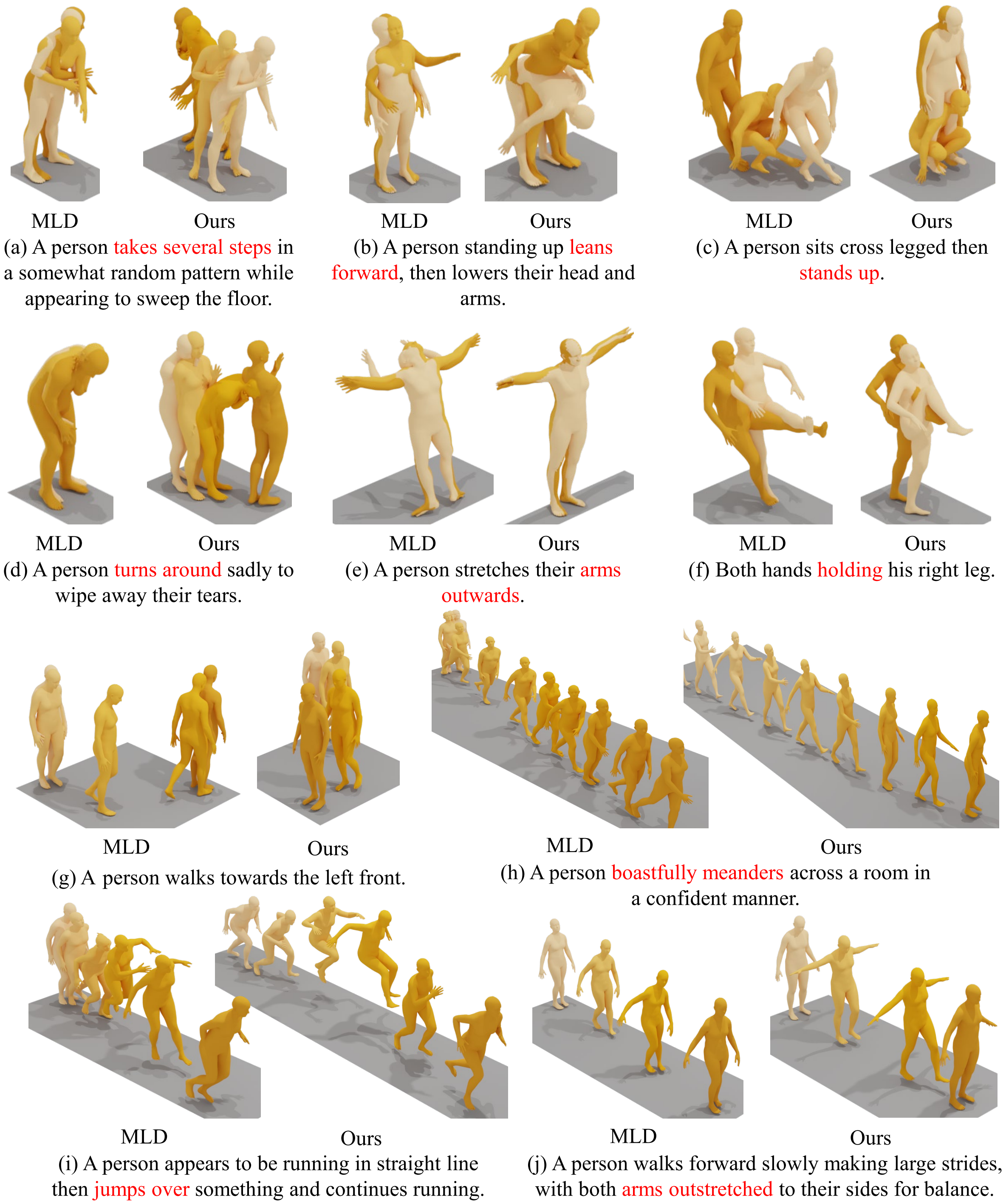
Bilingual Text-to-Motion Generation via Step-Aware Reward-Guided Alignment

Supplementary Material



MLD Ours
(a) A person takes several steps in a somewhat random pattern while appearing to sweep the floor.

MLD Ours
(b) A person standing up leans forward, then lowers their head and arms.

MLD Ours
(c) A person sits cross legged then stands up.

MLD Ours
(d) A person turns around sadly to wipe away their tears.

MLD Ours
(e) A person stretches their arms outwards.

MLD Ours
(f) Both hands holding his right leg.

MLD Ours
(g) A person walks towards the left front.

MLD Ours
(h) A person boastfully meanders across a room in a confident manner.

MLD Ours
(i) A person appears to be running in straight line then jumps over something and continues running.

MLD Ours
(j) A person walks forward slowly making large strides, with both arms outstretched to their sides for balance.

Figure S1. Visual comparison of motion generation results on the HumanML3D dataset. Our proposed BiMD method with ReAlign integration improves alignment between text descriptions and generated motions and enhances overall motion quality compared to MLD [1]. The red text denotes descriptions inconsistent with the generated motions.

