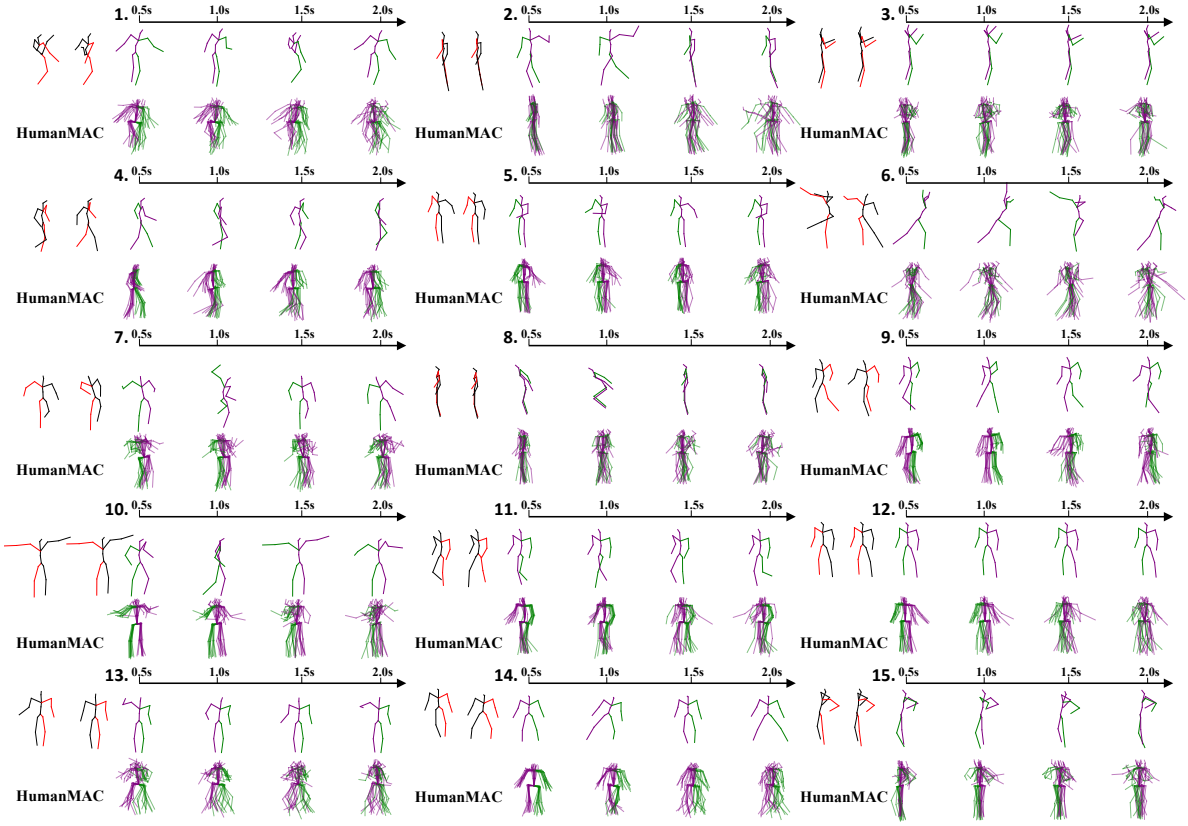
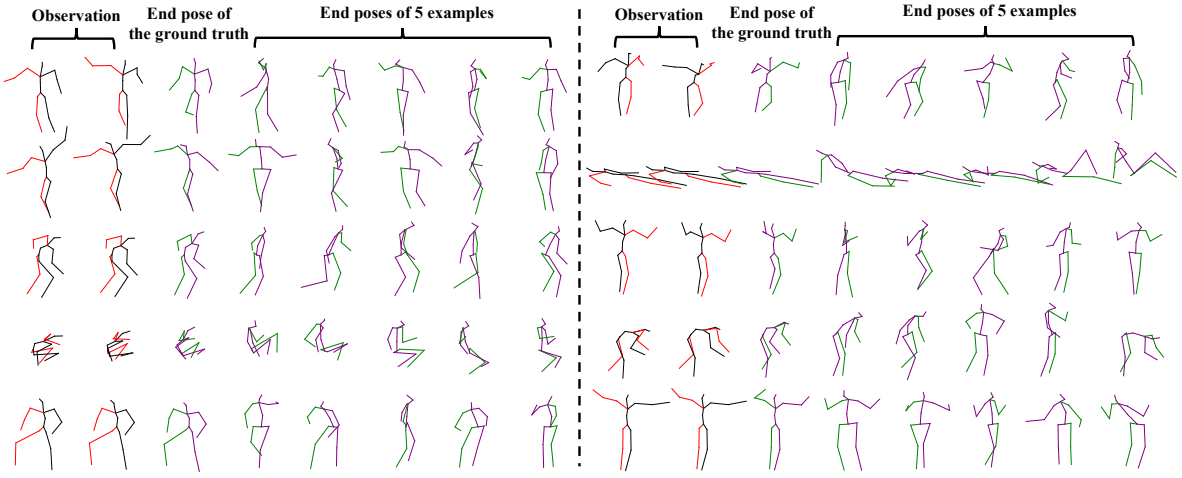


