MOTION-ALIGNED WORD EMBEDDINGS FOR TEXT-TO-MOTION GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing text-to-motion (T2M) generation models typically rely on pretrained large language models to encode textual inputs. However, these models, trained on generic text corpora, lack explicit alignment between motion-related words (e.g., "clockwise", "quickly") and human skeletal movements. This misalignment, fundamentally rooted in the word embedding layers, severely limits the ability of T2M models to understand and generalize fine-grained motion semantics. To tackle this issue, we propose Motion-Aligned Text Encoding (MATE), a novel framework that explicitly incorporates motion semantics into the word embedding layers of large language models to enhance text-motion alignment for motion generation. To address the challenge of inherent semantic entanglement in motion sequences, MATE introduces two key components: 1) a motion localization strategy that establishes localized correspondences between sub-texts and motion segments, enabling soft attention guidance for semantic localization; and 2) a motion disentanglement module that isolates word-specific motion semantics via contrastive kinematic prototypes, ensuring word-level alignment between linguistic and kinematic representations. Remarkably, language models enhanced with MATE can be seamlessly integrated into existing T2M methods, significantly surpassing state-of-the-art performance on two standard benchmarks with minimal modifications. Codes and pretrained models will be released upon acceptance.

1 INTRODUCTION

Text-to-motion (T2M) generation aims to synthesize sequences of human skeletal movements conditioned on textual descriptions Zhou et al. (2024a); Chi et al. (2024); Liu et al. (2024a); Fan et al. (2024); Wang et al. (2024). As a cross-modal generation task, T2M requires models to accurately translate textual descriptions into motion semantics and decode them into realistic human motions. However, existing T2M approaches still often exhibit limited cross-modal understanding. As shown in Fig. 1, while current models achieve the sentence-level alignment in example (1), they often struggle to robustly understand motion-related words such as "clockwise", leading to poor generalization and failure to generate plausible results for descriptions like "jogs in a clockwise motion" in example (2).

This limited word-level understanding largely stems from the limitations of text encoders, which process the textual inputs and directly determine the semantic information conveyed to the motion generator. While most T2M methods adopt pretrained large language models (LLMs) such as CLIP Radford et al. (2021) or DistilBERT Sanh et al. (2019) to leverage their strong textual understanding capabilities, these models are trained on general text corpora (or text-image pairs in the case of CLIP), lacking fine-grained alignment between motion-related words and human skeletal movements. In particular, the word embedding layers in LLMs fundamentally define the word semantics, which can differ substantially between linguistic and kinematic contexts. For instance, while "clockwise" functions linguistically as an adjective or adverb, in the kinematic domain it denotes a concrete rotational motion with a specific directional orientation. Without addressing such cross-modal word-level misalignment, LLMs struggle to encode motion-aware information effectively, inherently limiting the generation quality and generalization ability of T2M approaches.

To address this limitation, we propose Motion-Aligned Text Encoding (MATE), a novel framework that incorporates motion semantics into the word embedding layers of LLMs to enhance text-motion alignment for motion generation. MATE optimizes only the word embedding layers while freezing the subsequent layers, which retain strong contextual modeling abilities acquired during large-scale

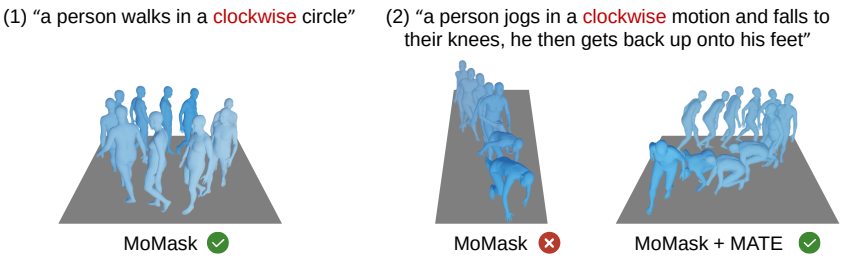


Figure 1: Examples generated by the state-of-the-art MoMask model Guo et al. (2024), with darker colors indicating motion progression. MoMask correctly produces “clockwise” motion in (1), but fails in (2), revealing limited robustness and generalization in capturing motion-related word-level semantics. Incorporating our Motion-Aligned Text Encoding (MATE) enables MoMask to produce the correct motion.

language model pretraining. We hypothesize that these higher layers can generalize effectively to motion semantics, as language and motion share structural properties, i.e., both consisting of compositional elements (e.g., words and actions) organized in temporal sequences.

However, incorporating word-specific motions semantics into word embeddings remains highly challenging and largely underexplored, primarily due to the intrinsic entanglement of motion semantics. Existing datasets typically provide only sentence-level annotations for entire motion sequences Guo et al. (2022a); Plappert et al. (2016), lacking explicit alignment between specific words and corresponding motion segments. This limitation restricts the model’s ability to temporally ground word-level semantics, particularly in sequences involving multiple compositional actions. More importantly, the semantics associated with related words are inherently intertwined, making it difficult to attribute distinct motion patterns to individual words, thereby limiting the model’s capacity for fine-grained semantic understanding.

To address these challenges, MATE introduces two key components: 1) A *motion localization* strategy that jointly decomposes paired textual descriptions and motion sequences into semantically aligned sub-units. This enables the construction of a soft attention prior that guides the temporal localization of word semantics; and 2) A *motion disentanglement* module that isolates word-specific motion semantics through two complementary mechanisms: *self-disentanglement*, which extracts shared semantics across related motions via contrastive kinematic prototypes; and *cross-disentanglement*, which enforces the exclusion of unrelated semantics, jointly ensuring semantic purity and inter-word discriminability. The disentangled motion semantics are then aligned with their corresponding word embeddings, effectively addressing the word-level misalignment inherent in LLMs.

MATE offers an resource-efficient solution for LLM fine-tuning by optimizing only word embedding layers, while maintaining broad compatibility with various LLMs. The MATE-enhanced LLMs can be seamlessly integrated into existing T2M methods with minimal architectural modifications. Extensive experiments demonstrate that MATE consistently improves text-motion alignment and generalization capability, significantly advancing the state of the art on standard benchmarks including HumanML3D Mahmood et al. (2019) and KIT Plappert et al. (2016). The main contributions of this work are summarized as follows:

1) To the best of our knowledge, MATE is the first framework to explicitly address the text-motion misalignment fundamentally rooted in the word embeddings of LLMs for motion generation. 2) We introduce a text-motion joint segmentation strategy that automatically establishes correspondences between sub-texts and motion segments, enabling action-level semantic localization for paired text-motion data. 3) We propose a motion disentanglement module that achieves word-level semantic disentanglement, mitigating the challenge of semantic entanglement in motion sequences. 4) Extensive experiments demonstrate that MATE-enhanced language models can be seamlessly integrated into existing T2M pipelines, yielding substantial performance improvements and significantly surpassing state-of-the-art results across two standard benchmarks.

2 Related Works

Text-to-Motion Generation typically involves two stages: text encoding and motion synthesis Shafir et al. (2024); Xie et al. (2024); Liu et al. (2023); Liang et al. (2024). Textual descriptions are first projected into a latent feature space and subsequently translated into motion sequences.

Most existing methods keep the text encoder frozen and primarily focus on improving the motion generation process, leveraging advanced architectures such as diffusion models Tevet et al. (2023); Zhang et al. (2023b); Jin et al. (2024a); Ren et al. (2023); Wang et al. (2023) and quantized variational autoencoders Chen et al. (2023); Zhang et al. (2023a); Van Den Oord et al. (2017); Dai et al. (2024). Another line of research seeks to enhance text encoding by constructing hierarchical semantic graphs over text embeddings to better capture fine-grained motion semantics Wang et al. (2023); Jin et al. (2024b). However, these methods typically rely on off-the-shelf text embeddings from pretrained large language models (LLMs), overlooking the inherent semantic gap between linguistic and kinematic representations caused by LLM itself. In contrast, our method introduces motion-aligned fine-tuning of LLMs, facilitating more accurate and robust modeling of motion semantics from textual inputs.

Text-Motion Retrieval aims to retrieve the most relevant motion given a text query, or vice versa. Existing approaches often focus on enhancing text-motion alignment through contrastive representation learning Yan et al. (2023); Yin et al. (2024); Guo et al. (2022b) or probabilistic divergence objectives such as KL divergence Petrovich et al. (2022). These methods usually utilize pretrained LLMs Tevet et al. (2022) or adapter-based enhancements Petrovich et al. (2022; 2023); Lu et al. (2024) to construct the joint embedding space, but the underlying cross-modal misalignment originating from pretrained language representations remains largely unresolved. Our work addresses this core limitation by improving motion semantic alignment within language models.

Large Language Model Fine-Tuning has emerged as a powerful paradigm for adapting general-purpose textual representations to downstream tasks such as domain adaptation Ding et al. (2023); Susnjak et al. (2025); Wei et al. (2023), image generation Li et al. (2024); Liu et al. (2024b); Ruiz et al. (2023), and video generation Rasheed et al. (2023); Wu et al. (2023). In the context of T2M, recent works LMM Zhang et al. (2024a), MotionGPT Jiang et al. (2023), AvatarGPT Zhou et al. (2024b), Motion-Agent Wu et al. (2025) unify multiple text-motion tasks within a single framework, often incorporating expanded token vocabularies and instruction tuning techniques Ouyang et al. (2022). Our method differs in two key aspects. First, rather than pretraining the entire language model, which typically demands substantial computational resources and large-scale integrated datasets, we focus specifically on the compact word embedding layers, offering a more data- and resource-efficient solution. Second, instead of introducing a standalone framework, our approach produces plug-and-play motion-aligned text encoders that can be directly integrated into various T2M models, yielding substantial performance improvements without changing their architectures.