This supplementary document presents additional visual results, detailed experimental setups, further outcomes, and insights into the construction of the bilingual HumanML3D dataset. It is organized as follows: Sec. A presents a visual comparison of motion generation results on the HumanML3D dataset, featuring the performance of our BiMD method with ReAlign integration alongside MLD [1]. Sec. B provides comprehensive details on the construction of the BiHumanML3D dataset, including its annotation design. Sec. C provides comprehensive details on the dataset, evaluation metrics, and additional experimental procedures and results. Sec. D presents the proofs and key results, encompassing Theorem 1, Theorem 2, and Theorem 3.

A. More Visualization

B. Details on Constructing Bilingual HumanML3D

This section outlines the pipeline for constructing the bilingual HumanML3D dataset. As depicted in Fig. S2, the pipeline comprises two main components: data collection and filtering, followed by an LLM-assisted annotation design.

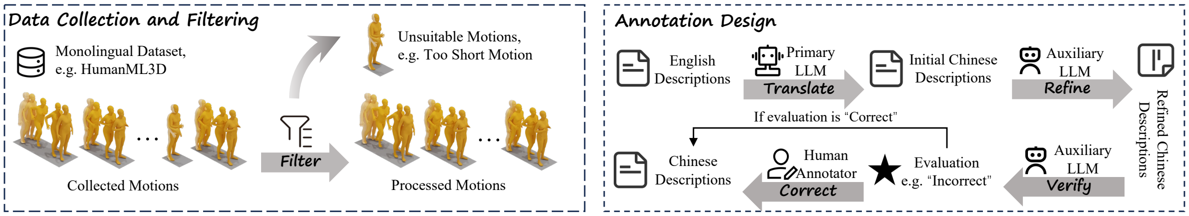

Data Collection and Filtering
Monolingual Dataset, e.g. HumanML3D
Unsuitable Motions, e.g. Too Short Motion
Filter
Collected Motions
Processed Motions
Annotation Design
English Descriptions
Primary LLM
Translate
Initial Chinese Descriptions
Auxiliary LLM
Refine
Refined Chinese Descriptions
If evaluation is "Correct"
Chinese Descriptions
Human Annotator
Correct
Evaluation e.g. "Incorrect"
Auxiliary LLM
Verify

Figure S2. Framework for constructing a bilingual HumanML3D dataset. The data collection and filtering process removes unsuitable motions, ensuring high-quality motion-text pairs for translation. The annotation pipeline begins with initial translation using DeepSeek [6], followed by refinement with Qwen [14] to address translation issues. Finally, human annotators manually verify and correct the translation with DeepSeek [6], ensuring linguistic and contextual accuracy.

B.1. Prompt Design

We present the prompt used in Annotation Design as follows. To construct the BiHumanML3D dataset, we employ an LLM-assisted annotation pipeline that translates English motion descriptions into a target language while preserving semantic accuracy, as detailed in the main text. The prompts shown here are designed to guide the LLMs through the three-stage pipeline, initial translation, refinement, and evaluation, ensuring high-quality bilingual annotations.

