LUNAR: LATENT UNITS WITH NATURAL-LANGUAGE ANCHORS FOR REASONING IN MOTION GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose LUNAR, a structured textual–latent reasoning framework built on a core principle: *use language to scope latent reasoning, and motion to ground it*. LUNAR organizes complex motion into atomic motions and represents each as a paired textual–latent reasoning unit. The textual block specifies the action intent and how it is realized across the body, defining the semantic scope of the latent block. The latent block is grounded in the corresponding motion segment through a pretrained text–motion space. After completing the reasoning trajectory, LUNAR predicts motion tokens that can be decoded into 3D motion by a frozen motion codec. We train LUNAR with progressive supervised fine-tuning followed by trajectory-level group relative policy optimization. Experiments show that LUNAR establishes state-of-the-art in-domain (ID) text–motion alignment on HumanML3D, achieving a Top-3 R-Precision of 0.838 and an MM-Dist of 2.714. In out-of-domain (OOD) evaluation across compositional, paraphrastic, and cross-dataset shifts, LUNAR consistently outperforms all compared baselines, improving overall R@3 from 0.517 to 0.593 in the T2M space and from 0.606 to 0.696 in the independent LaMP space. *Our code will be publicly available upon acceptance.*

1 INTRODUCTION

Text-to-motion (T2M) generation aims to synthesize human motion that faithfully realizes a natural-language instruction. A growing body of work represents motion as quantized tokens (Guo et al., 2022b; Zhang et al., 2023a; Guo et al., 2024; Pinyoanuntapong et al., 2024), allowing pretrained large language models (LLMs) to predict motion tokens from text using their language priors and instruction-following capabilities (Jiang et al., 2023; Wu et al., 2024; 2025b; Wang et al., 2026c). Recent methods introduce textual plans or reasoning traces to articulate complex motion semantics more explicitly (Jiang et al., 2024; Ouyang et al., 2025; Zheng et al., 2026).

Yet reasoning in language is not the same as representing motion realization: it can articulate what should happen while still omitting how the motion unfolds through the body. As illustrated in Figure 1, a textual plan may capture the coarse intent of lowering the body while failing to preserve the requested quadrupedal configuration and forward progression. This exposes a representation bottleneck: language is well suited for expressing discrete action structure, but not every aspect of motion realization is naturally or compactly verbalized.

A natural alternative is to reason through continuous latent states, as explored in recent multimodal latent-reasoning studies (Zhang et al., 2025; Wang et al., 2026b; Hao et al., 2024; Xu et al., 2025b). Yet continuous states introduce the opposite problem. Unlike language, an unconstrained latent state has no explicit semantic scope: it is unclear which part of a complex motion it represents, or whether it encodes motion-relevant information at all. This raises a central research question:

> *How can continuous latent reasoning be given explicit semantic scope and motion grounding, so that it complements textual reasoning instead of an unconstrained hidden channel?*

Concretely, this requires answering two coupled questions: *what* structure should the latent states be organized into and *what* motion content should each latent block represent?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

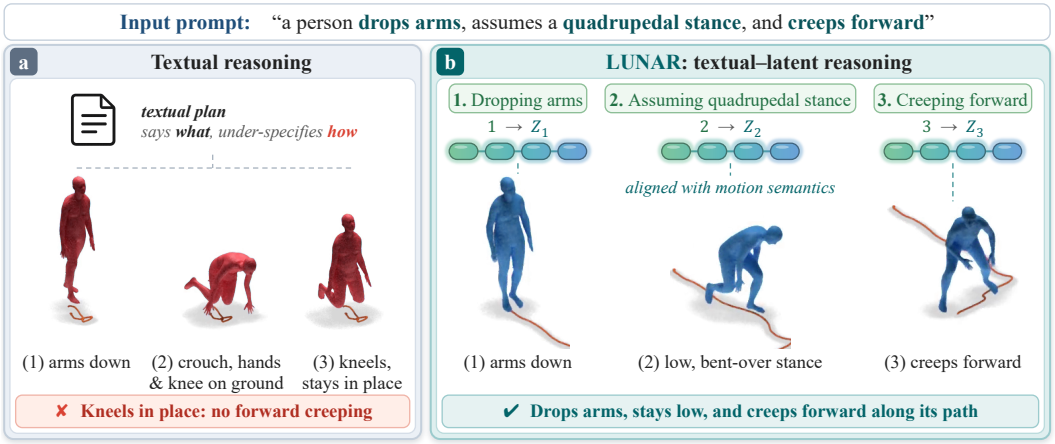


Figure 1: Illustrative comparison between textual reasoning and latent reasoning. Textual reasoning (left) simplifies the requested quadrupedal stance to a kneeling motion, missing *creeping forward*. Latent reasoning (right) correctly lowers the arms, assumes a quadrupedal stance, and creeps forward in a coherent and plausible sequence.

We answer this question with **LUNAR**, a structured textual–latent reasoning framework for T2M generation. LUNAR follows a core design principle: *use language to scope latent reasoning, and motion to ground it*. It organizes a complex motion into atomic motions, each corresponding to a short temporal unit with a coherent action intent. Each atomic motion is represented as a paired textual–latent reasoning unit, consisting of a textual block followed by its associated latent block. The textual block specifies the action intent and its realization across the body, defining the semantic scope of the latent block; the latent block is then grounded by aligning it with the corresponding motion segment in a pretrained text–motion space. Across atomic motions, these paired units form the reasoning trajectory from which LUNAR predicts the final motion tokens. To learn this reasoning process, we first use progressive supervised fine-tuning (SFT) for initialization, followed by trajectory-level group relative policy optimization (GRPO) over self-generated rollouts with sequence-level textual-motion rewards.

We evaluate LUNAR in both in-domain (ID) and out-of-domain (OOD) settings. For ID evaluation on HumanML3D, LUNAR achieves R-Precision of $0.554/0.748/0.838$ at Top-1/2/3 and an MM-Dist of 2.714, outperforming all compared methods in text–motion alignment. For OOD evaluation, we consider three complementary distribution shifts covering compositional instructions, paraphrased language, and cross-dataset motion transfer. Across these OOD settings, LUNAR improves overall R@3 from 0.517 to 0.593 in the standard T2M space and from 0.606 to 0.696 under the independent Language-Motion Pretraining (LaMP) evaluator (Li et al., 2025b).

Our main contributions are three-fold:

- We introduce LUNAR, a structured textual–latent reasoning method, organizing complex motion into atomic motions and represents each as a paired textual–latent reasoning unit.
- We develop a motion-grounded latent reasoning strategy in which language defines the semantic scope of each latent block, while the corresponding motion segment provides its grounding in a pretrained text–motion space.
- We show that LUNAR consistently improves text–motion alignment in both ID and OOD settings, including compositional, paraphrastic, and cross-dataset distribution shifts, with gains that also hold in the independent LaMP space.

2 Related Work

Text-to-motion generation. T2M generation has been studied with both continuous and discrete motion representations. Continuous methods model motion in trajectory or learned feature spaces.

TEMOS (Petrovich et al., 2022) learns text-conditioned motion distributions with a variational framework, while MLD (Chen et al., 2023) performs diffusion in a compressed motion latent space. MDM (Tevet et al., 2023) and MotionDiffuse (Zhang et al., 2022) instead apply diffusion directly to pose sequences under textual conditioning. Discrete methods first quantize motion using a VQ-VAE and then predict the resulting code sequences, either autoregressively, as in T2M-GPT (Zhang et al., 2023a), or through masked modeling, as in MoMask (Guo et al., 2024) and MMM (Pinyoanuntapong et al., 2024). These paradigms achieve strong motion quality and text alignment on benchmarks such as HumanML3D (Guo et al., 2022a). FineMoGen (Zhang et al., 2023b) models fine-grained spatiotemporal composition, while SINC (Athanasiou et al., 2023) uses language-derived action–body-part mappings to compose simultaneous motions. LUNAR focuses on language-scoped continuous reasoning units supervised by corresponding motion segments within an LLM generation trajectory.

Motion-language models and explicit reasoning in T2M. Motion-language models expose discrete motion codes or projected motion features to pretrained LLMs, allowing text and motion to be modeled within a shared sequential interface. MotionGPT (Jiang et al., 2023), MotionLLM (Wu et al., 2024), Motion-Agent (Wu et al., 2025b), and MotionGPT-2 (Wang et al., 2026c) demonstrate instruction-following motion understanding and generation under this formulation. More recent methods introduce explicit planning or reasoning before motion prediction. MotionChain (Jiang et al., 2024) incorporates conversational reasoning chains, IRG-MotionLLM (Li et al., 2025a) iteratively generates and refines motion, Motion-R1 (Ouyang et al., 2025) combines decomposed chain-of-thought reasoning with RL, and UniMo (Wang et al., 2026a) unifies multiple motion-reasoning tasks. Re2MoGen (Zheng et al., 2026) further explores open-vocabulary planning. Yet, textual planning and reasoning for motion can specify what should occur, but they do not fully describe how each motion is realized across the body and evolves over time.

Latent and multimodal reasoning. Recent multimodal reasoning methods move beyond fully verbalized intermediate reasoning by introducing modality-specific or continuous internal states. Visual Planning (Xu et al., 2025a), DeepEyes (Zheng et al., 2025), and MMSearch-R1 (Wu et al., 2025a) incorporate generated visual artifacts or tool outputs into subsequent reasoning. Latent Sketchpad (Zhang et al., 2025) and Monet (Wang et al., 2026b) maintain continuous visual representations, while Coconut (Hao et al., 2024) and SoftCoT (Xu et al., 2025b) propagate hidden states without decoding every intermediate step into language. These methods motivate nonverbal intermediate states. LUNAR specifies their motion-semantic scope through atomic-motion/body-group text and supervises them with corresponding motion segments in a pretrained text–motion space.

3 METHOD

3.1 FRAMEWORK OVERVIEW

Motion-token interface. LUNAR couples a decoder-only language-model backbone with a frozen VQ-VAE motion codec through a discrete motion-token interface. The codec comprises an encoder $\mathcal{E}_{\mathrm{vq}}$, a quantizer Q, and a decoder $\mathcal{D}_{\mathrm{vq}}$. Let $\mathcal{C}_{\mathrm{vq}}$ denote its codebook, with size $K_{\mathrm{vq}} = |\mathcal{C}_{\mathrm{vq}}|$. Given a target motion m, the encoder maps it to a sequence of spatio-temporal features, and the quantizer assigns each feature to its nearest codebook entry, yielding the target motion-token sequence $y^\star = (y_1^\star, \ldots, y_L^\star) = Q(\mathcal{E}_{\mathrm{vq}}(m))$. Each token $y_l^\star \in \{0, \ldots, K_{\mathrm{vq}} - 1\}$ represents four motion frames.

We instantiate the language-model backbone with Qwen3-1.7B (Yang et al., 2025) and augment its vocabulary with K_{vq} motion tokens, establishing a one-to-one correspondence between the added tokens and the VQ codebook entries. Given a text prompt x describing the target motion m, the language-model backbone performs reasoning and predicts a motion-token sequence $y = (y_1, \ldots, y_L)$. These tokens select the corresponding codebook vectors, which are decoded by the frozen VQ-VAE decoder to produce $\hat{m} = \mathcal{D}_{\mathrm{vq}}(y)$, a prompt-conditioned estimate of m.

Structured textual–latent reasoning. Before predicting y, LUNAR produces a prompt-level trace τ to identify and order atomic motions. It then generates paired textual–latent units for segments m_i^{seg}, $i = 1, \ldots, N$, each with a coherent action intent. The textual block (a_i, B_i) describes the action and its body-group realization, anchoring the semantic scope of the following latent block Z_i. Projected latent states are aligned with the corresponding motion segment in a frozen text–motion space, supplying motion-specific supervision to the intermediate trajectory.