3 METHOD

3.1 OVERVIEW

Word embedding layers, typically placed at the input of LLMs, function as a lexical lookup table that maps discrete word tokens to continuous vectors for contextual modeling. They play a crucial role in encoding word meanings and inter-word relationships. However, word-level semantics in linguistic domain often differ fundamentally from those in the kinematic domain. For example, "clockwise" and "anti-clockwise" are linguistically similar, due to sharing morphological structure and grammatical function. In contrast, they kinematically denote opposite directions of rotation and are incompatible, thus leading to significant cross-modality discrepancies.

To bridge this gap, unlike existing T2M methods that use pretrained LLMs without adaptation, this work introduces Motion-Aligned Text Encoding (MATE), a novel approach for explicitly aligning the word embedding layers in LLMs with word-specific motion semantics.

Specifically, we formulate the learning objective as follows: given a triplet $\{\boldsymbol{t}, \boldsymbol{m}, \boldsymbol{w}\}$ sampled from the training set, where $\boldsymbol{t}$ denotes a textual description, $\boldsymbol{m}$ is the corresponding motion sequence, and $\boldsymbol{w}$ is a word token sampled from $\boldsymbol{t}$, the goal is to align the textual semantics of $\boldsymbol{w}$ with its associated motion semantics expressed in $\boldsymbol{m}$ by optimizing its word embeddings in language models.

To achieve this alignment, as illustrated in Fig. 2, MATE comprises two key components: 1) Motion localization, which establishes correspondences between temporally aligned subtexts and motion segments for word semantic localization; 2) Motion disentanglement, which disentangles motion features that are semantically attributable to individual words for alignment with their corresponding word embeddings.

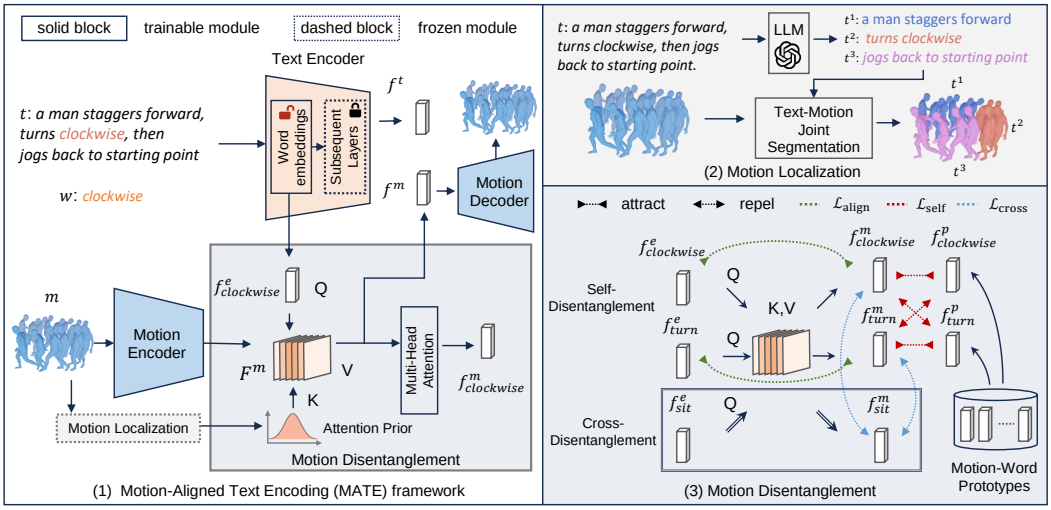


Figure 2: (1) Overview of the MATE framework, which comprises a text encoder (with trainable word embedding layers), a motion encoder, and a motion decoder, together with: (2) Motion localization, which establishes temporal text-motion correspondences to construct a Gaussian-shaped attention prior for guiding word-level semantic localization; and (3) Motion disentanglement, which employs multi-head attention to disentangle motion semantics of specific words for semantic alignment.

3.2 Text-Guided Motion Localization

A motion sequence is a complex integration of multiple word-level semantics, making fine-grained semantic alignment particularly challenging, especially in long sentences involving multiple words and actions. To tackle this issue, we adopt a coarse-to-fine semantic extraction strategy, with the first sub-goal of localizing the motion semantics corresponding to an individual word $\boldsymbol{w}$ within the motion sequence $\boldsymbol{m}$. However, most existing motion datasets Guo et al. (2022a); Plappert et al. (2016) provide only sentence-level annotations for entire sequences, lacking explicit supervision for word-level localization. To overcome this limitation, we propose a text-motion joint segmentation pipeline that automatically establishes correspondences between each sub-action described in the text and its counterpart segment in the motion sequence, as illustrated in Fig. 2 (2).

Specifically, we employ ChatGPT Roumeliotis & Tselikas (2023) to decompose each textual description $\boldsymbol{t}$ into a set of sub-texts $\boldsymbol{t}^1, \cdots, \boldsymbol{t}^N$ ($N \geq 1$), where each sub-text describes one or more temporally coherent actions. We then seek to segment the motion sequence $\boldsymbol{m}$ into N non-overlapping clips, each aligned with its corresponding sub-text. To this end, we formulate an optimal partitioning problem, where segment boundaries are adjusted to minimize the matching loss between each sub-text and its corresponding motion segment. Given a sentence decomposed into N sub-texts, the objective is defined as:

$$\min_{\{s_n, e_n\}} \sum_{n=1}^{N} 1 - \cos\left(\mathcal{E}_t(\boldsymbol{t}^n),\ \mathcal{E}_m(\boldsymbol{m}[s_n : e_n])\right), \tag{1}$$

where s_n and e_n denote the start and end frames of the n-th segment, constrained by $s_{n+1} = e_n$. Here, $\mathcal{E}_t$ and $\mathcal{E}_m$ are frozen text and motion encoders from a pretrained text-to-motion retrieval model Lu et al. (2024), and $\cos(\cdot, \cdot)$ denotes the cosine similarity between encoded text and motion features. An exhaustive search over all valid partitions is performed to identify the boundaries that best align motion segments with their respective sub-texts.

The obtained segmentation is not directly used as ground-truth localization, but instead serves as a soft prior to guide the discovery of the semantics of the word $\boldsymbol{w}$ within the motion sequence $\boldsymbol{m}$. Specifically, a motion encoder simultaneously extracts a sequence-level representation $\boldsymbol{f}^m \in \mathbb{R}^D$ and frame-level features $\boldsymbol{F}^m \in \mathbb{R}^{T \times D}$ from the motion sequence $\boldsymbol{m}$, where T denotes the number of frames and D is the feature dimension. We introduce a multi-head attention mechanism, in which the word embeddings serve as a query to explicitly attend to relevant motion features, formulated as:

$$\boldsymbol{f}^m_{\text{word}} = \text{MultiHead}(Q, K, V), \tag{2}$$

$$Q = \mathrm{Proj}(\mathrm{WE}(\boldsymbol{w})), \quad K = (1 + \lambda \cdot \mathrm{AttentionPrior}(t)) \odot \boldsymbol{F}^m, \quad V = \boldsymbol{F}^m. \tag{3}$$

Here, $\mathrm{WE}(\boldsymbol{w})$ is the trainable word embeddings of $\boldsymbol{w}$ from the text encoder, Proj is a linear projection layer, and $\mathrm{AttentionPrior}(t)$ is a temporal attention prior derived from segmentation information:

$$\mathrm{AttentionPrior}(t) = \exp(-\frac{(t - c_n)^2}{2\sigma_n^2}), \quad \text{with} \quad c_n = \frac{s_n + e_n}{2}, \quad \sigma_n = \frac{e_n - s_n}{2}. \tag{4}$$

where t is the frame index, and s_n, e_n are the start and end frames in the localized segment of $\boldsymbol{w}$, respectively. The Gaussian-shaped $\mathrm{AttentionPrior}(t)$ softly highlights frames near the center of the localized segment while smoothly attenuating distant frames, thereby improving robustness against localization errors introduced by the segmentation process.

3.3 Word-Guided Motion Disentanglement

While the above approach enables localization of word-level semantics, the semantics of related words remain highly intertwined. For instance, as shown in Fig. 2 (1) and (2), although the segment corresponding to "turn clockwise" can be identified, it remains challenging for the model to accurately distinguish between the semantics of "turn" and "clockwise", thereby hindering the precise understanding of individual words. To address this limitation, we propose a word-guided motion disentanglement approach that explicitly isolates motion semantics attributable to individual word units. Toward this goal, we introduce the following three criteria for effective motion disentanglement.

1) **Stability**: A given word query should consistently attend to *shared* motion features across different motions that exhibit the corresponding semantics (e.g., the features disentangled by the word "clockwise" from motions of "turn clockwise" and "jog in a clockwise circle" should remain similar.

2) **Discriminability**: Different word queries should result in semantically *distinct* motion features (e.g., the features disentangled by "turn" should be distinguishable from those by "clockwise").

3) **Rationality**: Disentanglement should yield meaningful features only when the motion sequence *contains* semantics associated with the queried word (e.g., querying "clockwise" from the motion of "walk forward" should not produce a meaningful feature representation).

To satisfy Criteria 1 and 2, we introduce a self-disentanglement mechanism based on prototype representations. Specifically, we predefine a set of motion-word prototypes consisting of K learnable vectors $\{\boldsymbol{f}_{w_k}^p\}_{k=1}^K$, each representing the motion semantics associated with a specific word $\boldsymbol{w}_k$. Suppose $\{\boldsymbol{t}_i, \boldsymbol{m}_i, \boldsymbol{w}_i\}$ the i-th triplet sample in a mini-batch, the disentangled motion features $\boldsymbol{f}_{w_i}^{m_i}$ satisfy the self-disentanglement loss $\mathcal{L}_{\text{self}}$:

$$\mathcal{L}_{\text{self}} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} -\log \frac{\exp(\cos(\boldsymbol{f}_{w_i}^{m_i}, \boldsymbol{f}_{w_i}^p)/\tau)}{\sum_{k=1}^K \exp(\cos(\boldsymbol{f}_{w_i}^{m_i}, \boldsymbol{f}_{w_k}^p)/\tau)}, \tag{5}$$

where $\mathcal{V}$ is the set of samples in a mini-batch. The loss $\mathcal{L}_{\text{self}}$ encourages $\boldsymbol{f}_{w_i}^{m_i}$ to be pulled closer to its corresponding prototype $\boldsymbol{f}_{w_i}^p$ while being pushed away from all other prototypes. Simultaneously, each prototype is optimized towards a stable and shared semantic representation across motions that express the semantics of $\boldsymbol{w}_i$. In contrast to conventional contrastive losses Radford et al. (2021); Oord et al. (2018) that operate at the batch level, the introduction of prototypes enables contrastive learning over dataset-wide word semantics, thereby enhancing the stability and discriminability of motion disentanglement.

