A. Additional Qualitative Results

We show additional qualitative comparisons for three editing tasks: style transfer (Figs. A4 to A7), body part replacement (Figs. A8 to A11), and fine-grained adjustment(Figs. A12 to A14). We highly recommend viewing our *project website* for compelling demonstrations across diverse scenarios.

B. Additional Implementation Details

B.1. Keypoint-Based Motion Representation

Our keypoint-based motion representation uses the first 22 joints from SMPL-X [41] as primary body joints. The two additional head joints and four finger joints (ring and index fingertips of both hands) correspond to the following SMPL-X indices:

- Joint 23: `left_eye_smplhf`.
- Joint 24: `right_eye_smplhf`.
- Joint 25: `left_index1`.
- Joint 34: `left_ring1`.
- Joint 40: `right_index1`.
- Joint 49: `right_ring1`.

These additional joints enable natural gaze behavior and head tracking through eye joints, while fingertip joints provide enhanced control over hand poses as end-effectors.

B.2. Keypoint Canonicalization and Nomalization

We canonicalize motion segments in a y-up coordinate system to simplify the learning space. For each training segment, we apply a transformation to the entire keypoint sequence that translates the first frame's pelvis to the horizontal origin (x and z) and rotates around the y-axis to align the character's initial forward direction with the positive z-axis. During inference, segments are merged through decanonicalization. Specifically, for segment i, we align it with segment $i-1$ by computing the transformation between their connecting frames (first frame of segment i and second-to-last frame of segment $i-1$) using the Kabsch algorithm on the rigid triangle formed by the pelvis and hip joints.

In addition to canonicalization, we normalize each spatial dimension (x, y, and z) of the keypoint data to the standardized range $[-1, 1]$ using channel-specific scaling factors. These factors are determined by the minimum and maximum values of each channel across the dataset. We capture 95% of the data range to compute these scaling factors with the outliers removed. During inference, we reverse this normalization by applying the inverse scaling factors to the model output.

B.3. Converting between Motion Representations

To convert SMPL-X parameters to keypoint representation, we perform forward kinematics using the official SMPL-X codebase, which transforms sequential pose parameters into 3D joint locations. We set hand and face parameters to zero vectors to focus on core body movements.

Converting keypoint representation to SMPL-X parameters involves a two-stage approach. First, we standardize each frame by translating the 28 keypoints to center the pelvis at the origin. The translated keypoints (84-dimensional input) are processed through a 3-layer MLP (512 hidden units, ReLU activation, layer normalization) to estimate the 66-dimensional SMPL-X body pose parameters, including global orientation. Second, we refine these initial body pose estimates and predict the global translation through optimization. We iteratively compute keypoint locations via SMPL-X forward passes and minimize the mean squared error between the computed and targeted keypoints. Optimization is performed for 120 iterations using the Adam optimizer [34] with a learning rate of 0.01.

B.4. Module Details

In our motion diffusion model, noisy motion frames from the canonicalized sequence $\mathcal{M}_t$ are encoded through an MLP encoder, where a single linear layer projects the input from 84 dimensions (28 joints $\times$ 3) to 512 dimensions. The original motion sequence $\mathcal{M}_{\text{ori}}$ is encoded through a separate MLP encoder with identical architecture. We implement a Transformer encoder [59] as the UNet backbone with 6 layers, 16 attention heads, and a dropout rate of 0.1. The encoded vectors from $\mathcal{M}_{\text{ori}}$ are added frame-wise to the encoded noisy motion to preserve reference motion information. A conditional token combines text condition embedding, progress indicator, and diffusion step embedding for temporal context. The Transformer encoder then processes the entire token sequence, followed by an MLP decoder projecting the output back to 84 dimensions.

For the body part coordinator D, we adopt a Transformer encoder with identical architecture to our main model. The transformer's outputs are mean-pooled temporally and processed by an MLP to classify whether the input motion is spatially composed. To ensure robustness during diffusion sampling, we inject random noise into the training keypoint sequences, with magnitude matching the noise levels of the last 20 diffusion steps.

B.5. Frame Rate

We downsample motion sequences to 10 FPS during training and inference for computational efficiency. For compatibility with standard evaluation protocols, the generated keypoint sequences are later upsampled to 20 FPS during SMPL-X conversion (Appendix B.3) to match the original dataset's frame rate.

B.6. Hyper-Parameters for Guidance

During inference, we apply classifier-free guidance [20] with weight $w = 3$ to enhance conditional signals through

linear extrapolation. For the body part coordinator, we set λ to 1.0 and apply classifier guidance during the final 20 steps of the auto-regressive sampling process.

C. Additional Experiment Details and Results

C.1. Training Details

In our experimental framework, all models undergo training for 1,500 epochs using the DDPM scheduler [21], with varying numbers of diffusion steps across different methods: our approach employs 100 steps, TMED [7] uses 300, and MDM-BP [56] requires 1,000, following their respective recommended configurations. We employ the AdamW optimizer [38] with a learning rate of 1e-4 and a weight decay of 0.01. The learning rate follows a linear decay schedule. During training, we use a batch size of 1024 sequences, with each sequence containing W frames. The training process is conducted on a setup of 4 NVIDIA RTX 3090 GPUs, with the entire training cycle completed within 36 hours. The model checkpoints are saved every 50 epochs, and we select the best model based on validation performance.

C.2. Adaption of Baselines

For baseline comparisons, we adapt MDM [56] with inpainting-based motion editing, where specific body parts are modified according to the provided masks. We enhance the baseline by supplying explicit masking information and initializing diffusion from the original motion sequence. We introduce an important modification to the standard MDM approach: while most of the diffusion process maintains strict masking constraints, we release these constraints during the final 20 diffusion steps, allowing the model to adjust the entire body. This modification enables natural whole-body adaptations that may be necessary for coherent motion synthesis. For TMED [7], we maintain strict adherence to the original implementation, utilizing the exact configurations and parameters as specified in the authors' codebase.

