MOGO: Residual Quantized Hierarchical Causal Transformer for Real-Time and Infinite-Length 3D Human Motion Generation

Anonymous submission

Abstract

Recent advances in transformer-based text-to-motion generation have significantly improved motion quality. However, achieving both real-time performance and long-horizon scalability remains an open challenge. In this paper, we present MOGO (Motion Generation with One-pass), a novel autoregressive framework for efficient and scalable 3D human motion generation. MOGO consists of two key components. First, we introduce MoSA-VQ, a motion scale-adaptive residual vector quantization module that hierarchically discretizes motion sequences through learnable scaling parameters, enabling dynamic allocation of representation capacity and producing compact yet expressive multi-level representations. Second, we design the RQHC-Transformer, a residual quantized hierarchical causal transformer that decodes motion tokens in a single forward pass. Each transformer block aligns with one quantization level, allowing hierarchical abstraction and temporally coherent generation with strong semantic flow. Compared to diffusion- and LLM-based approaches, MOGO achieves lower inference latency while preserving high motion fidelity. Moreover, its hierarchical latent design enables seamless and controllable infinite-length motion generation, with stable transitions and the ability to adaptively incorporate updated control signals at arbitrary points in time. To further enhance generalization and interpretability, we introduce Textual Condition Alignment (TCA), which leverages large language models with Chain-of-Thought reasoning to bridge the gap between real-world prompts and training data. TCA not only improves zero-shot performance on unseen datasets but also enriches motion comprehension for in-distribution prompts through explicit intent decomposition. Extensive experiments on HumanML3D, KIT-ML, and the unseen CMP dataset demonstrate that MOGO outperforms prior methods in generation quality, inference efficiency, and temporal scalability.

Introduction

Text-to-motion generation is a rapidly evolving research area with growing importance in virtual environments, such as gaming, AR/VR, and humanoid robotics (Zhu et al. 2023). Recent advances have leveraged large language model (LLM)-based techniques to generate high-quality 3D human motion from text descriptions, typically using vector-quantized variational autoencoders (VQ-VAE) and autoregressive decoding strategies (Guo et al. 2024; Zhang et al. 2023a; Pinyoanuntapong et al. 2024; Zhong et al. 2023). Meanwhile, diffusion-based approaches have also demonstrated strong generation performance, particularly in terms of motion fidelity and diversity (Tevet et al. 2022).

Despite these achievements, both diffusion- and LLM-based frameworks face practical limitations (Rombach et al. 2022; Chen et al. 2023; Brown et al. 2020; Raffel et al. 2020). Diffusion models often rely on iterative refinement processes, which introduce significant inference latency and hinder their suitability for real-time or interactive applications (Rombach et al. 2022). On the other hand, LLM-based models, although autoregressive, typically involve large parameter sizes and long context dependencies, leading to high memory and computation costs that challenge deployment on lightweight scenarios (Brown et al. 2020). A broader discussion of related research trends is provided in Section .

To address the challenges of motion generation quality, inference efficiency, and generalization, we propose MOGO, a unified and efficient transformer-based framework for expressive text-to-motion synthesis. MOGO generates high-fidelity motion sequences from textual input in a single forward pass, combining structural compactness, temporal coherence, and extendable decoding. The framework is composed of four key components, each targeting a core aspect of the generation pipeline:

- We propose **MoSA-VQ**, a motion structure-aware quantization framework with two key innovations: (i) a learnable modulation mechanism that adaptively scales residual magnitudes across quantization levels, and (ii) a novel cross-level decorrelation loss that enforces statistical orthogonality between quantized vectors and subsequent residuals. Together, these components enable more compact and hierarchically disentangled representations, leading to better reconstruction under constrained codebook capacity.
- **RQHC-Transformer (Residual Quantized Hierarchical Causal Transformer)** functions as the motion decoder. It autoregressively generates multi-layer motion tokens in a single forward pass, with each transformer block dedicated to decoding one level of the quantized hierarchy. This design ensures temporally coherent synthesis while preserving decoding efficiency and compatibility with interactive applications.
- **Infinite-Length Generation** is enabled by the au-

toregressive nature of MOGO's architecture. Unlike diffusion-based methods that require fixed-length sampling, MOGO supports seamless motion extension with consistent structure and fidelity, making it suitable for real-time or open-ended generation scenarios.

- **Textual Condition Alignment (TCA)** leverages large language models with Chain-of-Thought reasoning to bridge the gap between real-world user prompts and the model's training distribution. Beyond simple style normalization, TCA explicitly decomposes complex motion instructions into structured semantic cues, enabling more accurate motion interpretation. This enhances MOGO's generalization capability across both in-distribution and zero-/few-shot scenarios, while improving control fidelity and robustness to diverse user inputs.

Related Work

Human Motion Generation. Recent advances in human motion generation have enabled conditioning on modalities like text, audio, music, and images (Zhu et al. 2023). Early deterministic models (Ahuja and Morency 2019; Ghosh et al. 2021) often produced over-smoothed, unrealistic motions. To address this, stochastic approaches, such as GANs (Cai et al. 2018; Wang et al. 2020) and VAE-based models (Guo et al. 2022c; Petrovich, Black, and Varol 2021), improved motion diversity. Text-to-motion generation gained prominence with works like (Guo et al. 2022a), which used temporal VAEs to model text-motion distributions. Recently, diffusion-based methods (Kim, Kim, and Choi 2023; Zhang et al. 2024, 2023b; Kong et al. 2023; Chen et al. 2023; Tevet et al. 2022) and transformer-based approaches (Guo et al. 2024; Zhang et al. 2023a; Jiang et al. 2023) have led the field.