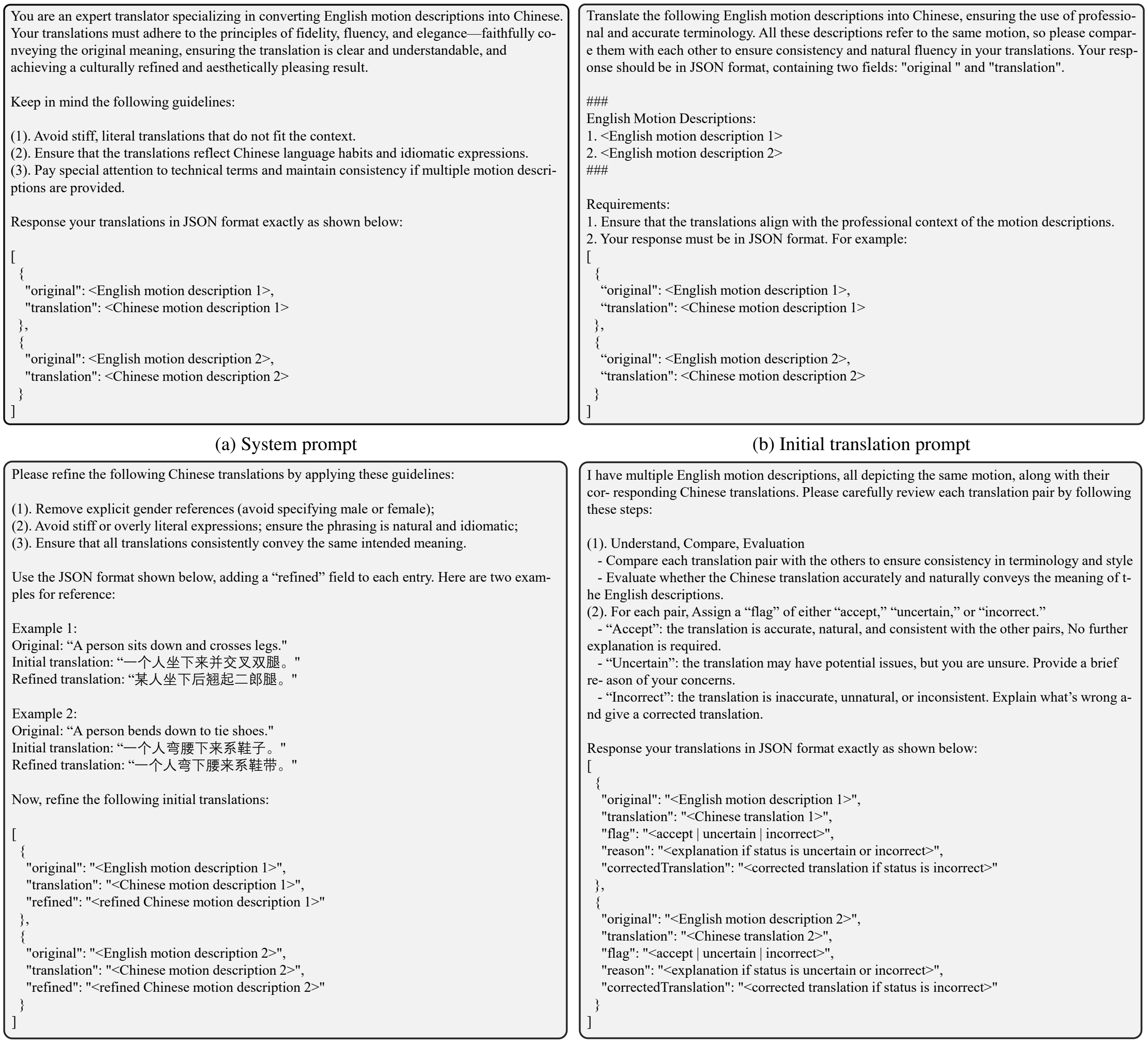
B.2. Details on Annotation Design

To extend monolingual datasets into bilingual form, we develop a robust LLM-assisted annotation pipeline that ensures high-quality translations while preserving motion semantics. As illustrated in Fig. 2, this pipeline consists of three key stages. We initialize the translation agent using the system prompt presented in Fig. S3 (a). In the first stage of initial translation, DeepSeek [6], a primary LLM, translates the original English text descriptions into the target language, guided by the prompt presented in Fig. S3 (b). In the second stage, a refinement prompt presented in Fig. S3 (c) is used with Qwen [14] to refine the translations, addressing issues such as gender bias, overly literal translations, and unnatural phrasing, thereby improving linguistic quality. In the final stage of correction, DeepSeek [6] and human annotators evaluate translation quality using the prompt presented in Fig. S3 (c). Descriptions flagged as "Uncertain" or "Incorrect" by the LLM are then manually reviewed and corrected by human annotators to ensure both linguistic and contextual accuracy.

C. Additional Experiment

C.1. Additional Experimental Details

Datasets HumanML3D [4] dataset comprises 14,616 motion sequences, each paired with one or more text descriptions, resulting in a total of 44,970 annotations. **KIT-ML** [12] dataset includes 3,911 motion sequences with 6,278 corresponding text descriptions. The proposed **BiHumanML3D** dataset extends the HumanML3D dataset into a bilingual dataset through an LLM-assisted annotation pipeline, incorporating bilingual text descriptions while preserving the original motion sequences.
Evaluation Metrics. Our experimental results are evaluated based on two key aspects: generation quality and alignment quality. For generation quality, we use Fréchet Inception Distance (FID) to measure the distributional difference between



You are an expert translator specializing in converting English motion descriptions into Chinese. Your translations must adhere to the principles of fidelity, fluency, and elegance—faithfully conveying the original meaning, ensuring the translation is clear and understandable, and achieving a culturally refined and aesthetically pleasing result.

Keep in mind the following guidelines:

(1). Avoid stiff, literal translations that do not fit the context.
(2). Ensure that the translations reflect Chinese language habits and idiomatic expressions.
(3). Pay special attention to technical terms and maintain consistency if multiple motion descriptions are provided.

Response your translations in JSON format exactly as shown below:

```
[
  {
   "original": <English motion description 1>,
   "translation": <Chinese motion description 1>
  },
  {
   "original": <English motion description 2>,
   "translation": <Chinese motion description 2>
  }
]
```

(a) System prompt

Translate the following English motion descriptions into Chinese, ensuring the use of professional and accurate terminology. All these descriptions refer to the same motion, so please compare them with each other to ensure consistency and natural fluency in your translations. Your response should be in JSON format, containing two fields: "original " and "translation".

```
###
English Motion Descriptions:
1. <English motion description 1>
2. <English motion description 2>
###
```

Requirements:
1. Ensure that the translations align with the professional context of the motion descriptions.
2. Your response must be in JSON format. For example:

```
[
  {
   “original": <English motion description 1>,
   “translation": <Chinese motion description 1>
  },
  {
   “original": <English motion description 2>,
   “translation": <Chinese motion description 2>
  }
]
```

(b) Initial translation prompt

Please refine the following Chinese translations by applying these guidelines:

(1). Remove explicit gender references (avoid specifying male or female);
(2). Avoid stiff or overly literal expressions; ensure the phrasing is natural and idiomatic;
(3). Ensure that all translations consistently convey the same intended meaning.