C.3. Dual Interpretation of the E2S Score

We argue that the interpretation of Edited-to-Source Retrieval (E2S) scores should be task-dependent.

For fine-grained adjustments (*e.g.*, modifying arm raise height), higher E2S scores are desirable as they indicate preserved motion characteristics with successful subtle modifications. Similarly, for MotionFix dataset [7] tasks which involve minor adjustments like refining limb positions and trajectories, high E2S scores demonstrate proper maintenance of source motion semantics.

However, for substantial editing tasks like body part replacement or style transfer, the E2S scores should align with the reference dataset's distribution rather than maximizing similarity to the source. In these cases, lower E2S scores may actually indicate successful editing, as the motion should significantly deviate from the source to reflect the intended modifications. The accuracy of these major changes should instead be evaluated through the Edited-to-Target Retrieval score, which measures alignment with the target characteristics.

C.4. Ablation Results of Classifier Guidance

In Fig. A1, we evaluate how body part coordinator performs across different hyper-parameters. The x-axis shows guidance strength λ, while the y-axis indicates the number of steps where classifier guidance is applied. We report both E2T AvgR (upper) and FID (lower) for the body part replacement task. Setting $\lambda = 1.0$ and applying 20 guidance steps produces optimal results.

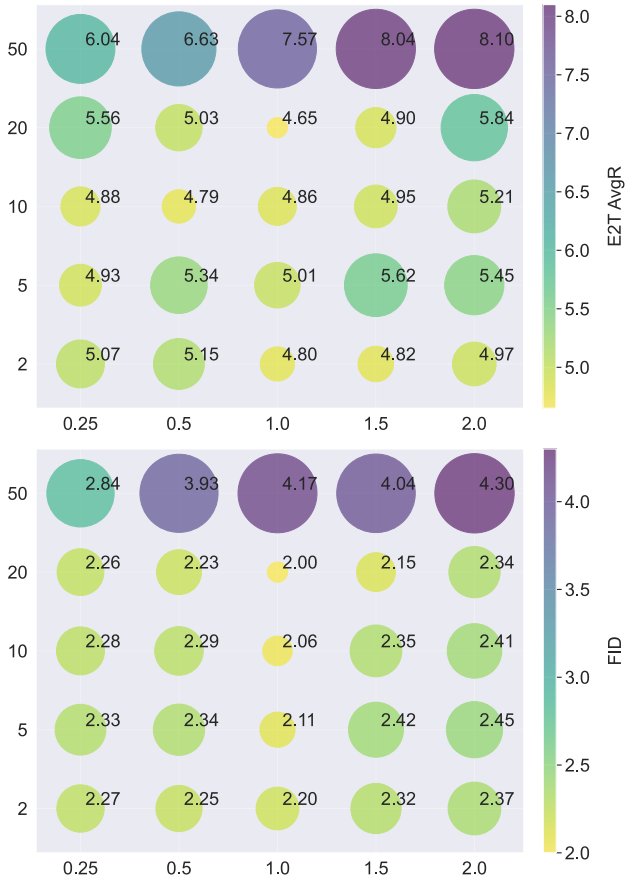


Figure A1. **Ablation results on classifier guidance.** We illustrate the E2T AvgR (upper) and FID (lower) performance of MotionReFit for the body part replacement task. The x-axis represents guidance strength, whereas the y-axis depicts guidance steps count.

C.5. Results of Fine-Grained Adjustment

Quantitative results in Tab. A1 demonstrate that our full method achieves superior performance across most metrics for the fine-grained adjustment task. The retrieval metrics reveal that the motion characteristics have been maintained, with successful fine-grained adjustments.

Table A1. **Quantitative comparison on fine-grained adjustment task.** For each metric, we repeat the evaluation 10 times. Arrows (→) indicate metrics where values closer to real data are better. **Bold** denotes best performance.

Method	FID↓	Diversity→	FS↑	Edited-to-Source Retrieval				Edited-to-Target Retrieval			
				R@1↑	R@2↑	R@3↑	AvgR↓	R@1↑	R@2↑	R@3↑	AvgR↓
Real Data	0.02	30.57	0.97	39.54	54.65	61.16	5.53	100.0	100.0	100.0	1.00
MDM-BP [56]	0.62	32.70	0.92	28.12	34.38	38.02	10.41	16.45	24.52	30.21	11.60
TMED [7]	0.33	31.13	0.94	60.16	72.66	82.03	2.66	29.69	44.01	52.08	6.97
TMED w/ MCM	0.33	31.42	0.94	62.8	74.78	87.0	2.61	32.22	45.03	54.83	6.56
Ours w/o MCM	0.34	**31.08**	**0.95**	81.77	92.45	93.49	1.48	34.11	48.70	57.03	5.77
Ours full	**0.29**	31.29	**0.95**	**85.16**	**92.97**	**95.31**	**1.38**	**42.45**	**56.25**	**62.76**	**5.12**

Table A2. **Ablation analysis for fine-grained adjustment.** Results show means across 10 evaluation runs, with **bold** indicating best result.