To satisfy Criterion 3, we further formulate a cross-disentanglement mechanism: the motion $\boldsymbol{m}_i$ is also queried by the word token $\boldsymbol{w}_j$ from the j-th sample, and if $\boldsymbol{m}_i$ does not contain the semantics of $\boldsymbol{w}_j$, the model is encouraged to produce motion features that are orthogonal to the reasonably disentangled features $\boldsymbol{f}_{w_i}^{m_i}$. The cross-disentanglement loss $\mathcal{L}_{\text{cross}}$ is defined as

$$\mathcal{L}_{\text{cross}} = \frac{1}{|\mathcal{N}|} \sum_{(i,j) \in \mathcal{N}} \left|\cos(\boldsymbol{f}_{w_i}^{m_i}, \boldsymbol{f}_{w_j}^{m_i})\right| + \left|\cos(\boldsymbol{f}_{w_i}^{m_i}, \boldsymbol{f}_{w_i}^{m_j})\right|,$$

where $\mathcal{N}$ is the set of negative pairs, and $\boldsymbol{f}_{w_j}^{m_i}$ denotes the disentangled motion features from the i-th sequence $\boldsymbol{m}_i$ with the word query $\boldsymbol{w}_j$ from the j-th sample. $\mathcal{L}_{\text{cross}}$ drives the model to discriminate whether the queried word semantics is expressed in the motion sequence to ensure the accuracy of motion disentanglement.

Table 1: Results on HumanML3D Mahmood et al. (2019). "↑", "↓" and "→" indicate that higher values, lower values, or values closer to real motion are better, respectively. Red and blue highlight the top two results.

Methods	R-Precision ↑			FID ↓	MM-Dist ↓	Diversity →	MModality ↑
	Top-1	Top-2	Top-3				
Real motions	$0.511^{\pm0.003}$	$0.703^{\pm0.003}$	$0.797^{\pm0.003}$	$0.002^{\pm0.000}$	$2.974^{\pm0.008}$	$9.503^{\pm0.065}$	-
GraphMotion Jin et al. (2024b)	$0.504^{\pm0.003}$	$0.699^{\pm0.002}$	$0.785^{\pm0.002}$	$0.116^{\pm0.004}$	$3.070^{\pm0.008}$	$9.692^{\pm0.067}$	$2.766^{\pm0.096}$
Motion Mamba Zhang et al. (2024c)	$0.502^{\pm0.003}$	$0.693^{\pm0.002}$	$0.792^{\pm0.002}$	$0.281^{\pm0.009}$	$3.060^{\pm0.058}$	$9.871^{\pm0.084}$	$2.294^{\pm0.058}$
ParCo Zou et al. (2024)	$0.515^{\pm0.003}$	$0.706^{\pm0.003}$	$0.801^{\pm0.002}$	$0.109^{\pm0.005}$	$2.927^{\pm0.008}$	$9.576^{\pm0.088}$	$1.382^{\pm0.060}$
CoMo Huang et al. (2024)	$0.502^{\pm0.002}$	$0.692^{\pm0.007}$	$0.790^{\pm0.002}$	$0.262^{\pm0.004}$	$3.032^{\pm0.015}$	$9.936^{\pm0.066}$	$1.013^{\pm0.046}$
BAMM Pinyoanuntapong et al. (2024a)	$0.522^{\pm0.003}$	$0.715^{\pm0.003}$	$0.808^{\pm0.003}$	$0.055^{\pm0.002}$	$2.936^{\pm0.077}$	$9.636^{\pm0.009}$	$1.732^{\pm0.055}$
MDM Tevet et al. (2023)	$0.320^{\pm0.005}$	$0.498^{\pm0.004}$	$0.611^{\pm0.007}$	$0.544^{\pm0.044}$	$5.566^{\pm0.027}$	$9.559^{\pm0.086}$	$2.799^{\pm0.072}$
+MATE (ours)	$0.509^{\pm0.002}$	$0.698^{\pm0.002}$	$0.797^{\pm0.003}$	$0.332^{\pm0.002}$	$3.057^{\pm0.063}$	$9.468^{\pm0.053}$	$2.773^{\pm0.062}$
MotionDiffuse Zhang et al. (2022)	$0.491^{\pm0.001}$	$0.681^{\pm0.001}$	$0.782^{\pm0.001}$	$0.630^{\pm0.001}$	$3.113^{\pm0.001}$	$9.410^{\pm0.049}$	$1.553^{\pm0.042}$
+MATE (ours)	$0.536^{\pm0.001}$	$0.721^{\pm0.001}$	$0.821^{\pm0.001}$	$0.234^{\pm0.002}$	$2.907^{\pm0.002}$	$9.446^{\pm0.081}$	$1.703^{\pm0.055}$
MMM Pinyoanuntapong et al. (2024b)	$0.515^{\pm0.002}$	$0.708^{\pm0.002}$	$0.804^{\pm0.002}$	$0.089^{\pm0.005}$	$2.926^{\pm0.007}$	$9.577^{\pm0.050}$	$1.226^{\pm0.035}$
+ MATE (ours)	$0.541^{\pm0.001}$	$0.729^{\pm0.003}$	$0.820^{\pm0.002}$	$0.069^{\pm0.003}$	$2.887^{\pm0.017}$	$9.562^{\pm0.088}$	$1.469^{\pm0.057}$
MoMask Guo et al. (2024)	$0.521^{\pm0.002}$	$0.713^{\pm0.002}$	$0.807^{\pm0.002}$	$0.045^{\pm0.002}$	$2.958^{\pm0.008}$	$9.632^{\pm0.072}$	$1.241^{\pm0.040}$
+ MATE (ours)	$0.550^{\pm0.002}$	$0.737^{\pm0.002}$	$0.832^{\pm0.002}$	$0.040^{\pm0.002}$	$2.811^{\pm0.007}$	$9.516^{\pm0.092}$	$1.369^{\pm0.036}$

3.4 Motion-Aligned Word Embedding

To align the disentangled motion features with corresponding word embeddings, we formulate an alignment loss $\mathcal{L}_{\text{align}}$ as

$$\mathcal{L}_{\text{align}} = \frac{1}{|2\mathcal{V}|}\sum_{i\in\mathcal{V}}(-\log\frac{\exp(\cos(\boldsymbol{f}^{e}_{w_i}, \boldsymbol{f}^{m_i}_{w_i})/\tau)}{\sum_{j\in\mathcal{V}}\exp(\cos(\boldsymbol{f}^{e}_{w_i}, \boldsymbol{f}^{m_j}_{w_j})/\tau)} - \log\frac{\exp(\cos(\boldsymbol{f}^{m_i}_{w_i}, \boldsymbol{f}^{e}_{w_i})/\tau)}{\sum_{j\in\mathcal{V}}\exp(\cos(\boldsymbol{f}^{m_i}_{w_i}, \boldsymbol{f}^{e}_{w_j})/\tau)}), \quad (6)$$

where $\boldsymbol{f}^{e}_{w_i} = \text{Proj}(\text{WE}(\boldsymbol{w}_i))$ is the projected word embeddings, consistent with the query in Eq (3). This loss adopts a symmetric InfoNCE formulation Oord et al. (2018) to encourage alignment between paired word embeddings and their motion semantics while simultaneously promoting separation between mismatched pairs. The motion-aligned word embedding loss is summarized as

$$\mathcal{L}_{\text{word}} = \mathcal{L}_{\text{self}} + \mathcal{L}_{\text{cross}} + \mathcal{L}_{\text{align}}. \quad (7)$$

However, the above approach primarily focuses on individual word-level alignment and overlooks the contextual dependencies among words. To tackle this issue, we further introduce a sentence-level alignment objective, which aligns the text feature vector $\boldsymbol{f}^t$ extracted from $\boldsymbol{t}$ with the corresponding motion feature vector $\boldsymbol{f}^m$ from $\boldsymbol{m}$ using an InfoNCE loss $\mathcal{L}_{\text{sent}}$. Additionally, $\boldsymbol{f}^m$ is passed through a motion decoder to reconstruct the original motion sequence $\boldsymbol{m}$, guided by a reconstruction loss $\mathcal{L}_{\text{rec}}$ to preserve detailed motion information. The overall training objective is:

$$\mathcal{L}_{\text{all}} = \mathcal{L}_{\text{rec}} + \lambda_1 \cdot \mathcal{L}_{\text{word}} + \lambda_2 \cdot \mathcal{L}_{\text{sent}}, \quad (8)$$

where λ_1 and λ_2 are weighting factors.

4 Experiments

4.1 Experiment Settings

Dataset. We conduct experiments on two standard human motion datasets: **HumanML3D** Mahmood et al. (2019) and **KIT** Plappert et al. (2016). HumanML3D contains 14,616 motion sequences, annotated with 44,970 textual descriptions, while KIT includes 3,911 motion sequences paired with 6,278 text descriptions.

Evaluation Protocols. Following the standard protocol Guo et al. (2022a), we adopt five evaluation metrics: **R-Precision** and **Multimodal Distance (MMDist)** measure how accurately generated motions match the text. **Frechet Inception Distance (FID)** evaluates the distributional similarity between generated and real motion features. **Diversity** computes the average Euclidean distance across 300 randomly sampled pairs of generated motions. **MultiModality (MModality)** reflects the variation of generated motions, calculated as the average distance among 10 motions generated from the same text.