Use the JSON format shown below, adding a “refined” field to each entry. Here are two examples for reference:

Example 1:
Original: “A person sits down and crosses legs."
Initial translation: “一个人坐下来并交叉双腿。"
Refined translation: “某人坐下后翘起二郎腿。"

Example 2:
Original: “A person bends down to tie shoes."
Initial translation: “一个人弯腰下来系鞋子。"
Refined translation: “一个人弯下腰来系鞋带。"

Now, refine the following initial translations:

```
[
  {
   "original": "<English motion description 1>",
   "translation": "<Chinese motion description 1>",
   "refined": "<refined Chinese motion description 1>"
  },
  {
   "original": "<English motion description 2>",
   "translation": "<Chinese motion description 2>",
   "refined": "<refined Chinese motion description 2>"
  }
]
```

(c) Refine translation prompt

I have multiple English motion descriptions, all depicting the same motion, along with their cor- responding Chinese translations. Please carefully review each translation pair by following these steps:

(1). Understand, Compare, Evaluation
  - Compare each translation pair with the others to ensure consistency in terminology and style
  - Evaluate whether the Chinese translation accurately and naturally conveys the meaning of the English descriptions.
(2). For each pair, Assign a “flag” of either “accept,” “uncertain,” or “incorrect.”
  - “Accept”: the translation is accurate, natural, and consistent with the other pairs, No further explanation is required.
  - “Uncertain”: the translation may have potential issues, but you are unsure. Provide a brief re- ason of your concerns.
  - “Incorrect”: the translation is inaccurate, unnatural, or inconsistent. Explain what’s wrong and give a corrected translation.

Response your translations in JSON format exactly as shown below:

```
[
  {
   "original": "<English motion description 1>",
   "translation": "<Chinese translation 1>",
   "flag": "<accept | uncertain | incorrect>",
   "reason": "<explanation if status is uncertain or incorrect>",
   "correctedTranslation": "<corrected translation if status is incorrect>"
  },
  {
   "original": "<English motion description 2>",
   "translation": "<Chinese translation 2>",
   "flag": "<accept | uncertain | incorrect>",
   "reason": "<explanation if status is uncertain or incorrect>",
   "correctedTranslation": "<corrected translation if status is incorrect>"
  }
]
```

(d) Evaluate translation prompt

Figure S3. **Prompts Utilized in the LLM-Assisted Annotation Process for the BiHumanML3D Dataset.** This figure presents the prompts guiding the translation of English motion descriptions into a target language with semantic accuracy, covering initial translation, refinement, and evaluation stages. The system prompt establishes foundational instructions, setting the tone and context for the LLM to ensure consistent bilingual annotations, as detailed in the main text.

high-level features of generated and real motions, and Diversity to assess motion diversity by calculating variation among generated motions. For alignment quality, R-Precision evaluates motion-retrieval precision, assessing matching quality between generated motions and text descriptions, while Multi-Modal Distance (MM Dist) quantifies the distance between motions and their corresponding text descriptions.

C.2. Details on Cross-lingual Alignment.

To achieve cross-lingual alignment, we follow the configuration outlined in AltCLIP [2], utilizing the XLM-Base model [3] as the backbone of our student model. We fine-tune the student model by incorporating an additional fully connected layer to derive sentence embeddings, facilitating alignment across languages. The optimization process employs the AdamW algorithm [7], enhanced by a cosine decay learning rate scheduler with a 500-step linear warm-up phase. Training is conducted for 50 epochs with a batch size of 128 and a learning rate set to 10^{-4}.

C.3. Details on Step-Aware Reward Model training.

Representation Loss. We adopt the foundational loss framework proposed by [9], which integrates multiple components to guide the training process. This framework is formalized as:

$$\mathcal{L}_R(\varphi; \mathbf{x}_t, c) = \mathcal{L}_{\text{rencos}} + \lambda_1 \mathcal{L}_{\text{KL}} + \lambda_2 \mathcal{L}_{\text{E}}, \tag{S1}$$

where $\mathcal{L}_{\text{recons}}$ denotes the reconstruction loss, $\mathcal{L}_{\text{KL}}$ represents the Kullback-Leibler (KL) divergence loss, $\mathcal{L}_{\text{E}}$ indicate the cross-modal embedding loss, and λ_1, λ_2 are weight parameters to balance their contributions. The first component, $\mathcal{L}_{\text{recons}}$, evaluates the accuracy of motion reconstruction from text or motion inputs by applying a smooth L1 loss to quantify the discrepancy between reconstructed and ground-truth motions. The second component, $\mathcal{L}_{\text{KL}}$, comprises four terms: two regularize the encoded distributions of motion $\mathcal{N}(\mu^M, \Sigma^M)$ and text $\mathcal{N}(\mu^T, \Sigma^T)$ to align with a standard normal distribution $\mathcal{N}(0, I)$, while the remaining two foster similarity between the text and motion distributions. The third component, $\mathcal{L}_{\text{E}}$, ensures alignment between the latent representations of text z^T and motion z^M, utilizing a smooth L1 loss to minimize their differences.

