

Figure 6: Text annotation UI for OmniMotion.

In the supplementary materials, we provide the following:

- A static webpage ([index.html](#)) containing visualizations of generation results, dataset examples, comparisons, ablation studies, and failure cases.
- Representative samples from our data annotation process.
- Core algorithmic code implementation.
- Detailed documentation covering annotation procedures, evaluation protocols, prompt augmentation methods, and current limitations, in current document.

A Annotation Details

Our text annotation interface is presented in Fig. 6. We first visualize all motions using a 3D character to help annotators better understand the motion content. However, since the character may walk out of the camera view and inter-penetration artifacts sometimes occur, we also display the motions using stick-figure representations that remain centered in the camera view. These two visualizations are synchronized and presented simultaneously to annotators. Annotators can also flag low-quality motions during the annotation process.

B Evaluation Details

B.1 Baseline Implementation.

All baseline models on OmniMotion dataset leverage the T5-base model for extracting word-level features from text descriptions and are trained using a single NVIDIA RTX A6000 GPU.

For MDM [34], we use an 8-layer transformer decoder where the text encoding is injected via cross-attention layers. The model is trained for 600K steps with a batch size of 1024 using a diffusion process with $T = 1000$ steps. For T2M-GPT [37], we first learn a codebook size of 1024×512 with a downsampling rate of 4. Then, we model a sequence of codebook indices via an 18-layer transformer. During training, text embeddings and motions are concatenated and processed as input, and a random portion of the ground-truth code indices is replaced with random ones to improve robustness. The model is trained for 600K steps with a batch size of 128. For StableMoFusion [14], we use a Conv1D-based U-Net incorporating residual cross-attention to align motion features with word-level semantics, along with group normalization. The model is trained for 500K iterations with $T = 1000$ denoising steps and a batch size of 1024. For MARDM [23], we first encode motion into a latent representation using a 3-layer ResNet-based auto-encoder. These motion latents are then modeled using a masked autoregressive transformer with a dimension of 1024 and 16 attention heads, where text encodings are injected via cross-attention layers. The model is trained for 600K steps with a batch size of 128.

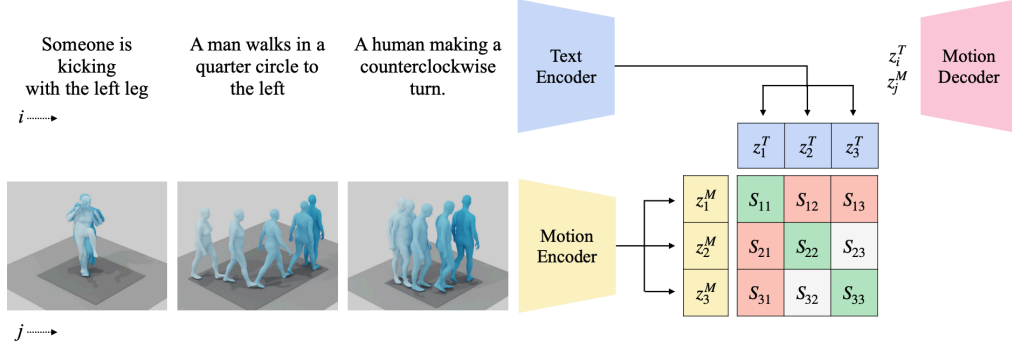


Figure 7: **Architecture of the evaluation model** [26]. Three network components are trained with two main goals: multimodal alignment and reconstruction. The cosine similarity between motion embeddings and text embeddings from positive pairs (green) is maximized, while similarity for negative pairs is minimized. Meanwhile, both embeddings are required to reconstruct the corresponding motion sequence through the motion decoder. Image adapted from TMR [26].

B.2 Evaluation Model

Our evaluation model accounts for both motion fidelity and text-motion alignment. We adopt the TMR framework, as shown in Figure 7. This framework comprises three network components: a motion encoder that encodes motion sequences into global vectors, a text encoder that encodes text sequences into global vectors, and a motion decoder that reconstructs motions from either motion or text vectors. All three networks are 6-layer transformers with a latent dimension of 256, 4 attention heads, and a feedforward hidden size of 1024. The T5-base model first extracts word-level features from texts. For motions, we use only the essential 148-dimensional root motion and local rotational features. All encoders output Gaussian distribution parameters (mean and log-variance), from which vectors are sampled. We append two temporal timesteps at the end of the sequence input for outputting these vectors.

This evaluation model is trained with a compound loss:

$$\mathcal{L}_{\text{tmr}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}} + \lambda_{\text{E}} \mathcal{L}_{\text{E}} + \lambda_{\text{NCE}} \mathcal{L}_{\text{NCE}},$$

where \mathcal{L}_{rec} measures the motion reconstruction given text or motion input (via a smooth L1 loss). A KL-divergence loss \mathcal{L}_{KL} regularizes each embedding distribution to be close to a unitary Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and also encourage these two distributions to be close to each other. \mathcal{L}_{E} enforces both mean vectors to be similar to each other. Finally, a InfoNCE [24] loss is used for constrastive learning of motion-text batches with batch size of 64. We set λ_{E} , λ_{KL} , and λ_{NCE} to $1e-5$, $1e-5$, and 0.1. For more model details, we recommend to read the original TMR work [26].