LLM-Based Motion Generation Models. LLM-based architectures have become a cornerstone of text-to-motion generation, leveraging their ability to model sequential data and adapt to varied tasks (Guo et al. 2024; Zhang et al. 2023a; Pinyoanuntapong et al. 2024; Zhong et al. 2023; Jiang et al. 2023). Models like MoMask (Guo et al. 2024) and MMM (Pinyoanuntapong et al. 2024) use masked token modeling to produce high-quality motions but struggle with extendable output and generalization in low-data or out-of-distribution settings due to their bidirectional design. Conversely, autoregressive models like T2M-GPT (Zhang et al. 2023a) and AttT2M (Zhong et al. 2023) enable sequential generation, making them suitable for real-time applications and scalable with larger datasets, though they often sacrifice some motion quality. Efforts like MotionGPT (Jiang et al. 2023) integrate multimodal language modeling but face challenges in achieving high-fidelity motion outputs. Our MOGO framework addresses these issues by combining efficient encoding through the MoSA-VQ with a single-pass autoregressive transformer, ensuring both high-quality motion and interactive capabilities.

Hierarchical Transformers. Hierarchical transformer architectures excel in domains like NLP (Nawrot et al. 2022; Pappagari et al. 2019), image generation (Ding et al. 2022), and vision tasks (Liu et al. 2021; Chen et al. 2022). By processing data at multiple abstraction levels, these models enhance representation capacity and scalability. For instance, Swin Transformers (Liu et al. 2021) support scalable high-resolution vision, while CogView2 (Ding et al. 2022) enables high-fidelity image synthesis. In motion generation, hierarchical transformers are underexplored, especially for autoregressive frameworks with residual quantization. Our RQHC-Transformer in MOGO leverages hierarchical modeling to efficiently process multi-layer motion tokens, improving quality and generalization.

Motion Tokenization. Discretizing continuous motion data into tokens v'ia vector quantization is central to transformer-based motion generation. TM2T (Guo et al. 2022b) introduced VQ-VAE to map motions to discrete sequences. T2M-GPT (Zhang et al. 2023a) enhanced token quality with exponential moving average and codebook reset techniques. AttT2M (Zhong et al. 2023) improved quantization through body-part-aware encoding. MoMask (Guo et al. 2024) advanced this with residual vector quantization (RVQ), producing multi-level tokens for better reconstruction quality. Compared to prior work, our approach introduces learnable feature scaling into the residual vector quantization process, allowing each quantization level to adaptively adjust the magnitude of its residuals. This scaled RVQ mechanism stabilizes the residual distribution across quantization stages, mitigates the risk of later stages collapsing to noise, and ensures more effective codebook utilization. As a result, the model benefits from improved reconstruction fidelity, better token expressiveness, and enhanced training stability.

Methods

Overview of the MOGO Framework

Our proposed framework, MOGO, as showned in Figure 1, consists of four key components designed for efficient and expressive text-to-motion generation. First, MoSA-VQ discretizes motion sequences into multi-level latent tokens using hierarchical residual quantization with adaptive scaling. Second, RQHC-Transformer autoregressively decodes the quantized representations with a stack of level-aligned transformer blocks. Third, MOGO supports infinite-length generation via seamless autoregressive extension. Lastly, we introduce TCA to enhance generalization to real-world prompts. We elaborate on each component in the following subsections.

MoSA-VQ: Motion Scale-Adaptive Residual VQ-VAE

To transform continuous human motion into discrete representations suitable for autoregressive generation, we build upon residual vector quantized variational autoencoders (RVQ-VAEs) and propose MoSA-VQ. While standard RVQ-based approaches (Guo et al. 2024; Zhong et al. 2023) hierarchically encode motion features through fixed-stage quantization, they often rely on manually tuned feature ranges and static scaling behavior, which limit their adaptability and quantization efficiency—especially in the presence of diverse and complex motions.