Method	FID↓	Edited-to-Source Retrieval				Edited-to-Target Retrieval			
		R@1↑	R@2↑	R@3↑	AvgR↓	R@1↑	R@2↑	R@3↑	AvgR↓
1% MCM	0.34	81.77	92.45	93.49	1.48	34.11	48.70	57.03	5.77
5% MCM	0.37	**86.72**	**95.57**	**97.14**	**1.30**	34.17	50.00	57.81	5.65
10% MCM	0.31	82.81	92.71	95.31	1.42	37.24	51.30	59.11	5.32
20% MCM	0.29	85.68	91.93	94.27	1.45	39.06	52.08	60.68	5.36
12% data	0.32	81.51	91.67	94.53	1.56	40.10	58.07	**67.71**	4.74
24% data	0.31	82.03	92.19	95.83	1.42	41.93	**59.11**	67.45	**4.71**
60% data	0.30	84.90	92.45	96.09	1.38	41.67	55.47	63.54	5.02
Ours full	**0.29**	85.16	92.97	95.31	1.38	**42.45**	56.25	62.76	5.12

Table A3. **Quantitative comparison with TMED [7] on MotionFix using the full dataset [7].** Results show means across 10 evaluation runs, with **bold** indicating best result.

Method	FID↓	FS↑	Edited-to-Source Retrieval				Edited-to-Target Retrieval			
			R@1↑	R@2↑	R@3↑	AvgR↓	R@1↑	R@2↑	R@3↑	AvgR↓
Real Data	0.010	0.98	20.83	33.66	40.47	33.13	64.36	88.75	95.56	1.74
MDM-BP [56]	0.145	0.90	30.21	36.82	40.47	106.05	8.69	14.71	18.36	180.99
TMED [7]	0.129	0.92	22.41	34.45	40.57	31.42	**14.51**	21.72	28.73	56.63
Ours	**0.120**	**0.96**	**43.77**	**56.72**	**64.13**	**24.09**	14.13	**23.52**	**30.53**	**54.06**

Tab. A2 presents quantitative comparisons between our method and ablation variants on the fine-grained adjustment task. While increasing the MotionCutMix Ratio generally enhances results, we find that a lower ratio of 5% actually achieves optimal performance, outperforming higher ratios including 100%. This phenomenon can be attributed to the inherent consistency of editing patterns across fine-grained motion adjustments. Additionally, our experiments show that varying the size of the annotated dataset produces only marginal differences in performance metrics. This finding suggests that our method achieves effective generalization even with a smaller annotated dataset, likely because our large-scale training set already encompasses a comprehensive range of fine-grained adjustment scenarios.

Figs. A12 to A14 showcase visual comparisons between our method and ablations across diverse editing instructions, demonstrating our full method's superiority in producing precise and natural motion edits.

C.6. Results on the MotionFix Dataset

Evaluation Settings For TMED [7] compatibility, we use a 22-keypoint representation aligned with the SMPL model [37], instead of the 28-keypoint SMPL-X format used in our main method. The conversion process between keypoint representation and SMPL parameters remains similar to the one described in Appendix B.3.

For our auto-regressive framework, we preprocess the MotionFix dataset by segmenting continuous motions into clips and applying canonicalization. For retrieval-based metrics evaluation, we use the original TMR checkpoint [43] to ensure consistent comparison with previously reported results.

Comparison on the Entire Test Set Tab. A3 shows full-scale evaluation results on the MotionFix benchmark comparing our method against TMED and MDM baselines. Consistent with the batch-wise evaluation, our method demonstrates superior performance in both E2T scores for

Table A4. **Breakdown of inference time on a single RTX 3090 GPU.** Our optimal setting achieves real-time inference speed.

Window size	Diffusion sampling	Body part coordinator	SMPL-X optimization	Total (seconds)	FPS
2-frame	0.142	0.014	0.067	0.223	8.97
8-frame	0.355	0.036	0.106	0.497	16.10
16-frame	0.474	0.046	0.126	0.646	24.76

editing accuracy and E2S scores for motion preservation. Most notably, we achieve substantially higher foot contact scores, indicating significantly improved physical plausibility and overall motion quality.

For detailed qualitative comparisons and motion visualizations that further illustrate these improvements, we direct readers to Appendix A.

C.7. Real-Time Inference

In Tab. A4, we provide a breakdown of inference time on a single RTX 3090 GPU. Despite the auto-regressive nature, inference with a 16-frame window size (our optimal setting) achieves real-time speed. Furthermore, the motion coordinator is applied only during the final few diffusion steps, adding minimal overhead to the overall computation.

D. Additional Details on the STANCE Dataset

D.1. Body Part Replacement

Our body part replacement subset extends HumanML3D [18] through a two-phase annotation process capturing both body part participation and detailed motion descriptions.

Mask Annotation The first phase focuses on mask annotation, where we developed specialized visualization software to streamline the annotation process. As shown in Fig. A2, this tool renders HumanML3D motion sequences in 3D and offers annotators a selection of predefined body part masks and their combinations. Annotators can play, pause, and scrub through the animation while making their selections based on direct visualization of the motions. For each sequence, annotators identify which body parts are actively participating in meaningful movements, as opposed to parts that remain relatively static or perform only supporting motions. This visual-based annotation approach distinguishes our dataset from previous works that rely solely on language model interpretation of text descriptions to determine body part involvement [6]. We employed five trained annotators who processed sequences from HumanML3D, resulting in multiple mask annotations per sequence.