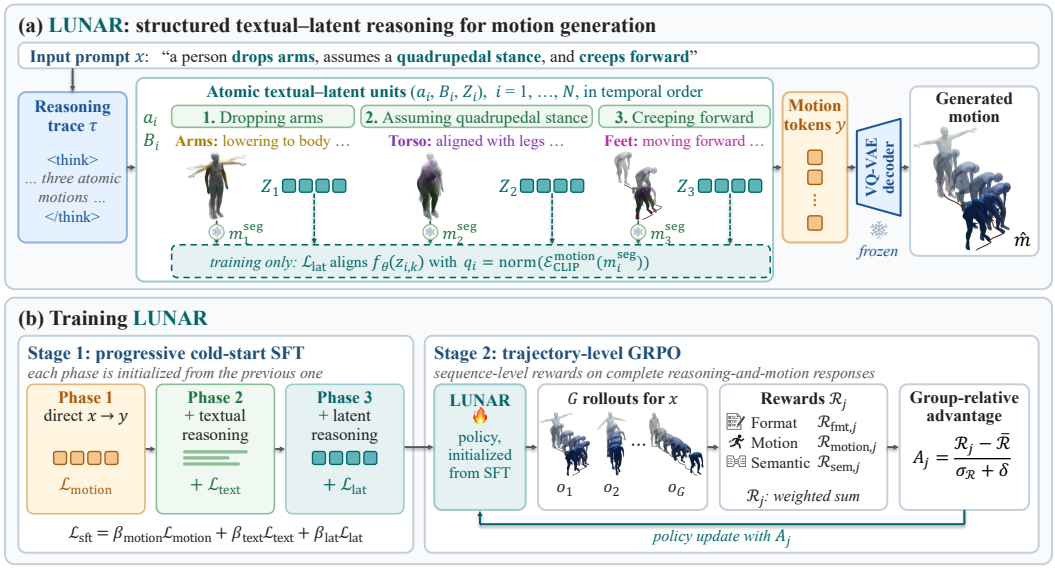


Figure 2: Overview of LUNAR. (a) Generation follows a prompt-level trace τ, temporally ordered textual–latent units (a_i, B_i, Z_i), and motion tokens y, decoded by a frozen VQ-VAE. Each latent block has four continuous states; the dashed training-only path aligns their projections with the corresponding ground-truth motion segment. (b) Progressive cold-start SFT introduces motion prediction, textual reasoning, and latent alignment in sequence. GRPO then refines complete responses using group-relative advantages from format, motion, and semantic rewards.

Figure 3 provides a concrete serialized example of the interleaved textual–latent trajectory. The overall generation process is

$$\begin{aligned} x \quad \rightarrow \quad & \tau \rightarrow (a_1, B_1, Z_1) \rightarrow \cdots \\ & \rightarrow (a_N, B_N, Z_N) \rightarrow y \rightarrow \hat{m}. \end{aligned} \tag{1}$$

Two-stage training. LUNAR is trained in two stages, as illustrated in Figure 2. Stage 1 uses progressive cold-start supervised fine-tuning (SFT) to establish the structured reasoning-and-motion generation process through teacher-forced imitation of reference outputs. Stage 2 applies group relative policy optimization (GRPO) to refine self-generated trajectories, using sequence-level rewards to improve the reasoning and resulting motion beyond supervised imitation.

We detail the model components and training objectives in the remainder of this section, with additional technical details provided in Appendix A.

3.2 Atomic-Motion Reasoning in Language

Each atomic motion is described at two complementary levels. The atomic-motion description a_i summarizes the overall intent of the temporal segment, including its principal action, displacement, and orientation. LUNAR further expands this summary into a body-group realization:

$$B_i = \{b_{i,g}\}_{g \in \mathcal{G}}, \tag{2}$$

where $\mathcal{G}$ comprises ten groups: the head, torso, left and right arms, left and right hands, left and right legs, and left and right feet. Each field $b_{i,g}$ specifies how the corresponding body group participates in the segment, including its movement or maintained state, its functional role in support or contact, and its coordination with the other groups.

For example, in the decomposition of *a person drops arms, assumes a quadrupedal stance, and creeps forward*, a_1 is *dropping arms*, the atomic-motion description for the first segment. Its B_1 localizes this motion by specifying that the arms lower from their raised positions while the relaxed hands follow them. This decomposition turns an atomic-motion description into an explicit body-wise execution plan.

User prompt a person **drops arms**, assumes a **quadrupedal stance**, and **creeps forward**

Assistant response **<think>** ... three atomic motions ... **</think>**

<output>

1. Dropping arms:
Left arm:
lowering from raised position to side
...
Left hand:
relaxed, following left arm
<LS> $z_{1,1}, \ldots, z_{1,K_{\text{lat}}}$ **<LE>**

2. Assuming quadrupedal stance:
Torso:
aligned with legs
...
Left leg:
bent at knee
<LS> $z_{2,1}, \ldots, z_{2,K_{\text{lat}}}$ **<LE>**

3. Creeping forward:
Torso:
upright with slight forward lean
...
Left foot:
flat on ground, moving forward
<LS> $z_{3,1}, \ldots, z_{3,K_{\text{lat}}}$ **<LE>**

</output>

<motion> **<Motion_315>** **<Motion_315>** ... **</motion>**

Figure 3: Abbreviated structured response. Blue identifies structural tags, green identifies the three atomic motions and representative body-group fields, and orange identifies quantized motion tokens. Each textual block (a_i, B_i) is immediately followed by $z_{i,1}, \ldots, z_{i,K_{\text{lat}}}$. <LS> and <LE> denote <latent_start> and <latent_end>, respectively.

3.3 Atomic-Motion Reasoning in Latent Space

The preceding textual reasoning in language provides a semantic anchor for each atomic-motion unit. However, some aspects of how the motion unfolds are difficult to express explicitly in language. We therefore introduce a latent block Z_i after each textual block (a_i, B_i) to capture complementary motion-specific information. To ensure that these latent states represent the intended atomic motion rather than unconstrained hidden features, we align them with the corresponding motion segment in a frozen CLIP-style text–motion semantic space.

Latent representation. The latent block associated with atomic motion i is:

$$Z_i = \langle \texttt{latent_start} \rangle \, z_{i,1}, \ldots, z_{i,K_{\text{lat}}} \, \langle \texttt{latent_end} \rangle. \tag{3}$$

The special tokens <latent_start> and <latent_end> delimit the block. We use $K_{\text{lat}} = 4$ context-dependent states per atomic motion. At inference, each projected state is mapped back through g_θ and fed to the next decoding step; latent SFT instead uses target feedback $g_\theta(q_i)$ (Appendix A).

Cross-space semantic alignment. Let $\mathcal{E}_{\text{CLIP}}^{\text{motion}}$ and $\mathcal{E}_{\text{CLIP}}^{\text{text}}$ be the frozen motion and text encoders (Guo et al., 2022a). A learnable projector maps each language-model state into their shared space: $h_{i,k} = f_\theta(z_{i,k})$. During training, the frozen motion branch maps the paired ground-truth segment m_i^{seg} to the normalized latent target $q_i = \text{norm}(\mathcal{E}_{\text{CLIP}}^{\text{motion}}(m_i^{\text{seg}}))$. We assign ordered text units to non-overlapping motion segments using velocity-valley boundaries, with uniform partition as a fallback (Appendix A). Thus, q_i is motion-derived; this heuristic assignment is not a semantic boundary annotation. Concurrent actions may overlap, so the one-to-one assignment approximates their temporal structure. Let $\mathcal{P} = \{(i,k) \mid 1 \le i \le N,\ 1 \le k \le K_{\text{lat}}\}$ index all latent slots. We supervise every projected state using the target for its corresponding atomic motion:

$$\mathcal{L}_{\text{lat}} = \frac{1}{|\mathcal{P}|} \sum_{(i,k)\in\mathcal{P}} \|h_{i,k} - q_i\|_2^2. \tag{4}$$

3.4 Progressive Cold-Start SFT

Learning the complete reasoning-and-motion trajectory from scratch is challenging because the model must simultaneously acquire motion-token prediction, textual motion reasoning, and latent reasoning. We therefore train LUNAR with a progressive three-phase supervised curriculum. The first phase establishes the motion-token interface by learning direct text-to-motion prediction. The second phase introduces the textual reasoning while retaining motion-token supervision. The final phase further incorporates the latent reasoning guided by the alignment in the CLIP-style text-motion

space. Each phase is initialized from the preceding one.

$$\mathcal{L}_{\text{sft}} = \beta_{\text{motion}}\mathcal{L}_{\text{motion}} + \beta_{\text{text}}\mathcal{L}_{\text{text}} + \beta_{\text{lat}}\mathcal{L}_{\text{lat}}, \tag{5}$$

where $\mathcal{L}_{\text{motion}}$ supervises motion-token prediction, $\mathcal{L}_{\text{text}}$ supervises textual reasoning, and $\mathcal{L}_{\text{lat}}$ enforces motion-semantic latent alignment. The coefficients β_{motion}, β_{text}, and β_{lat} weight the three objectives. Each phase activates only the loss terms associated with its introduced components.

3.5 Trajectory-Level Reinforcement Learning

Progressive cold-start SFT establishes the structured reasoning-and-motion generation process through teacher-forced imitation. However, its token-level objectives do not directly optimize complete responses generated by the model. In Stage 2, we therefore apply GRPO (Shao et al., 2024) to refine the full reasoning trajectory and the resulting motion using sequence-level rewards.

For each prompt x, the rollout policy samples a group of G complete responses $\{o_j\}_{j=1}^{G}$. Each response receives a reward $\mathcal{R}_j = \mathcal{R}(x, o_j)$, from which we compute the group-relative advantage:

$$\bar{\mathcal{R}} = \frac{1}{G}\sum_{j=1}^{G}\mathcal{R}_j, \qquad A_j = \frac{\mathcal{R}_j - \bar{\mathcal{R}}}{\sigma_{\mathcal{R}} + \delta}, \tag{6}$$

where $\sigma_{\mathcal{R}}$ is the standard deviation of the group rewards and $\delta > 0$ ensures numerical stability. Following standard GRPO, a positive A_j increases the likelihood of response o_j, whereas a negative A_j decreases it.

Each response receives a *format reward*, *motion reward* and *semantic reward*. Let y_j be the predicted motion-token sequence in o_j and $\hat{m}_j = \mathcal{D}_{\text{vq}}(y_j)$ its decoded motion; x and m remain the input prompt and target motion defined above.

Format reward. Invalid formatting can make the generated response unparsable or prevent the motion tokens from being decoded correctly. We therefore define a format indicator $\mathbf{1}_{\text{fmt},j} = \mathbf{1}_{\text{fmt}}(o_j)$, which equals one only when response o_j satisfies all required structural constraints, including valid tag ordering, complete body-group fields, the prescribed latent slots, and a parseable motion-token sequence. The format reward is then defined as:

$$\mathcal{R}_{\text{fmt},j} = \mathbf{1}_{\text{fmt},j}. \tag{7}$$

Motion reward. It measures the fidelity between the generated and reference motions:

$$\mathcal{R}_{\text{motion},j} = \mathbf{1}_{\text{fmt},j}\, C(\mathcal{E}_{\text{CLIP}}^{\text{motion}}(\hat{m}_j), \mathcal{E}_{\text{CLIP}}^{\text{motion}}(m)), \tag{8}$$

where we adopt a normalized cosine similarity function $C(u, v) = (1 + \cos(u, v))/2$.

Semantic reward. It measures the alignment between the generated motion and the input text:

$$\mathcal{R}_{\text{sem},j} = \mathbf{1}_{\text{fmt},j}\, C(\mathcal{E}_{\text{CLIP}}^{\text{motion}}(\hat{m}_j), \mathcal{E}_{\text{CLIP}}^{\text{text}}(x)). \tag{9}$$

The final trajectory reward is:

$$\mathcal{R}_j = \lambda_{\text{motion}}\mathcal{R}_{\text{motion},j} + \lambda_{\text{sem}}\mathcal{R}_{\text{sem},j} + \lambda_{\text{fmt}}\mathcal{R}_{\text{fmt},j}, \tag{10}$$

where λ_{motion}, λ_{sem}, and λ_{fmt} are the weight terms.

4 Experiments

4.1 Experimental Settings

Training datasets. All training stages use the HumanML3D training split (Guo et al., 2022a). The direct-motion SFT stages uses its caption–motion pairs. To construct supervision for textual reasoning, DeepSeek-R1 (DeepSeek-AI, 2025) generates internal prompt-level reasoning traces, atomic-motion descriptions, and ten body-group fields, while Qwen3-235B (Yang et al., 2025) serves as a verifier for faithfulness, completeness, atomicity and clarity. The augmented HumanML3D dataset provides the reasoning traces, target VQ motion-token sequences, and segment-level motion-semantic targets used for latent alignment. Neither the teacher nor the verifier is used at inference.

Table 1: Comparison of text-to-motion generation on HumanML3D. R-Precision measures text–motion retrieval accuracy at Top-1/2/3; FID measures distributional fidelity, MM-Dist measures text–motion embedding distance, and Diversity measures motion variation. Arrows indicate preferred directions. The best and second-best results in each column are highlighted in **bold** and underline, respectively. The shaded row denotes LUNAR.