Contrastive Loss for Text-Motion Alignment. To improve the alignment of text and motion within our step-aware reward model, we implement the InfoNCE loss formulation [8], a method previously utilized in [8, 10]. This loss is mathematically expressed as:

$$\mathcal{L}_{\text{C}} = -\frac{1}{2N} \sum_{i=1}^{N} \left(\log \frac{\exp\left(S_{ii}/\tau\right)}{\sum_j \exp\left(S_{ij}/\tau\right)} + \log \frac{\exp\left(S_{ii}/\tau\right)}{\sum_j \exp\left(S_{ji}/\tau\right)} \right), \tag{S2}$$

where $S_{ij} = \cos(z_i^T, z_j^M)$ represents the cosine similarity between text latent embedding z_i^T and motion latent embedding z_j^M, and $\tau$is the temperature hyperparameter that adjusts the distribution's softness. The loss is applied to a batch of N positive latent pairs $(z_1^T, z_1^M), \ldots, (z_N^T, z_N^M)$, where negative pairs are identified as (z_i^T, z_j^M) for $i \neq j$. captures the pairwise cosine similarities across all combinations, facilitating the alignment process. Additionally, we employ a filtering mechanism for negative pairs: only those with a cosine similarity between sentence embeddings F_i^T and F_j^T exceeding a predefined threshold are retained as valid negatives, where F^T denotes the input text's sentence embeddings.

Training Details for Step-Aware Reward Model. We employ the SkipTransformer [1] as the foundational architecture for our step-aware reward model, consisting of a transformer encoder processing both text and motion inputs, alongside a motion decoder. Each component features 9 layers and 4 attention heads, with the latent space dimension fixed at 256. The training process incorporates a maximum timestep of 1000, a noisy motion probability of 0.5, and a negative filtering threshold of 0.9 to regulate the selection of negative samples.

For model training, we adhere to the TMR framework [10], employing a composite loss function expressed as a weighted combination $\mathcal{L}_{\text{R}} + \lambda_{\text{NCE}} \mathcal{L}_{\text{NCE}}$. Optimization is performed using the AdamW algorithm [7], configured with a learning rate of 10^{-4} and a batch size of 128, while other hyperparameters are consistent with those specified in the TMR framework [10].

Details on BiHumanML3D Evaluator Training. For the training of the BiHumanML3D evaluator, we opt for a more powerful and adaptable pretrained multilingual large language model, MultilingualBERT [11], as the sentence encoder to derive token-level features, moving away from the use of Chinese Word2Vec. The remaining training configurations align with the specifications provided in Guo et al. [4].

Method	Language	R Precision ↑			FID ↓	MM Dist ↓	Diversity →
		Top 1	Top 2	Top 3			
Real	CN	0.543	0.732	0.821	0.002	3.338	10.750
	EN	0.531	0.721	0.811	0.002	3.211	10.760
	BL	0.535	0.724	0.815	0.002	3.110	10.748

Table 1. Performance of our evaluator on the BiHumanML3D dataset for different language settings. The table presents the evaluation results for Chinese (CN), English (EN), and Bilingual (BL) language settings, corresponding to the language of the text descriptions.

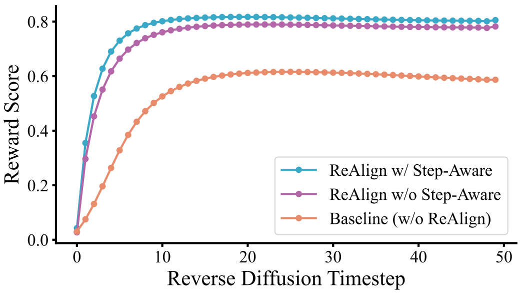


Figure S4. Average reward score for each timestep during the reverse diffusion process

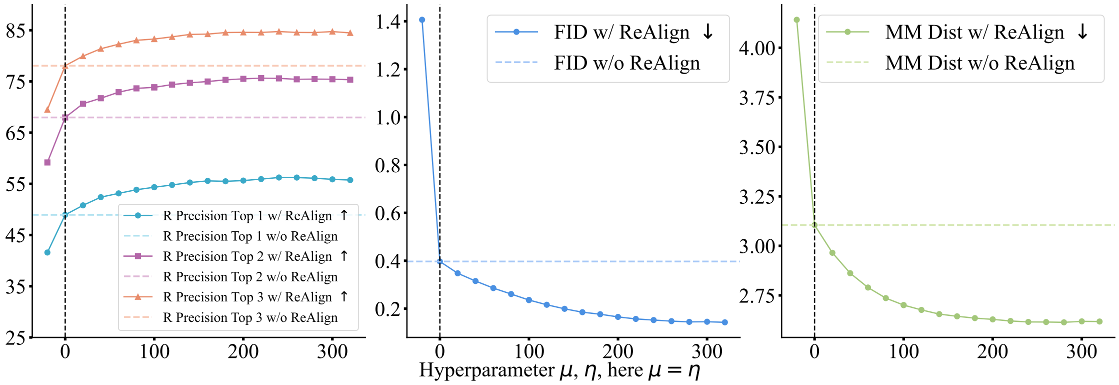


Figure S5. Impact of hyperparameters μ and η (with $\mu = \eta$) on step-aware reward alignment (ReAlign): R Precision for Top k (left), FID (middle), and MM Dist (right) are evaluated under varying μ and η, with $\mu = \eta = 0$ serving as the baseline (dashed line).

C.4. Additional Experimental Results

Effectiveness of Step-Aware Strategy We evaluate the effectiveness of step-aware realignment on the HumanML3D dataset using our BiMD, as shown in Fig. S4. The figure presents the average reward scores, defined in Eq. (11), across reverse diffusion timesteps, comparing ReAlign with step-aware, ReAlign without step-aware, and a baseline without ReAlign. ReAlign with step-aware consistently achieves higher reward scores than both the baseline and ReAlign without step-aware throughout the diffusion process. Additionally, it demonstrates a sustained improvement over ReAlign without step-aware, emphasizing its step-aware realignment capability as a vital factor in enhancing motion generation quality.