Detailed Annotation The second phase involves creating detailed descriptions for the movements of designated body parts. We initialize this process using GPT-4 to obtain the

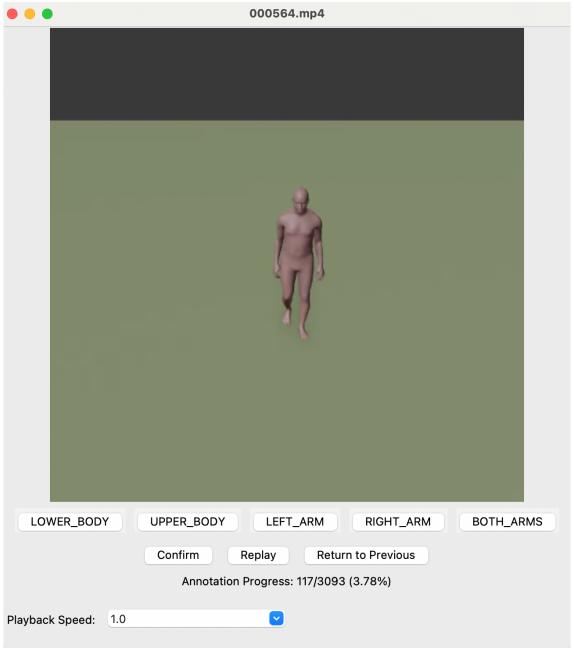


Figure A2. **Screenshot of our annotation software.**

original HumanML3D motion descriptions and specific instructions to focus on particular body parts while excluding others. For example, given a motion described as "a person walks forward while waving their arms," and focusing on the arms, the LLM might generate "waves arms enthusiastically from side to side." These initial descriptions then undergo careful refinement by human annotators who enhance their accuracy, naturalness, and linguistic diversity. This combined approach leverages both automated assistance and human expertise to create approximately 13,000 rich, precise annotations of body part movements. Each motion sequence receives 2-4 different body part-specific descriptions, creating a diverse set of potential editing targets.

D.2. Motion Style Transfer

We construct a motion style transfer dataset by professionally recreating sequences from HumanML3D [18] using the high-fidelity Vicon motion capture system. In our capture sessions, we enlisted trained performers who were instructed to replay selected HumanML3D sequences while incorporating specific style variations. They first familiarized themselves with the original motions through video playback and practice sessions, then executed each motion multiple times with different stylistic interpretations. The capture setup consisted of 12 Vicon cameras operating at 30 fps, positioned strategically around a 6×6 meter capture volume. Performers wore a standard 53-marker FrontWaist set for full-body tracking, ensuring accurate capture of subtle stylistic nuances.

We focused on distinct style categories: proud, old, play-

ful, depressed, and angry, with each performer interpreting these styles based on provided style guidelines. From 180 base motions selected from HumanML3D, we captured each motion in all five styles, resulting in a dataset of 900 high-quality motion sequences after post-processing and cleanup. Each sequence is paired with its original HumanML3D counterpart and annotated with detailed descriptions of the stylistic differences, creating style transfer triplets suitable for training and evaluation.

D.3. Fine-Grained Adjustment

We introduce a novel text-to-motion generation approach for obtaining semantically consistent motion pairs. We curate 5,000 base instructions spanning common human actions (walking, running, dancing, sports activities). For each instruction, we generate the initial motion using MLD's standard sampling process [12]. To create variants, we additionally apply Gaussian noise ($\sigma = 0.1$) to the latent space, creating 16 slightly different but semantically consistent variations for each base motion. These variants maintain the core action while exhibiting subtle differences in execution style, speed, or range of movement.

The variants are then paired one-to-one, creating 8 pairs per instruction. Trained annotators carefully examine each pair and describe the specific modifications needed to transform the original motion into the edited motion. The annotations focus on precise, actionable descriptions such as "bend the knees more deeply," "perform the arm swing with greater force," or "slow down the spinning movement slightly." To ensure dataset quality and clarity, we implement a rigorous filtering process where triplets with unnatural motions (*e.g.*, physically implausible movements) or unclear editing descriptions are discarded. Additionally, we maintain a balanced distribution across different motion categories and editing types to prevent dataset bias.

This systematic approach results in a large-scale dataset of 16,000 annotated triplets, each consisting of an original motion, an edited motion, and a clear instruction for the required modification. The dataset covers a wide range of fine-grained adjustments, including changes in motion amplitude, speed, force, and spatial positioning of body parts.

E. Compositional Applications

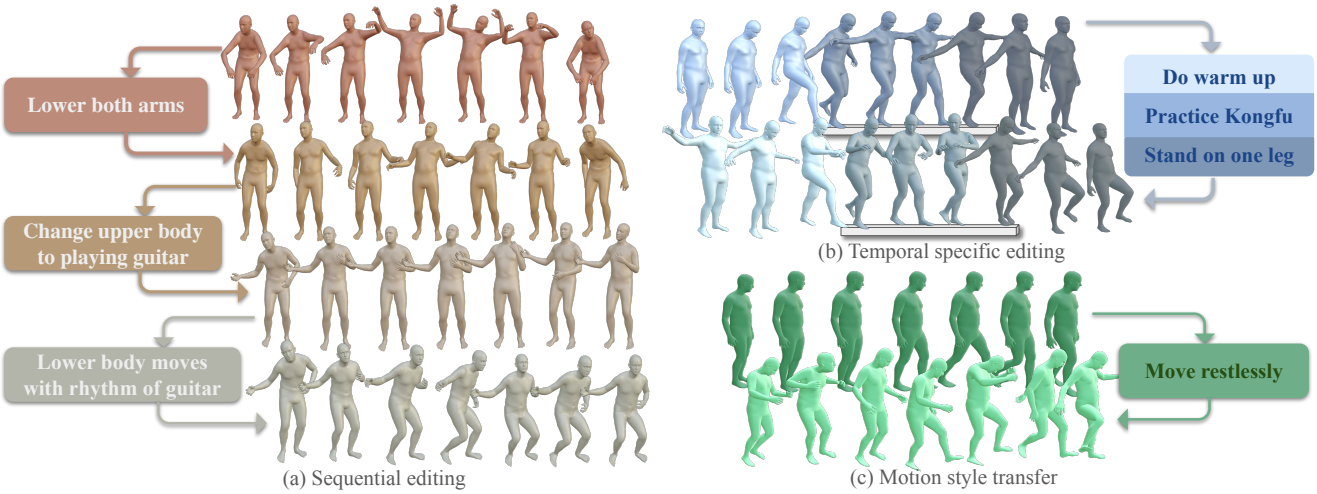
As shown in Fig. A3, our method enables both interactive editing and complex compositional motion generation, advancing beyond simple motion modifications. This capability distinguishes our approach from prior works that address only specific editing scenarios or isolated modifications.