Method	R-Precision ↑			FID ↓	MM-Dist ↓	Diversity ↑
	Top 1	Top 2	Top 3			
Non-LLM based text-to-motion methods						
MDM (2023)	$0.320 \pm .005$	$0.498 \pm .004$	$0.611 \pm .007$	$0.544 \pm .044$	$5.566 \pm .027$	$9.559 \pm .086$
MLD (2023)	$0.481 \pm .003$	$0.673 \pm .003$	$0.772 \pm .002$	$0.473 \pm .013$	$3.196 \pm .010$	$9.724 \pm .082$
MotionDiffuse (2022)	$0.491 \pm .001$	$0.681 \pm .001$	$0.782 \pm .001$	$0.630 \pm .001$	$3.113 \pm .001$	$9.410 \pm .049$
T2M (2022a)	$0.457 \pm .002$	$0.639 \pm .003$	$0.740 \pm .003$	$1.067 \pm .002$	$3.340 \pm .008$	$9.188 \pm .002$
TM2T (2022b)	$0.424 \pm .003$	$0.618 \pm .003$	$0.729 \pm .002$	$1.501 \pm .017$	$3.467 \pm .011$	$8.589 \pm .076$
T2M-GPT (2023a)	$0.491 \pm .003$	$0.680 \pm .003$	$0.775 \pm .002$	$\underline{0.116 \pm .004}$	$3.118 \pm .011$	$9.761 \pm .081$
MoMask (2024)	$\underline{0.521 \pm .002}$	$0.713 \pm .002$	$0.807 \pm .002$	$\mathbf{0.045 \pm .002}$	$2.958 \pm .008$	$9.620 \pm .064$
LLM-based text-to-motion methods						
MotionGPT (2023)	$0.492 \pm .003$	$0.681 \pm .003$	$0.778 \pm .002$	$0.232 \pm .008$	$3.096 \pm .008$	$9.528 \pm .071$
MotionChain (2024)	$0.504 \pm .003$	$0.695 \pm .003$	$0.790 \pm .003$	$0.248 \pm .009$	$3.033 \pm .010$	$9.470 \pm .075$
MotionLLM (2024)	$0.515 \pm .004$	$0.691 \pm .003$	$0.801 \pm .004$	$0.230 \pm .009$	$2.967 \pm .020$	$\underline{9.908 \pm .102}$
MotionGPT-2 (2026c)	$0.496 \pm .002$	$0.691 \pm .003$	$0.782 \pm .004$	$0.191 \pm .004$	$3.080 \pm .013$	$9.860 \pm .026$
Motion-R1 (2025)	$0.515 \pm .003$	$\underline{0.719 \pm .002}$	$\underline{0.818 \pm .002}$	$0.201 \pm .004$	$\underline{2.854 \pm .010}$	$\mathbf{10.026 \pm .075}$
LUNAR (Ours)	$\mathbf{0.554 \pm .005}$	$\mathbf{0.748 \pm .002}$	$\mathbf{0.838 \pm .004}$	$0.247 \pm .023$	$\mathbf{2.714 \pm .015}$	$9.836 \pm .202$

Evaluation datasets. We evaluate in-domain generation on the official HumanML3D evaluation set of 4,648 motions, following the standard T2M protocol with repeat-3 reporting. To examine whether semantic alignment transfers beyond the language and motion of HumanML3D, we additionally construct three complementary OOD splits. *OOD-Comp* evaluates temporal and concurrent compositions of familiar actions using 200 deterministic prompts constructed by combining either two source captions (150 prompts) or three source captions (50 prompts). *OOD-Para* evaluates linguistic variation using 196 LLM-generated paraphrased prompts. *OOD-Motion-X* evaluates cross-dataset transfer using 566 paired single-person examples from Motion-X (Lin et al., 2023); we remove multi-person captions and captions that match HumanML3D, and convert all motions to the 263-D representation used by the evaluators. The overall OOD score is the unweighted mean of the three split-level retrieval scores. Appendix B.1 provides further construction and evaluation details.

Baselines and comparison protocol. For in-domain evaluation, we compare with MDM (Tevet et al., 2023), MLD (Chen et al., 2023), MotionDiffuse (Zhang et al., 2022), T2M (Guo et al., 2022a), TM2T (Guo et al., 2022b), T2M-GPT (Zhang et al., 2023a), MoMask (Guo et al., 2024), MotionGPT (Jiang et al., 2023), MotionChain (Jiang et al., 2024), MotionLLM (Wu et al., 2024), MotionGPT-2 (Wang et al., 2026c), and Motion-R1 (Ouyang et al., 2025). The OOD comparison covers three representative T2M paradigms: MLD for latent diffusion generation, MoMask for VQ-VAE generation, and Motion-Agent (Wu et al., 2025b) for LLM-driven motion modeling.

Evaluation metrics. R-Precision@1/2/3 and multimodal distance (MM-Dist) measure text–motion alignment through retrieval accuracy and embedding distance, respectively. Fréchet inception distance (FID) evaluates distributional fidelity to the reference motions as a proxy for motion quality, while Diversity measures variation among generated motions. To reduce dependence on a single semantic encoder, we evaluate OOD R@3 in both the standard T2M (Guo et al., 2022a) and the independent LaMP (Li et al., 2025b) spaces.

Implementation details. LUNAR uses Qwen3-1.7B (Yang et al., 2025) as its language-model backbone, the frozen Motion-Agent VQ-VAE motion tokenizer with a 512-entry codebook (Wu et al., 2025b), and a pretrained CLIP-style T2M semantic encoder (Guo et al., 2022a) for latent targets and semantic reward scoring. Full-parameter SFT uses a learning rate of 5×10^{-5} for up to 300 epochs in Phase 1 and up to 80 epochs in each of Phases 2 and 3, with early stopping. GRPO is trained for 3 epochs using a learning rate of 5×10^{-7} and a group size of $G = 5$ responses per

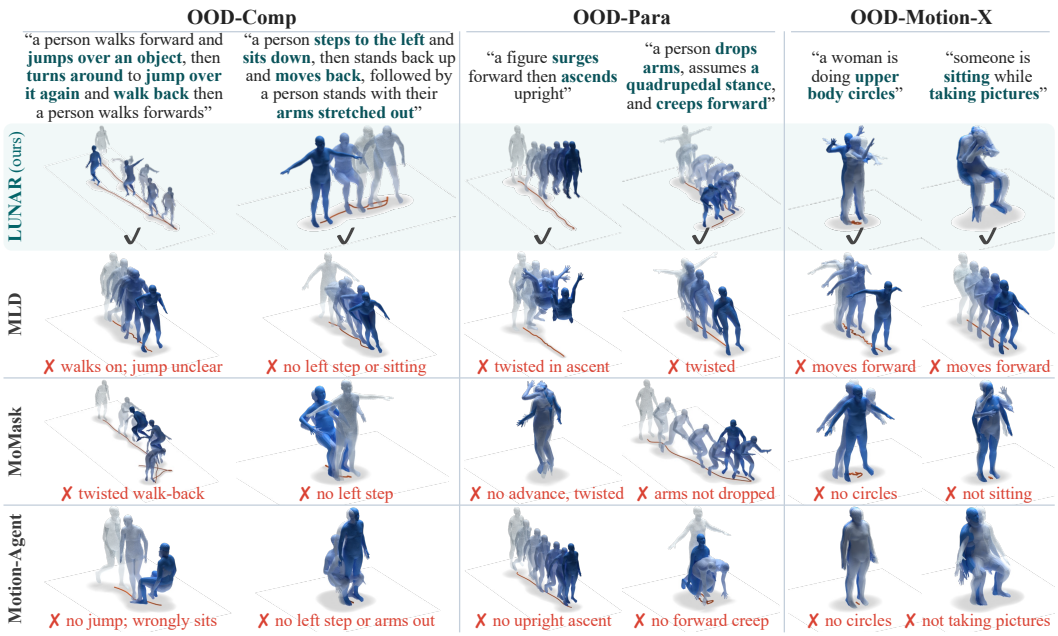


Figure 4: Qualitative comparisons on OOD-Comp (multi-step composition), OOD-Para (linguistic variation), and OOD-Motion-X (cross-dataset transfer), with two prompts per split. Highlighted phrases specify the requested actions and constraints; the shaded row shows LUNAR. Overlaid poses depict each generated sequence. Checkmarks in the shaded row indicate action coverage, while red annotations with crosses identify missed actions or inconsistent body configurations.

Table 2: Comparison of OOD retrieval under T2M and LaMP. Comp., Para., and Motion-X cover composition, paraphrasing, and cross-dataset transfer; Overall is their macro-average R@3. Central values are greedy scores; ± gives approximate 95% normal half-widths from a separate 20-pass stochastic study, not intervals for the greedy scores (Appendix B.2). In each column, **bold** and underline mark the best and second-best greedy scores, respectively; shading denotes LUNAR.

Method	Overall		Comp.		Para.		Motion-X	
	T2M	LaMP	T2M	LaMP	T2M	LaMP	T2M	LaMP
MLD	.440±.005	.413±.005	.451±.008	.415±.009	.459±.012	.406±.009	.412±.006	.418±.004
MoMask	.480±.004	.551±.005	.525±.007	.622±.007	.513±.006	.535±.009	.401±.004	.496±.006
Motion-Agent	.517±.004	.606±.004	.504±.007	.598±.007	.635±.005	.730±.009	.413±.005	.491±.005
LUNAR	**.593**±.004	**.696**±.004	**.654**±.007	**.770**±.008	**.653**±.005	**.748**±.007	**.471**±.006	**.571**±.005

prompt. Evaluation uses a single GRPO checkpoint with greedy decoding and deterministic latent-mean feedback, without response reranking (complete settings in Appendix *Training Configuration*).

4.2 Quantitative Results

In-Domain Quantitative Results. Table 1 reports R-Precision of 0.554/0.748/0.838 at Top-1/2/3 and MM-Dist of 2.714 for LUNAR, improving over the baselines listed there. These metrics indicate stronger text–motion alignment within the evaluated comparison.

FID measures marginal distributional fidelity, and LUNAR's 0.247 remains higher than MoMask's 0.045 and T2M-GPT's 0.116. Thus, the alignment gains do not establish superior distributional motion quality. Conversely, low FID does not ensure that individual actions or body-specific constraints are realized, as the qualitative examples illustrate. Diversity measures dataset-level variation; neither it nor retrieval establishes human-perceived naturalness or physical correctness.

Table 3: Cumulative ablation on HumanML3D (ID) and three OOD splits. Body, Lat., and RL denote body-group text, continuous latent reasoning, and reinforcement learning, respectively. ID $\pm$ values are approximate 95% normal half-widths over three evaluator rounds on cached outputs; OOD scores are greedy macro-averages (Appendix B.5). In each metric column, **bold** and underline mark the best and second-best results, respectively; shading denotes LUNAR.

Variant	Modules			ID		OOD R@3 ↑	
	Body	Lat.	RL	FID ↓	R@3 ↑	T2M	LaMP
Baseline	–	–	–	$\underline{0.290 \pm .021}$	$0.776 \pm .001$	0.515	0.586
Baseline-RL	–	–	✓	$0.439 \pm .014$	$0.810 \pm .009$	0.567	0.647
Baseline-RL-Reasoning	✓	–	✓	$0.480 \pm .028$	$\underline{0.825 \pm .004}$	$\underline{0.576}$	$\underline{0.665}$
LUNAR	✓	✓	✓	$\mathbf{0.247 \pm .023}$	$\mathbf{0.838 \pm .004}$	**0.593**	**0.696**

Out-of-Domain Quantitative Results. Table 2 reports greedy-reference R@3 for each OOD split and their equal-weight average. LUNAR improves overall T2M/LaMP R@3 over Motion-Agent from 0.517/0.606 to 0.593/0.696. Since LaMP is separate from training, this agreement suggests that the gains extend beyond the T2M space used for supervision.

On OOD-Comp, LUNAR exceeds Motion-Agent by 0.150/0.172 in T2M/LaMP R@3. Novel temporal and concurrent action compositions test action order, simultaneity, and execution details. On OOD-Para, LUNAR reaches R@3 of 0.653/0.748. However, matched original/paraphrase diagnostics do not establish a robustness advantage over Baseline-RL-Reasoning (Appendix C.5). On OOD-Motion-X, LUNAR achieves the best R@3 of 0.471/0.571, suggesting transfer across language and motion-data domains. Split FID results are in Appendix Table 6; response-budget and cost diagnostics are in Appendix C.