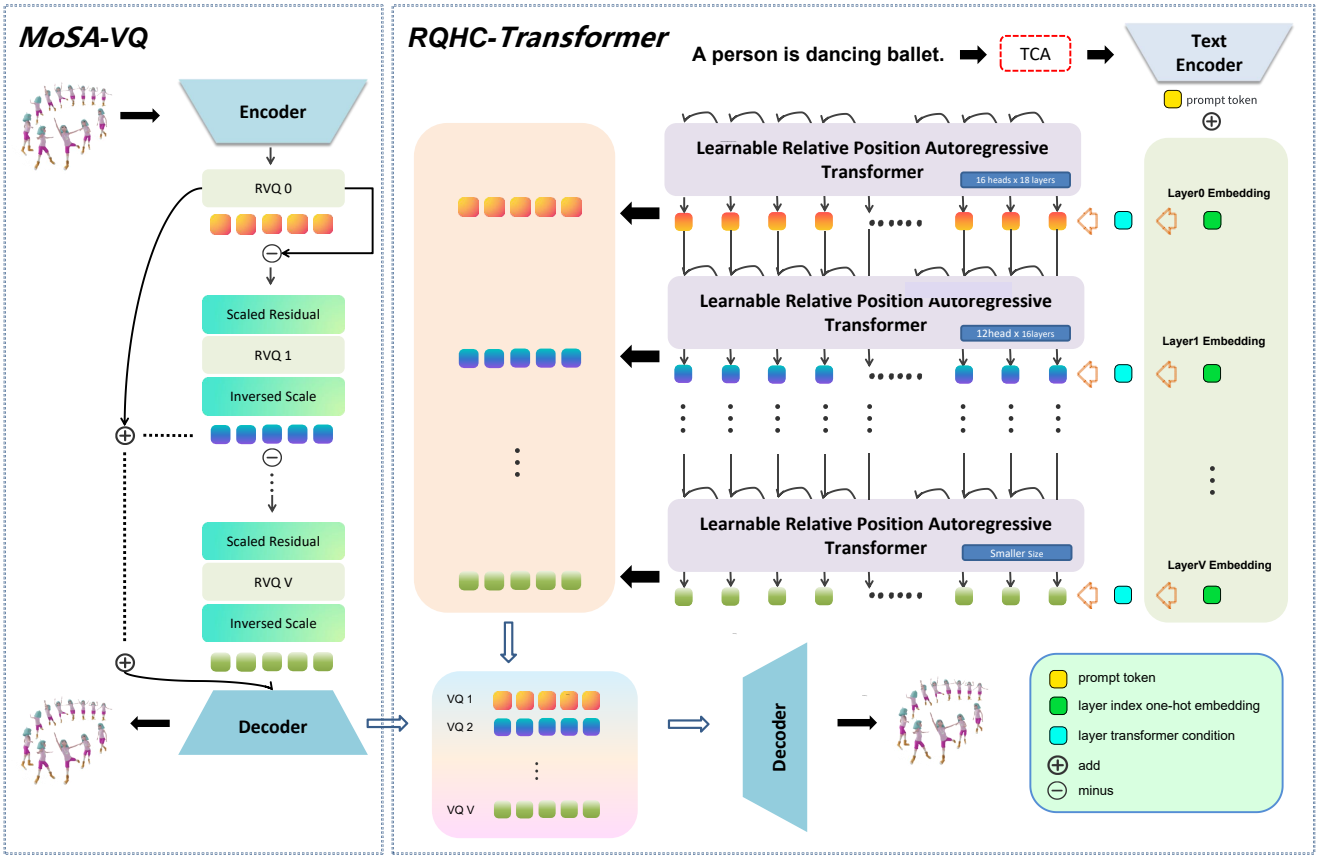

Figure 1: Overview of the proposed MOGO framework. *Left*: We first encode the motion sequence using a hierarchical MoSA-VQ module with learnable feature scaling, indexing a multi-level codebook. *Right*: Then, tokens are autoregressively generated by a RQHC-Transformer with relative positional attention. Finally, the predicted token sequence is quantized and decoded into a full 3D human motion sequence.

Adaptive Hierarchical Quantization. To address these limitations, we introduce a learnable feature scaling mechanism into each quantization level. Specifically, each level is equipped with its own learnable scale and bias parameters, which dynamically adjust the amplitude and offset of the input residual features before quantization. This adaptive design allows each level to allocate representational capacity where needed, resulting in more effective codebook usage and higher reconstruction accuracy without the need for handcrafted calibration.

At each hierarchical quantization level $l \in \{0, 1, \ldots, L\}$, the residual vector is first transformed using a learnable affine mapping. We denote this process as:

$$\mathbf{q}^l = \mathcal{Q}(\mathbf{W}^l \mathbf{r}^l + \mathbf{b}^l), \quad \boldsymbol{\phi}^l = (\mathbf{W}^l)^{-1}(\mathbf{q}^l - \mathbf{b}^l), \qquad (1)$$

where $\mathbf{W}^l \in \mathbb{R}^{d \times d}$ is a diagonal scaling matrix, $\mathbf{b}^l \in \mathbb{R}^d$ is a learnable bias vector, $\mathcal{Q}(\cdot)$ denotes nearest-neighbor quantization via a codebook, and $\boldsymbol{\phi}^l$ represents the decoded contribution at level l.

This hierarchical quantization scheme enables progressive residual refinement. Unlike conventional residual vector quantization (RVQ) frameworks that apply uniform scaling across all levels, our design introduces level-specific adaptive modulation through learnable parameters $\mathbf{W}^l$ and $\mathbf{b}^l$. This flexibility allows coarse levels to capture global structure, while deeper layers focus on subtle motion or spatial details. The inverse affine transformation ensures consistent feature recovery, and stride-1 convolutions in both encoder and decoder maintain temporal continuity without aliasing artifacts. We have

$$\mathbf{r}^{l+1} = \mathbf{r}^l - \boldsymbol{\phi}^l, \quad \hat{\mathbf{z}} = \sum_{l=0}^{L} \boldsymbol{\phi}^l. \qquad (2)$$

Training Objective. Our full training objective combines four terms: motion reconstruction, codebook commitment, lightweight modulation regularization, and a novel cross-level decorrelation loss.

We adopt an ℓ_1 reconstruction loss to ensure accurate motion recovery, and a standard commitment loss to encourage the encoder residuals $\mathbf{r}^l$ to remain close to their quantized counterparts $\boldsymbol{\phi}^l$. A mild regularization is also applied to the per-layer scaling and bias parameters $(\mathbf{W}^l, \mathbf{b}^l)$, following prior work on stable quantization with learnable modulation.

Additionally, we introduce a cross-level decorrelation loss, which explicitly penalizes statistical redundancy between the quantized vector $\boldsymbol{\phi}^l$ and the subsequent residual $\mathbf{r}^{l+1}$:

$$\mathcal{L}_{\text{decor}} = \sum_{l=0}^{L-1} \left\| \text{Cov}\left(\boldsymbol{\phi}^l, \mathbf{r}^{l+1}\right) \right\|_F^2 .$$

This loss is motivated by the principle that, in an ideal residual quantization hierarchy, each level should encode novel information orthogonal to what has already been captured. A nonzero covariance between $\boldsymbol{\phi}^l$ and $\mathbf{r}^{l+1}$ indicates overlapping representation subspaces, leading to inefficient coding and excessive reliance on deeper quantizers. By minimizing their batch-wise covariance, we enforce an approximate statistical orthogonality between levels, which encourages disentangled information flow and hierarchical efficiency.

From an information-theoretic perspective, this decorrelation loss can be viewed as promoting minimal mutual information between consecutive encoding layers, thus maximizing the effective bit utilization of each quantized token. Empirically, we find this regularization yields more compact latent codes and improves reconstruction quality under constrained codebook budgets. The final loss function is:

$$\mathcal{L}_{\text{vq}} = \gamma \sum_{l=0}^{L} \underbrace{\left(\left\| \mathbf{W}^l - \mathbf{I} \right\|_F^2 + \left\| \mathbf{b}^l \right\|_2^2 \right)}_{\text{modulation reg.}} + \beta \sum_{l=0}^{L} \underbrace{\left\| \mathbf{r}^l - \text{sg}[\boldsymbol{\phi}^l] \right\|_2^2}_{\text{commitment}}$$
$$+ \lambda \sum_{l=0}^{L-1} \underbrace{\left\| \text{Cov}\left(\boldsymbol{\phi}^l, \mathbf{r}^{l+1}\right) \right\|_F^2}_{\text{cross-level decorrelation}} + \underbrace{\|\mathbf{m} - \hat{\mathbf{m}}\|_1}_{\text{reconstruction}} . \quad (3)$$

Here, β, λ and γ are weight factors.