Table 2: Results on KIT Plappert et al. (2016), using the same notations as in Table 1.

Methods	R-Precision ↑			FID ↓	MM-Dist ↓	Diversity →	MModality ↑
	Top-1	Top-2	Top-3				
Real motions	$0.424^{\pm0.005}$	$0.649^{\pm0.006}$	$0.779^{\pm0.006}$	$0.031^{\pm0.004}$	$2.788^{\pm0.012}$	$11.08^{\pm0.097}$	-
GraphMotion Jin et al. (2024b)	$0.429^{\pm0.007}$	$0.648^{\pm0.006}$	$0.769^{\pm0.006}$	$0.313^{\pm0.013}$	$3.076^{\pm0.022}$	$11.12^{\pm0.135}$	$3.627^{\pm0.113}$
Motion Mamba Zhang et al. (2024c)	$0.419^{\pm0.006}$	$0.645^{\pm0.005}$	$0.765^{\pm0.006}$	$0.307^{\pm0.041}$	$3.021^{\pm0.025}$	$11.02^{\pm0.098}$	$1.678^{\pm0.064}$
ParCo Zou et al. (2024)	$0.430^{\pm0.004}$	$0.649^{\pm0.007}$	$0.772^{\pm0.006}$	$0.453^{\pm0.027}$	$2.820^{\pm0.028}$	$10.95^{\pm0.094}$	$1.245^{\pm0.022}$
CoMo Huang et al. (2024)	$0.422^{\pm0.009}$	$0.638^{\pm0.007}$	$0.765^{\pm0.011}$	$0.332^{\pm0.009}$	$2.873^{\pm0.021}$	$10.95^{\pm0.196}$	$1.249^{\pm0.008}$
BAMM Pinyoanuntapong et al. (2024a)	$0.436^{\pm0.007}$	$0.660^{\pm0.006}$	$0.791^{\pm0.005}$	$0.200^{\pm0.011}$	$2.714^{\pm0.016}$	$10.91^{\pm0.097}$	$1.517^{\pm0.058}$
MDM Tevet et al. (2023)	$0.164^{\pm0.004}$	$0.291^{\pm0.004}$	$0.396^{\pm0.004}$	$0.497^{\pm0.021}$	$9.191^{\pm0.022}$	$10.85^{\pm0.109}$	$1.907^{\pm0.214}$
+ MATE (ours)	$0.407^{\pm0.006}$	$0.608^{\pm0.005}$	$0.723^{\pm0.007}$	$0.297^{\pm0.026}$	$2.978^{\pm0.046}$	$10.93^{\pm0.112}$	$1.988^{\pm0.194}$
MotionDiffuse Zhang et al. (2022)	$0.417^{\pm0.004}$	$0.621^{\pm0.004}$	$0.739^{\pm0.004}$	$1.954^{\pm0.062}$	$2.958^{\pm0.005}$	$11.10^{\pm0.143}$	$0.730^{\pm0.013}$
+ MATE (ours)	$0.432^{\pm0.005}$	$0.644^{\pm0.004}$	$0.763^{\pm0.005}$	$0.965^{\pm0.077}$	$2.852^{\pm0.005}$	$11.12^{\pm0.104}$	$1.204^{\pm0.013}$
MMM Pinyoanuntapong et al. (2024b)	$0.404^{\pm0.005}$	$0.621^{\pm0.005}$	$0.744^{\pm0.004}$	$0.316^{\pm0.028}$	$2.977^{\pm0.019}$	$10.91^{\pm0.101}$	$1.232^{\pm0.039}$
+ MATE (ours)	$0.422^{\pm0.008}$	$0.642^{\pm0.004}$	$0.770^{\pm0.007}$	$0.253^{\pm0.017}$	$2.815^{\pm0.026}$	$10.38^{\pm0.101}$	$1.533^{\pm0.044}$
MoMask Guo et al. (2024)	$0.433^{\pm0.007}$	$0.656^{\pm0.005}$	$0.781^{\pm0.005}$	$0.204^{\pm0.011}$	$2.779^{\pm0.022}$	$10.88^{\pm0.099}$	$1.131^{\pm0.043}$
+ MATE (ours)	$0.443^{\pm0.006}$	$0.669^{\pm0.005}$	$0.798^{\pm0.007}$	$0.197^{\pm0.015}$	$2.732^{\pm0.014}$	$10.96^{\pm0.098}$	$1.683^{\pm0.041}$

Implementation Details. The text encoder in the MATE framework could be various large language models, such as CLIP Radford et al. (2021), DistilBERT Sanh et al. (2019), etc. The number of word prototypes (K) is automatically decided by the number of words included in the training set. Specifically, we perform lemmatization on textual descriptions and summarize the vocabulary, establishing K= 5,161 and 1,191 prototypes for HumanML3D and KIT, respectively. These prototypes are initialized with their corresponding word embeddings in the pretrained language models.

After training MATE, we integrate the MATE-enhanced text encoder into existing T2M models in place of their original encoders, and retrain the models from scratch. The training and inference procedures strictly follow the official implementations of the T2M models, without any modifications. To ensure statistical reliability, we perform 20 rounds of inference and report the averaged results. Unless otherwise stated, we use MoMask Guo et al. (2024) with the MATE-enhanced CLIP as the default T2M model for evaluation. Additional details are provided in the supplementary materials.

4.2 Comparison with State-of-the-Art Methods

Quantitative Comparison. Table 1 and Table 2 present the performance of state-of-the-art (SOTA) T2M models. The "+MATE" variants of MDM Tevet et al. (2023), MotionDiffuse Zhang et al. (2022), MMM Pinyoanuntapong et al. (2024b) and MoMask Guo et al. (2024) are obtained by retraining the original models with the pretrained CLIP text encoders replaced by MATE-enhanced CLIP, where only the word embedding layers are updated. Despite this minimal modification, our approach consistently yields substantial improvements across all evaluation metrics on both benchmarks. Notably, while the performance gains on HumanML3D are significant, the improvements on KIT are relatively modest due to the smaller dataset size, which constrains the optimization of word embeddings in large language models.

Visualization Comparison. Fig. 3 compares motion sequences generated by different SOTA methods. MATE accurately distinguishes fine-grained semantics, such as "kick one time with the right leg" and "three times with the left leg" in the upper example, and faithfully captures key textual descriptions like "counterclockwise circle" and "yawn" in the lower example, demonstrating the superiority of our approach in fine-grained motion semantic understanding and text-motion alignment.

4.3 Ablation Study

Loss Functions. Table 3 presents the ablation study of the loss functions. Removing $\mathcal{L}_{\text{self}}$, $\mathcal{L}_{\text{cross}}$, or $\mathcal{L}_{\text{align}}$ leads to varying degrees of performance degradation, highlighting their complementary and essential contributions. Specifically, $\mathcal{L}_{\text{self}}$ and $\mathcal{L}_{\text{cross}}$ promote the discriminability and effectiveness of word-level disentangled motion semantics, while $\mathcal{L}_{\text{align}}$ is critical for integrating these semantics into the word embeddings. Additionally, $\mathcal{L}_{\text{sent}}$ plays an important role by enforcing alignment between entire sentences and motion sequences, thereby modeling contextual dependencies across words. Removing $\mathcal{L}_{\text{rec}}$ also results in slight performance degradation, as motion reconstruction can enhance the informativeness of motion features.

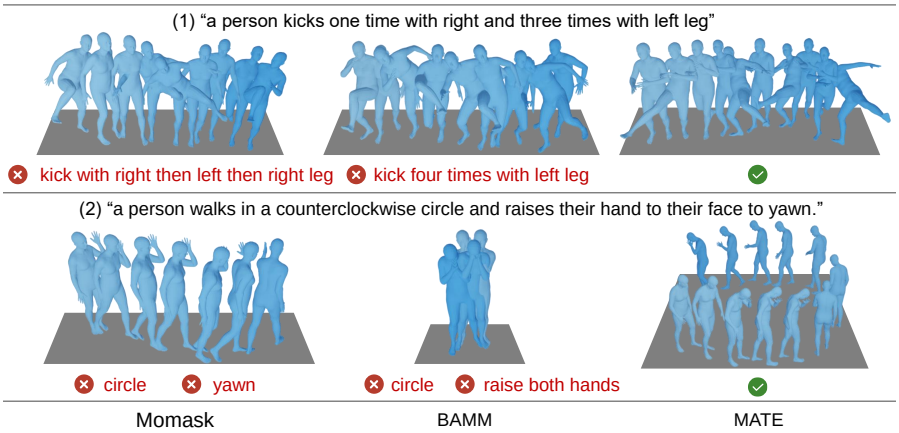


Figure 3: Comparison of motion sequences generated by different SOTA methods, with semantic misalignments highlighted in red.

Table 3: Ablation study of loss functions on HumanML3D by removing losses in Eq. (7) or (8).

Loss Removed	Top-1 ↑	FID ↓	MM-Dist ↓
$\mathcal{L}_{\text{self}}$	0.498	0.339	2.982
$\mathcal{L}_{\text{cross}}$	0.533	0.044	2.934
$\mathcal{L}_{\text{align}}$	0.519	0.049	2.954
$\mathcal{L}_{\text{sent}}$	0.324	0.524	2.983
$\mathcal{L}_{\text{rec}}$	0.547	0.042	2.819
Full Model	**0.550**	**0.040**	**2.811**

Table 4: Evaluations of optimizing different layers of CLIP text encoder on HumanML3D.