C.5. Additional Ablation Study

Impact on Step-Aware Reward Alignment. We examine the impact of hyperparameters μ and η on the step-aware reward alignment (ReAlign) framework using our Bilingual Motion Diffusion (BiMD) model on the HumanML3D dataset, as outlined in Eq. (11), under the condition $\mu = \eta$. Fig. S5 presents the variations in R Precision for Top 1, 2, and 3, alongside FID and MM Dist, with $\mu = \eta = 0$ serving as the baseline (marked by the black dashed line), reflecting the scenario without ReAlign. Negative values of μ and η invert the ReAlign mechanism, leading to degraded performance, where FID escalates to 1.400, R Precision@3 declines to 40.0%, and MM Dist rises to 4.000. In contrast, positive values of μ and η enhance performance, lowering FID from 0.397 to 0.178 and MM Dist from 3.000 to 2.714, highlighting the essential role of ReAlign in improving motion generation quality.

D. Theories

D.1. Proof of Theorem 1

Theorem 1. *When using the ideal sampling distribution $p_t^I(\mathbf{x}|c)$ in Eq. (5) to replace the vanilla sampling distribution $p_t(\mathbf{x}|c)$, the reverse SDE becomes:*

$$\mathbf{dx} = \Big[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla\big(\log p_t(\mathbf{x}|c) + \log p_t^r(\mathbf{x}|c)\big)\Big]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}. \tag{S3}$$

Recall the definition of ideal sampling distribution $p_t^I(\mathbf{x}|c)$ in Eq. (5):

$$p_t^I(\mathbf{x}|c) = \frac{p_t(\mathbf{x}|c)p_t^r(\mathbf{x}|c)}{Z(c)}, \tag{S4}$$

where $p_t^r(\mathbf{x}|c)$ is reward distribution, and $Z(c) = \int p_t(\mathbf{x}|c)p_t^r(\mathbf{x}|c)\mathrm{d}\mathbf{x}$ is a normalizing constant. Additionally, for the reverse process, the trajectory sampling is defined as [13]:

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla \log p_t(\mathbf{x})]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \tag{S5}$$

where $\nabla \log p_t(\mathbf{x})$ is the score function of $p_t(\mathbf{x})$, directing sampling toward higher-density regions. We prove Theorem 1 here.

Proof. By replacing the sampling distribution $p_t(\cdot)$ with ideal sampling distribution $p_t^I(\cdot)$, the reverse SDE becomes:

$$\begin{aligned}\mathrm{d}\mathbf{x} &= [\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla \log p_t^I(\mathbf{x}|c)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\ &= \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla \left(\log p_t\left(\mathbf{x}|c\right) + \log p_t^r\left(\mathbf{x}|c\right) - \log Z\left(c\right)\right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\ &\overset{\textcircled{1}}{=} \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla \left(\log p_t\left(\mathbf{x}|c\right) + \log p_t^r\left(\mathbf{x}|c\right)\right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}\end{aligned} \tag{S6}$$

where $\textcircled{1}$ holds since $Z(c)$ is a constant and $\nabla_x \log Z(c) = 0$. The proof is completed. □

D.2. Proof of Theorem 2

Theorem 2. *Given the reward distribution $p_t^r(\mathbf{x}|c)$ defined in Eq. (12), the reverse SDE can be rewritten as:*

$$\mathbf{dx} = \Big[\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla\big(\log p_t(\mathbf{x}|c) + R\left(\mathbf{x}_t, c\right)\big)\Big]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}. \tag{S7}$$

Recall the definition of the reward distribution $p_t^r(\mathbf{x}_t|c)$ over noised motion is defined as:

$$p_t^r(\mathbf{x}_t|c) = \frac{\exp\left(R\left(\mathbf{x}_t, c\right)\right)}{Z^r(c)}. \tag{S8}$$

Here, $Z^r(c) = \int \exp(R_\varphi(\mathbf{x}, c))\mathrm{d}\mathbf{x}$ is for normalization. We prove Theorem 2 here.

Proof. By introducing the reward distribution $p_t^r(\mathbf{x}_t|c)$ into Eq. (S5), the reverse SDE can be rewritten as:

$$\begin{aligned}\mathrm{d}\mathbf{x} &= \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla \left(\log p_t\left(\mathbf{x}|c\right) + \log p_t^r\left(\mathbf{x}|c\right)\right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\ &= \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla \left(\log p_t\left(\mathbf{x}|c\right) + R\left(\mathbf{x}|c\right) - \log Z^r(c)\right)\right]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\ &\overset{\textcircled{1}}{=} \Big[\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla\big(\log p_t(\mathbf{x}|c) + R\left(\mathbf{x}_t, c\right)\big)\Big]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w},\end{aligned} \tag{S9}$$

where $\textcircled{1}$ holds since $Z^r(c)$ is a constant and $\nabla_x \log Z^r(c) = 0$. The proof is completed. □