RQHC-Transformer: Residual Quantized Hierarchical Causal Transformer

To fully leverage the high-quality hierarchical representations produced by MoSA-VQ, we propose the Residual Quantized Hierarchical Causal Transformer (RQHC-Transformer), an autoregressive decoder structurally aligned with the residual quantization process. In contrast to single-resolution or per-layer decoding approaches, RQHC jointly models all quantization levels in a residual-aware and level-aware manner. Moreover, we adopt a causal Transformer design, which—compared to bidirectional transformers, LLM-style decoders, or iterative diffusion models—offers efficient one-pass inference, supports streamable generation, and naturally enables long-horizon synthesis. The coarse-to-fine decoding progression preserves structural semantics, mitigates token interference, and promotes stable, coherent motion generation with improved interpretability.

This hierarchical causal architecture progressively refines coarse motion structures, preserving semantic consistency across abstraction levels and reducing token-level interference. Crucially, its design is structurally aligned with the residual quantization process of MoSA-VQ, enabling stable information flow across layers. Together, this synergy allows MOGO to extend motion from any given frame with temporal continuity and semantic fidelity, laying the foundation for real-time, controllable infinite-length generation.

Hierarchical Causal Generation. To generate the token sequence at quantization level l, we construct an input sequence $\mathbf{s}^l$ that conditions on both the textual prompt and the residual context from lower levels:

$$\mathbf{s}^l = [\mathbf{p} + \mathbf{q}_{\text{emb}}, \mathbf{t}_l^{1:n}], \quad (4)$$

where $\mathbf{p}$ is the CLIP-based text prompt embedding, and $\mathbf{q}_{\text{emb}}$ is the embedding representing the current quantization level. The sequence $\mathbf{t}_l^{1:n}$ is computed by aggregating token embeddings across all previous levels at each position:

$$\mathbf{t}_l^{1:n} = \left[\text{Embed}(t_l^1),\ \text{Embed}(t_l^2),\ \ldots,\ \text{Embed}(t_l^n)\right] . \quad (5)$$

Here, n denotes the number of motion tokens (i.e., temporal steps) in the sequence. Each t_l^i is the i-th quantized token at layer l, and $\text{Embed}(\cdot)$ maps each token to its corresponding embedding vector.

Relative Positional Encodings. To effectively handle long motion sequences, we incorporate a relative positional encoding scheme into our causal attention layers. Compared to absolute positional encodings, this approach better preserves attention consistency for varying sequence lengths and enhances the model's ability to capture long-range dependencies.

Given an input token sequence, the attention score between token i and token j is computed as:

$$\mathbf{A}_{i,j}^{\text{rel}} = \mathbf{q}_i^\top \mathbf{k}_j + \mathbf{q}_i^\top \phi_{i-j} + u^\top \mathbf{k}_j + v^\top \phi_{i-j}, \quad (6)$$

where $\mathbf{q}_i$ and $\mathbf{k}_j$ are the query and key embeddings at position i and j, and ϕ_{i-j} is a learned relative position embedding, and u and v are learnable global bias vectors. The final output of the self-attention is:

$$\mathbf{a}_l^n = \text{Softmax}(\mathbf{A}_l^n)\mathbf{V}_l^n, \quad (7)$$

where $\mathbf{A}_l^n$ is the attention score matrix incorporating relative positional encoding at layer n of the l-th quantization level. $\mathbf{V}_l^n$ denotes the value matrix at the same layer.

Training Objective. To learn temporally coherent and hierarchically consistent token generation, we adopt a multi-layer autoregressive training objective defined over all quantization levels. The loss function is written as:

$$\mathcal{L}_{\text{ce}} = -\mathbb{E}_{(\mathbf{t},c)\sim\mathcal{D}} \left[\sum_{l=1}^{L} \log p_\theta(\mathbf{t}^l \mid \mathbf{t}_<^l, c) \right], \quad (8)$$

where $\mathcal{L}_{\text{ce}}$ denotes the total cross-entropy loss computed over all quantization levels. The training data $\mathcal{D}$ consists of pairs $(\mathbf{t}, c)$, where $\mathbf{t} = \{\mathbf{t}^1, \ldots, \mathbf{t}^L\}$ is the set of discrete motion token sequences across L quantization levels, and c is the global conditioning signal derived from the input text and the residual context from coarser levels. Each sequence $\mathbf{t}^l = (t_1^l, t_2^l, \ldots, t_T^l)$ corresponds to the motion tokens at level l, with T being the number of time steps. The term $\mathbf{t}_<^l$ denotes the causal context of $\mathbf{t}^l$, i.e., all tokens preceding the current position. The model p_θ, parameterized by weights θ, is trained to predict each token autoregressively based on its past tokens and the shared condition c. This objective encourages consistent hierarchical structure and smooth temporal evolution across all levels.

Infinite-Length Generation

By leveraging MoSA-VQ's temporally discrete codes and RQHC's autoregressive decoding, our framework naturally supports infinite-length motion generation from arbitrary time steps. During extension, the model conditions on previously generated tokens (excluding the original prompt embedding), while allowing the text prompt to be substituted dynamically—enabling seamless continuation under updated instructions.

Flexible Continuation with Prompt Switching. At any generation step n, given an existing motion token sequence $\mathbf{t}_l^{1:n-1} = (t_l^1, t_l^2, \ldots, t_l^{n-1})$ at quantization level l, we continue autoregressive generation from step n onward using the RQHC-Transformer. To support runtime intervention, we allow prompt switching by replacing the original text condition $\mathbf{p}_{\text{old}}$ with a new prompt $\mathbf{p}_{\text{new}}$, yielding an updated conditioning vector $c = \mathbf{p}_{\text{new}} + \mathbf{q}_{\text{emb}}$.