E.1. Time-Variant Motion Editing

We enable time-variant motion editing through different text instructions. Users can independently modify distinct motion segments by applying different instructions to specific frame ranges. For instance, users can specify "raise right hand higher" for the first 25 frames, followed by "lower the right hand" for subsequent frames. This fine-grained control is implemented by iteratively calling the auto-regressive model with the first instruction until frame 25, then continuing with the second instruction from frame 25 onward.

E.2. Interactive Motion Modification

Our model supports interactive motion modification by using previously edited motions as input for subsequent processes. Users can build upon earlier edits by feeding the modified motion back into the model with new instructions. For example, after raising an arm, users can further adjust its position by applying additional modifications to the edited motion. This sequential editing process enables progressive refinement until the desired motion is achieved.



(a) Sequential editing (b) Temporal specific editing (c) Motion style transfer

Figure A3. **Compositional applications performed by our method.**

E.3. Compositional Motion Generation

Our model enables compositional motion generation through time-variant motion editing and interactive motion modifications. Starting with a base motion, users can layer multiple actions by applying sequential edits. For instance, to create a motion of simultaneously drinking water and reading, users first modify a standing pose with "drink water" followed by "reading the book using the other hand" applied to the resulting motion.

F. Limitations and Future work

While our method demonstrates strong performance across various editing tasks, it does have several notable limitations that warrant discussion. (i) Our approach shows reduced effectiveness when handling complex temporal dependencies in motion sequences, such as sequential actions (*e.g.*, a number of crouch-stand cycles). (ii) Our model struggles with instructions that require comprehension of spatial relationships (*e.g.*, return to the starting point after forward movement). (iii) While the model performs well on editing patterns similar to those in the training data, its behavior with novel or significantly different editing instructions remains unexplored.

Future work could focus on: (i) Enhancing the model's spatial-temporal understanding to better handle more complex motion sequences and editing instructions (*e.g.*, adopting motion representations from works that separately consider body parts). (ii) Incorporating physics-based constraints to ensure physical plausibility in extreme editing cases.