D.3. Proof of Theorem 3

Theorem 3. *Given a reverse SDE defined in Eq. (13), adopting standard DDPM settings [5, 13] where $\mathbf{f}(\mathbf{x}, t) = -\frac{1}{2}\bar{\beta}_{t+\Delta t}\mathbf{x}_t$, $g(t) = \sqrt{\beta_{t+\Delta t}}$, and $\bar{\beta}_t = \frac{\beta_{t+\Delta t}}{\Delta t}$, with time steps $N \to \infty$ and step size $\Delta t = \frac{1}{N}$, the reward-guided denoising process is given by:*

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\bar{\mathbf{x}}_{t-1} + \sqrt{\beta_t}\epsilon\right) + \frac{\beta_t}{\sqrt{\alpha_t}}\nabla R\left(\mathbf{x}_t,\, c\right), \tag{S10}$$

where $\bar{\mathbf{x}}_{t-1} = \mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t, c)$, β_t and α_t are the noise schedule parameters, $\epsilon_\theta(\cdot)$ represents the diffusion model network, and ϵ is Gaussian noise sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Proof. The proof begins with the continuous-time formulation of SDE in Eq. (S8):

$$\mathbf{dx} = \Big[\mathbf{f}(\mathbf{x},t) - g(t)^2\nabla\big(\log p_t(\mathbf{x}|c) + R\left(\mathbf{x}_t, c\right)\big)\Big]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}. \tag{S11}$$

Given a reverse SDE defined in Eq. (13), adopting standard DDPM settings [5, 13] where $\mathbf{f}(\mathbf{x},t) = -\frac{1}{2}\bar{\beta}_{t+\Delta t}\mathbf{x}_t$, $g(t) = \sqrt{\beta_{t+\Delta t}}$, and $\bar{\beta}_t = \frac{\beta_{t+\Delta t}}{\Delta t}$, with time steps $N \to \infty$ and step size $\Delta t = \frac{1}{N}$, the discrete-time formulation of SDE is rewritten as:

$$\begin{aligned} \mathrm{d}\mathbf{x} &= \Big[\mathbf{f}(\mathbf{x},t) - g(t)^2\nabla\left(\log p_t\left(\mathbf{x}|c\right) + \nabla R\left(\mathbf{x}_t, c\right)\right)\Big]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \\ \Rightarrow \mathbf{x}_{t+\Delta t} - \mathbf{x}_t &= \Big[\mathbf{f}(\mathbf{x},t) - g(t)^2\nabla\left(\log p_{t+\Delta t}\left(\mathbf{x}|c\right) + R\left(\mathbf{x}_{t+\Delta t},\, c\right)\right)\Big]\Delta t + g(t)\sqrt{\Delta t}\epsilon, \end{aligned} \tag{S12}$$

By substituting $\mathbf{f}(\mathbf{x},t) = -\frac{1}{2}\bar{\beta}_{t+\Delta t}\mathbf{x}_t$, $g(t) = \sqrt{\beta_{t+\Delta t}}$ into Eq. (S12), we have:

$$\begin{aligned} \mathbf{x}_t \overset{①}{=} & \mathbf{x}_{t+\Delta t} - \Big[\mathbf{f}(\mathbf{x},t) - g(t)^2\nabla\left(\log p_{t+\Delta t}\left(\mathbf{x}|c\right) + R\left(\mathbf{x}_{t+\Delta t},\, c\right)\right)\Big]\Delta t + g(t)\sqrt{\Delta t}\epsilon \\ = & \mathbf{x}_{t+\Delta t} + \frac{1}{2}\bar{\beta}_{t+\Delta t}\mathbf{x}_t\Delta t + \beta_{t+\Delta t}\nabla\left(\log p_{t+\Delta t}\left(\mathbf{x}|c\right) + R\left(\mathbf{x}_{t+\Delta t},\, c\right)\right)\Delta t + \sqrt{\beta_{t+\Delta t}}\sqrt{\Delta t}\epsilon \\ = & \frac{1}{1-\frac{1}{2}\bar{\beta}_{t+\Delta t}\Delta t}\Big[\mathbf{x}_{t+\Delta t} + \beta_{t+\Delta t}\nabla\left(\log p_{t+\Delta t}\left(\mathbf{x}|c\right) + R\left(\mathbf{x}_{t+\Delta t},\, c\right)\right)\Delta t + \sqrt{\beta_{t+\Delta t}}\sqrt{\Delta t}\epsilon\Big] \\ = & \frac{1}{1-\frac{1}{2}\beta_{t+\Delta t}}\Big[\mathbf{x}_{t+\Delta t} + \beta_{t+\Delta t}\nabla\left(\log p_{t+\Delta t}\left(\mathbf{x}|c\right) + R\left(\mathbf{x}_{t+\Delta t},\, c\right)\right)\Delta t + \sqrt{\beta_{t+\Delta t}}\sqrt{\Delta t}\epsilon\Big] \\ \overset{②}{\approx} & \frac{1}{\sqrt{1-\beta_{t+\Delta t}}}\Big[\mathbf{x}_{t+\Delta t} + \beta_{t+\Delta t}\nabla\left(\log p_{t+\Delta t}\left(\mathbf{x}|c\right) + R\left(\mathbf{x}_{t+\Delta t},\, c\right)\right)\Delta t + \sqrt{\beta_{t+\Delta t}}\sqrt{\Delta t}\epsilon\Big] \\ \overset{③}{=} & \frac{1}{\sqrt{1-\beta_{t+\Delta t}}}\Big[\mathbf{x}_{t+\Delta t} + \beta_{t+\Delta t}\Delta t\left(-\frac{1}{\sqrt{1-\bar{\alpha}_{t+\Delta t}}}\epsilon_\theta(x_{t+\Delta t}, t+\Delta t, c) + \nabla R\left(\mathbf{x}_{t+\Delta t},\, c\right)\right) + \sqrt{\beta_{t+\Delta t}}\sqrt{\Delta t}\epsilon\Big] \end{aligned} \tag{S13}$$