Trainable Layers	Parameters	Top-1 ↑	FID ↓
No trainable layers	0M	0.521	0.045
Word embedding layers	3.2M	**0.550**	**0.040**
Subsequent layers	37M	0.022	7.611
All layers	40.2M	0.014	9.468
Adapter (LoRA) Zhang et al. (2024b)	0.4M	0.525	0.051

Optimization of Different Layers. Table 4 compares the effects of optimizing different layers of CLIP within the MATE framework. Fine-tuning subsequent layers or the entire model markedly increases the number of trainable parameters, leading to severe overfitting and degraded generation performance due to the limited size of motion datasets. Instead, MATE restricts optimization to the word embedding layers, effectively aligning word-level semantics while preserving the strong contextual representations captured by the subsequent frozen layers. We also evaluate the LoRA, which is a commonly used LLM fine-tuning strategy Zhang et al. (2024b) by introducing additional lightweight layers while keeping the pretrained model frozen. However, it does not lead to notable performance improvements, suggesting its limited alignment ability in our setting.

Integration with Different Language Models. We tried constructing MATE with CLIP Radford et al. (2021) and DistilBERT Sanh et al. (2019), two of the most commonly adopted language models in T2M methods, as the text encoder. As shown in Table 5, incorporating MATE with either model consistently leads to remarkable performance gains, showing the strong compatibility and generalization capability of MATE across different language models.

Attention Prior. The attention prior, based on motion localization, is evaluated in Table 6. “No prior” denotes using the raw motion features as keys in Eq. (3) without any attention prior, making word-level semantic extraction from full sequences challenging. “Binary ($s_n \leq t \leq e_n$)” applies a hard binary mask, assigning 1 to frames within the target segment and 0 elsewhere, which is highly sensitive to segmentation errors. “Gaussian” is the soft attention prior defined in Eq. (4). “Gaussian ($s_n \leq t \leq e_n$)” restricts the prior within the segment, with zero attention outside. “Gaussian ($0 \leq t \leq T$)” extends the prior across the entire sequence, softly emphasizing the target region while gradually attenuating attention to neighboring frames, thus improving robustness to localization noise and achieving the best generation results.

4.4 Visualization Results

Motion Consistency with Word Change. To demonstrate that MATE effectively learns word-level semantic understanding, we present examples in Fig. 4, where individual words in the text prompts

Table 5: Evaluation of integrating with different large language models on HumanML3D.

Text Encoder	Momask		MDM	
	Top-1 ↑	FID ↓	Top-1 ↑	FID ↓
CLIP Radford et al. (2021)	0.521	0.045	0.491	0.630
+MATE	**0.550**	**0.040**	0.536	**0.234**
DistilBERT Sanh et al. (2019)	0.513	0.053	0.493	0.615
+MATE	0.546	0.045	**0.542**	0.244

Table 6: Ablation of the attention prior in Eq. (4) on KIT.

Attention Prior	Top-1↑	FID↓	MM-Dist↓
No prior	0.428	0.253	2.794
Binary ($s_n \leq t \leq e_n$)	0.431	0.217	2.746
Gaussian ($s_n \leq t \leq e_n$)	0.439	0.198	2.766
Gaussian ($0 \leq t \leq T$)	**0.443**	**0.197**	**2.732**

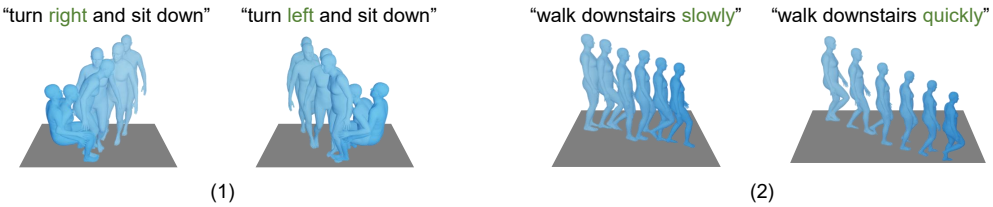


Figure 4: Motions generated by MATE when individual words in the text prompts are replaced with their antonyms. In subfigure (2), both the left and right motions span the same time period, while the right motion descends more stairs and exhibits a faster pace.

are replaced with their antonyms. MATE accurately captures the semantic differences between "right" and "left" as well as "slowly" and "quickly," and generates motions that are semantically aligned with the corresponding words, highlighting its robust fine-grained word-level understanding.

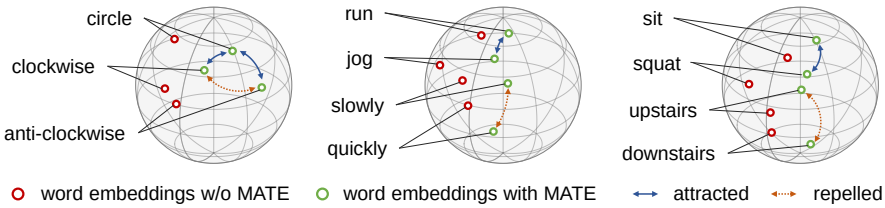


Figure 5: Word embedding distributions on the unit sphere, visualized by DOSNES Lu et al. (2019). MATE brings kinematically related words closer together while separating antonyms.

Distributions of Word Embeddings. To better understand the mechanism underlying the performance gains of our approach, we visualize the effect of MATE on word embeddings in Fig. 5. MATE draws together the embeddings of kinematically related motion words (e.g., "clockwise" and "circle", "run" and "jog") while pushing apart those with contrasting semantics (e.g., "quickly" and "slowly", "upstairs" and "downstairs"). This suggests that MATE structurally regularizes the word embedding space, promoting a closer alignment with motion semantics.

5 Limitations

1) Although our method focuses on word-level semantic alignment, certain words (e.g., "position", "starting", "area") inherently lack clear kinematic semantics or rely on contextual information. Future work will explore selective word-level semantic modeling strategies and the incorporation of contextualized queries to better handle such cases. 2) Word frequencies are imbalanced in the motion descriptions, which is not explicitly considered in this initial exploration of word embedding fine-tuning. An important direction for future work is to explore re-weighting or adaptive updating strategies to mitigate this imbalance.

6 Conclusion

In this work, we have proposed a systematic framework that integrates word-level motion localization, semantic disentanglement and alignment, addressing the text-motion misalignment fundamentally rooted in the word embeddings of large language models for motion generation. Our approach not only demonstrates substantial improvements over state-of-the-art performance on two benchmarks, but also highlights the strong potential of word embedding fine-tuning for enabling motion-aware language modeling.

REFERENCES

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18000–18010, 2023.

Seunggeun Chi, Hyung-gun Chi, Hengbo Ma, Nakul Agarwal, Faizan Siddiqui, Karthik Ramani, and Kwonjoon Lee. M2d2m: Multi-motion generation from text with discrete diffusion models. *European Conference on Computer Vision*, 2024.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. *European Conference on Computer Vision*, 2024.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

Ke Fan, Junshu Tang, Weijian Cao, Ran Yi, Moran Li, Jingyu Gong, Jiangning Zhang, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. Freemotion: A unified framework for number-free text-to-motion synthesis. *European Conference on Computer Vision*, 2024.

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*, pp. 5152–5161, 2022a.

Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, pp. 580–597, 2022b.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1900–1910, 2024.

Yiming Huang, Weilin Wan, Yue Yang, Chris Callison-Burch, Mark Yatskar, and Lingjie Liu. Como: Controllable motion generation through language guided pose code editing. In *European Conference on Computer Vision*, pp. 180–196, 2024.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36, 2023.

Peng Jin, Hao Li, Zesen Cheng, Kehan Li, Runyi Yu, Chang Liu, Xiangyang Ji, Li Yuan, and Jie Chen. Local action-guided motion diffusion model for text-to-motion generation. *European Conference on Computer Vision*, 2024a.

Peng Jin, Yang Wu, Yanbo Fan, Zhongqian Sun, Wei Yang, and Li Yuan. Act as you wish: Fine-grained control of motion diffusion model with hierarchical semantic graphs. *Advances in Neural Information Processing Systems*, 36, 2024b.

Yanyu Li, Xian Liu, Anil Kag, Ju Hu, Yerlan Idelbayev, Dhritiman Sagar, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Textcraftor: Your text encoder can be image quality controller. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7985–7995, 2024.

Han Liang, Jiacheng Bao, Ruichi Zhang, Sihan Ren, Yuecheng Xu, Sibei Yang, Xin Chen, Jingyi Yu, and Lan Xu. Omg: Towards open-vocabulary motion generation via mixture of controllers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 482–493, 2024.

Hanchao Liu, Xiaohang Zhan, Shaoli Huang, Tai-Jiang Mu, and Ying Shan. Programmable motion generation for open-set motion control tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1399–1408, 2024a.

Jinpeng Liu, Wenxun Dai, Chunyu Wang, Yiji Cheng, Yansong Tang, and Xin Tong. Plan, posture and go: Towards open-world text-to-motion generation. *arXiv preprint arXiv:2312.14828*, 2023.

Zhixuan Liu, Peter Schaldenbrand, Beverley-Claire Okogwu, Wenxuan Peng, Youngsik Yun, Andrew Hundt, Jihie Kim, and Jean Oh. Scoft: Self-contrastive fine-tuning for equitable image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10822–10832, 2024b.

Shunlin Lu, Ling-Hao Chen, Ailing Zeng, Jing Lin, Ruimao Zhang, Lei Zhang, and Heung-Yeung Shum. Humantomato: Text-aligned whole-body motion generation. In *International Conference on Machine Learning*, 2024.

Yao Lu, Jukka Corander, and Zhirong Yang. Doubly stochastic neighbor embedding on spheres. *Pattern Recognition Letters*, 128:100–106, 2019.

Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5442–5451, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision*, pp. 480–497, 2022.

Mathis Petrovich, Michael J Black, and Gül Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9488–9497, 2023.

Ekkasit Pinyoanuntapong, Muhammad Usama Saleem, Pu Wang, Minwoo Lee, Srijan Das, and Chen Chen. Bamm: bidirectional autoregressive motion model. In *European Conference on Computer Vision*, pp. 172–190, 2024a.

Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. Mmm: Generative masked motion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1546–1555, 2024b.

Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763, 2021.

Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6545–6554, 2023.