Figure 12: Flexible controllable motion prediction. We split the human skeleton into 5 parts: left leg, right leg, left arm, right arm, and torso. In each row, we show the end poses of 10 examples of various motions when part-control certain body parts.



(a) Motion sequences visualization. The first row is the ground truth. The second row indicates a sample of 10 predictions.



(b) End pose visualization. We visualize the end pose of the prediction from three random examples.

Figure 13: Visualization results of zero-shot adaption ability on the AMASS dataset. The *red-black* skeletons and *green-purple* skeletons denote the observed and predicted motions respectively.

# Examples	w/o optimization	w/ optimization	Speedup
100	4.87±0.05	**0.56±0.05**	↑ 869.6%
500	20.55±0.21	**3.10±0.81**	↑ 662.9%
1000	43.29±0.25	**6.88±1.14**	↑ 672.8%
2000	78.51±0.16	**12.49±0.48**	↑ 628.6%
5000	179.78±0.69	**29.95±4.04**	↑ 600.6%

Table 10: Engineering optimization simulation. We randomly generate a certain number of examples for evaluation. We show the results without (w/o) and with (w/) our engineering optimization for comparison.

Method	w/o optimization	w/ optimization	Speedup
DLow [89]	∼13 h	**∼ 52 s**	∼1k×
Ours	-	**∼16 mins**	◇

Table 11: Overall evaluation optimization comparison. The symbol '-' indicates the computation time is longer than 1 day. The symbol '◇' means that the speed improvement is significant.

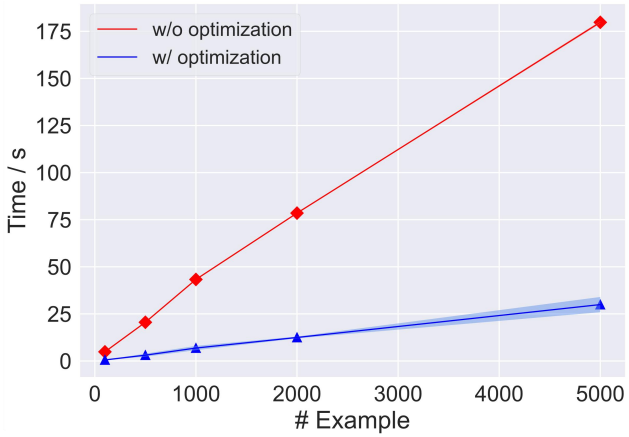


Figure 14: Engineering optimization simulation. We show the results without (w/o) and with (w/) our engineering optimization for comparison. Our implementation improves the speed significantly.