While the previous tokens $\mathbf{t}_l^{1:n-1}$ are preserved, the updated prompt influences subsequent decoding. At each step n, RQHC predicts the next token t_l^n based on the causal context and the new condition:

$$t_l^n \sim p_\theta(t_l^n \mid \mathbf{t}_l^{1:n-1}, c), \quad \forall n > n{-}1. \tag{9}$$

This per-step decoding continues autoregressively and can be extended to arbitrary lengths, enabling real-time motion continuation from any point. Crucially, the updated prompt takes effect immediately from step n, without requiring regeneration of prior tokens. Only after completing the desired generation (of any length) do we perform motion decoding to reconstruct full-body trajectories. This design enables flexible, prompt-controllable extension with temporal coherence and efficient computation.

Why This Works. This flexible and prompt-adaptive generation capability stems from the close architectural synergy between MoSA-VQ and RQHC-Transformer. MoSA-VQ's residual quantization provides a structured representation where coarse global motion is captured in early levels and finer dynamics are encoded hierarchically. This layered encoding ensures stability during long-horizon generation, as new content naturally extends existing motion without disrupting overall structure. In advanced, RQHC's hierarchical causal structure enables prompt substitution at any point by conditioning future generation on preserved prior tokens, ensuring smooth semantic transitions. Together, these properties support real-time extension, runtime editing, and infinite-length motion generation under evolving conditions.

Text Condition Alignment for Improved Motion Decoding

Real-world user prompts for motion generation often deviate from the structured annotations seen during training, varying in style, verbosity, and semantic complexity. Notably, many instructions contain abstract or composite motions that are difficult to execute when treated as a single condition, posing a fundamental challenge to robust generation.

To bridge this gap, we propose TCA, an inference-time strategy that leverages a pretrained large language model $\mathcal{T}$ to enhance both interpretability and executability of free-form prompts—without modifying model parameters. TCA performs two key functions: (1) *style normalization*, aligning user input with the model's linguistic prior, and (2) *instruction decomposition*, splitting complex intents into atomic motion steps.

Formally, given an arbitrary user prompt $\mathbf{c}_{\text{raw}} \in \mathbb{U}$, TCA first produces a normalized instruction $\mathbf{c}_{\text{norm}} \in \mathbb{S}$, then applies Chain-of-Thought prompting over $\mathcal{T}$ to decompose it as:

$$[\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(K)}] = \mathcal{T}(\mathbf{c}_{\text{norm}} \mid \Psi_{\text{CoT}}), \tag{10}$$

where Ψ_{CoT} encodes few-shot exemplars and reasoning patterns.

Each $\mathbf{c}^{(k)}$ denotes a semantically atomic sub-instruction, enabling the motion model to synthesize sequences in a stepwise or joint manner with improved controllability, precision, and temporal coherence.

By externally structuring complex instructions into executable units, TCA enhances generation quality across both seen and unseen prompts—without requiring any fine-tuning or additional supervision.

Experiments

We comprehensively evaluate our MOGO framework on text-to-motion generation tasks, focusing on motion quality, text-motion alignment, and generalization. Additional experimental results and ablation studies are provided in the appendix for further analysis.

Datasets and Evaluation Metrics

Datasets We train and evaluate MOGO on two widely used benchmarks, HumanML3D (Guo et al. 2022a) and KIT-ML (Plappert, Mandery, and Asfour 2016), following the data splits of T2M (Guo et al. 2022a). To assess out-of-distribution generalization, we further evaluate zero-shot performance on CMP (Liao et al. 2024), which contains non-daily, combat-style motions. Prompt engineering is disabled during CMP evaluation to ensure fairness.

Evaluation Metrics We adopt standard metrics from (Guo et al. 2022a). *FID* (Frechet Inception Distance) measures motion realism and serves as a primary evaluation metric. *R-Precision* evaluates text-motion alignment (Top-1, 2, 3). *MM-Dist* computes the distance between motion and text embeddings. *MultiModality* measures the variance of motions generated from the same text prompt.

Training Details The codebook is sized at 8192×128 with 6 quantization layers, using stride-1 convolutions and 0.2 dropout. It is trained with AdamW (learning rate 2×10^{-4}, batch size 512) for 2000 iterations on an NVIDIA 4090 GPU. RQHC-Transformer has 6 sub-modules aligned with RVQ layers, with head counts $[16, 12, 6, 2, 2, 2]$ and layer counts $[18, 16, 8, 4, 2, 2]$, and a model dimension of 1024. For HumanML3D, it is trained on an A800-80G with cosine decay from 2.5×10^{-5} to 3×10^{-6}, batch size 32, for 1500 epochs. For KIT-ML, a V100-32G is used with a learning rate from 3×10^{-5} to 3×10^{-6}, batch size 48.

Table 1: Comparison with Motion Generation Models. **Bold Boxed** = Best, **Bold** = Second Best, Underline = Third Best.