Figure A4. **Comparison with baselines and ablations on style transfer.**

Figure A5. **Comparison with baselines and ablations on style transfer.**

Figure A6. **Comparison with baselines and ablations on style transfer.**

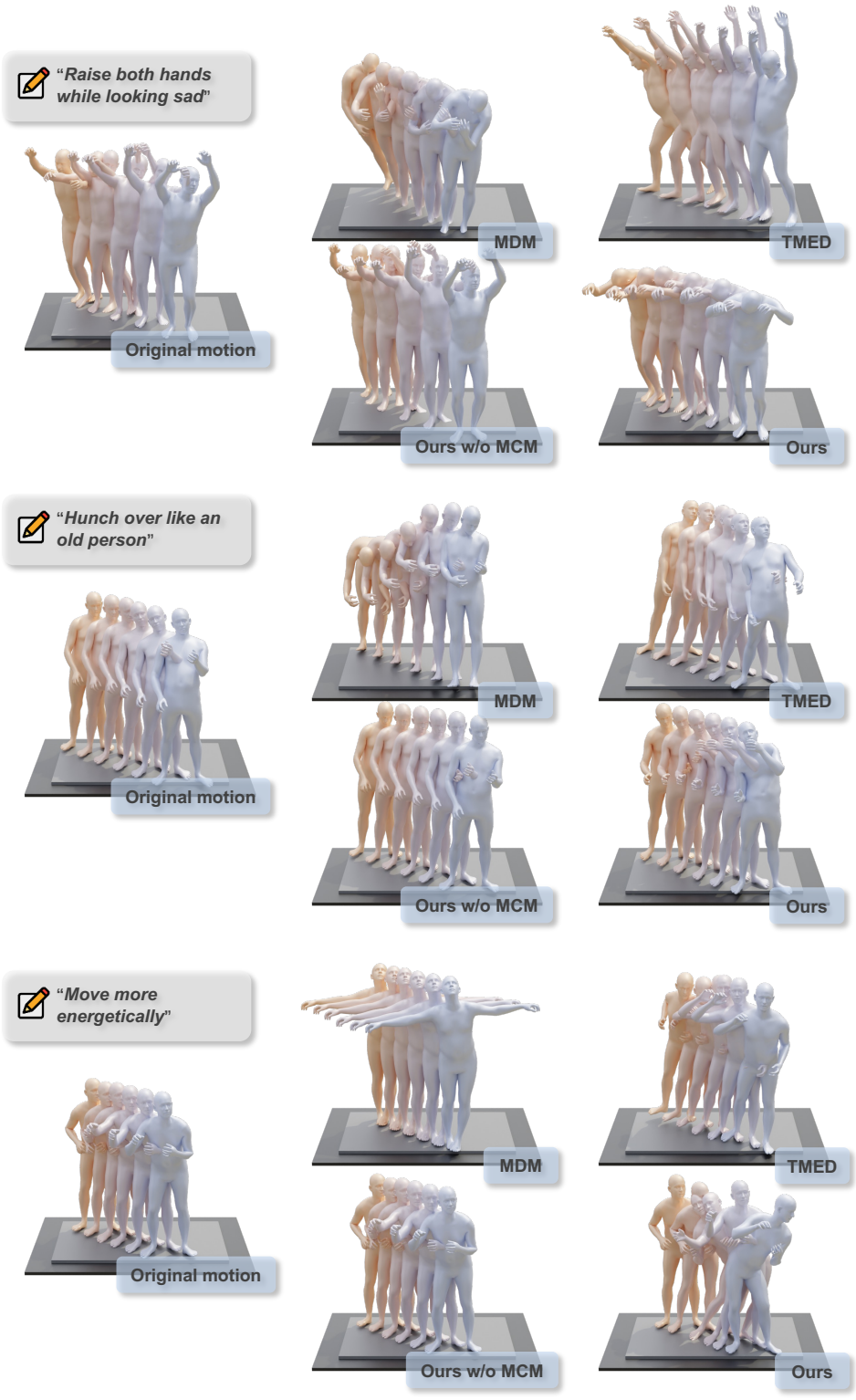


Figure A7. **Comparison with baselines and ablations on style transfer.**

Figure A8. **Comparison with baselines and ablations on body part replacement.**

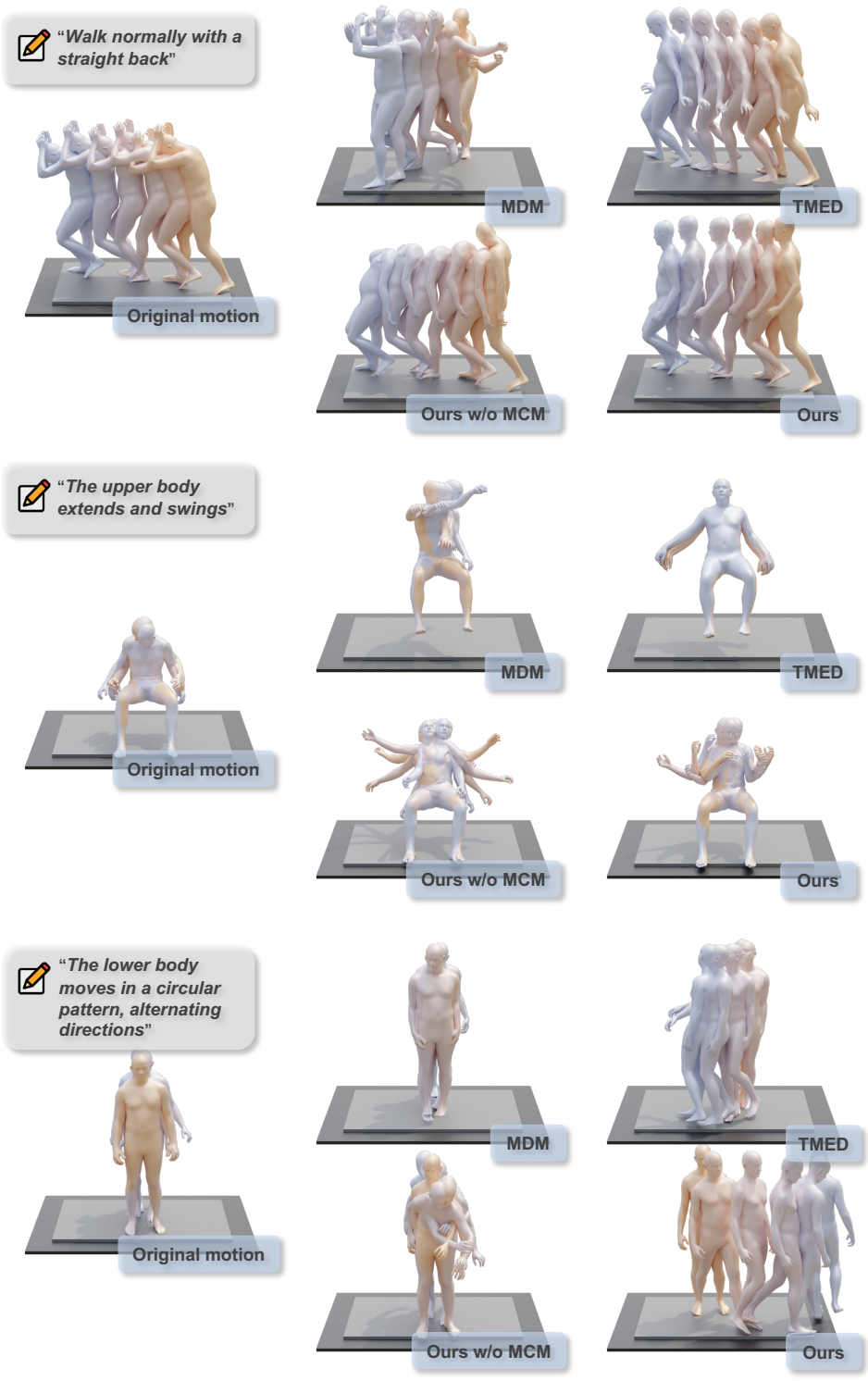


Figure A9. **Comparison with baselines and ablations on body part replacement.**

Figure A10. **Comparison with baselines and ablations on body part replacement.**

Figure A11. **Comparison with baselines and ablations on body part replacement.**