where ① holds since $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a Gaussian noise, i.e., $\epsilon = -\epsilon$, ② holds since the following relationship satisfies:

$$\sqrt{1-\beta_{t+\Delta t}} = 1 - \frac{1}{2}\beta_{t+\Delta t} + \mathcal{O}(\beta_{t+\Delta t}^2), \tag{S14}$$

and ③ holds since the score-based models meet the relationship [13]:

$$\nabla\log p_{t+\Delta t}\left(\mathbf{x}|c\right) = -\frac{1}{\sqrt{1-\bar{\alpha}_{t+\Delta t}}}\epsilon_\theta(x_{t+\Delta t}, t+\Delta t, c). \tag{S15}$$

Let $\Delta t = 1$, we have:

$$\begin{aligned} \mathbf{x}_t = & \frac{1}{\sqrt{1-\beta_{t+1}}}\Big[\mathbf{x}_{t+1} + \beta_{t+1}\left(-\frac{1}{\sqrt{1-\bar{\alpha}_{t+1}}}\epsilon_\theta(x_{t+\Delta t}, t+\Delta t, c) + \nabla R\left(\mathbf{x}_{t+1},\, c\right)\right) + \sqrt{\beta_{t+1}}\epsilon\Big] \\ = & \frac{1}{\sqrt{\alpha_{t+1}}}\Big[\mathbf{x}_{t+1} - \frac{\beta_{t+1}}{\sqrt{1-\bar{\alpha}_{t+1}}}\epsilon_\theta(x_{t+\Delta t}, t+1, c) + \beta_{t+1}\nabla R\left(\mathbf{x}_{t+1},\, c\right) + \sqrt{\beta_{t+1}}\epsilon\Big] \\ = & \frac{1}{\sqrt{\alpha_{t+1}}}\Big[\mathbf{x}_{t+1} - \frac{\beta_{t+1}}{\sqrt{1-\bar{\alpha}_{t+1}}}\epsilon_\theta(x_{t+\Delta t}, t+1, c) + \sqrt{\beta_{t+1}}\epsilon\Big] + \frac{\beta_{t+1}}{\sqrt{\alpha_{t+1}}}\nabla R\left(\mathbf{x}_{t+1},\, c\right) \\ = & \frac{1}{\sqrt{\alpha_{t+1}}}\Big[\bar{\mathbf{x}}_{t+1} + \sqrt{\beta_{t+1}}\epsilon\Big] + \frac{\beta_{t+1}}{\sqrt{\alpha_{t+1}}}\nabla R\left(\mathbf{x}_{t+1},\, c\right), \end{aligned} \tag{S16}$$

where $\bar{\mathbf{x}}_{t+1} = \mathbf{x}_{t+1} - \frac{\beta_{t+1}}{\sqrt{1-\bar{\alpha}_{t+1}}}\epsilon_\theta(x_{t+1}, t+1, c)$. The proof is completed. □

D.4. Discussion about Eq. (15)

A critical insight from Theorem 3 is that the influence of the reward gradient $\nabla R(\mathbf{x}_t, c)$ increases with timestep t, owing to the typical scheduling of β_t and α_t in DDPM. While this reinforcement is beneficial in later timesteps where structure has emerged, it may destabilize early denoising stages, where the motion remains heavily corrupted by noise. In such cases, the denoiser $\epsilon_\theta(\mathbf{x}_t, t, c)$ plays a crucial role in recovering coarse motion structures, and an excessive reward signal could disrupt this process. To mitigate this instability, we remove the weight $\frac{\beta_t}{\sqrt{\alpha_t}}$ from the reward term, leading to a revised denoising process:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\bar{\mathbf{x}}_{t-1} + \sqrt{\beta_t} \epsilon \right) + \nabla R(\mathbf{x}_t, c). \tag{S17}$$

This modification ensures that the reward signal remains a guiding force across all timesteps without overwhelming the early denoising stages, preserving the balance between semantic alignment and motion coherence.

References

[1] 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*, pages 18000–18010, 2023. 1, 2, 4

[2] Zhongzhi Chen, Guang Liu, Bo-Wen Zhang, Fulong Ye, Qinghong Yang, and Ledell Wu. Altclip: Altering the language encoder in clip for extended language capabilities. *arXiv preprint arXiv:2211.06679*, 2022. 3

[3] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019. 3

[4] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5142–5151, 2022. 2, 4

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 6, 7

[6] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 2

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[8] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 4

[9] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision*, pages 480–497. Springer, 2022. 4

[10] Mathis Petrovich, Michael J. Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *International Conference on Computer Vision (ICCV)*, 2023. 4

[11] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*, 2019. 4

[12] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016. 2

[13] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 6, 7

[14] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 2