Datasets	Methods	R Precision ↑			FID ↓	MultiModal Dist ↓	MultiModality ↑
		Top 1	Top 2	Top 3			
HumanML3D	MotionDiffuse (Zhang et al. 2024)	0.491±0.001	0.681±0.001	0.782±0.001	0.630±0.001	3.113±0.001	1.553±0.042
	T2M-GPT† (Zhang et al. 2023a)	0.491±0.003	0.680±0.002	0.775±0.002	0.116±0.004	3.118±0.011	1.856±0.011
	Fg-T2M (Wang et al. 2023)	0.492±0.002	0.683±0.003	0.783±0.003	0.243±0.019	3.109±0.007	1.614±0.049
	AttT2M† (Zhong et al. 2023)	0.499±0.005	0.690±0.006	0.786±0.004	0.112±0.004	3.038±0.016	2.452±0.043
	MotionGPT† (Jiang et al. 2023)	0.492±0.003	0.681±0.003	0.778±0.002	0.232±0.008	3.096±0.009	2.008±0.084
	MoMask (Guo et al. 2024)	**0.521**±0.002	**0.713**±0.002	**0.807**±0.002	**0.045**±0.002	2.958±0.008	1.241±0.040
	MMM (Pinyoanuntapong et al. 2024)	0.504±0.002	0.696±0.003	0.794±0.004	0.080±0.004	2.998±0.007	1.226±0.035
	MotionAnything (Zhang et al. 2025)	**0.546**±0.002	**0.735**±0.002	**0.829**±0.002	**0.028**±0.001	**2.859**±0.010	**2.705**±0.060
	MOGO†	0.515±0.003	0.709±0.003	0.801±0.003	0.064±0.002	2.951±0.008	2.108±0.070
	MOGO with TCA†	0.527±0.007	0.722±0.008	0.827±0.012	0.038±0.003	**2.849**±0.003	**2.344**±0.037
KIT-ML	MotionDiffuse (Zhang et al. 2024)	0.417±0.004	0.621±0.004	0.739±0.004	1.954±0.062	2.958±0.005	0.730±0.013
	T2M-GPT† (Zhang et al. 2023a)	0.416±0.006	0.627±0.006	0.745±0.006	0.514±0.029	3.007±0.023	1.570±0.039
	Fg-T2M (Wang et al. 2023)	0.418±0.005	0.626±0.004	0.745±0.004	0.571±0.047	3.114±0.015	1.019±0.029
	AttT2M† (Zhong et al. 2023)	0.413±0.006	0.632±0.006	0.751±0.006	0.870±0.039	3.039±0.016	2.281±0.043
	MotionGPT† (Jiang et al. 2023)	0.366±0.005	0.558±0.004	0.680±0.005	0.510±0.004	3.527±0.021	**2.328**±0.117
	MoMask (Guo et al. 2024)	**0.433**±0.007	**0.656**±0.005	**0.781**±0.005	**0.204**±0.011	2.779±0.022	1.131±0.043
	MMM (Pinyoanuntapong et al. 2024)	0.381±0.005	0.590±0.006	0.718±0.005	0.429±0.019	3.146±0.019	1.105±0.026
	MotionAnything (Zhang et al. 2025)	**0.449**±0.007	**0.678**±0.004	**0.802**±0.006	**0.131**±0.003	**2.705**±0.024	1.374±0.060
	MOGO†	0.420±0.007	0.634±0.007	0.754±0.007	0.313±0.016	2.957±0.029	2.063±0.066
	MOGO with TCA†	0.447±0.023	0.668±0.016	0.801±0.007	0.191±0.009	2.849±0.007	**2.273**±0.073
CMP (zero-shot)	T2M-GPT† (Zhang et al. 2023a)	0.061±0.003	0.103±0.005	0.147±0.006	16.092±0.099	4.179±0.049	2.118±0.033
	AttT2M† (Zhong et al. 2023)	0.065±0.004	0.109±0.008	0.147±0.008	18.403±0.071	4.048±0.017	2.208±0.019
	MotionGPT† (Jiang et al. 2023)	0.050±0.002	0.094±0.002	0.133±0.003	15.654±0.183	4.431±0.021	**5.535**±0.259
	MoMask (Guo et al. 2024)	0.062±0.003	0.108±0.005	0.150±0.004	24.351±0.205	4.817±0.022	1.651±0.050
	MMM (Pinyoanuntapong et al. 2024)	0.067±0.004	0.116±0.008	0.154±0.008	17.087±0.313	4.360±0.017	2.802±0.011
	MOGO†	0.071±0.003	0.124±0.004	0.183±0.004	10.388±0.171	3.847±0.029	4.562±0.066
	MOGO with TCA†	**0.122**±0.006	**0.227**±0.011	**0.304**±0.004	**6.873**±0.073	**3.040**±0.014	**4.299**±0.013

Table 2: Comparison of reconstruction quality between our VAE design and prior motion VAEs.

HumanML3D		KIT-ML	
Method	**FID ↓**	**Method**	**FID ↓**
M2DM	$0.063^{\pm 0.001}$	M2DM	$0.413^{\pm 0.009}$
T2M-GPT	$0.070^{\pm 0.001}$	T2M-GPT	$0.472^{\pm 0.011}$
MoMask	$0.019^{\pm 0.001}$	MoMask	$0.112^{\pm 0.002}$
MMM	$0.075^{\pm 0.001}$	MMM	$0.641^{\pm 0.014}$
MOGO (ours)	$\mathbf{0.013}^{\pm 0.001}$	**MOGO (ours)**	$\mathbf{0.037}^{\pm 0.001}$

Comparison to State-of-the-art Approaches

As shown in Figure 2, motions produced by our model are more semantically consistent with the input descriptions and exhibit smoother, more stable transitions without noticeable spatial drift. Additional qualitative visualizations, including extended motion generation results, are provided in the Appendix.

Reconstruction Quality of Motion Encoder We compare our VAE design with existing motion VAEs—M2DM, T2M-GPT, MoMask, and MMM—on HumanML3D and KIT-ML (Table 2). MOGO achieves the best results on both benchmarks, with an FID of 0.013 on HumanML3D, outperforming MoMask (0.019) and others. On the more challenging KIT-ML dataset, MOGO achieves an FID of 0.037, significantly lower than MoMask (0.112), T2M-GPT (0.472), and MMM (0.641). These results demonstrate the strong reconstruction capability of our VAE across diverse motion distributions.

Quantitative Results. MOGO demonstrates strong performance across HumanML3D, KIT-ML, and zero-shot CMP benchmarks, particularly on FID and R@3. As showned in Table 1, On HumanML3D, our base model achieves an FID of 0.038, outperforming T2M-GPT (0.116), MotionGPT (0.232), and MotionDiffuse (0.630). With TCA, FID slightly increases to 0.064, while R@3 improves from 0.801 to 0.827. On KIT-ML, MOGO attains an FID of 0.191 and R@3 of 0.801, competitive with MotionAnything (FID 0.131, R@3 0.802). Under the zero-shot CMP setting, FID drops to 10.388 (6.873 with TCA), outperforming MotionGPT (15.654) and T2M-GPT (16.092), while R@3 reaches 0.304, over twice that of the best baseline MMM (0.154).