4.3 Qualitative Results

Figure 4 compares four models on six prompts, with two examples from each OOD split. On OOD-Comp, LUNAR realizes more of the requested multi-stage sequences, including the jump–turn–walk and the step–sit–stand progression and transition, whereas the baselines often omit intermediate actions or fail to reach the requested state. On OOD-Para, LUNAR captures the forward surge followed by an upright ascent and the transition from lowered arms to a low, bent-over stance with forward creeping. On OOD-Motion-X, it preserves the localized upper-body circular gesture and combines a seated posture with the requested picture-taking arm configuration, while the baselines frequently replace these constraints with generic movement or an incorrect body pose. These examples provide concrete instances of the semantic and temporal omissions reflected in the quantitative retrieval results. More results are in Appendix F and the supplement.

4.4 Ablation Study

Table 3 follows a cumulative path: Baseline predicts motion tokens directly; Baseline-RL adds GRPO; Baseline-RL-Reasoning adds atomic-motion/body-group text; and LUNAR adds continuous motion-semantic latent reasoning. GRPO raises ID R@3 by 0.034 and OOD T2M/LaMP R@3 by 0.052/0.061, establishing a stronger sequence-level baseline. Textual reasoning further raises these metrics by 0.015 and 0.009/0.018. LUNAR improves over the textual variant by 0.013 and 0.017/0.031. These point estimates favor LUNAR; a separate paired OOD diagnostic at a larger response budget has intervals spanning zero under both evaluators (Appendix C.4).

5 Conclusion

We introduced LUNAR, a structured textual–latent reasoning framework that grounds latent motion representations through atomic-motion body-group descriptions and motion-semantic alignment. Experiments on HumanML3D and three OOD splits demonstrate consistent improvements in text–motion alignment, while ablation point estimates support the complementary roles of textual and latent reasoning. Future work may incorporate richer interaction-aware patterns, explicit contact constraints, and finer temporal grounding to improve the fidelity of complex, interaction-rich T2M.

AI USE STATEMENT

Generative AI was not used for research ideation or research design. GPT-5.6 Sol was used for language editing. OpenAI tools also assisted with manuscript drafting and revision, and the interpretation and presentation of our experimental results. DeepSeek-R1 generated atomic-motion reasoning annotations and paraphrased evaluation prompts, and Qwen3-235B performed automated verification of these generated data. The data-construction procedures and prompt specifications are described in Appendices A.3, B.1, and G. These automated checks do not replace author verification. We take responsibility for the final text, analyses, claims, and artifacts, including those developed with AI assistance.

ETHICS STATEMENT

This work studies text-conditioned motion generation using existing HumanML3D and Motion-X data; it does not report a new human-subject experiment. The source datasets and LLM-generated annotations may reflect limited action coverage and biases in descriptions of human movement. Automated annotation verification does not establish correctness or fairness. Generated motions could be misused to portray actions that did not occur and should be identified as synthetic. The method's limited modeling of contact and interaction also means that improved retrieval does not establish safe physical behavior. Downstream use and redistribution should respect the licenses and access conditions of the source datasets and pretrained models.

REPRODUCIBILITY STATEMENT

Appendix A specifies motion tokenization, atomic-motion supervision, the continuous-feedback recurrence, and training objectives. Appendix B documents OOD data construction, retrieval pools, training and decoding settings, checkpoint selection, and uncertainty calculations. Appendix C describes response budgets, validity criteria, inference timing, and paired comparisons, while Appendix D details the latent-state analyses. Self-contained teacher and verifier prompt specifications and examples are provided in Appendix G. We distinguish the original greedy benchmarks, stochastic-generation reruns, and evaluator repetitions on cached outputs, and state their respective limitations. Code will be made publicly available.

REFERENCES

Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol. SINC: Spatial composition of 3D human motions for simultaneous action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9984–9995, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Athanasiou_SINC_Spatial_Composition_of_3D_Human_Motions_for_Simultaneous_Action_ICCV_2023_paper.html.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, Jingyi Yu, 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.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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. Springer, 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.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 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*, 36:20067–20079, 2023.

Biao Jiang, Xin Chen, Chi Zhang, Fukun Yin, Zhuoyuan Li, Gang Yu, and Jiayuan Fan. Motionchain: Conversational motion controllers via multimodal prompts. In *European Conference on Computer Vision*, pp. 54–74. Springer, 2024.

Yuan-Ming Li, Qize Yang, Nan Lei, Shenghao Fu, Ling-An Zeng, Jian-Fang Hu, Xihan Wei, and Wei-Shi Zheng. IRG-MotionLLM: Interleaving motion generation, assessment and refinement for text-to-motion generation. *arXiv preprint arXiv:2512.10730*, 2025a.

Zhe Li, Weihao Yuan, Yisheng He, Lingteng Qiu, Shenhao Zhu, Xiaodong Gu, Weichao Shen, Yuan Dong, Zilong Dong, and Laurence Yang. LaMP: Language-motion pretraining for motion generation, retrieval, and captioning. In *International Conference on Learning Representations*, 2025b.

Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-X: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 36:25268–25280, 2023.

Runqi Ouyang, Haoyun Li, Zhenyuan Zhang, Xiaofeng Wang, Zeyu Zhang, Zheng Zhu, Guan Huang, Sirui Han, and Xingang Wang. Motion-R1: Enhancing motion generation with decomposed chain-of-thought and RL binding. *arXiv preprint arXiv:2506.10353*, 2025.

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. Springer, 2022.

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, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025. doi: 10.1145/3689031.3696075.

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

Guocun Wang, Kenkun Liu, Jing Lin, Guorui Song, Jian Li, and Xiaoguang Han. UniMo: Unified motion generation and understanding with chain of thought. *arXiv preprint arXiv:2601.12126*, 2026a.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond image and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12030–12040, 2026b.

Yuan Wang, Di Huang, Yaqi Zhang, Wanli Ouyang, Jile Jiao, Xuetao Feng, Dan Xu, and Shixiang Tang. MotionGPT-2: A general-purpose motion-language model for motion generation and understanding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026c.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. MMSearch-R1: Incentivizing LMMs to search. *arXiv preprint arXiv:2506.20670*, 2025a.

Qi Wu, Yubo Zhao, Yifan Wang, Yu-Wing Tai, and Chi-Keung Tang. MotionLLM: Multimodal motion-language learning with large language models. *arXiv preprint arXiv:2405.17013*, 2024.

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. In *International Conference on Learning Representations*, 2025b.

Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. Visual planning: Let's think only with images. *arXiv preprint arXiv:2505.11409*, 2025a.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. *arXiv preprint arXiv:2502.12134*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Huanyu Zhang, Wenshan Wu, Chengzu Li, Ning Shang, Yan Xia, Yangyu Huang, Yifan Zhang, Li Dong, Zhang Zhang, Liang Wang, Tieniu Tan, and Furu Wei. Latent sketchpad: Sketching visual thoughts to elicit multimodal reasoning in MLLMs. *arXiv preprint arXiv:2510.24514*, 2025.

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*, pp. 14730–14740, 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, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. FineMoGen: Fine-grained spatio-temporal motion generation and editing. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://papers.neurips.cc/paper_files/paper/2023/hash/2d52879ef2ba487445ca2e143b104c3b-Abstract-Conference.html.

Jiakun Zheng, Ting Xiao, Shiqin Cao, Xinran Li, Zhe Wang, and Chenjia Bai. Re^{2}MoGen: Open-vocabulary motion generation via LLM reasoning and physics-aware refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1378–1387, 2026.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing thinking with images via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

TECHNICAL APPENDIX

A ADDITIONAL METHOD DETAILS

A.1 MOTION TOKENIZATION AND RESPONSE FORMAT

Let $m = (m_t)_{t=1}^{T}$ be the paired reference motion used in the main paper, represented by 263-dimensional features over 22 joints at 20 frames per second. The frozen VQ-VAE motion codec comprises the encoder $\mathcal{E}_{\mathrm{vq}}$, quantizer Q, and decoder $\mathcal{D}_{\mathrm{vq}}$. Its codebook is $\mathcal{C}_{\mathrm{vq}} = \{e_r\}_{r=0}^{K_{\mathrm{vq}}-1}$ with $K_{\mathrm{vq}} = 512$. Given encoder features $u = (u_1, \ldots, u_L) = \mathcal{E}_{\mathrm{vq}}(m)$, each position is assigned to its nearest codebook entry:

$$y_l^\star = \arg\min_{0 \le r < K_{\mathrm{vq}}} \|u_l - e_r\|_2, \qquad \bar{u}_l = e_{y_l^\star}. \tag{11}$$

This gives the target sequence $y^\star = (y_1^\star, \ldots, y_L^\star) = Q(\mathcal{E}_{\mathrm{vq}}(m))$ and the reconstruction

$$m_{\mathrm{rec}} = \mathcal{D}_{\mathrm{vq}}(\bar{u}_1, \ldots, \bar{u}_L). \tag{12}$$

Let $\bar{u} = (\bar{u}_1, \ldots, \bar{u}_L)$, let p and p_{rec} denote the corresponding reference and reconstructed joint positions, and let sg denote stop-gradient. Following Motion-Agent, pre-training combines feature reconstruction, joint-position reconstruction, and encoder commitment:

$$\mathcal{L}_{\mathrm{VQ}} = \|m - m_{\mathrm{rec}}\|_1 + \lambda_p \|p - p_{\mathrm{rec}}\|_1 + \beta_{\mathrm{vq}} \|u - \mathrm{sg}[\bar{u}]\|_2^2. \tag{13}$$

Here λ_p and β_{vq} weight the position and commitment terms, respectively. The codec uses four-frame temporal downsampling. Its codebook is maintained by exponential-moving-average updates and codebook reset, rather than by a separate gradient-based codebook loss (Wu et al., 2025b). The encoder, codebook, and decoder remain frozen throughout LUNAR SFT and RL; LUNAR learns to predict the discrete code indices, and $\mathcal{D}_{\mathrm{vq}}$ reconstructs the final 3D motion.

The Qwen3-1.7B vocabulary is extended with $\{$<Motion_0>, ..., <Motion_511>$\}$ and the structural delimiters used in Figure 3. A complete response follows

$$\begin{aligned} x \to\ & \langle \texttt{think} \rangle\, \tau\, \langle \texttt{/think} \rangle\, \langle \texttt{output} \rangle \\ & (a_1, B_1, Z_1) \cdots (a_N, B_N, Z_N)\, \langle \texttt{/output} \rangle \\ & \langle \texttt{motion} \rangle\, y\, \langle \texttt{/motion} \rangle. \end{aligned} \tag{14}$$

Here τ is the model's internal prompt-level reasoning trace, which identifies and orders the atomic motions and guides construction of the subsequent structured trajectory. In the serialized response, each <latent_pad> marks one position at which a continuous state $z_{i,k}$ is formed; the listings therefore show placeholder tokens, whereas the method notation and Figure 3 show the corresponding latent states.

Added-token initialization. The 512 language-model motion-token embeddings are distinct from the frozen VQ codebook vectors. At the original vocabulary expansion, new rows were initialized from a multivariate normal using the existing language-model embeddings' mean and covariance. The progressive SFT phases inherit the trained motion-token rows; they are not reinitialized when latent reasoning is introduced. At latent SFT initialization, all 512 motion rows and six existing structural rows were inherited from the textual-reasoning checkpoint, while three new latent-structure rows were freshly initialized. Input embeddings and the language-model output head share weights in the Qwen3 configuration. During latent SFT, gradients on the original vocabulary rows are masked; added structural and motion rows use 30 times the backbone learning rate. The archived initialization log specifies the distribution family but does not retain the original numerical covariance.

A.2 MOTION-CODE USAGE AND RECONSTRUCTION

We audit the frozen codec on real HumanML3D motions and on LUNAR outputs from the repeated-generation study in Appendix B.2. The real-motion scan encodes one full motion per split ID after restricting lengths to $40 \le T < 200$ frames and truncating to a multiple of four. This yields 22,326 training motions and 4,198 held-out motions. The generated population contains 18,789 outputs

Table 4: Motion-code usage in encoded HumanML3D motions and LUNAR OOD generations pooled over 20 passes. Tokens and Codes denote the total token count and occupied entries out of 512, respectively; H denotes token entropy in nats, and $\exp(H)$ is effective perplexity. Top-10 share is the token fraction assigned to the ten most frequent codes. The shaded row denotes LUNAR; the populations differ in prompts, lengths, and sampling.

Population	Tokens	Codes	H	$\exp(H)$	Top-10 share
Real train	799,280	512/512	5.932	377.00	10.37%
Real test	150,152	512/512	5.920	372.34	10.53%
LUNAR OOD	790,947	512/512	5.451	232.99	21.20%

accepted by both evaluators out of 19,240 attempts on fixed OOD prompts over 20 generation passes; these repeated attempts are not independent source examples.