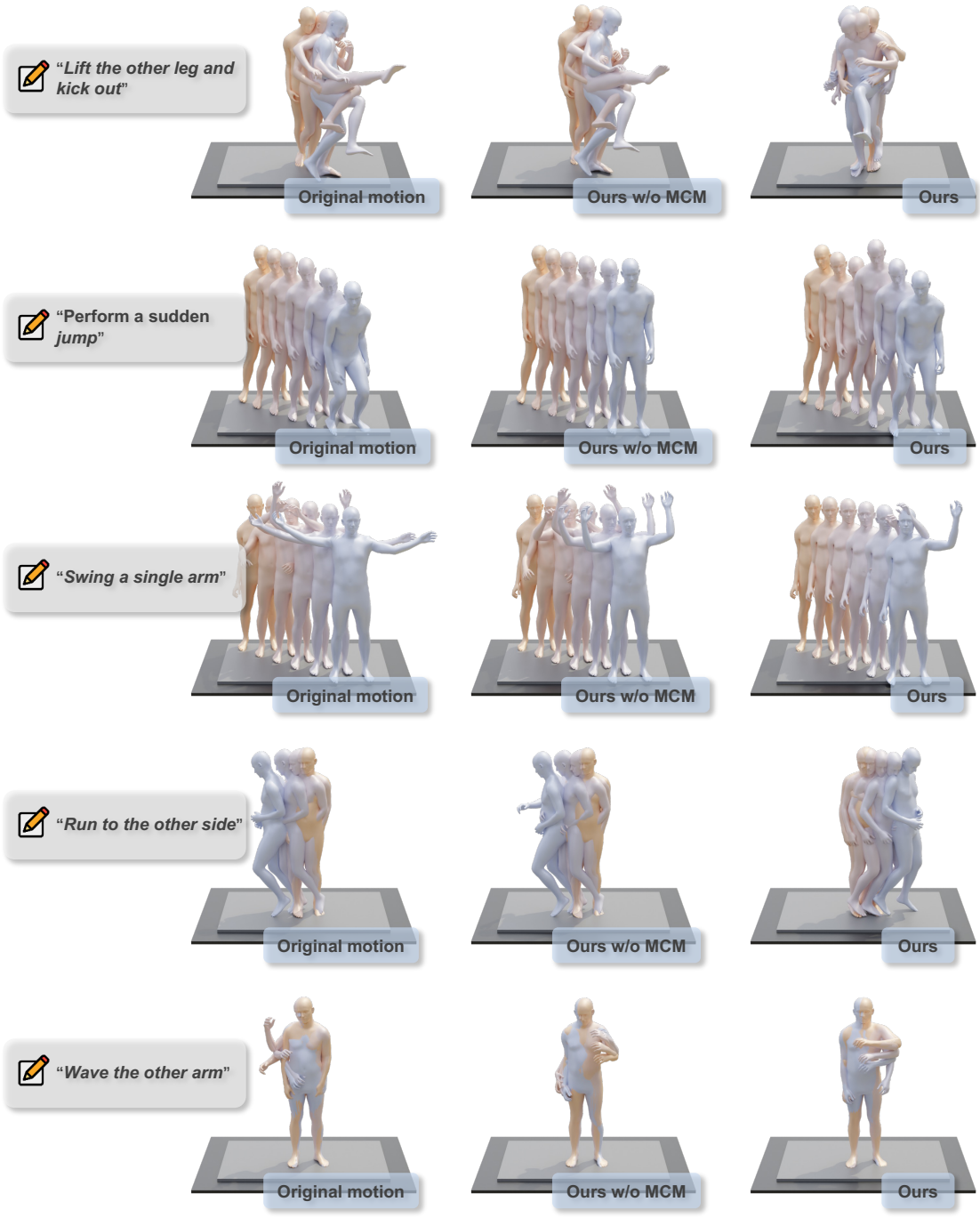


Figure A12. **Comparison with ablations on fine-grained adjustment.**

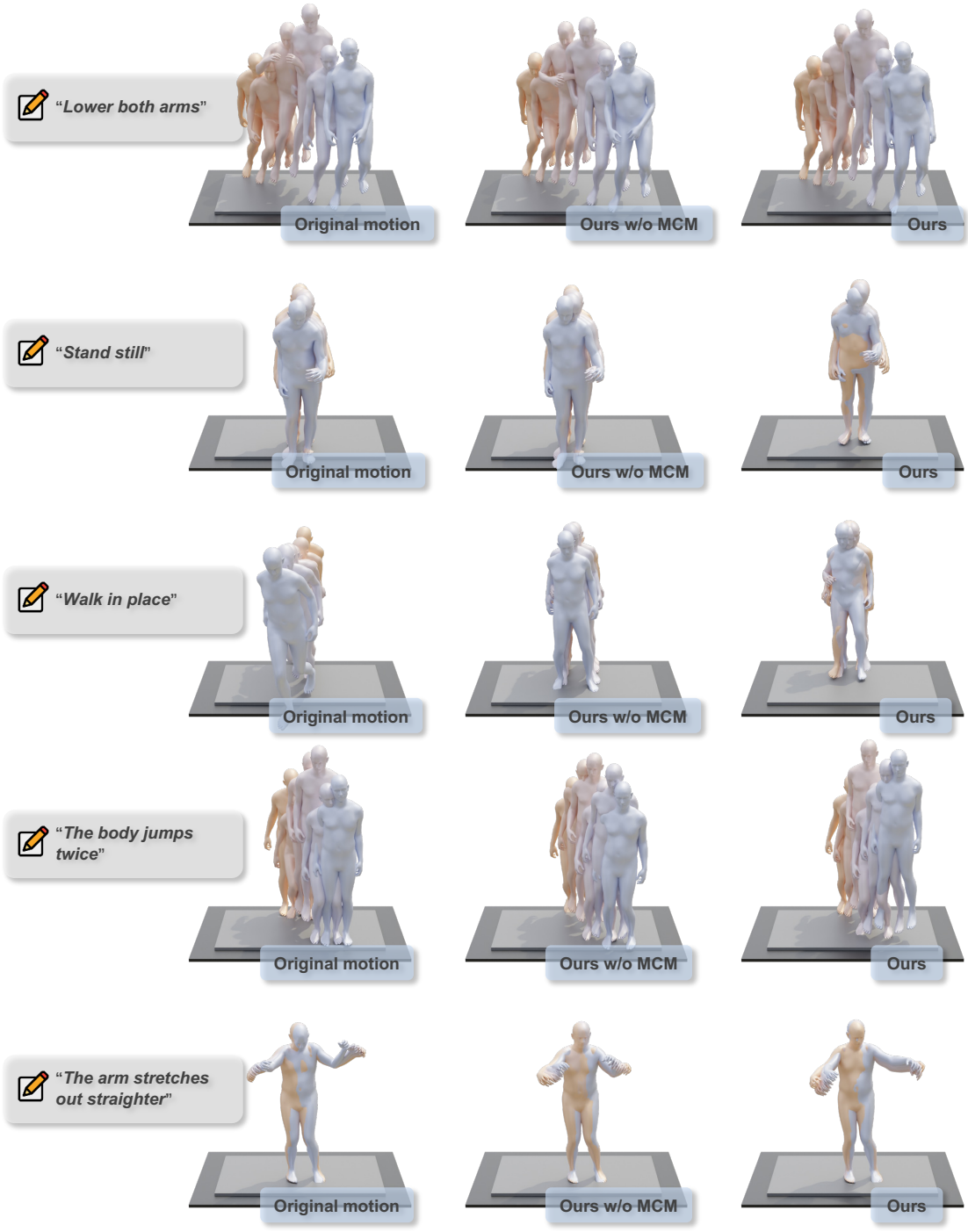