Overall, our method consistently achieves top FID and R@3 scores across benchmarks. The TCA module further enhances semantic alignment, especially under distribution shifts, validating the effectiveness of structured condition interpretation.

Ablation Study

Ablation Study on VAE Depth We evaluate the impact of quantization depth by varying the number of layers while fixing the codebook size to 8192×128 (Table 3). A single-layer quantizer performs poorly (FID: 0.070), indicating limited capacity. Increasing to 3 layers significantly improves performance (FID: 0.021), and 6 layers yields the

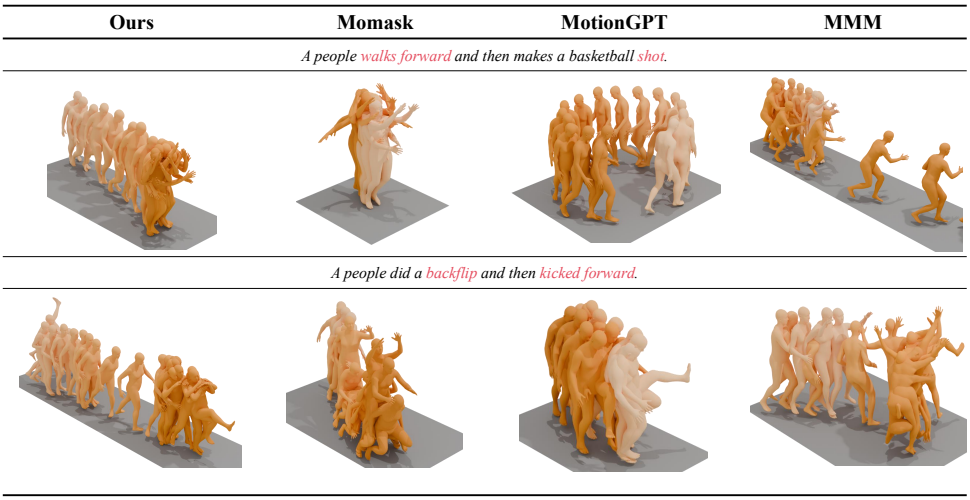

Figure 2: Qualitative comparison of generated motions produced by different models. Compared to prior methods Momask, MotionGPT, and MMM.

Table 3: The impact of different quantization layers on model reconstruction quality when the codebook size is 8192×128. Bold face indicates the best result.

Depth	FID↓	R TOP1 ↑	MM-Dist↓
1	$0.070^{\pm 0.001}$	$0.502^{\pm 0.001}$	$2.999^{\pm 0.006}$
3	$0.021^{\pm 0.001}$	$0.508^{\pm 0.001}$	$2.992^{\pm 0.008}$
6	$\mathbf{0.016}^{\pm 0.001}$	$0.510^{\pm 0.001}$	$2.989^{\pm 0.007}$
7	$0.016^{\pm 0.001}$	$0.509^{\pm 0.001}$	$2.996^{\pm 0.003}$

best overall results (FID: **0.016**, R@1: 0.510, MM-Dist: 2.989). Adding a 7th layer offers no further gain and slightly worsens MM-Dist, suggesting diminishing returns. Overall, 6-layer quantization provides the best balance between expressiveness and generalization.

Ablation Study on Layer-Head Configuration We study the impact of Transformer depth and attention head configurations on HumanML3D (Table 4). Starting from a uniform setup (`[4, 4, 4, 4, 4, 4]` layers, `[6, 6, 6, 6, 6, 6]` heads), the model yields suboptimal results (FID 0.101, MM-Dist 3.359). As we adopt deeper, hierarchically structured layers and progressively narrower heads, performance improves consistently. The best configuration (`[18, 16, 8, 4, 2, 2]` layers, `[16, 12, 6, 2, 2, 2]` heads) achieves the lowest FID (**0.079**), highest R@1 (**0.505**), and lowest MM-Dist (**3.002**).

These findings suggest that allocating more depth in early stages and gradually reducing head width enables better modeling of both global semantics and local temporal details. Compared to uniformly scaled variants, hierarchical designs yield stronger R@1 and MM-Dist, highlighting the importance of architectural balance.

For more ablation details, please check our Appendix.

Table 4: Impact of model depths and head configurations on HumanML3D results (codebook size: 8192×128). Bold indicates the best performance.

Heads	Layers	FID↓	R@1↑	MM-Dist↓
[4, 4, 4, 4, 4, 4]	[6]	$0.242^{\pm 0.009}$	$0.397^{\pm 0.005}$	$4.132^{\pm 0.013}$
[4, 4, 4, 4, 4, 4]	[6, 6, 6, 6, 6, 6]	$0.101^{\pm 0.004}$	$0.461^{\pm 0.003}$	$3.359^{\pm 0.009}$
[12, 6, 4, 2, 2, 2]	[16, 8, 6, 4, 2, 2]	$0.077^{\pm 0.003}$	$0.473^{\pm 0.003}$	$3.260^{\pm 0.010}$
[16, 8, 4, 2, 2, 2]	[18, 10, 6, 4, 2, 2]	$0.058^{\pm 0.003}$	$0.490^{\pm 0.002}$	$3.168^{\pm 0.009}$
[16, 12, 4, 2, 2, 2]	[18, 16, 6, 4, 2, 2]	$0.044^{\pm 0.004}$	$0.508^{\pm 0.003}$	$3.063^{\pm 0.010}$
[16, 12, 6, 2, 2, 2]	[18, 16, 8, 4, 2, 2]	$\mathbf{0.038}^{\pm 0.002}$	$\mathbf{0.515}^{\pm 0.003}$	$\mathbf{2.951}^{\pm 0.008}$

Conclusion

We presented MOGO, a one-pass autoregressive framework for high-quality and real-time 3D human motion generation. By combining the MoSA-VQ module for scale-adaptive residual vector quantization with the RQHC-Transformer for hierarchical causal decoding, MOGO achieves efficient and temporally coherent motion synthesis. The integration of TCA mechanism further enhances semantic alignment and enables better generalization in zero- and few-shot scenarios. Our framework supports low-latency, frame-by-frame inference, making it well-suited for real-time and interactive applications. Extensive experiments demonstrate that MOGO consistently outperforms prior transformer-based methods across multiple benchmarks in terms of generation quality and text-motion alignment, establishing a strong foundation for future research on controllable, editable, and interactive motion generation.