In inference, we employ the evaluation metrics designed in [10]. We increase the pool size for R-precision to 100 and directly use the mean vectors of the latent distributions as embedding vectors

C LLM-based Prompt Augmentation

Dataset Augmentation. During training, we enhance data diversity by employing an LLM to rewrite human-provided annotations, generating paraphrased versions with varied linguistic structures while preserving core semantics. This approach ensures each motion sequence is associated with multiple textual descriptions, improving model robustness. The prompt instructions for ChatGPT are provided in Tab. 5.

Inference-time Prompt Augmentation. During inference, the LLM rewrites each input prompt into a richly detailed description, incorporating explicit motion cues such as body posture, timing, and stylistic elements. This expanded form more effectively guides motion generation models. The instructions for ChatGPT are provided in Tab. 6.

Task Description The goal is to rewrite textual descriptions from a text-to-motion dataset to correct typos and grammar while rephrasing them for better readability and fluency. The key requirement is that the semantics of the described human motion must be **preserved exactly**, with no loss or modification of motion detail. You may vary the sentence structure, use synonyms, merge or split phrases, or improve flow, but **the described body movements and temporal order must remain unchanged**.

Instructions

1. Correct all spelling, grammar, and stylistic issues in the input text.
2. Rephrase the input to make it clearer, more fluent, and more readable.
3. You **must not** remove, simplify, or alter the described body movements.
4. Keep all motions, ordering, and temporal logic intact.
5. Add mild clarifications only if they help with motion clarity.

Examples

Original: The person takes two steps forward, starting with his left foot, bends down and reaches down with his right hand to the floor, rises with his arms out to the sides and steps back, and abruptly takes a step forward with his right foot with his arms bent at the elbows, and shaking both arms slowly steps forward standing in a fighting stance and provoking a fight.

Augmented: The person steps forward twice, beginning with the left foot. They bend down, reaching the floor with their right hand. Rising, they extend arms sideways and step back. Suddenly, they step forward with the right foot, elbows bent, arms shaking. They stand in a fighting stance, slowly advancing, as if provoking a fight.

Original: The person stands with their legs wide apart. Then they take two steps back and slightly to the left, lowering their head down and raising their left hand to their head. Then they lower their left hand and take two steps to the right, stopping. Then walks forward, turning to the left and waving their arms.

Augmented: The person stands with legs spread wide apart. They move two steps backward and slightly to the left while lowering their head and raising their left hand to touch it. Afterward, they lower their left hand, take two steps to the right, and pause. Then they walk forward, turning towards the left while waving their arms in a fluid motion.

Table 5: Prompt instruction for grammar-correcting and semantically-preserving text augmentation.

D Limitation

We present several representative failure motions in the static webpage. Here we discuss limitations from both data and model perspectives.

Dataset. Despite extensive calibration and post-processing of the collected motions, quality issues rooted in the inertial-based mocap suit persist. For example, global positions may lack precision, and jitters can occur during fast or complex motions. Additionally, we are unable to capture highly skilled motions such as cartwheels, backflips, or outdoor activities (e.g., climbing).

Model. Opportunities for improving text-to-motion models also remain. As MoMask++ relies on VQ, quantization errors inevitably degrade motion quality. We observe that MoMask++ struggles with rare motion patterns or uncommon text prompts. Furthermore, it does not yet maintain physical plausibility, such as proper foot contacts.

Task Description: Your task is to rewrite text prompts of user inputs for a text-to-motion generation model inference. This model generates 3D human motion data from text, you need to understand the intent of the user input and describe how the human body should move in detail, and give me the proper duration of the motion clip, usually from 4 to 12 seconds.

Instructions:

1. Make sure the rewritten prompts describe the human motion without major information loss.
2. Be related to human body movements—the tool is not able to generate anything else.
3. The rewritten prompt should be around 60 words, no more than 100.
4. Use a clear, descriptive, and precise tone.
5. Be creative and make the motion interesting and expressive.
6. Feel free to add physical movement details.

Examples:

Input: Shooting a basketball.

Rewrite: The person stands neutrally, then leans forward, spreading their legs wide. They simulate basketball dribbling with hand gestures, moving their hips side to side. The left hand performs dribbling actions. They pause, turn left, put the right leg forward, and squat slightly before simulating a basketball shot with a small jump.

Length: 8 seconds

Input: Zombie walk.

Rewrite: The person shuffles forward with a stiff, dragging motion, one foot scraping the ground as it moves. His arms hang loosely by its sides, occasionally jerking forward as it staggers with uneven steps.

Length: 6 seconds

Table 6: Instructions for re-writing casual user prompts.