Both real-motion splits and the pooled generated outputs use all 512 codes (Table 4); individual generation passes use 496–505. Generated usage is more concentrated: the ten most frequent codes account for 21.20% of generated tokens, compared with 10.37% and 10.53% in the real-motion scans. Full pooled coverage therefore does not imply uniform usage or explain differences between these populations.

A separate codec reconstruction diagnostic encodes and decodes test-loader motions and compares reconstructed versus real T2M embeddings. With seed 1234, batch size 32, and the incomplete final batch dropped, reconstruction FID is 0.0630 on 4,640 loader records; real/reconstructed diversity is 9.7943/9.7932. This loader samples captions and crops, so its records differ from the 4,198 unique full motions used in the frequency scan. The result provides an empirical reconstruction reference under T2M, not a lower bound on generation FID or a measure of physical correctness. These diagnostics characterize the chosen discrete motion-output interface; a trained continuous-output comparison remains future work, distinct from LUNAR's continuous reasoning feedback.

A.3 Atomic-Motion Textual Supervision

For each accepted training example, the teacher emits an ordered sequence of N units in the main-paper (a_i, B_i) format. Representing left and right limbs separately preserves asymmetric actions and support configurations; segment-level displacement and orientation are expressed by a_i and the ten body-group fields without an additional root field.

DeepSeek-R1 generates the internal reasoning trace τ and the ordered (a_i, B_i) blocks from each HumanML3D caption. Qwen3-235B verifies faithfulness, completeness, atomicity, clarity, and coverage of all ten fields, and only accepted traces are used for SFT. Self-contained teacher and verifier specifications, including the in-prompt example, appear in Appendix G.

Training-corpus statistics. The eligible training corpus contains 22,326 motion files and 66,729 available caption rows. Teacher-derived atomic-unit counts per caption have mean 2.508, median 2, 95th percentile 5, and maximum 19, counted from numbered units within `<output>`. A training access samples one caption for a retained motion, so 66,729 is the available caption population rather than the number of independent examples in one epoch. These textual unit counts characterize the four-state-per-unit reasoning budget; they do not validate the semantic boundaries of the reference-motion segments.

A.4 Atomic-Motion Latent Targets and Feedback

HumanML3D provides a paired caption and full reference motion, but it does not provide atomic-motion boundary annotations. For an accepted N-unit textual trajectory, the teacher decomposition supplies the number and order of the reasoning units, but not their frame-level timing. We therefore impose a monotonic one-to-one assignment between the ordered units and N non-overlapping temporal segments of the paired reference motion m, with boundaries $0 = t_0 < t_1 < \cdots < t_N = T$.

To estimate these boundaries, we compute a frame-wise motion-energy trace from the local-velocity feature channels given by the Python slice `193:259` of the 263-D HumanML3D representation and

smooth it with a five-frame moving average. Local minima of this trace form candidate transition boundaries. We require every segment to contain at least

$$\Delta_{\min} = \max\left(4, \left\lfloor \frac{T}{2N} \right\rfloor\right)$$

frames, where four frames correspond to one motion-token interval. Among the feasible candidates, we select the $N-1$ boundaries with the smallest summed motion energy while preserving temporal order and the minimum-length constraint. If no feasible candidate set exists, we divide the motion uniformly into N segments; when $N = 1$, the complete motion is used. Each selected segment is normalized with the same HumanML3D mean and standard deviation used by the frozen semantic encoder. Among the 66,729 eligible training-caption rows, 55,863 use velocity-valley boundaries, 9,279 contain one atomic unit, and 1,587 use uniform fallback. The larger 87,338-record target archive spans all data splits and is not the training-caption denominator. These frequencies document target construction, not semantic boundary accuracy.

The resulting heuristic reference-motion segment assigned to reasoning unit i is

$$m_i^{\text{seg}} = (m_{t_{i-1}+1}, \ldots, m_{t_i}). \tag{15}$$

The frozen motion branch of the CLIP-style T2M semantic encoder maps this segment to the normalized target

$$q_i = \operatorname{norm}\big(\mathcal{E}_{\text{CLIP}}^{\text{motion}}(m_i^{\text{seg}})\big) \in \mathbb{R}^{512}. \tag{16}$$

Thus q_i is computed from a velocity-aligned segment of the paired reference motion, not from an annotated atomic boundary, generated motion, or teacher text. This assignment assumes that the teacher's ordered units admit a monotonic temporal alignment and remains approximate for overlapping or concurrent actions.

Following the main-paper alignment objective, we use $K_{\text{lat}} = 4$ slots per block, project each language-model state as $h_{i,k} = f_\theta(z_{i,k}) \in \mathbb{R}^{512}$, and supervise all four states against the same target q_i. Their distinct causal positions nevertheless allow context-dependent states before the trajectory continues.

Let $d_{\text{proj}} = \max(d_h, 512)$. The projector and reverse projector are two independently parameterized, two-layer GELU MLPs with the following exact stacks:

$$\begin{aligned} f_\theta = {} & \mathrm{LN}_{512} \circ \mathrm{Linear}_{d_{\text{proj}} \to 512} \circ \mathrm{GELU} \\ & \circ \mathrm{Linear}_{d_h \to d_{\text{proj}}} \circ \mathrm{LN}_{d_h}, \\ g_\theta = {} & \mathrm{Linear}_{d_{\text{proj}} \to d_h} \circ \mathrm{GELU} \\ & \circ \mathrm{Linear}_{512 \to d_{\text{proj}}} \circ \mathrm{LN}_{512}\,. \end{aligned} \tag{17}$$

Thus, the forward path ends with a 512-dimensional LayerNorm, whereas the reverse path ends directly in the d_h-dimensional language-model input space. During teacher-forced latent SFT, the target-side feedback $g_\theta(q_i)$ is supplied to the following decoding step, allowing subsequent reasoning and motion-token losses to train the feedback path. During deterministic generation, the unavailable target is replaced by the model prediction:

$$e_{i,k}^{\text{fb}} = \begin{cases} g_\theta(q_i), & \text{teacher-forced latent SFT}, \\ g_\theta(h_{i,k}), & \text{deterministic generation}. \end{cases} \tag{18}$$

Consequently, latent slots within Z_i condition one another, and the completed block conditions the next $(a_{i+1}, B_{i+1}, Z_{i+1})$ unit or, for the final block, the motion-token sequence. At inference, the model receives only x and generates the full trajectory autoregressively. Complete responses are shown in Appendix Figures 7 and 8.

A.5 GRPO Objective and Structural Validation

The main paper defines the group-relative advantage A_j for each sampled response o_j, $j = 1, \ldots, G$. At an ordinary token position t, its policy likelihood is categorical. At latent slot (i, k), the projected state $h_{\theta,j,i,k} = f_\theta(z_{j,i,k})$ parameterizes a Gaussian semantic-state action $\tilde{h}_{j,i,k}$ with fixed standard deviation σ_h, where $I \in \mathbb{R}^{512 \times 512}$ is the identity matrix:

$$p_{\theta,j,t} = \begin{cases} \pi_\theta(o_{j,t} \mid x, o_{j,<t}), & t \text{ is a token position}, \\ \mathcal{N}\left(\tilde{h}_{j,i,k}; h_{\theta,j,i,k}, \sigma_h^2 I\right), & t \text{ is latent slot } (i,k). \end{cases} \tag{19}$$

The sampled state is fed back through $g_\theta(\tilde{h}_{j,i,k})$ during RL rollout. Let $p_{\mathrm{old},j,t}$ be the corresponding likelihood under the rollout policy and let $|o_j|$ count its policy decisions. The likelihood ratio is

$$\rho_{j,t}(\theta) = \frac{p_{\theta,j,t}}{p_{\mathrm{old},j,t}}. \tag{20}$$

Following standard GRPO (Shao et al., 2024), we maximize

$$\begin{aligned} \ell_{j,t}(\theta) = \; & \min\big(\rho_{j,t}A_j,\ \mathrm{clip}(\rho_{j,t}, 1-\epsilon, 1+\epsilon)A_j\big) \\ & - \beta_{\mathrm{KL}} D_{j,t}^{\mathrm{KL}} \end{aligned} \tag{21}$$

and average this surrogate over the group and all policy positions:

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \frac{1}{G}\sum_{j=1}^{G}\frac{1}{|o_j|}\sum_{t}\ell_{j,t}(\theta). \tag{22}$$

where ϵ is the clipping radius and β_{KL} weights the divergence $D_{j,t}^{\mathrm{KL}}$ from the fixed SFT reference policy. Including the latent-state density allows trajectory-level rewards to optimize both discrete response tokens and the semantic feedback path.

The format reward in the main paper is a single structural-validity indicator. It requires correctly ordered reasoning, output, latent, and motion delimiters; at least one atomic motion; all ten body-group fields for each motion; exactly K_{lat} latent slots after each (a_i, B_i) block; and a parseable motion-token sequence. This same indicator gates the motion and semantic rewards because a malformed response cannot be decoded and scored reliably. No separate latent-format reward is used in the paper formulation.

B Detailed Experimental Setup

B.1 OOD Suite Construction and Protocol

The OOD suite separates three sources of distribution shift that can expose failures in structured motion reasoning. *OOD-Comp* tests unseen temporal and concurrent combinations of familiar actions. We select one HumanML3D evaluation caption per source motion, retain captions containing 3–25 words, and deterministically construct 150 prompts by pairing two source captions and 50 prompts by combining three source captions. The selected caption pairs or triples are joined with relations such as "then," "and then," "while," and "followed by." Because an individual source caption may already describe multiple actions, "two-caption" and "three-caption" denote the construction units rather than the number of atomic actions in the resulting prompt. The resulting 200 prompts are validated and deduplicated.

OOD-Para tests whether the same motion semantics are preserved under unfamiliar wording. It contains 196 accepted paraphrases of held-out captions. DeepSeek-R1 (DeepSeek-AI, 2025) serves as the teacher model that generates candidate rewrites, and Qwen3-235B (Yang et al., 2025) serves as the verifier model that rejects candidates that add, omit, or alter motion content. The rewrite prompt requires semantic equivalence. The source captions span different action complexities, but paraphrasing does not introduce new actions. Self-contained rewrite and verification prompt specifications, including the rewrite example, are reproduced in Appendix G.

OOD-Motion-X tests transfer across both language and motion-data domains. It is a deterministic paired subset of 566 single-person examples from Motion-X (Lin et al., 2023). We retain the original 322-D SMPL-X body tracks, exclude captions that explicitly describe multiple people, and remove normalized caption matches against the complete HumanML3D text corpus. The selected motions are resampled and converted to the 263-D representation expected by the T2M and LaMP evaluators.

OOD-Comp and OOD-Para are unpaired prompt sets, so their FID reference statistics use HumanML3D evaluation motions. OOD-Motion-X instead uses its 566 paired ground-truth motions as the target-domain reference. We therefore report FID separately for each split and evaluator space rather than combining values with different reference distributions.

Retrieval pools and aggregation. For the greedy-reference OOD scores, OOD-Comp and OOD-Para are evaluated as source subsets of a shared candidate set containing their 200 and 196 prompts plus 170 auxiliary distractor captions. OOD-Motion-X uses its separate 566-caption pool. Within each pool, the evaluator retains caption–motion pairs with valid embeddings, uses local seed 0 to shuffle them into complete 32-candidate batches, and averages retrieval scores over 20 shuffles; each shuffle discards an incomplete tail. Overall R@3 is the equal-weight mean of the Comp., Para., and Motion-X source scores, not a pooled 566-caption rate. Structural-format validity is checked separately from embedding validity.

B.2 Repeated-Generation OOD Study

We additionally generate 20 fresh outputs per fixed prompt with seeds 20260926–20260945, using the same manifests and checkpoints across passes. Each pass applies the retrieval-pool construction above and averages 20 evaluator shuffles. This outer generation repetition is separate from the inner candidate-pool repetition. Motion-Agent and LUNAR use text-token temperature 0.2 and top-p 0.9, with response caps of 150 and 4,096, respectively. MLD uses its native sampler with guidance 7.5, 50 steps, and a 196-frame output length; MoMask uses its length estimator, 18 sampling steps, temperature 1.0, and top-k ratio 0.9. These stochastic settings differ from the original greedy-reference evaluation.

For pass scores $r_1, \ldots, r_{20}$, we report the mean $\bar{r}$ and the recorded approximate 95% normal half-width

$$h = \frac{1.96}{\sqrt{20}} \sqrt{\frac{1}{20} \sum_{j=1}^{20} (r_j - \bar{r})^2}. \tag{23}$$

Overall is computed within each pass before applying this summary, so its half-width is not the average of the three split half-widths. Table 5 pairs each stochastic mean with its own half-width. Table 2 instead places these half-widths beside the original greedy scores to display reference performance and stochastic rerun uncertainty together. They are not confidence intervals centered on those greedy scores and do not quantify paired differences, prompt-family variation, or independent-training-seed variation.

Table 5: Repeated-generation OOD retrieval under T2M and LaMP. Entries are mean R@3 $\pm h$ over 20 stochastic generation passes, where h is the approximate 95% normal half-width defined in Equation 23. Comp., Para., and Motion-X denote the three OOD splits; Overall is their macro-average computed within each pass. These means correspond to the half-widths shown in Table 2. The best and second-best means in each column are highlighted in **bold** and underline, respectively; shading denotes LUNAR.

Method	Overall		Comp.		Para.		Motion-X	
	T2M	LaMP	T2M	LaMP	T2M	LaMP	T2M	LaMP
MLD	.428±.005	.396±.005	.438±.008	.378±.009	.462±.012	.408±.009	.385±.006	.401±.004
MoMask	.487±.004	.563±.005	.515±.007	.616±.007	.530±.006	.571±.009	.415±.004	.503±.006
Motion-Agent	.506±.004	.591±.004	.501±.007	.573±.007	.605±.005	.710±.009	.411±.005	.489±.005
LUNAR	**.589**±.004	**.686**±.004	**.638**±.007	**.758**±.008	**.663**±.005	**.749**±.007	**.466**±.006	**.553**±.005

MLD, MoMask, and Motion-Agent yield 200/196/566 motions valid under both T2M and LaMP in every pass. For LUNAR, the corresponding ranges are 196–199/200, 192–194/196, and 546–551/566. Retrieval is conditional on valid outputs, so invalidity can change candidate-pool membership. LUNAR has the highest stochastic mean in all reported columns under both evaluators, consistent with the ordering of the greedy references. These marginal intervals do not establish a paired or causal advantage over the internal textual-reasoning control.

B.3 OOD Distribution-Fidelity Results

Appendix Table 6 complements the main-paper retrieval metrics with split-specific marginal distribution fidelity in the T2M and LaMP evaluator spaces.

Table 6: Comparison of OOD distributional fidelity using FID in the T2M and LaMP embedding spaces; lower values are better. Each entry reports T2M / LaMP. Comp. and Para. use HumanML3D reference motions, whereas Motion-X uses paired target-domain references. T2M and LaMP values are reported to three and two decimal places, respectively. The best and second-best results within each evaluator and split are highlighted in **bold** and underline, respectively; shading denotes LUNAR.

Method	FID (T2M / LaMP) ↓		
	Comp	Para	Motion-X
MLD	5.270/691.90	3.110/574.54	7.630/1136.75
MoMask	6.730/327.32	1.580/144.63	**5.260/1125.99**
Motion-Agent	**4.750**/263.95	**0.550/97.49**	6.069/1254.41
LUNAR	4.870/**256.87**	0.860/123.98	6.212/1269.07

B.4 Training Configuration

Supervised training. SFT tunes backbone parameters and added-token embeddings with bfloat16 mixed precision, chat-template prompts, gradient checkpointing, cosine learning-rate decay, 100 warmup steps, maximum gradient norm 2.0, weight decay 0.01, embedding learning-rate scale 30, label smoothing 0.1, and top-5 checkpoint retention. Phase 1 direct-motion SFT uses batch size 32, learning rate 5×10^{-5}, motion-loss weight 1.0, at most 300 epochs, early-stopping patience 10, validation every 5 epochs, and 256 validation-generation tokens. Phase 2 textual-reasoning SFT resumes from that checkpoint with effective batch size 32 (batch size 4 and gradient accumulation 8), the same learning rate, motion-loss weight 0.5, maximum target length 4096, at most 80 epochs, patience 12, and validation every 5 epochs. Phase 3 latent-reasoning SFT uses the same sequence settings, $K_{\text{lat}} = 4$ serialized `<latent_pad>` positions per atomic-motion block, latent-alignment weight 0.01, and a 500-step linear warmup for that weight.

Reinforcement learning and decoding. RL uses the VERL implementation of GRPO (Sheng et al., 2025) on HumanML3D prompts. The format, motion, and semantic reward weights are 0.2, 0.4, and 0.4, respectively. The actor learning rate is 5×10^{-7}, each batch contains 256 prompts with $G = 5$ sampled responses per prompt, the proximal policy optimization (PPO) mini-batch size (Schulman et al., 2017) is 128, maximum prompt and response lengths are 512 and 4096, and the KL coefficient is 0.001. Rollouts use temperature 0.7, top-p 0.95, top-k 50, and semantic-action standard deviation $\sigma_h = 10$. Training runs for three epochs, saves every 100 steps, and validates every 25 steps. Validation and the original greedy-reference benchmarks use greedy motion-token decoding with deterministic latent-mean feedback, without candidate reranking or reward-based selection. The stochastic study in Appendix B.2 and the diagnostics below specify their own decoding settings. The main benchmark tables use one GRPO-optimized textual–latent reasoning checkpoint. Additional diagnostics specify their checkpoint steps and decoding budgets in Appendix C.

B.5 Checkpoint Selection and ID Uncertainty

All reported checkpoints were fixed before the repeated-generation study. The external generators use the supplied MLD and Motion-Agent checkpoints and the local MoMask transformer/residual checkpoints. The internal ablation follows the progressive SFT lineage; its RL checkpoints are Baseline-RL step 400, Baseline-RL-Reasoning step 780, and LUNAR step 700. The latent SFT checkpoint was selected by T2M validation FID; LUNAR step 700 was selected post hoc from the completed RL sweep using canonical ID FID and headline T2M retrieval, not the subsequent 20-pass scores.

The ID entries in Table 3 use three evaluator rounds around cached greedy generations. Caption choice, batch order, crops, and metric sampling can vary across rounds; no new generation or training is performed. The reported half-width is $1.96 s_{\text{pop}}/\sqrt{3}$, with s_{pop} the population standard deviation of the three round scores. These descriptive intervals should not be interpreted as training-seed uncertainty. The historical OOD ablation entries remain point estimates; the separate paired diagnostic below covers only LUNAR and Baseline-RL-Reasoning.

The frozen T2M encoder checkpoint used for segment targets and rewards also supplies the canonical T2M evaluation space. LaMP is separate from training and provides an OOD evaluator cross-check. A same-output ID LaMP comparison and blinded human fidelity/naturalness judgments remain unavailable; neither retrieval nor FID alone establishes physical correctness or human-perceived quality.

C Additional Evaluation Diagnostics

These diagnostics use fixed existing checkpoints: LUNAR at step 700, Baseline-RL-Reasoning at step 780, and, for timing, Baseline-RL at step 400. They introduce no new training and use the method names from Table 3. Each experiment specifies its own cohort and response limit; these results do not replace the main benchmark tables or quantify training-seed variation.

C.1 Response Budget and Output Validity

We compare response caps of 2,048 and 8,192 tokens on a fixed auxiliary set of 566 prompts, separate from the three-dataset suite in Table 2. Both methods use one greedy generation pass per budget, batch size 32, and one H200 per actor. Table 7 reports outputs accepted by both the T2M and LaMP evaluators. This valid-motion count differs from strict response-format compliance; passing the evaluator does not imply that all reasoning fields satisfy the full format specification.

Table 7: Effect of the response-token budget on valid-motion coverage for the auxiliary 566-prompt set. Each entry reports the number of outputs accepted by both T2M and LaMP out of all attempted prompts, using one greedy generation pass at each budget. This measure captures evaluation coverage rather than motion quality or strict response-format compliance. The shaded row denotes LUNAR.

Method	2,048 tokens	8,192 tokens
Baseline-RL-Reasoning	491/566	558/566
LUNAR	546/566	561/566

Increasing the response budget improves valid-output coverage for both methods, with a larger increase for Baseline-RL-Reasoning. Thus, the decoding limit can materially affect which outputs enter evaluation. These counts describe coverage, not motion quality, and do not isolate the effect of latent reasoning. Comparisons of quality should report both the response budget and the valid evaluation population.

C.2 Measured Inference Cost

We compare all three methods sequentially on one NVIDIA H200 using the same 40 fixed captions, Qwen3-1.7B backbone, greedy decoding, batch size 1, and a 2,048-token response cap. Two warmup prompts precede three measured passes. Timing includes prompt construction, generation, parsing, VQ decoding, and transfer of the decoded motion to CPU; model loading and evaluator scoring are excluded.

Table 8: Measured inference cost on 40 fixed captions using one H200, batch size 1, greedy decoding, and a 2,048-token response cap. Latency is the median of three pass-average times per prompt, including generation, parsing, VQ decoding, and CPU transfer but excluding model loading and evaluator scoring. Output tokens and Motion codes are totals over the 40 prompts; Valid reports accepted outputs. The shaded row denotes LUNAR.

Method	Latency (s)	Output tokens	Motion codes	Valid
Baseline-RL	0.083	1,653	1,573	40/40
Baseline-RL-Reasoning	2.927	61,280	1,309	33/40
LUNAR	2.034	42,643	1,647	40/40

LUNAR incurs substantially more generation time than direct motion prediction, while its measured latency is lower than Baseline-RL-Reasoning in this pilot. The reasoning variants produce different response lengths, and Baseline-RL-Reasoning has seven invalid outputs at this cap. Therefore, the comparison does not isolate latent-step overhead or establish a general speed–quality ranking. The three passes repeat the same 40 prompts and are not independent training or prompt samples. GPU-wide memory readings include runtime and KV-cache reservations, so we do not interpret them as model memory requirements or infer consumer-GPU deployment from theoretical FLOPs.

C.3 Reasoning Length and Structural Coverage

We audit every LUNAR attempt in the 20-pass OOD study (Appendix B.2). Numbered units are counted within `<output>`. Prefix length is obtained by retokenizing the saved response before its first `<motion>` marker; it includes reasoning prose, structural delimiters, and serialized latent markers. If the marker is absent, the complete response is counted. These counts describe saved response structure, not the contents or computational cost of continuous hidden states.

Table 9: Reasoning length and output validity of LUNAR across 20 generation passes on each OOD split. Mean units counts numbered atomic motions; Mean prefix counts retokenized response tokens before the first motion marker, including structural and latent markers. Both means include all attempts. Valid/attempted reports dual-evaluator acceptance, whereas Format failures reports violations of the strict response structure, with percentages relative to all attempts. These are separate validity criteria.

Split	Mean units	Mean prefix	Valid/attempted	Format failures
OOD-Comp	3.125	1,292.8	3,938/4,000	68 (1.70%)
OOD-Para	2.033	952.6	3,866/3,920	56 (1.43%)
OOD-Motion-X	1.682	850.9	10,985/11,320	351 (3.10%)

Composition prompts produce longer prefixes and more numbered units in this diagnostic (Table 9). Across 19,240 attempts, 36 have unequal numbered-unit and complete four-slot-block counts; all 451 responses missing `<motion>` are evaluator-invalid but remain in the prefix statistics. The mean emitted VQ-code counts among valid outputs are 46.47, 38.19, and 41.91 for Comp., Para., and Motion-X. The saved text does not retain generation finish reasons, so it does not establish an exact response-cap-hit rate. Repeating fixed prompts increases measurement coverage without creating additional independent source examples or validating the semantics of generated decompositions.

C.4 Paired OOD Comparison at a Larger Response Budget

A separate diagnostic compares LUNAR and Baseline-RL-Reasoning at an 8,192-token response cap on the 200/196/566 Comp./Para./Motion-X prompts. It conditions on 196/190/545 common-valid cases and uses identical candidate assignments for both methods. With generated outputs held fixed, 10,000 paired prompt/clip bootstrap resamples give macro R@3 differences (LUNAR minus Baseline-RL-Reasoning) of $+0.0178$ $[-0.0117, +0.0479]$ under T2M and $+0.0213$ $[-0.0094, +0.0515]$ under LaMP, with 95% percentile intervals. Both include zero. These intervals concern a paired contrast at a different response budget, not the stochastic pass means or the historical greedy ablation points. They are conditional on common validity and fixed retrieval pools, and omit training-seed uncertainty. The cumulative training lineages also differ in supervision, rewards, and update history, so this comparison does not isolate the causal contribution of continuous feedback.

C.5 Matched Original–Paraphrase Evaluation

To assess sensitivity to wording directly, we pair each distinct original caption with its first active paraphrase, selected before inference. The 196 OOD-Para captions yield 192 distinct original/paraphrase pairs. At a 2,048-token cap, 174 pairs are valid for both methods under both caption conditions. Both methods are scored using identical 20 deterministic shuffles of 32-candidate retrieval pools. We compute 95% percentile bootstrap intervals using 10,000 paired-original-caption resamples (seed 20260926), keeping candidate-pool assignments fixed.

Table 10: Matched original–paraphrase comparison on 174 pairs valid for both methods and caption conditions at a 2,048-token cap. Δ denotes paraphrase minus original R@3, so negative values indicate degradation. The final column reports $\Delta_{\text{LUNAR}} - \Delta_{\text{Baseline-RL-Reasoning}}$ with 95% percentile intervals from 10,000 paired-caption bootstrap resamples. The shaded column denotes LUNAR; both difference intervals include zero.

Evaluator	Δ_{LUNAR}	$\Delta_{\text{Baseline-RL-Reasoning}}$	Difference [95% CI]
T2M	-0.1941	-0.1944	$+0.0003\ [-0.0690, +0.0723]$
LaMP	-0.1381	-0.1019	$-0.0363\ [-0.1200, +0.0462]$

Both methods degrade under paraphrasing, and neither evaluator establishes a difference in degradation. This exploratory result does not demonstrate equivalence. It is conditional on common validity and fixed retrieval pools; the intervals omit training-seed and source-family uncertainty. Independent semantic-equivalence annotation remains unavailable. Of 122 originals with multiple candidate IDs, 115 have only plain/mirrored variants of one numeric motion root. Seven of the 192 pairs remain associated with multiple roots, including five of the 174 common-valid pairs; the other 169 common-valid pairs each have a distinct root. This identifier audit narrows the ambiguity but does not convert the existing 174-pair intervals into source-family-grouped intervals or validate paraphrase equivalence. The analysis therefore qualifies absolute OOD-Para scores without establishing a general robustness advantage.

External reasoning baseline. Motion-R1 (Ouyang et al., 2025) is a closely related external baseline: it generates an ordered textual decomposition into physical sub-actions before predicting motion tokens, and combines cold-start SFT with GRPO using format, motion, and semantic rewards. Its published examples include movement directions and involved body parts, but do not prescribe a fixed ten-body-group schema or a continuous latent block for every sub-action.

LUNAR organizes the intermediate trajectory into paired units: each atomic-motion description and its ten body-group fields define the scope of a following continuous latent block, whose states are supervised by the corresponding motion segment and fed back into autoregressive decoding. Baseline-RL-Reasoning retains our atomic-motion and body-group textual structure without these latent blocks. It therefore provides an internal control for the added latent component under our formulation; differences in reasoning supervision and response structure prevent interpreting it as a reproduction of Motion-R1. This structural comparison alone does not establish a motion-quality advantage.

The main obstacle to faithfully retraining Motion-R1 is the unavailability of its MotionCoT training data: the validated description–decomposed-CoT–motion triplets produced by its data engine. At the time of this revision, the official repository did not release these triplets, the data-generation/filtering implementation, runnable inference code, or model checkpoints.[1] Although the paper describes the prompting and quality-control procedure, regenerating annotations would introduce different reasoning traces and filtering decisions, changing the supervision used to train the baseline. Missing code and checkpoints additionally prevent direct OOD evaluation of the official model. We therefore include Motion-R1's published ID results and identify our textual-reasoning comparison explicitly as an internal baseline.

D Latent Unit Analysis

We use *latent unit* to denote the four-slot block Z_i ($K_{\text{lat}} = 4$) paired with atomic motion i. We analyze these units along two complementary axes. Semantic alignment asks whether their projected states encode the intended atomic-motion content; intervention asks whether the latent feedback path affects subsequent motion-token prediction. The former establishes semantic grounding, whereas the latter distinguishes an active conditioning signal from an inert intermediate state. Direct latent-space comparisons use the frozen T2M space in which the targets q_i are defined. LaMP is used only

[1] `https://github.com/GigaAI-Research/Motion-R1` and its artifact-release issues, accessed 26 September 2026.

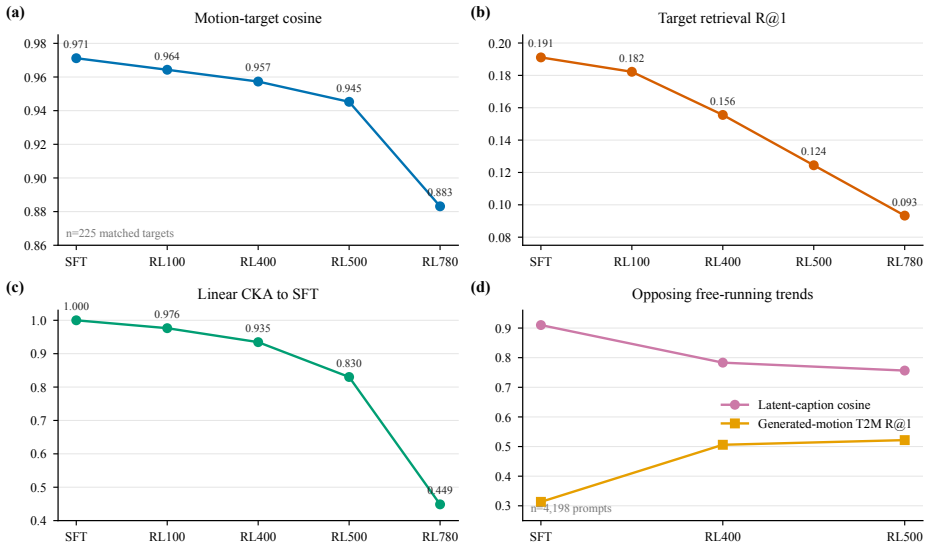


Figure 5: Semantic structure and generation quality of the learned latent units. Panel (a) reports mean teacher-forced cosine between each projected latent state and its assigned atomic-motion target. Panel (b) reports target-retrieval R@1 among the 225 matched targets. Panel (c) reports linear CKA between each checkpoint's projected states and the corresponding SFT states. Panels (a)–(c) use the same 225 target-bearing atomic-motion instances matched across checkpoints. Panel (d) compares latent–caption cosine with generated-motion T2M R@1 under free-running decoding on 4,198 HumanML3D prompts.

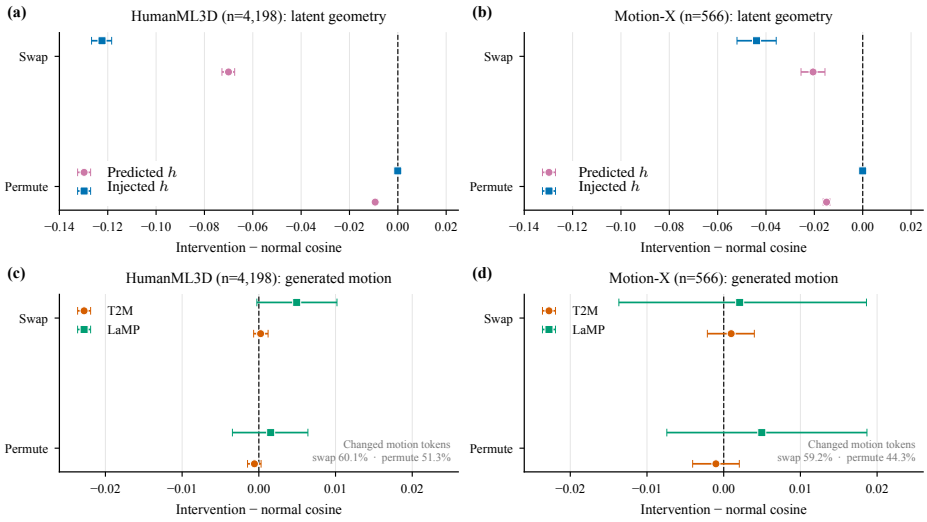


Figure 6: Latent-path interventions. For each prompt and metric s, the x-axis reports the paired difference $\Delta s = s_{\text{intervention}} - s_{\text{normal}}$. Swap injects projected content from another sample; Permute reorders the four within-block slots. Panels (a) and (b) compare the model-predicted state h^{pred} with the injected state h^{inj} passed through the reverse projector. Panels (c) and (d) report caption–generated-motion cosine under T2M and LaMP. Points are paired means, bars are motion-ID-clustered 95% bootstrap intervals, and annotations give the proportion of prompts with at least one changed motion token.

as an independent evaluator of generated motions; its embedding space is not compared directly with the 512-dimensional projected latent states.

D.1 Semantic Alignment Across SFT and RL

The 225 matched target-bearing instances in Figure 5 yield 900 projected latent states across their four slots. Target R@1 uses the 225 associated targets as candidates, and linear centered-kernel alignment (CKA) uses SFT as the representation reference.

Figure 5(a) shows that SFT gives the latent units a strong motion-semantic anchor: mean teacher-forced motion-target cosine is 0.971 at SFT, remains 0.945 at RL500, and decreases to 0.883 at RL780. Figure 5(b) reports target-retrieval R@1, which is 0.191 at SFT, far above the chance rate of $1/225 \approx 0.004$, and remains 0.124 at RL500. Thus the projected states contain target-specific information rather than merely occupying a common motion-like region. Figure 5(c) measures how that representation changes: linear CKA to SFT decreases from 1.000 at SFT to 0.830 at RL500 and 0.449 at RL780, showing that RL reorganizes, rather than simply preserves, the SFT representation.

This reorganization does not imply that the units lose their utility. Figure 5(d) compares two free-running metrics on all 4,198 prompts: generated-motion T2M R@1 increases from 0.313 at SFT to 0.506 at RL400 and 0.522 at RL500, while latent–caption cosine decreases from 0.910 to 0.783 and 0.756. Together, the four panels show that SFT learns a semantically grounded latent unit and trajectory-level RL adapts its geometry jointly with the downstream generation objective. Absolute target alignment therefore verifies learned semantic structure, whereas its change during RL should not be treated as a proxy for final motion quality.

D.2 Causal Use of Latent Units

To test whether the decoder uses a latent unit, we intervene on the RL500 latent path while holding the prompt, model parameters, and decoding procedure fixed; no textual or motion token is edited. The swap and permutation controls defined in Figure 6 retain essentially the same format validity and motion coverage as normal decoding. We exclude zeroing, discrete-pad replacement, block suppression, and target-oracle replacement because they sharply reduce valid-motion coverage and are therefore not quality-equivalent controls.

At each slot, the model produces z^{pred} and projects it to $h^{\mathrm{pred}} = f_\theta(z^{\mathrm{pred}})$. Normal decoding feeds $g_\theta(h^{\mathrm{pred}})$ to the next step; an intervention instead feeds $g_\theta(h^{\mathrm{inj}})$. Panels (a) and (b) measure the caption cosine of these projected states in the normalized 512-dimensional T2M semantic space. The injected state verifies the signal delivered to subsequent decoding, while the predicted state captures the autoregressive response at later slots. Swap decreases the predicted/injected caption cosine by $0.070/0.122$ on HumanML3D and $0.021/0.044$ on Motion-X. This verifies that the intervention changes the semantic-space signal and that downstream latent predictions react; it does not by itself test whether the generated motion acquires the source example's semantics. Permutation leaves pooled injected cosine unchanged by construction but shifts predicted-state cosine by -0.009 and -0.015, demonstrating sensitivity to within-block order beyond the pooled representation.

Panels (c) and (d) trace the same edits to generated-motion embeddings under T2M and LaMP. Swap changes at least one decoded motion token for 60.1% of HumanML3D prompts and 59.2% of Motion-X prompts; permutation changes 51.3% and 44.3%, respectively. Since all other decoding inputs are fixed, these changes establish that the autoregressive decoder actively consumes the latent feedback path. The any-change rate is a sensitivity diagnostic, not a measure of effect magnitude or targeted semantic transfer.

The mean signed caption–motion cosine differences remain near zero because the untargeted prompt-level effects can vary in direction and cancel; all T2M and LaMP intervals span zero. Together with target alignment, retrieval, and the main-paper ablation, the interventions establish grounded and behaviorally active conditioning, not independently controllable semantic factors.

E Limitations and Future Work

LUNAR inherits the representational scope of its frozen HumanML3D-style motion codec. It models whole-body movement but does not explicitly encode scene geometry, object state, contact events, or detailed hand and finger articulation. Prompts requiring precise manipulation or

environmental interaction can therefore be semantically plausible at the body level while remaining incomplete at the contact level.

The reasoning supervision and semantic objectives are also approximations. Teacher-generated atomic-motion decompositions may be ambiguous, and the frozen T2M space used for motion-semantic latent alignment and automatic rewards cannot cover every physical or fine-grained semantic constraint. Format, motion, and semantic rewards improve structured generation and caption alignment, but they do not guarantee physical correctness. Future work can extend the codec, latent targets, and rewards with interaction-aware representations, contact constraints, and finer temporal supervision.