Zeping Ren, Shaoli Huang, and Xiu Li. Realistic human motion generation with cross-diffusion models. *European Conference on Computer Vision*, 2023.

Konstantinos I Roumeliotis and Nikolaos D Tselikas. Chatgpt and open-ai models: A preliminary review. *Future Internet*, 15(6):192, 2023.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22500–22510, 2023.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a generative prior. In *International Conference on Learning Representations*, 2024.

Teo Susnjak, Peter Hwang, Napoleon Reyes, Andre LC Barczak, Timothy McIntosh, and Surangika Ranathunga. Automating research synthesis with domain-specific large language model fine-tuning. *ACM Transactions on Knowledge Discovery from Data*, 19(3):1–39, 2025.

Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *European Conference on Computer Vision*, pp. 358–374, 2022.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *International Conference on Learning Representations*, 2023.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Yin Wang, Zhiying Leng, Frederick WB Li, Shun-Cheng Wu, and Xiaohui Liang. Fg-t2m: Fine-grained text-driven human motion generation via diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22035–22044, 2023.

Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Move as you say, interact as you can: Language-guided human motion generation with scene affordance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yixuan Wei, Han Hu, Zhenda Xie, Ze Liu, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Improving clip fine-tuning performance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5439–5449, 2023.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7623–7633, 2023.

Qi Wu, Yubo Zhao, Yifan Wang, Xinhang Liu, Yu-Wing Tai, and Chi-Keung Tang. Motion-agent: A conversational framework for human motion generation with llms. *International Conference on Learning Representations*, 2025.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *International Conference on Learning Representations*, 2024.

Sheng Yan, Yang Liu, Haoqiang Wang, Xin Du, Mengyuan Liu, and Hong Liu. Cross-modal retrieval for motion and text via droptriple loss. In *Proceedings of the ACM International Conference on Multimedia in Asia*, pp. 1–7, 2023.

Kangning Yin, Shihao Zou, Yuxuan Ge, and Zheng Tian. Tri-modal motion retrieval by learning a joint embedding space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1596–1605, 2024.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023a.

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.

Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 364–373, 2023b.

Mingyuan Zhang, Daisheng Jin, Chenyang Gu, Fangzhou Hong, Zhongang Cai, Jingfang Huang, Chongzhi Zhang, Xinying Guo, Lei Yang, Ying He, et al. Large motion model for unified multi-modal motion generation. *European Conference on Computer Vision*, 2024a.

Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. Motiongpt: Finetuned llms are general-purpose motion generators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7368–7376, 2024b.

Zeyu Zhang, Akide Liu, Ian Reid, Richard Hartley, Bohan Zhuang, and Hao Tang. Motion mamba: Efficient and long sequence motion generation. In *European Conference on Computer Vision*, pp. 265–282, 2024c.

Wenyang Zhou, Zhiyang Dou, Zeyu Cao, Zhouyingcheng Liao, Jingbo Wang, Wenjia Wang, Yuan Liu, Taku Komura, Wenping Wang, and Lingjie Liu. Emdm: Efficient motion diffusion model for fast, high-quality motion generation. *European Conference on Computer Vision*, 2024a.

Zixiang Zhou, Yu Wan, and Baoyuan Wang. Avatargpt: All-in-one framework for motion understanding planning generation and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1357–1366, 2024b.

Qiran Zou, Shangyuan Yuan, Shian Du, Yu Wang, Chang Liu, Yi Xu, Jie Chen, and Xiangyang Ji. Parco: Part-coordinating text-to-motion synthesis. In *European Conference on Computer Vision*, pp. 126–143, 2024.

A LLM USAGE STATEMENT

In this paper, we employed Large Language Models (LLMs) exclusively for language polishing and grammatical refinement of the manuscript. The LLMs were not involved in formulating research ideas, designing methodology, conducting experiments, analyzing results, or drawing conclusions. All scientific contributions, including problem formulation, technical approach, experiments, and analysis, were conceived and carried out solely by the authors.

B ADDITIONAL EVALUATION RESULTS

B.1 USER STUDY

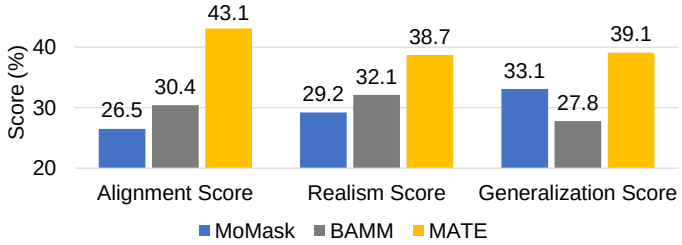


Figure 6: Statistical results of user study.

Questionnaire Design. We conducted a user study to evaluate the subjective quality of motions generated by state-of-the-art methods. The study consisted of a questionnaire containing 40 groups of questions. Each group included a textual description and three motion sequences generated by MoMask Guo et al. (2024), BAMM Pinyoanuntapong et al. (2024a), and MATE. 20 textual descriptions were randomly sampled from HumanML3D. For each description, one half of the corresponding motions were generated by models trained on HumanML3D Mahmood et al. (2019), and the other half by models trained on KIT Plappert et al. (2016), in order to evaluate the generalization ability across datasets. Another 20 descriptions were sampled from KIT and evaluated using the same protocol. In total, 20 descriptions served as cross-dataset samples for assessing generalization. For each question group, users were asked to answer the following two questions (multiple selections allowed): 1) Which motion sequence best aligns with the textual description? 2) Which motion sequence appears most realistic?

Questionnaire Administration. The questionnaire survey was distributed through public channels of several academic and social groups. The majority of respondents were undergraduate and graduate students, as well as researchers. Importantly, a portion of them had backgrounds in computer vision and were familiar with evaluating AI-generated results, which ensured a reasonable level of expertise among participants.

Results Analysis. We collected responses from 24 respondents. The statistical results are shown in Fig. 6. Alignment and realism scores were computed over all 40 groups based on answers to questions 1) and 2), respectively. The generalization score was calculated over the 20 cross-dataset groups using responses to question 1). MATE significantly outperforms MoMask and BAMM across all three metrics, with particularly notable gains in alignment score, demonstrating its superiority in text-motion alignment, generation fidelity, and generalization ability.

B.2 EVALUATION ON DIVERSE TEXT-MOTION TASKS

Beyond text-to-motion generation, we further evaluate MATE on text–motion retrieval, motion inpainting, and motion editing tasks to demonstrate the broad utility and generalizability of our approach.

Text-Motion Retrieval. We compare MATE with existing text-motion retrieval models, including TEMOS Petrovich et al. (2022), TMR Petrovich et al. (2023), and LAVIMO Yin et al. (2024), under the standard text-motion retrieval setting. As shown in Table 7, MATE significantly improves retrieval accuracy compared to the baseline trained without the $\mathcal{L}_{\text{word}}$ loss, demonstrating the benefit

Table 7: Top-1 text-motion mutual retrieval accuracy on HumanML3D. Baseline is MATE trained without $\mathcal{L}_{\text{word}}$. The evaluation protocols (a), (b), and (d) follow the settings provided in Yin et al. (2024).

Retrieval	Methods	(a) All	(b) All with threshold	(d) Small batches
Text-Motion	TEMOS Petrovich et al. (2022)	2.12	5.21	40.49
	TMR Petrovich et al. (2023)	5.68	11.60	67.16
	LAVIMO Yin et al. (2024)	**6.37**	**12.94**	68.58
	Baseline	2.34	2.46	23.65
	MATE (ours)	6.03	11.42	**71.24**
Motion-Text	TEMOS Petrovich et al. (2022)	3.86	5.48	39.96
	TMR Petrovich et al. (2023)	**9.95**	13.20	67.97
	LAVIMO Yin et al. (2024)	9.72	**13.89**	68.64
	Baseline	2.05	1.98	21.66
	MATE (ours)	6.78	11.93	**69.25**

Table 8: Evaluation on motion inpainting and editing tasks on HumanML3D.

Tasks	Methods	Top 1 ↑	FID ↓	MM-Dist ↓
Motion Inpainting	MMM Pinyoanuntapong et al. (2024b)	0.523	0.071	2.910
	+MATE	**0.538**	**0.066**	**2.884**
Motion Editing	MMM Pinyoanuntapong et al. (2024b)	0.500	**0.103**	2.972
	+MATE	**0.521**	0.115	**2.934**

of explicitly optimizing word embeddings for enhancing text-motion alignment. While MATE outperforms all compared methods under protocol (d), it does not achieve competitive performance under protocols (a) and (b). This can be attributed to the different objectives: retrieval models aim to maximize feature separability for retrieval accuracy, whereas MATE prioritizes semantically rich and decodable representations that directly benefit motion generation.

Motion Inpainting and Editing. As shown in Table 8, MATE consistently improves performance on motion inpainting and editing tasks when integrated with the MMM model Pinyoanuntapong et al. (2024b). Word-level misalignment is a fundamental limitation in text-conditioned motion generation, so addressing this issue benefits a wide range of related tasks.

B.3 Distributions of Disentangled Features

To intuitively evaluate the accuracy of motion disentanglement, we visualize the distributions of the disentangled motion features in Fig. 7. In Fig. 7 (1), triangular, circular, and star-shaped markers of the same color (corresponding to the same word) form distinct clusters, while features associated with different words remain well separated. This indicates that the disentangled features are both discriminative and semantically aligned with their respective word embeddings and prototypes.

In Fig. 7 (2), circular and rhombus-shaped markers of the same color are expected to be distinguishable, as meaningful disentanglement should occur only when the motion sequence ***contains*** the semantics of the queried word. This behavior is clearly observed for words with clear kinematic meaning, such as "upstairs" and "pick", demonstrating that our approach does not indiscriminately extract semantics from unrelated motions, which is essential for ensuring the disentanglement accuracy.