Figure A13. **Comparison with ablations on fine-grained adjustment.**

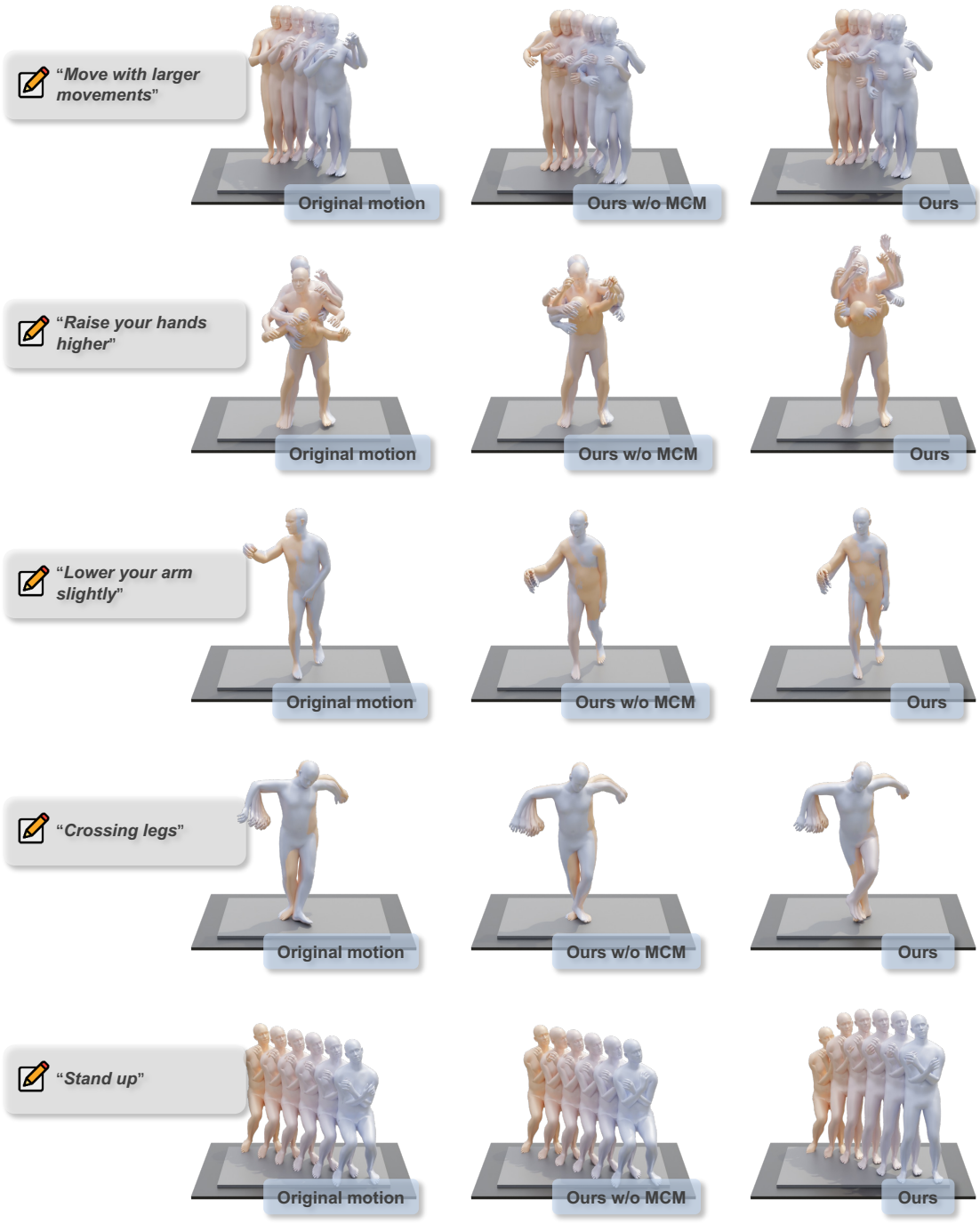


Figure A14. **Comparison with ablations on fine-grained adjustment.**