F Complete Structured Reasoning Trajectories

This appendix reproduces two complete ID responses from a human-reviewed reasoning shortlist. They are illustrative rather than random or exhaustive and are not used for quantitative conclusions. The accompanying supplementary package provides eight additional verbatim responses; together, the ten examples cover four HumanML3D, two OOD-Para, three OOD-Comp, and one OOD-Motion-X prompt. Selection prioritizes faithful temporal decomposition, consistent laterality and direction, complete coverage of the ten body groups, and agreement between the internal trace and structured output.

Appendix Figure 7 separates a static overhead-arm pose from the coordinated arm descent and rightward pivot. Appendix Figure 8 preserves three ordered phases—backward travel, forward travel, and a two-foot jump—while assigning an explicit role to every body group. In locomotion segments, the left- and right-side fields describe each limb's participation across the full segment; parallel wording therefore denotes an alternating gait rather than same-frame synchronous steps. Both ID examples retrieve their paired motion at rank 1 under the T2M evaluator. Blue identifies structural reasoning and latent tags, green identifies body-group fields, and orange identifies the motion block and tokens.

Figure 7: Complete LUNAR response for the HumanML3D prompt "person has both arms above head before throwing them down and turning to the right."

```
<think>
We are given the motion description: "person has both arms above head before throwing them
     down and turning to the right"
 This motion can be broken down into two atomic motions:
 1. Having both arms above head (the starting position)
 2. Throwing the arms down and turning to the right

 However, note that the first part is a static position and the second part is a dynamic
     motion. We must describe the body groups for each atomic motion.

 For the first atomic motion (having both arms above head):
  - Head: Likely facing forward, neutral.
  - Torso: Upright and stable.
  - Left arm: Raised above head, extended upward.
  - Right arm: Raised above head, extended upward.
  - Left hand: Relaxed, following the arm movement.
  - Right hand: Relaxed, following the arm movement.
  - Left leg: Standing, supporting the body.
  - Right leg: Standing, supporting the body.
  - Left foot: Flat on the ground.
  - Right foot: Flat on the ground.

 For the second atomic motion (throwing the arms down and turning to the right):
  - Head: Turning to the right to look in the direction of the throw.
  - Torso: Rotating to the right and leaning slightly to the right.
  - Left arm: Lowering downward and extending to the right.
  - Right arm: Lowering downward and extending to the right.
  - Left hand: Relaxed, following the arm movement.
  - Right hand: Relaxed, following the arm movement.
  - Left leg: Stepping to the right to facilitate the turn.
  - Right leg: Pivoting on the ball of the foot to turn the torso to the right.
  - Left foot: Lifting and stepping to the right.
  - Right foot: Pivoting on the ball of the foot.
```

```
 Note: The throwing motion involves the arms and hands moving downward and to the right. The
     legs and feet are involved in the turn and stepping.

 Therefore, we have two atomic motions.
</think>
<output>1. Having both arms above head:
Head: Facing forward, neutral
Torso: Upright and stable
Left arm: Raised above head, extended upward
Right arm: Raised above head, extended upward
Left hand: Relaxed, following the arm movement
Right hand: Relaxed, following the arm movement
Left leg: Standing, supporting the body
Right leg: Standing, supporting the body
Left foot: Flat on the ground
Right foot: Flat on the ground
<latent_start><latent_pad><latent_pad><latent_pad><latent_pad><latent_end>

2. Throwing arms down and turning to the right:
Head: Turning to the right to look in the direction of the throw
Torso: Rotating to the right and leaning slightly to the right
Left arm: Lowering downward and extending to the right
Right arm: Lowering downward and extending to the right
Left hand: Relaxed, following the arm movement
Right hand: Relaxed, following the arm movement
Left leg: Stepping to the right to facilitate the turn
Right leg: Pivoting on the ball of the foot to turn the torso to the right
Left foot: Lifting and stepping to the right
Right foot: Pivoting on the ball of the foot
<latent_start><latent_pad><latent_pad><latent_pad><latent_pad><latent_end>
</output><motion><Motion_308><Motion_308><Motion_69><Motion_181><Motion_141><Motion_64><
     Motion_102><Motion_128><Motion_128><Motion_128><Motion_266><Motion_420><Motion_319><
     Motion_101><Motion_27><Motion_27><Motion_27><Motion_90><Motion_308><Motion_308><
     Motion_308></motion>
```

Figure 8: Complete LUNAR response for the HumanML3D prompt “a person moves backwards then forwards then jumps.”

```
<think>
We are given the motion description: "a person moves backwards then forwards then jumps."
 This description has three distinct phases: moving backwards, moving forwards, and jumping.
 Therefore, we can break it down into three atomic motions.

 However, note that the jumping motion is a separate atomic motion that occurs after the two
     movements.

 Let's describe each atomic motion in terms of the ten body groups.

 For the first atomic motion: moving backwards.
 - Head: Likely facing forward or slightly down to see the ground.
 - Torso: Leaning backward slightly to maintain balance.
 - Left arm: Swinging backward and forward to counterbalance the movement.
 - Right arm: Swinging backward and forward to counterbalance the movement.
 - Left hand: Relaxed, following the arm movement.
 - Right hand: Relaxed, following the arm movement.
 - Left leg: Stepping backward, bent at the knee to push off the ground.
 - Right leg: Stepping backward, bent at the knee to push off the ground.
 - Left foot: Pushing off the ground, then lifting and stepping backward.
 - Right foot: Pushing off the ground, then lifting and stepping backward.

 For the second atomic motion: moving forwards.
 - Head: Facing forward.
 - Torso: Leaning forward slightly to maintain balance.
 - Left arm: Swinging forward and backward to counterbalance the movement.
 - Right arm: Swinging forward and backward to counterbalance the movement.
 - Left hand: Relaxed, following the arm movement.
 - Right hand: Relaxed, following the arm movement.
 - Left leg: Stepping forward, bent at the knee to push off the ground.
 - Right leg: Stepping forward, bent at the knee to push off the ground.
 - Left foot: Pushing off the ground, then lifting and stepping forward.
 - Right foot: Pushing off the ground, then lifting and stepping forward.

 For the third atomic motion: jumping.
 - Head: Facing forward.
 - Torso: Upright or slightly leaning forward.
 - Left arm: Swinging upward and then downward to generate momentum.
```

```
 - Right arm: Swinging upward and then downward to generate momentum.
 - Left hand: Relaxed, following the arm movement.
 - Right hand: Relaxed, following the arm movement.
 - Left leg: Bending at the knee to lower the body, then extending to push off the ground and
     jump.
 - Right leg: Bending at the knee to lower the body, then extending to push off the ground
    and jump.
 - Left foot: Pushing off the ground and then leaving the ground to jump.
 - Right foot: Pushing off the ground and then leaving the ground to jump.

 Note: The jumping motion is a single continuous motion. We are describing the key movements
    during the jump.

 Let's write the output accordingly.
</think>
<output>1. Moving backwards:
Head: Facing forward or slightly down
Torso: Leaning backward slightly
Left arm: Swinging backward and forward
Right arm: Swinging backward and forward
Left hand: Relaxed, following arm movement
Right hand: Relaxed, following arm movement
Left leg: Stepping backward, bent at knee
Right leg: Stepping backward, bent at knee
Left foot: Pushing off ground, then lifting and stepping backward
Right foot: Pushing off ground, then lifting and stepping backward
<latent_start><latent_pad><latent_pad><latent_pad><latent_pad><latent_end>

2. Moving forwards:
Head: Facing forward
Torso: Leaning forward slightly
Left arm: Swinging forward and backward
Right arm: Swinging forward and backward
Left hand: Relaxed, following arm movement
Right hand: Relaxed, following arm movement
Left leg: Stepping forward, bent at knee
Right leg: Stepping forward, bent at knee
Left foot: Pushing off ground, then lifting and stepping forward
Right foot: Pushing off ground, then lifting and stepping forward
<latent_start><latent_pad><latent_pad><latent_pad><latent_pad><latent_end>

3. Jumping:
Head: Facing forward
Torso: Upright or slightly leaning forward
Left arm: Swinging upward and then downward
Right arm: Swinging upward and then downward
Left hand: Relaxed, following arm movement
Right hand: Relaxed, following arm movement
Left leg: Bending at knee to lower body, then extending to push off ground and jump
Right leg: Bending at knee to lower body, then extending to push off ground and jump
Left foot: Pushing off ground and then leaving ground to jump
Right foot: Pushing off ground and then leaving ground to jump
<latent_start><latent_pad><latent_pad><latent_pad><latent_pad><latent_end>
</output><motion><Motion_341><Motion_393><Motion_509><Motion_137><Motion_331><Motion_329><
    Motion_509><Motion_171><Motion_145><Motion_319><Motion_268><Motion_465><Motion_123><
    Motion_78><Motion_4><Motion_307><Motion_500><Motion_169><Motion_316><Motion_17><
    Motion_506><Motion_169><Motion_169><Motion_154><Motion_33><Motion_458><Motion_142><
    Motion_319><Motion_170><Motion_25><Motion_154><Motion_386><Motion_166><Motion_494><
    Motion_431><Motion_171><Motion_173><Motion_367><Motion_367><Motion_367><Motion_367></
    motion>
```

G DATA-CONSTRUCTION PROMPTS AND EXAMPLES

The four listings reproduce, in order, the atomic-motion teacher, atomic-motion verifier, OOD-Para rewrite, and OOD-Para verifier specifications, including their placeholders, required return formats, and in-prompt examples. At data-construction time, `[MOTION DESCRIPTION]` and `[CANDIDATE OUTPUT]` are replaced by the current example.

```
## Instruction
You generate cold-start reasoning data for human motion
understanding. Given a motion description:

1. Think step by step about how to break it into simple,
   independent atomic motions.
   - Show this reasoning inside <think>...</think> tags.
   - If the description is already atomic, segmentation
     may be skipped.
2. For every atomic motion, give a concise description
   for all ten body groups:
   Head, Torso, Left arm, Right arm, Left hand,
   Right hand, Left leg, Right leg, Left foot, Right foot.

## Output Format
<think>
[step-by-step decomposition]
</think>
1. Atomic Motion 1:
Head: ...
Torso: ...
Left arm: ...
Right arm: ...
Left hand: ...
Right hand: ...
Left leg: ...
Right leg: ...
Left foot: ...
Right foot: ...
2. Atomic Motion 2:
[repeat all ten body groups]

## Example Input
ties their shoelaces

## Example Output
1. Bend at the waist:
Head: Tilted downward
Torso: Bent forward at waist
Left arm: Extended toward left shoe
Right arm: Extended toward right shoe
Left hand: Reaching for left shoelace
Right hand: Reaching for right shoelace
Left leg: Slightly bent to stabilize
Right leg: Slightly bent to stabilize
Left foot: Flat on ground
Right foot: Flat on ground
2. Reach both hands toward shoes:
Head: Still tilted downward
Torso: Remains bent
Left arm: Grasping left lace
Right arm: Grasping right lace
Left hand: Holding left lace end
Right hand: Holding right lace end
Left leg: Stabilizing
Right leg: Stabilizing
Left foot: Unchanged
Right foot: Unchanged

## Motion Description
[MOTION DESCRIPTION]

## Output
Provide the output in the specified format, ensuring that
each atomic motion is clearly described.
```

```
Below are a user-provided motion description and a
generated decomposition.

## Motion Description
[MOTION DESCRIPTION]

## Generated Output
[CANDIDATE OUTPUT]

Verify the output using all of the following criteria:
1. Atomic actions: every listed motion is a simple,
   independent action derived from the description.
2. Body-group coverage: every action clearly describes
   head, torso, left/right arms, left/right hands,
   left/right legs, and left/right feet.
3. Faithfulness: descriptions reflect plausible motion.
4. Completeness: no required action is missing.
5. Clarity: descriptions are unambiguous and focused.

Return only one score:
1 if every criterion is met; 0 otherwise.
```

```
## Instruction
Rewrite the human-motion description in a different form
while preserving exactly the same semantics. Use new
wording and return only the concise rewritten description,
with no commentary.

## Example Input
ties their shoelaces

## Example Output
bends forward and secures the laces on their shoes

## Motion Description
[MOTION DESCRIPTION]

## Rewritten Description
```

```
Below are an original motion description and its rewrite.

## Original Motion Description
[MOTION DESCRIPTION]

## Rewritten Motion Description
[CANDIDATE OUTPUT]

Verify the rewrite using all of the following criteria:
1. Semantic equivalence: it describes exactly the same
   action.
2. Clarity: it is clear and unambiguous.
3. Conciseness: it is succinct and does not add or omit
   details.

Return only one score:
1 if the rewrite is semantically equivalent and clear;
0 otherwise.
```