In contrast, for more ambiguous words like "quickly", some overlap between markers is occasionally observed, reflecting reduced robustness for semantically vague terms. This suggests that it remains challenging to disentangle vague semantics such as "quickly" from ***mixed*** motion sequences that both contain and do not contain the "quickly" semantics. Nevertheless, it is important to note that during training, disentanglement for such vague terms is applied ***only*** to motion sequences whose corresponding text explicitly ***contains*** the word "quickly", thereby ensuring the accuracy of the learned disentangled semantics.

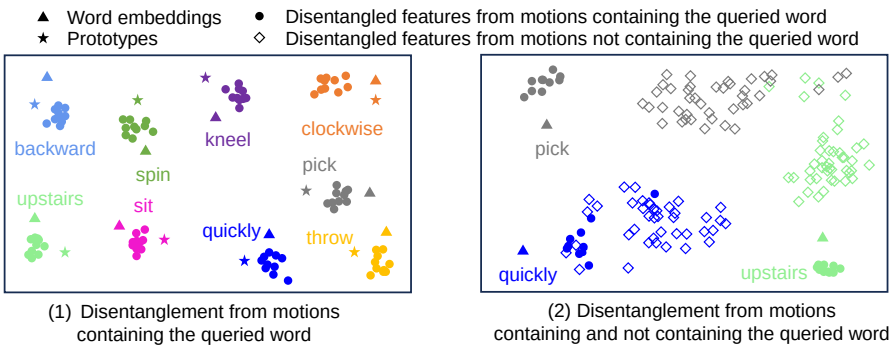


Figure 7: Visualization of motion disentanglement. • indicates motion features disentangled from motions that semantically contain the queried word (*e.g.*, using "upstairs" as query for motions like "walk upstairs"). ◊ indicates motion features disentangled from motions that do not exhibit the queried word's semantics (e.g., using "upstairs" as query for motions like "sit down"). The features are L2-normalized and subsequently visualized using t-SNE.

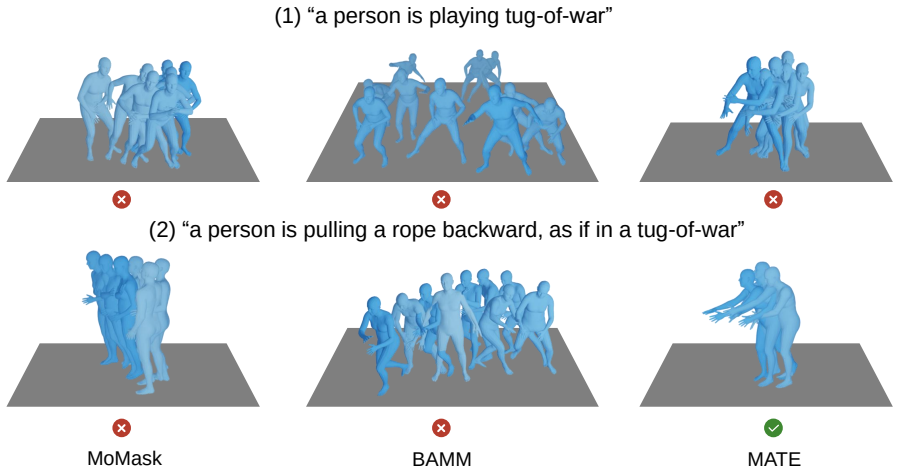


Figure 8: Examples of generalizing state-of-the-art methods to unseen words. "Tug-of-war" is not present in the training vocabulary. MATE generates a more plausible motion among the three motions in (1), and a more accurate and contextually aligned motion in (2) when provided with more textual details. Corresponding motion videos are available in the supplementary materials.

B.4 Generalization to Unseen Words

MATE optimizes only the embeddings of words that appear in the training set. To evaluate its generalization capability, we test MATE on unseen words and present qualitative examples in Fig. 8. In the textual description "a person is playing tug-of-war", the word "tug-of-war" does not appear in the training vocabulary. Compared to MoMask Guo et al. (2024) and BAMM Pinyoanuntapong et al. (2024a), MATE generates a more plausible motion of a person bending over and appearing to pull something with their hands in place, making it the most consistent with a tug-of-war motion among the three.

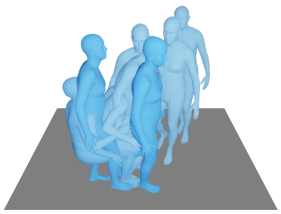
When the textual description is further specified as "a person is pulling a rope backward, as if in a tug-of-war", MATE generates the motion that best aligns with the input text, depicting a person pulling a rope while slightly moving backward in place. This result demonstrates MATE's superior generalization ability to unseen words, particularly when provided with more detailed contextual information.

This improvement can be attributed to two main factors. First, MATE develops a robust understanding of seen words (e.g., "pulling", "rope", "backward") by fine-tuning their embeddings within the language model. The fine-tuning introduces only slight modifications to the word embeddings,

thereby preserving the structural integrity of the original representations learned by the language model. Second, the subsequent layers of the language model remain unchanged, maintaining their strong contextual modeling capabilities. By combining the locally adapted word embeddings for seen words with the inherent generalization ability of large language models, MATE improves generalization performance on unseen words, especially when the textual descriptions are contextually rich. More examples are provided in the supplementary video materials.

B.5 Failure cases

The example on the right shows a failure case in which the generated motion fails to accurately reflect the phrase "walks back to starting point". This illustrates a limitation of MATE in capturing the semantics of abstract words like "starting", which require more contextual and temporal cues. Our method focuses on extracting word semantics from the target motion segment, while reducing attention to temporally distant content. Future work will incorporate richer contextual cues to improve the semantic understanding of words that depend on broader temporal context.



"a person takes steps forward, sits down, and then walks back to starting point."

Figure 9: Failure case in capturing the semantics of "starting point."

C Implementation Details

Text-Motion Joint Segmentation. We use ChatGPT (gpt-4-turbo) to decompose the textual descriptions. An example prompt is shown below:

```
"a man staggers forward, turns clockwise, then jogs back to starting
point."
Goal: Split the above sentence with \n.
Each contains simultaneous one or more motions.
Return the sentence in which the actions are arranged in the order
they occur.
Notes:
1. Each word may be used only once.
2. Each phrase must include at least one verb; do not introduce any
new words.
3. Adverbial modifiers like "as if", "while" should remain in the
same sentence
as the action they describe.
4. If all actions occur simultaneously, return the original sentence.
```

If the sentence is split into more than one sentence, they are fed into a pretrained text-to-motion retrieval model Lu et al. (2024) with the paired motion sequence to identify motion segment boundaries by minimizing the objective defined in Eq. (1).

To improve the efficiency of the exhaustive search, we assume that the semantics of an action remain relatively stable for at least 0.5s. Based on this assumption, we constrain each segment's length to be a multiple of 0.5s and apply a fixed sliding window with a stride of 0.5s. This corresponds to 10 frames in HumanML3D and 6 frames in KIT. If the final segment is shorter than 0.5s, it is merged with the preceding segment.

Examples of decomposed sentences and the corresponding motion segments are provided in the supplementary video materials.

Model Structures. The text encoder adopts the original architecture of large language models without any modifications. The motion encoder and decoder follow the design presented in Petrovich et al. (2022). The motion encoder simultaneously extracts frame-level features $\boldsymbol{F}^m \in \mathbb{R}^{T\times D}$ and a sequence-level feature $\boldsymbol{f}^m \in \mathbb{R}^D$, where the feature dimension D is fixed at 512. This dimensionality is consistently used across motion features, prototype representations $\boldsymbol{f}^p_{w_k}$, and the linear projection

layer Proj. The $\mathrm{MultiHead}$ module is an 8-head attention mechanism with an embedding dimension of 512 and a dropout rate of 0.1.

Training Details. Text-motion joint segmentation is performed as a preprocessing step prior to training. The MATE framework is trained on the training sets of the respective datasets using the AdamW optimizer. The learning rate is set to 1e-5 for the word embedding layer and 1e-4 for all other layers. Training is performed on one RTX 4090 GPU (24 GB) with a batch size of 64 for 100 epochs.

To construct the word prototypes, we first lemmatize the textual descriptions and build the vocabulary accordingly, resulting in $K = 5,161$ and $1,191$ prototypes for HumanML3D and KIT, respectively. Each word in the input text is lemmatized to match its corresponding prototype. For example, "walks" and "walking" are both lemmatized to "walk" and share the same prototype.

The loss $\mathcal{L}_{\text{sent}}$ follows a similar formulation to Eq. (6), where paired text and motion features $\boldsymbol{f}^t$ and $\boldsymbol{f}^m$ serve as positive examples, and mismatched pairs are treated as negatives. The loss $\mathcal{L}_{\text{rec}}$ is defined as an L1 loss between the original motion sequence and the reconstructed sequence.

The hyperparameters are set as follows: the weight of the attention prior loss is $\lambda = 0.1$; the weights for $\mathcal{L}_{\text{word}}$ and $\mathcal{L}_{\text{sent}}$ are $\lambda_1 = \lambda_2 = 0.1$; and all temperature parameters are set to $\tau = 0.05$.

During the cross-disentanglement process, when computing the loss $\mathcal{L}_{\text{cross}}$, the set of negative pairs $\mathcal{N}$ is constructed as follows: For the i-th textual description $\boldsymbol{t}_i$ and its associated word semantics $\boldsymbol{w}_i$, we compute the cosine similarity between the feature of $\boldsymbol{w}_i$ and those of all other textual descriptions $\boldsymbol{t}_j$ within the batch (where $j \in \mathcal{V}$ and $j \neq i$). The top 8 samples with the lowest similarity scores are selected to form the negative set for the i-th instance. This strategy ensures that the selected negatives are the least likely to share the same target word semantics $\boldsymbol{w}_i$, thereby enhancing the reliability of the cross-disentanglement process.

D Additional Ablation Study

D.1 Analysis of Text-Motion Joint Segmentation

Table 9: Quantitative evaluation of text-motion joint segmentation results.

Metrics	HumanML3D	KIT
Text decomposition accuracy (%)	97.0	98.5
Motion segmentation errors (sec)	0.73	0.42

Quantitative Evaluation. To quantitatively evaluate text–motion joint segmentation, we manually assess the text decomposition results and annotate ground-truth segmentation boundaries for 200 motion sequences randomly sampled from the HumanML3D and KIT datasets, respectively. We then compute the average accuracy of text decomposition and the boundary errors between our predictions and the ground truth, with results reported in Table 9. The decomposition accuracy reaches 97.0% and 98.5% on the two datasets, demonstrating the strong text-processing capability of large language models within our approach. The small boundary errors (0.73s and 0.42s) further indicate that our method achieves precise alignment, with deviations well within a second, between decomposed texts and segmented motion clips. Notably, while some noise may be introduced around the segmentation boundaries, our Gaussian-based soft attention mechanism leverages soft localization signals rather than rigid hard segmentation, thereby improving robustness to such noise.

Qualitative Analysis. We present examples of text-motion joint segmentation in Fig. 10. The textual decomposition generated by ChatGPT is generally accurate. The primary source of error arises from imprecise boundaries predicted by the segmentation module. For instance, the boundaries in (1) are correctly identified, whereas in (2), the transition between "walks three steps forward" and "halts and sits down" is slightly misaligned.

Such boundary inaccuracies are anticipated in our framework. Instead of applying a hard binary mask to isolate target segments, MATE leverages the segmentation output as a soft attention prior, modeled using Gaussian-shaped attenuation. This approach mitigates the impact of boundary errors

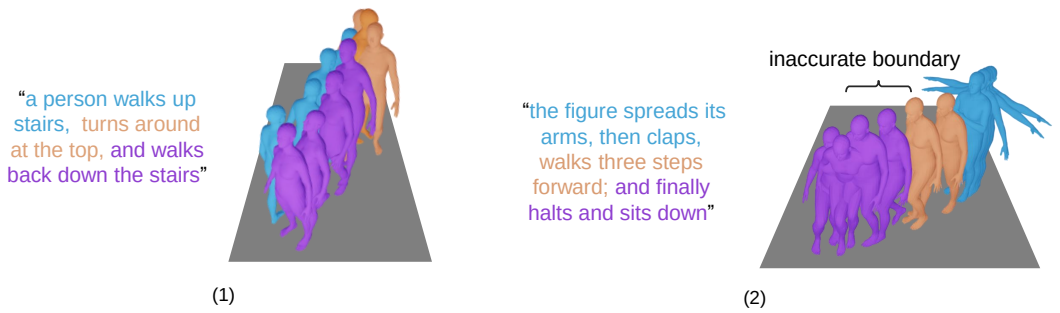


Figure 10: Examples of text-motion joint segmentation. Different colors represent the decomposed sub-texts and their corresponding motion segments. The segmentation in (1) is accurate, while in (2), the third motion segment corresponding to "halts and sits down" includes the partial semantics of the second sub-text "walks three steps", which reflects a minor boundary imprecision. Corresponding motion videos are available in the supplementary materials.

and enhances robustness to segmentation imprecision, ultimately improving alignment quality. The quantitative comparison of different attention strategies is shown in Table 6 in the main paper.

Segment Unit Length. As described in Sec. C, our approach constrains each segment length to be a multiple of 0.5s to improve computational efficiency. The effect of the segment unit length is reported in Table 10. Results show that 0.25s and 0.5s yield comparable performance, whereas larger units ($\geq$0.75s) cause slight degradation. We adopt 0.5s as a balanced setting that maintains segmentation quality while improving processing speed.

Table 10: Evaluation of segment unit length.

Segment Unit Length (s)	Top 1 ↑	MM-Dist ↓
0.25	0.546	**2.804**
0.5	**0.550**	2.811
0.75	0.538	2.826
1.0	0.541	2.847

D.2 Analysis of Word-Level and Sentence-Level Contributions

MATE directly optimizes word embeddings and indirectly optimizes sentence-level features. To investigate their respective contributions to performance, we conduct experiments with MMM Pinyoanuntapong et al. (2024b), which explicitly separates the effects of word embeddings and sentence features, thereby providing a suitable framework for this analysis. Specifically, we selectively replace the word embeddings and sentence features in CLIP within MMM with their MATE-enhanced counterparts, and report the results in Table 11. The results reveal three key findings:

1) Enhancing either word embeddings or sentence features independently improves performance. 2) Sentence-level enhancement yields slightly larger gains, likely due to the stronger influence of sentence features compared to word embeddings in MMM. 3) Joint enhancement at both levels leads to the best performance, confirming that MATE improves generation by simultaneously enhancing both representations.

Table 11: Evaluation of enhancing word embeddings or sentence features in MMM Pinyoanuntapong et al. (2024b) on HumanML3D.

Word Embeddings	Sentence Features	Top 1 ↑	FID ↓	MM-Dist ↓
CLIP	CLIP	0.515	0.089	2.926
MATE	CLIP	0.522	0.098	2.913
CLIP	MATE	0.530	0.081	2.909
MATE	MATE	**0.541**	**0.069**	**2.887**

D.3 Comparison with Text-Motion Retrieval Methods

We also explore optimizing word embeddings of large language models using text-motion retrieval models, specifically TEMOS Petrovich et al. (2022) and TMR Petrovich et al. (2023). We fine-tune word embeddings of the CLIP text encoder using the alignment strategies proposed in these works. The fine-tuned encoder then replaces the original text encoder in MoMask, which is retrained with this modification. As shown on the right, neither TEMOS nor TMR leads to consistent or significant performance improvements for MoMask. This is likely because both methods focus on sentence-level alignment, without explicitly optimizing individual word embeddings to capture fine-grained, motion-specific semantics.

Methods	Top 1 ↑	FID ↓	MM-Dist ↓
MoMask Guo et al. (2024)	0.433	0.204	2.779
+TEMOS Petrovich et al. (2022)	0.427	0.226	2.772
+TMR Petrovich et al. (2023)	0.434	0.235	2.767
+MATE	**0.443**	**0.197**	**2.732**

Table 12: Comparison of optimizing word embeddings with different text-motion retrieval models on the KIT dataset.

D.4 Analysis of Word Frequency

We present the distribution of word frequencies on HumanML3D in Table 13. A small proportion of words occur very frequently, while the majority appear infrequently, indicating a clear word imbalance. To address this issue, we conducted an initial exploration using a classical re-weighting strategy Cui et al. (2019) that decreases updates for high-frequency words and increases them for low-frequency words, with results reported in Table 14. However, this strategy results in performance degradation, which we attribute to the unique characteristics of our task compared with standard long-tail learning.

In our setting, word embeddings are aligned with motion semantics through fine-tuning. This process requires a delicate balance between adapting to motion-specific semantics and preserving pretrained language priors, a balance that varies depending on word frequency.

1) High-frequency words. These benefit from abundant samples, which allow the construction of more stable motion prototypes. As described in our stability principle and Eq. (5), we align word embeddings with prototypes that capture shared semantics across many instances rather than single examples, thereby reducing bias and instability. Suppressing updates for high-frequency words can therefore weaken the learning of generalized motion semantics.

2) Low-frequency words. These are more susceptible to noisy supervision due to limited data. Simply increasing their updates may lead to overfitting or semantic drift away from their original language priors. In such cases, limiting updates can better preserve semantic stability and reduce the risk of unreliable adaptation.

Overall, achieving an optimal trade-off remains challenging and requires more careful strategy design. As future work, we plan to develop adaptive re-weighting strategies that not only account for word frequency but also dynamically assess the reliability of motion semantics during updates.

Table 13: Word Frequency Statistics.

Frequency	>1,000	100-1,000	10-100	<10
Words	5.3%	23.5%	47.6%	23.6%

Table 14: Evaluation of reweighting.

Methods	Top 1 ↑	FID ↓	MM-Dist ↓
MATE	**0.550**	**0.040**	**2.811**
MATE with re-weighting	0.536	0.066	2.924

D.5 Analysis of Hyper-Parameters

Fig. 11 illustrates the effect of key hyper-parameters in the MATE framework. In the left subfigure, λ_1 and λ_2 denote the weight factors for the word-level loss $\mathcal{L}_{\text{word}}$ and the sentence-level loss $\mathcal{L}_{\text{sent}}$, respectively. Compared to λ_2, λ_1 has a more pronounced impact on the generation results, indicating that the model is more sensitive to variations in word-level alignment than in sentence-level alignment. In the right subfigure, λ controls the strength of the attention prior. A relatively small value may fail to provide sufficient motion localization guidance, while an overly large value can compromise the model's robustness to inaccuracies in the prior. The best performance is observed when $\lambda = 0.1$. The

temperature parameter τ used in $\mathcal{L}_{\text{self}}$, $\mathcal{L}_{\text{align}}$, and $\mathcal{L}_{\text{sent}}$, modulates the contrast between positive and negative sample similarities. Empirically, the model achieves optimal performance when $\tau = 0.05$.

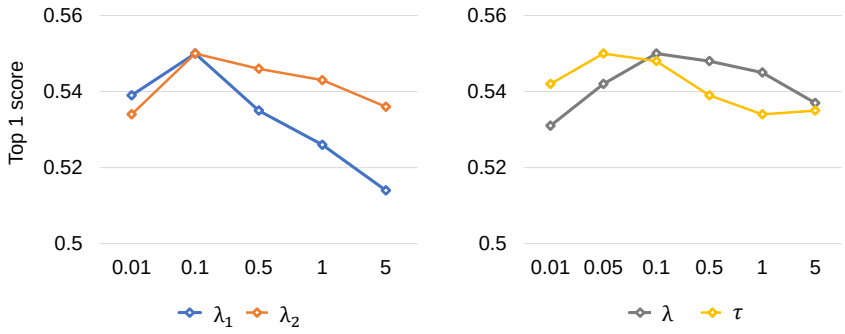


Figure 11: Analysis of hyper-parameters on HumanML3D dataset.