References

Ahuja, C.; and Morency, L.-P. 2019. Language2pose: Natural language grounded pose forecasting. In *2019 International conference on 3D vision (3DV)*, 719–728. IEEE.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cai, H.; Bai, C.; Tai, Y.-W.; and Tang, C.-K. 2018. Deep video generation, prediction and completion of human action sequences. In *Proceedings of the European conference on computer vision (ECCV)*, 366–382.

Chen, R. J.; Chen, C.; Li, Y.; Chen, T. Y.; Trister, A. D.; Krishnan, R. G.; and Mahmood, F. 2022. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16144–16155.

Chen, X.; Jiang, B.; Liu, W.; Huang, Z.; Fu, B.; Chen, T.; and Yu, G. 2023. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 18000–18010.

Ding, M.; Zheng, W.; Hong, W.; and Tang, J. 2022. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *Advances in Neural Information Processing Systems*, 35: 16890–16902.

Ghosh, A.; Cheema, N.; Oguz, C.; Theobalt, C.; and Slusallek, P. 2021. Synthesis of compositional animations from textual descriptions. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1396–1406.

Guo, C.; Mu, Y.; Javed, M. G.; Wang, S.; and Cheng, L. 2024. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1900–1910.

Guo, C.; Zou, S.; Zuo, X.; Wang, S.; Ji, W.; Li, X.; and Cheng, L. 2022a. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5152–5161.

Guo, C.; Zuo, X.; Wang, S.; and Cheng, L. 2022b. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, 580–597. Springer.

Guo, C.; Zuo, X.; Wang, S.; Liu, X.; Zou, S.; Gong, M.; and Cheng, L. 2022c. Action2video: Generating videos of human 3d actions. *International Journal of Computer Vision*, 130(2): 285–315.

Jiang, B.; Chen, X.; Liu, W.; Yu, J.; Yu, G.; and Chen, T. 2023. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36: 20067–20079.

Kim, J.; Kim, J.; and Choi, S. 2023. Flame: Free-form language-based motion synthesis & editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8255–8263.

Kong, H.; Gong, K.; Lian, D.; Mi, M. B.; and Wang, X. 2023. Priority-centric human motion generation in discrete latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14806–14816.

Liao, Y.; Fu, Y.; Cheng, Z.; and Wang, J. 2024. AnimationGPT: An AIGC tool for generating game combat motion assets. https://github.com/fyyakaxyy/AnimationGPT.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Nawrot, P.; Tworkowski, S.; Tyrolski, M.; Kaiser, Ł.; Wu, Y.; Szegedy, C.; and Michalewski, H. 2022. Hierarchical Transformers Are More Efficient Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, 1559–1571.

Pappagari, R.; Żelasko, P.; Villalba, J.; Carmiel, Y.; and Dehak, N. 2019. Hierarchical Transformers for Long Document Classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 838–844. Singapore: IEEE. ArXiv:1910.10781.

Petrovich, M.; Black, M. J.; and Varol, G. 2021. Action-conditioned 3D human motion synthesis with transformer VAE. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10985–10995.

Pinyoanuntapong, E.; Wang, P.; Lee, M.; and Chen, C. 2024. Mmm: Generative masked motion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1546–1555.

Plappert, M.; Mandery, C.; and Asfour, T. 2016. The kit motion-language dataset. *Big data*, 4(4): 236–252.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10684–10695. New Orleans, LA, USA: IEEE. ArXiv:2112.10752.

Tevet, G.; Raab, S.; Gordon, B.; Shafir, Y.; Cohen-Or, D.; and Bermano, A. H. 2022. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*.

Wang, Y.; Leng, Z.; Li, F. W.; Wu, S.-C.; and Liang, X. 2023. Fg-t2m: Fine-grained text-driven human motion generation via diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22035–22044.

Wang, Z.; Yu, P.; Zhao, Y.; Zhang, R.; Zhou, Y.; Yuan, J.; and Chen, C. 2020. Learning diverse stochastic human-action generators by learning smooth latent transitions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 12281–12288.

Zhang, J.; Zhang, Y.; Cun, X.; Zhang, Y.; Zhao, H.; Lu, H.; Shen, X.; and Shan, Y. 2023a. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14730–14740.

Zhang, M.; Cai, Z.; Pan, L.; Hong, F.; Guo, X.; Yang, L.; and Liu, Z. 2024. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE transactions on pattern analysis and machine intelligence*, 46(6): 4115–4128.

Zhang, M.; Guo, X.; Pan, L.; Cai, Z.; Hong, F.; Li, H.; Yang, L.; and Liu, Z. 2023b. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 364–373.

Zhang, Z.; Wang, Y.; Mao, W.; Li, D.; Zhao, R.; Wu, B.; Song, Z.; Zhuang, B.; Reid, I.; and Hartley, R. 2025. Motion anything: Any to motion generation. *arXiv preprint arXiv:2503.06955*.

Zhong, C.; Hu, L.; Zhang, Z.; and Xia, S. 2023. Attt2m: Text-driven human motion generation with multi-perspective attention mechanism. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 509–519.

Zhu, W.; Ma, X.; Ro, D.; Ci, H.; Zhang, J.; Shi, J.; Gao, F.; Tian, Q.; and Wang, Y. 2023. Human motion generation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4): 2430–2449.

Reproducibility Checklist

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) yes

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) yes

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) yes

2.4. Proofs of all novel claims are included (yes/partial/no) yes

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) yes

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) yes

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) yes

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) yes

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) yes

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) yes

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) yes

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) yes

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) yes

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) yes

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) yes

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) yes

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) yes

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond single-

dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) yes

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes