BEYOND MOCAP: SCALING MOTION TOKENIZERS WITH SYNTHETIC HUMAN MOTION FOR GENERATIVE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Human motion generation is fundamentally limited by the diversity of available motion data. Existing motion capture datasets are dominated by common, repetitive actions, leaving rare, compositional, and highly dynamic motions underrepresented. As a result, learned representations capture only a restricted motion vocabulary and generalize poorly beyond common motions. In this work, we propose a framework for expanding the motion representation space with large-scale synthetic human motion. The framework combines a data generation pipeline that produces diverse, physically plausible motion sequences beyond the distribution of existing datasets with a redesigned VQ-VAE tokenizer that adapts to this expanded motion space. Unlike conventional tokenizers trained on narrow data distributions, our approach jointly scales the training distribution and the discrete codebook, enabling the model to capture a significantly richer set of motion primitives. We demonstrate that the resulting motion vocabulary achieves broader coverage and improved compositionality, reducing reconstruction FID by 34.2%, and transfers effectively to existing generation frameworks, including MotionGPT, yielding a 20.2% average FID improvement across existing backbones without architectural changes. These results highlight motion representation as a key bottleneck and show that scaling synthetic data together with representation learning is an effective path toward more generalizable motion generation.

1 INTRODUCTION

Recent advances in human motion generation, particularly those based on discrete latent representations such as VQ-VAE tokenizers van den Oord et al. (2017) and autoregressive transformers (e.g., MotionGPT Zhang et al. (2024b)), have achieved strong performance in tasks such as text-to-motion Guo et al. (2022a); Zhang et al. (2024b) and motion continuation. However, their capabilities remain constrained by the motion tokenizers on which they rely. These tokenizers are typically trained on motion capture (MoCap) datasets such as Human3.6M Ionescu et al. (2013) and AMASS Mahmood et al. (2019), whose distributions are dominated by common actions such as walking, sitting, and simple interactions. Consequently, the learned codebooks cover only a limited portion of human motion, making rare, complex, and highly dynamic movements difficult to represent or generate. Since downstream generators operate within this learned motion vocabulary, increasing model capacity alone cannot recover motions outside its support.

Scaling MoCap data is challenging because collecting high-quality motion is expensive, labor-intensive, and requires specialized equipment and controlled environments. This makes it difficult to capture rare, dangerous, or highly creative motions such as acrobatics and stylized performances, resulting in limited diversity and poor long-tail coverage Guo et al. (2022a). This limitation makes the motion tokenizer a critical bottleneck: as the interface between continuous motion and discrete generative models, it determines the effective vocabulary available for generation. A richer and better-structured token space can improve motion diversity, compositionality, and controllability, similar to the benefits of stronger discrete representations observed in vision and language Esser et al. (2021); Razavi et al. (2019). Thus, advancing motion generation requires scaling not only the generator, but also the motion distribution and its underlying representation (Fig. 1).

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.

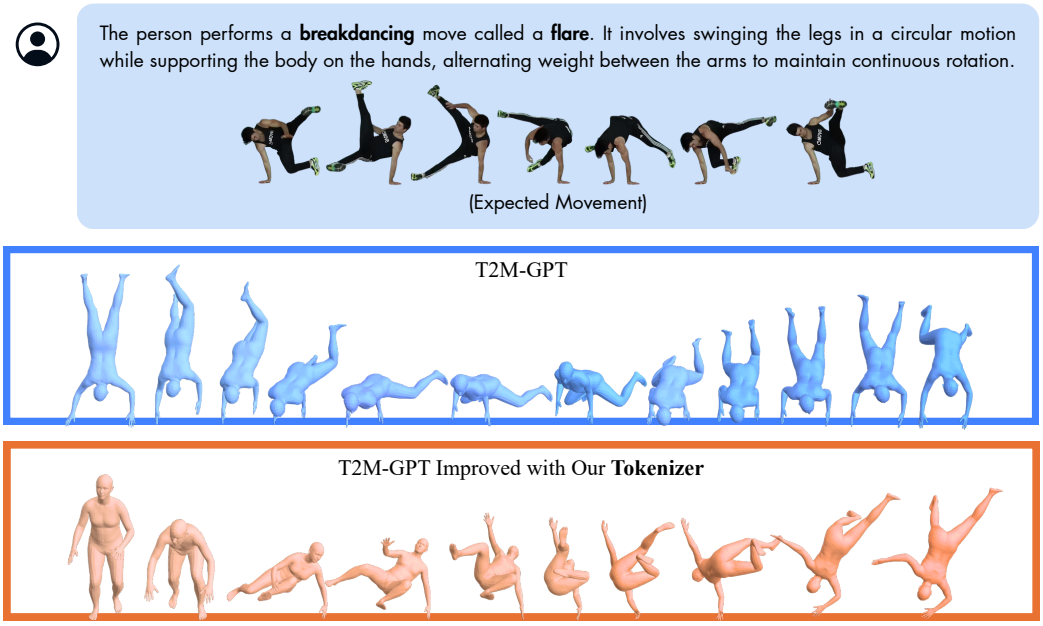


Figure 1: Given a complex textual description, existing text-to-motion models such as T2M-GPT Zhang et al. (2023) struggle to reproduce the intended dynamics due to the limited expressiveness of MoCap-trained tokenizers. Our enhanced VQ-VAE, trained on large-scale synthetic motion, expands the motion vocabulary and enables more faithful motion synthesis, highlighting the importance of scaling the motion representation.

In this work, we address this limitation by jointly expanding the motion data distribution and tokenizer capacity through large-scale synthetic human motion. We introduce a scalable motion generation pipeline that synthesizes diverse and physically plausible sequences beyond the scope of existing MoCap datasets. To ensure physical plausibility, we incorporate constraints derived from kinematics, dynamics, and contact consistency, following common practices in motion synthesis and physics-aware modeling Holden et al. (2016); Rempe et al. (2021). At the same time, we systematically explore the long tail of motion space to generate rare, extreme, and highly compositional sequences that are underrepresented or absent in real-world datasets. Building on this enriched distribution, we redesign the VQ-VAE tokenizer with an expanded and better-utilized codebook, enabling it to represent a substantially broader spectrum of motion primitives.

To transfer the improved representation to downstream generation, we adapt existing autoregressive motion generation frameworks to the new token space without changing their architectures. We re-tokenize the motion side of the training data with our augmented tokenizer and continue fine-tuning pretrained generators on the resulting token sequences. This provides a simple and architecture-agnostic recipe for upgrading existing discrete-token motion generation models, while directly measuring the impact of the expanded motion representation on downstream generation.

Extensive experiments show that jointly scaling the motion distribution and tokenizer substantially improves motion representation and downstream generation. Our tokenizer achieves better reconstruction, higher codebook utilization, and broader coverage of motion space, while generators built on the augmented tokenizer consistently improve across multiple downstream models. Notably, these gains are obtained without architectural changes, demonstrating that expanding the underlying motion representation can directly extend the capabilities of existing motion generation systems.

Contributions. We summarize our contributions as follows:

- We propose a joint scaling paradigm for motion representation learning, showing that expanding the motion distribution and tokenizer capacity together is essential for overcoming the limited support of MoCap-based motion representations.
- We introduce a scalable synthetic motion pipeline and an augmented VQ-VAE tokenizer that jointly expand the coverage and capacity of the learned motion vocabulary, enabling representation of rare, compositional, and highly dynamic motions.

- We demonstrate broad transferability of the learned representation, improving reconstruction and downstream generation across multiple existing motion generation models without modifying their architectures, with particularly consistent gains on unseen and long-tail motions.

2 Related Work

Human Motion Generation Models. Recent advances in human motion generation have improved realism and controllability. However, most methods assume the training distribution adequately covers the target motion space, which fails for rare, complex, or weakly represented actions.

Early language-to-motion works, including Text2Action Ahn et al. (2017), Language2Pose Ahuja & Morency (2019), and Action2Motion Guo et al. (2020), established text-motion mappings but were limited by small datasets and simple models. Modern datasets such as KIT Motion-Language Plappert et al. (2016) and the CMU Motion Capture Database provide richer supervision, yet still suffer from long-tail limitations. Existing approaches include diffusion-based models Tevet et al. (2022a); Shafir et al. (2024); Chen et al. (2023) and discrete-token autoregressive frameworks Zhang et al. (2023); Guo et al. (2024), achieving strong results on benchmarks such as HumanML3D Guo et al. (2022a). Yet, scaling model capacity alone yields diminishing returns when the data distribution is limited Lu et al. (2025), and fails when target motions lie outside its support. To improve semantic alignment, recent works integrate large language models, such as MotionGPT Jiang et al. (2024a) and Motion-Agent Wu et al. (2025). While effective at high-level consistency, they do not address motion representation and remain constrained by available motion primitives. Compositional approaches Athanasiou et al. (2022); Shafir et al. (2024) improve long-horizon structure but are similarly bounded by motion coverage.

In contrast to prior work focusing on architecture or conditioning, we identify the *motion tokenizer* as a key bottleneck. Limited data induces a restricted motion vocabulary, constraining the motions that generative models can represent. This motivates jointly expanding the training distribution and the representation space.

Discrete Motion Representations and Tokenization. Discrete motion representations enable language-model-style generation by mapping continuous motion into token sequences. Most methods adopt VQ-VAE-based tokenizers van den Oord et al. (2017), as used in MotionGPT Jiang et al. (2024a) and T2M-GPT Zhang et al. (2023), typically with small codebooks (e.g., $\sim$512 entries) Zhang et al. (2023); Jiang et al. (2024a); Guo et al. (2024). However, human motion exhibits richer temporal structure and stronger long-tail variability, making such limited codebooks a representational bottleneck. Diverse motions are compressed into a small set of discrete codes, reducing expressiveness and leading to under-utilization or collapse, often mitigated by heuristics such as EMA updates or code resetting Zhang et al. (2023). Recent work identifies token capacity as a key scaling factor. ScaMo Lu et al. (2025) shows performance depends jointly on model, data, and token scale, while MotionCtrl Cao et al. (2025), MotionRL Liu et al. (2024), and MotionChain Jiang et al. (2024b) highlight the role of richer representations in improving controllability and reasoning.

Building on this idea, we look at motion tokenization from the perspective of scaling data. When we expand the motion distribution with synthetic data, standard codebooks are no longer enough. To address this, we increase the size of the codebook, study how well it works through ablation experiments, and introduce a loss-weighting strategy to keep it stable and avoid collapse. This allows the model to better cover the larger motion space and produce improved results.

Data Limitations and Synthetic Motion Generation Data augmentation is well studied in pose-related tasks, especially in static or frame-wise settings. Methods such as PoseAug Gong et al. (2021) and EvoSkeleton Li et al. (2020) expand pose distributions via spatial transformations or structural perturbations. Extending these techniques to motion sequences is non-trivial, as valid motion requires temporal coherence, physical plausibility, and long-range consistency. Sequence-level augmentation for motion generation remains underexplored. MotionAug Maeda & Ukita (2022) focuses on motion prediction rather than generation. Recent data-scaling and self-improving pipelines Gillman et al. (2024); Guo et al. (2025); Cao et al. (2025) emphasize dataset quality and scale, but rely mainly on curation or filtering rather than generating new motion sequences.

In contrast, we introduce a synthetic motion generation pipeline that expands the range of motions the model can learn. It works on a structured representation that keeps motions physically valid while

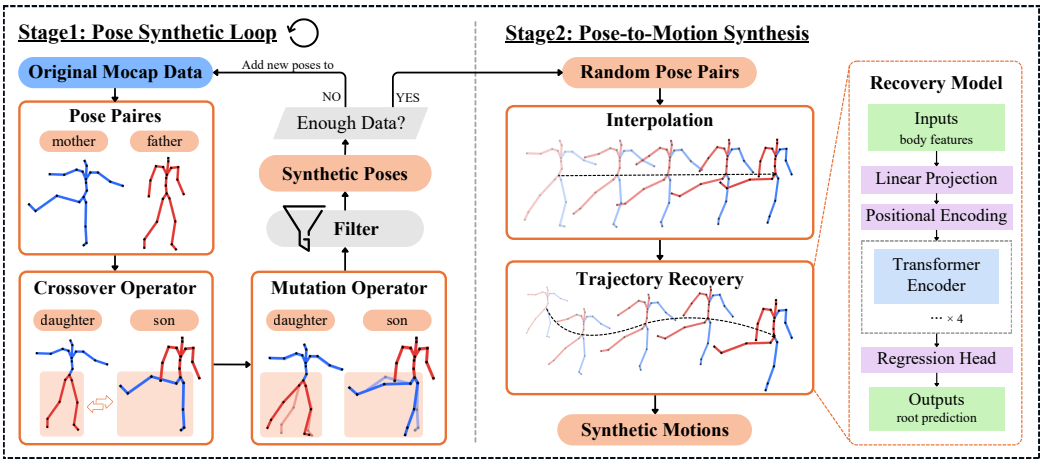


Figure 2: Overview of the motion synthesis pipeline. We generate diverse poses via crossover and mutation, filter them using a physical prior, and construct motion sequences through interpolation and trajectory recovery.

allowing controlled randomness, so it can generate diverse and realistic sequences, including rare and combined motions that do not appear in datasets such as AMASS Mahmood et al. (2019) and KIT Plappert et al. (2016). When combined with a scaled tokenizer, this data increases the effective motion vocabulary for downstream models, allowing them to generate motions beyond the original training distribution and addressing a limitation not handled by prior work.

3 METHODS

Overview. We address the limited motion vocabulary induced by MoCap data by jointly scaling (1) the motion data distribution and (2) the capacity of the motion tokenizer. Our framework consists of two key components: (i) a synthetic motion generation pipeline that expands the coverage of human motion space, and (ii) an augmented VQ-VAE tokenizer that adapts to this expanded distribution. The resulting tokenizer can be directly integrated into existing motion generation models.

Our design is guided by three principles: (1) *validity*: generated motions must remain physically plausible; (2) *diversity*: the pipeline should explore the long-tail of motion space; (3) *compatibility*: synthesized data should align with the representation used by downstream models.

3.1 MOTION SYNTHESIS

Our motion synthesis pipeline consists of two main stages (Fig. 2): (1) pose synthesis via structured stochastic exploration, and (2) pose-to-motion conversion for generating motion sequences.

The pose synthesis stage is inspired by genetic algorithms Holland (1992) and builds upon Li et al. (2020). Generated poses are filtered using a dynamics pose prior Akhter & Black (2015), ensuring physically plausible skeletons. We then construct motion sequences by interpolating between valid poses and recover global trajectories using a lightweight transformer. The pipeline is implemented in the SMPL representation Loper et al. (2023) and converted to HumanML3D format Guo et al. (2022a).

Hierarchical Pose Representation. We represent each pose as a kinematic tree rooted at the pelvis, where each bone is parameterized in a local coordinate frame using spherical coordinates. This representation decouples orientation from bone length, enabling structured manipulation while preserving anatomical consistency.

Structured Exploration via Crossover and Mutation. To expand the pose distribution, we draw inspiration from genetic algorithms and perform stochastic exploration through crossover and mutation operations. Given two poses, crossover exchanges subtrees at randomly selected limb roots (e.g., arms or legs), recombining semantically meaningful body parts while preserving kinematic structure. Mutation further perturbs bone orientations with Gaussian noise while keeping bone lengths fixed.

Importantly, these operations are not arbitrary perturbations: they preserve the compositional structure of human motion while enabling exploration of configurations that are unlikely to appear in MoCap data. This allows the pipeline to systematically populate underrepresented regions of the pose space.

Pose Filtering. To ensure physical plausibility, we enforce anatomical constraints using a pose prior that restricts joint angles to feasible ranges. Invalid poses are discarded. This step ensures that the expanded distribution remains within the manifold of valid human poses, preventing the tokenizer from modeling unrealistic artifacts.

Pose-to-Motion Conversion. We construct motion sequences by interpolating between valid poses using spherical interpolation. While simple, this process plays a critical role: it exposes the model to transitions between poses that are rare or absent in real datasets, thereby expanding the coverage of motion dynamics. This effectively enriches the transition manifold, which is essential for generative models that compose motion tokens over time.

Trajectory Recovery. The interpolated motions lack global root displacement. We recover their global trajectory based on the underlying motion dynamics, using the Transformer-based recovery model shown in Fig. 2.

We represent a motion sequence by decoupling the root information from the rest of the body as

$$\mathbf{m}_{1:T} = \{\mathbf{m}_t\}_{t=1}^{T}, \quad \mathbf{m}_t = [\mathbf{r}_t, \mathbf{b}_t] \in \mathbb{R}^{263}, \tag{1}$$

where $\mathbf{r}_t \in \mathbb{R}^4$ denotes the root features, consisting of frame-wise rotational and planar translational increments, along with the root height, i.e., $\mathbf{r}_t = (\omega_t, v_t^x, v_t^z, h_t)$; $\mathbf{b}_t \in \mathbb{R}^{259}$ represents the body features.

We use the body component $\mathbf{b}_{1:T}$ as input to predict the corresponding root dynamics. The model first projects the input body features into a latent space, followed by positional encoding and a multi-layer Transformer encoder Vaswani et al. (2017) to capture long-range temporal dependencies. The final root features are obtained through a lightweight prediction head. The training objective, and quantitative evaluation are provided in Appendix H.6.

Given the predicted root sequence, the global trajectory is recovered in two components. The root orientation is obtained by integrating the angular velocity, while the global position is computed by accumulating the rotated planar velocities.

3.2 Motion Tokenization

We adopt a vector-quantized variational autoencoder (VQ-VAE) Guo et al. (2022a); Zhang et al. (2023) to discretize motion sequences into a finite set of motion tokens, enabling language-model-style generation.

Given a motion sequence $\mathbf{m}_{1:T} \in \mathbb{R}^{T \times D}$, an encoder E maps it to a latent representation:

$$\mathbf{z}_{1:T/N} = E(\mathbf{m}_{1:T}), \tag{2}$$

where N is the temporal downsampling factor.

Vector Quantization. Each latent vector $\mathbf{z}_t$ is quantized by nearest-neighbor lookup in a codebook $\mathcal{C} = \{\mathbf{c}_k\}_{k=1}^{K}$:

$$\hat{\mathbf{z}}_t = \arg\min_{\mathbf{c}_k \in \mathcal{C}} \|\mathbf{z}_t - \mathbf{c}_k\|_2^2. \tag{3}$$

The quantized sequence $\hat{\mathbf{z}}_{1:T/N}$ is then decoded to reconstruct the motion:

$$\hat{\mathbf{m}}_{1:T} = D(\hat{\mathbf{z}}_{1:T/N}). \tag{4}$$

Training Objective. The tokenizer is trained using a combination of reconstruction and commitment losses:

$$\mathcal{L}_{\text{vq}} = \|\mathbf{m} - \hat{\mathbf{m}}\|_1 + \alpha\|\mathbf{p} - \hat{\mathbf{p}}\|_1 + \beta\|\mathbf{z} - \text{sg}[\hat{\mathbf{z}}]\|_2^2, \tag{5}$$

where $\mathbf{p}$ denotes joint positions, $\text{sg}[\cdot]$ is the stop-gradient operator, and α, β are weighting factors. We adopt exponential moving average (EMA) updates and periodic codebook reset strategies following Zhang et al. (2023) to improve codebook utilization.

Augmented Training with Synthetic Motions. To expand the coverage of motion patterns, we train the tokenizer on a mixture of real and synthesized data:

$$\mathcal{D} = (1 - \lambda)\mathcal{D}_{\text{real}} + \lambda\mathcal{D}_{\text{syn}}, \tag{6}$$

where λ controls the contribution of synthetic data.

This augmentation exposes the tokenizer to rare and compositional motion patterns that are underrepresented in MoCap datasets, improving its ability to represent the long-tail of motion space.

Codebook Scaling. As the diversity of the training distribution increases, a fixed-size codebook becomes insufficient to represent fine-grained motion primitives. We therefore increase the codebook size K to match the expanded distribution. From a quantization perspective, a larger codebook reduces reconstruction error by providing a finer partition of the latent space. However, excessively large K may lead to under-utilization. In practice, we identify an operating regime where codebook utilization remains high while reconstruction quality and downstream performance improve.

By jointly scaling the training data and codebook capacity, the tokenizer transitions from a bottleneck to a flexible interface between motion signals and generative models. This results in a richer and more compositional motion vocabulary, which directly benefits downstream motion generation.

3.3 Adapting Motion Generation Models

Autoregress Motion Generation. Autoregressive motion generation models cast motion synthesis as discrete sequence modeling. Given a motion sequence $\mathbf{x} = (x_1, \ldots, x_T)$ in continuous space, a motion tokenizer first maps it into a sequence of discrete codes $\mathbf{z} = (z_1, \ldots, z_L)$, where each $z_\ell \in \{1, \ldots, K\}$ indexes an entry in a learned codebook of size K. The generator then models the token sequence autoregressively as

$$p(\mathbf{z} \mid \mathbf{c}) = \prod_{\ell=1}^{L} p(z_\ell \mid z_{<\ell}, \mathbf{c}), \tag{7}$$

where $\mathbf{c}$ denotes the conditioning signal, such as a text description. Motion generation is thus reduced to next-token prediction in a discrete vocabulary, analogous to autoregressive language modeling.

Training Procedure. We first re-tokenize the training motions with our augmented tokenizer and use the resulting token sequences as the new supervision targets, while keeping the paired text unchanged. Starting from the pretrained checkpoint, we then continue fine-tuning the original autoregressive generator without changing its architecture. Notably, the amount of training data remains the same, as we only reuse existing text-motion pairs with the new tokenizer for fine-tuning. The newly synthesized motions are not used in this training, since we do not generate new text pairs for them.

4 Experiments

We design our experiments to answer three questions. **(i)** Does our synthetic data expand the motion distribution beyond MoCap, and do the resulting motions remain anatomically valid? **(ii)** Does training on this expanded distribution yield a better tokenizer, both in- and out-of-distribution? **(iii)** Does the improved tokenizer transfer to downstream generation models without changing their architecture? We address each question in Sec. 4.2, Sec. 4.3, and Sec. 4.4 respectively.

4.1 Experiment Setup

HumanML3D. HumanML3D Guo et al. (2022a) is a large-scale 3D human motion dataset, which contains 14,616 motion sequences and paired text annotations. We build our synthetic motion data based on the HumanML3D and generate approximately $64\times$ additional synthetic motions using our synthesis pipeline, resulting in a large-scale augmented dataset. We use HumanML3D as the baseline to analyze the distributional differences introduced by our synthesized data. We train the motion tokenizer on the augmented HumanML3D dataset. For downstream motion generation, we adopt the standard $80\%/5\%/15\%$ train/val/test split and train models on the training set.

Motion-X. Motion-X++ Zhang et al. (2025) is a large-scale multimodal 3D whole-body human motion dataset, which contains 120.5K motion sequences spanning diverse scenarios such as music,

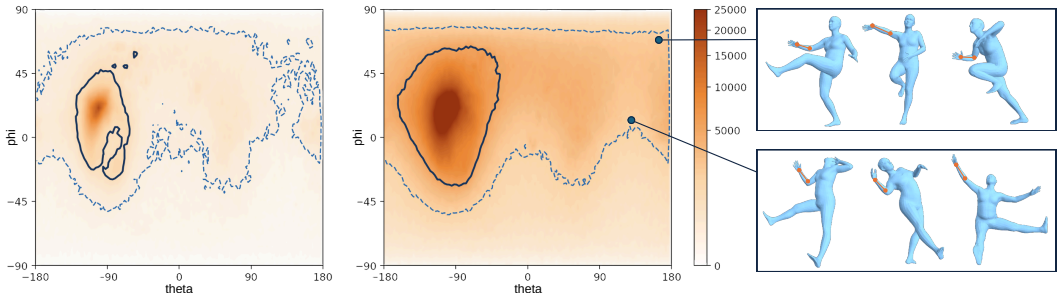


Figure 3: **Bone Orientation Distribution (Elbow to Wrist).** We visualize the angular distribution of the right forearm using spherical coordinates, where the horizontal axis represents θ and the vertical axis represents ϕ. **Left:** HumanML3D. **Right:** Our synthesized data. The solid contours encloses the core 50% of the data distribution, while the dashed contour encloses 90% of the samples, representing the majority of the dataset.

kungfu, and performance. Compared to HumanML3D, Motion-X++ exhibits significantly higher diversity in motion patterns and semantic annotations, making it a challenging benchmark for evaluating generalization. We use Motion-X++ solely for evaluation without further training to assess the cross-dataset generalization capability of our tokenizer and downstream generation models.

Evaluation metrics. For tokenizer evaluation, we report *Mean Per Joint Position Error (MPJPE)* for reconstruction accuracy. For motion generation evaluation, following Guo et al. (2022a), we use the pre-trained motion/text feature extractor of Guo et al. (2022a) and report five standard metrics: *R-Precision* (Top-1, Top-2, Top-3) for text-motion retrieval consistency; *Fréchet Inception Distance (FID)* for distributional fidelity; *Multimodal Distance (MM-Dist)* for text-motion feature alignment; and *Diversity* for generation variety. For each metric, we repeat the evaluation 20 times and report the average with 95% confidence interval.

4.2 Synthetic Data Evaluation

We assess our synthetic data along two axes: **coverage** of the motion distribution and **validity** of the resulting poses. These two measurements establish the precondition for the rest of our experiments

Coverage. We examine the angular distribution of every bone vector in the SMPL skeleton under each bone's local coordinate frame. Aggregated over all 21 bones, every single bone shows a non-trivial expansion, and none of them shrinks. Fig. 3 visualizes one representative bone, showing that the synthetic distribution covers a substantially larger region of orientations and populates areas that are sparsely visited or entirely absent in HumanML3D. Detailed coverage statistics are provided in Appendix D, Tab. 5

Validity. Coverage is only useful if the new poses remain anatomically valid. We filter synthesized poses using the dynamics pose prior Akhter & Black (2015) and retain those whose validity score is no lower than that of their parent pose. Under this criterion, 67.33% of generated poses are kept for training. We sample poses on the right of Fig. 3 from the newly explored regions, which indicate that the expanded coverage corresponds to plausible, though relatively rare, human configurations rather than anatomically impossible ones.

Beyond pose-level filtering, we evaluate temporal quality using physics-related metrics, as the synthetic motions are used for tokenizer training rather than text–motion alignment. Results and evaluation details are provided in Appendix E.

4.3 Evalution of Motion Tokenizer

In-distribution. On HumanML3D, reconstruction FID drops from 0.07 to $\mathbf{0.046}$ ($\mathbf{-34.2}\%$) while every retrieval metric improves (Tab. 1 (a)).

Out-of-distribution. We evaluate the tokenizer on the unseen dataset Motion-X++ and its task-specific subsets, which lie farther from the support of training data. As shown in Tab. 1 (b), MPJPE drops on all tasks, suggesting that the expanded data improves the tokenizer's generalization and

Metric	T2M-GPT	**Ours**
FID ↓	0.070	**0.046** (−34.2%)
Top-1 ↑	0.501	**0.504** (+0.01%)
MM-Dist ↓	3.072	**3.043** (−0.01%)
Diversity ↑	9.593	**9.728** (+0.02%)

(a) In-distribution on HumanML3D

Subset	# samples	T2M-GPT ↓	**Ours** ↓
Overall	25,865	272.73	**252.37** (−7.5%)
Daily	12,042	244.70	**231.98** (−5.2%)
Sports	6,944	322.65	**282.74** (−12.4%)
Dance	3,394	180.77	**161.80** (−10.5%)

(b) Out-of- distribution on Motion-X++ subsets

Table 1: Our tokenizer enhances T2M-GPT both in-distribution on HumanML3D, and out-of-distribution on Motion-X++ subsets unseen during training. The complete per-subset breakdown in (b) is reported in Appendix J.

representation capacity by extending coverage beyond the original HumanML3D support, rather than merely densifying the existing distribution.

Joint Scaling Ablation. We independently vary synthetic motion augmentation and codebook size to disentangle their contributions (Table 2). These results support the complementary benefits of expanding the motion training distribution and tokenizer capacity in the evaluated setting.

Setting	Size	Synthetic	FID ↓
Baseline	512	No	0.070
Codebook scaling	2048	No	0.146
Synthetic data scaling	512	Yes	0.141
Full method (Ours)	2048	Yes	**0.046**

Table 2: Joint scaling ablation.

4.4 Evaluation of Motion Generation

Method	FID ↓	R-Precision ↑			MM-Dist ↓	Diversity	MModality ↑
		Top-1	Top-2	Top-3			
MDM Tevet et al. (2022b)	$0.544^{\pm.044}$	$0.320^{\pm.005}$	$0.498^{\pm.004}$	$0.611^{\pm.007}$	$5.566^{\pm.027}$	$9.559^{\pm.086}$	$2.799^{\pm.072}$
MLD Chen et al. (2023)	$0.473^{\pm.013}$	$0.481^{\pm.003}$	$0.673^{\pm.003}$	$0.772^{\pm.002}$	$3.196^{\pm.010}$	$9.724^{\pm.082}$	$2.413^{\pm.079}$
MotionDiffuse Zhang et al. (2024a)	$0.630^{\pm.001}$	$0.491^{\pm.001}$	$0.681^{\pm.001}$	$0.782^{\pm.001}$	$3.113^{\pm.001}$	$9.410^{\pm.049}$	$1.553^{\pm.042}$
T2M Guo et al. (2022a)	$1.067^{\pm.002}$	$0.457^{\pm.002}$	$0.559^{\pm.007}$	$0.740^{\pm.003}$	$3.340^{\pm.008}$	$9.188^{\pm.002}$	$2.090^{\pm.083}$
TM2T Guo et al. (2022b)	$1.501^{\pm.017}$	$0.424^{\pm.003}$	$0.618^{\pm.003}$	$0.729^{\pm.002}$	$3.467^{\pm.011}$	$8.589^{\pm.076}$	$2.424^{\pm.093}$
MoMask Guo et al. (2024)	$0.045^{\pm.002}$	$0.521^{\pm.002}$	$0.713^{\pm.002}$	$0.807^{\pm.002}$	$2.958^{\pm.008}$	$9.620^{\pm.064}$	$1.241^{\pm.040}$
MotionChain Jiang et al. (2024b)	$0.248^{\pm.009}$	$0.504^{\pm.003}$	$0.617^{\pm.002}$	$0.790^{\pm.003}$	$3.033^{\pm.010}$	$9.470^{\pm.075}$	$1.727^{\pm.014}$
MotionGPT-2 Wang et al. (2024)	$0.191^{\pm.004}$	$0.496^{\pm.002}$	$0.691^{\pm.003}$	$0.782^{\pm.004}$	$3.080^{\pm.013}$	$9.860^{\pm.026}$	$2.137^{\pm.022}$
Motion-R1 Ouyang et al. (2026)	$0.201^{\pm.004}$	$0.515^{\pm.003}$	$0.719^{\pm.002}$	$0.818^{\pm.002}$	$2.854^{\pm.010}$	$10.206^{\pm.075}$	$2.137^{\pm.105}$
T2M-GPT Zhang et al. (2023)	$0.116^{\pm.004}$	$\mathbf{0.491^{\pm.003}}$	$\mathbf{0.680^{\pm.003}}$	$\mathbf{0.775^{\pm.002}}$	$\mathbf{3.118^{\pm.011}}$	$9.761^{\pm.081}$	$1.856^{\pm.011}$
T2M-GPT (Ours)	$\mathbf{0.097^{\pm.005}}$	$0.464^{\pm.003}$	$0.647^{\pm.002}$	$0.747^{\pm.002}$	$3.271^{\pm.009}$	$\mathbf{9.594^{\pm.081}}$	$\mathbf{2.266^{\pm.095}}$
MotionGPT Jiang et al. (2024a)	$0.232^{\pm.008}$	$\mathbf{0.492^{\pm.003}}$	$\mathbf{0.681^{\pm.003}}$	$\mathbf{0.778^{\pm.002}}$	$\mathbf{3.096^{\pm.008}}$	$\mathbf{9.528^{\pm.071}}$	$2.008^{\pm.084}$
MotionGPT (Ours)	$\mathbf{0.176^{\pm.007}}$	$0.432^{\pm.003}$	$0.611^{\pm.003}$	$0.710^{\pm.003}$	$3.578^{\pm.011}$	$9.578^{\pm.082}$	$\mathbf{4.845^{\pm.163}}$
MotionAgent Wu et al. (2025)	$0.230^{\pm.009}$	$\mathbf{0.515^{\pm.004}}$	$\mathbf{0.691^{\pm.003}}$	$\mathbf{0.801^{\pm.004}}$	$\mathbf{2.967^{\pm.020}}$	$9.908^{\pm.102}$	$2.142^{\pm.014}$
MotionAgent (Ours)	$\mathbf{0.184^{\pm.009}}$	$0.453^{\pm.003}$	$0.637^{\pm.002}$	$0.738^{\pm.002}$	$3.318^{\pm.009}$	$\mathbf{9.665^{\pm.076}}$	$\mathbf{2.904^{\pm.230}}$

Table 3: **Comparison on HumanML3D.** The green rows indicate results with our synthetic data and augmented tokenizer; bold values indicate the better result within each baseline/ours pair.

Quantitative Results. Tab. 3 shows a consistent pattern: our synthetic-data training and augmented tokenizer mainly improve motion fidelity. After replacing the original tokenizer while keeping the generator architecture unchanged, FID drops from 0.116 to 0.097 for T2M-GPT (**16.4%**), from 0.232 to 0.176 for MotionGPT (**24.1%**), and from 0.230 to 0.184 for MotionAgent (**20.0%**). This improvement is consistent across all three backbones, suggesting that the gain comes from better motion representation rather than model-specific tuning. We observe a decline in text-motion alignment. Our diagnostic analyzes suggest that this is partly due to weaker text controllability but more substantially to distribution bias in the HumanML3D evaluator. Detailed analyses are provided in Appendix I. Overall, our method enhances motion realism and diversity while keeping other metrics comparable, supporting tokenizer scaling for better generation.

To further assess generalization beyond the training distribution, we evaluate all three backbones on 100 OOD and long-tail prompts covering rare and challenging motions. The full quantitative results and prompt set are provided in Appendix C.

Qualitative Results. We further present qualitative visualization results in Fig. 4. The example prompts include uncommon motions such as sports, yoga, and kung-fu. These are out-of-distribution prompts that do not exist in the training data. In each example, the result improved with our method

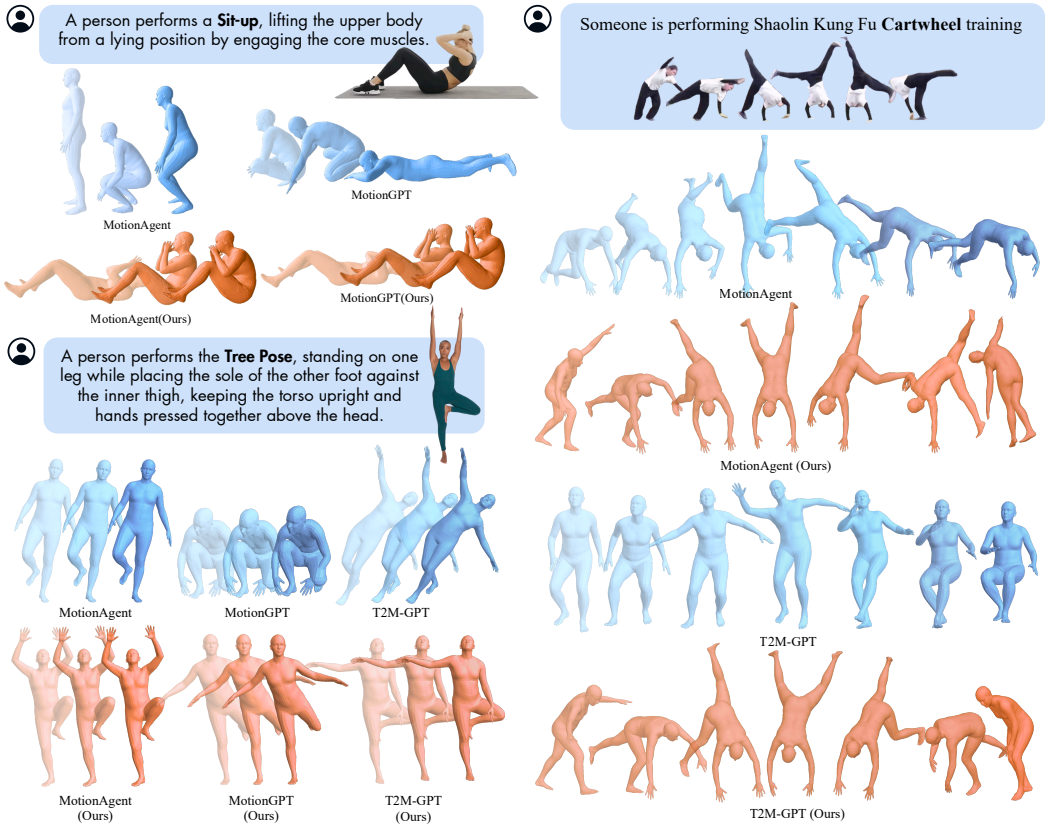


Figure 4: Qualitative motion generation results on challenging prompts, including rare actions such as sports, yoga, and kung-fu. For each example, the original model output is shown in blue, and the result from our method is shown in orange below.

is shown in orange, below the original model result shown in blue. Our method produces visibly better motions, supporting our main claim that the proposed motion synthesis pipeline broadens motion-space coverage and improves the generalization of downstream generation models. More qualitative examples can be found in the Appendix B

To further assess generalization beyond the training distribution, we evaluate all three backbones on 100 OOD and long-tail prompts covering rare and challenging motions. The full quantitative results and prompt set are provided in Appendix C.

5 Conclusion

In this work, we present a motion data synthesis pipeline, Beyond MoCap, that both expands motion-space coverage and scales the codebook accordingly. This simple change improves tokenizer reconstruction quality on both in-distribution and out-of-distribution generalization tasks, and transfers consistently to downstream motion generation models. These results demonstrate that better motion generation is not solely a problem of scaling the generator, but also requires scaling the motion distribution and the tokenizer together.

As with any other work, our method has several limitations. Our synthetic pipeline currently focuses only on body motion, without modeling finer-grained hand or facial dynamics. Furthermore, we focus on expanding motion diversity without pairing the synthesized motions with corresponding text annotations. While current expansion significantly improves the realism and diversity metrics of motion generation, it is foreseeable that incorporating text labels would further enhance the semantic alignment capability of the model, leading to more comprehensive improvements across all aspects of generation performance.

AI USE STATEMENT

In this work, we used generative AI tools for drafting and editing portions of the manuscript, improving writing clarity and readability, and help with implementation debugging. We have not used generative AI tools for proving mathematical claims, and generative AI-assisted idea collections are not applicable to this work. All AI-assisted content was reviewed and verified by the authors. The authors are responsible for the final content, claims, and results presented in this paper.

ETHICS STATEMENT

This work does not involve human subjects, user studies, or the collection of new motion-capture data from individuals. The synthetic motion data used in this work are generated computationally and are not intended to represent or identify specific individuals. We use existing motion datasets and synthetic human motion solely for research on motion representation learning and generative modeling. We acknowledge that generative models of human motion can potentially be used in applications beyond the scope of this work, and we encourage responsible use of the released models and data.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the synthetic motion generation pipeline, motion representation, VQ-VAE architecture, training procedure, and evaluation protocols in the paper and supplementary material. We also report the datasets, model configurations, and experimental settings used to train and evaluate the proposed tokenizer and its transfer to existing motion generation frameworks. We will release the source code, trained models, and relevant configuration files upon publication to facilitate reproduction of our results.

REFERENCES

Hyemin Ahn, Timothy Ha, Yunho Choi, Hwiyeon Yoo, and Songhwai Oh. Text2action: Generative adversarial synthesis from language to action. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *International Conference on 3D Vision (3DV)*, 2019.

Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol. TEACH: Temporal Action Compositions for 3D Humans. In *International Conference on 3D Vision (3DV)*, 2022.

Bin Cao, Sipeng Zheng, Ye Wang, Lujie Xia, Qianshan Wei, Qin Jin, Jing Liu, and Zongqing Lu. Motionctrl: A real-time controllable vision-language-motion model. In *IEEE International Conference on Computer Vision (ICCV)*, 2025.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Nate Gillman, Michael Freeman, Daksh Aggarwal, Chia-Hong Hsu, Calvin Luo, Yonglong Tian, and Chen Sun. Self-correcting self-consuming loops for generative model training. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

Kehong Gong, Jianfeng Zhang, and Jiashi Feng. Poseaug: A differentiable pose augmentation framework for 3d human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *ACM International Conference on Multimedia (MM)*, 2020.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 (ECCV)*, 2022b.

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

Chuan Guo, Inwoo Hwang, Jian Wang, and Bing Zhou. Snapmogen: Human motion generation from expressive texts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. In *SIGGRAPH*, 2016.

John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE TPAMI*, 2013.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

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 (ECCV)*, 2024b.

Shichao Li, Lei Ke, Kevin Pratama, Yu-Wing Tai, Chi-Keung Tang, and Kwang-Ting Cheng. Cascaded deep monocular 3d human pose estimation with evolutionary training data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Xiaoyang Liu, Yunyao Mao, Wengang Zhou, and Houqiang Li. Motionrl: Align text-to-motion generation to human preferences with multi-reward reinforcement learning. *arXiv preprint arXiv:2410.06513*, 2024.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023.

Shunlin Lu, Jingbo Wang, Zeyu Lu, Ling-Hao Chen, Wenxun Dai, Junting Dong, Zhiyang Dou, Bo Dai, and Ruimao Zhang. Scamo: Exploring the scaling law in autoregressive motion generation model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Takahiro Maeda and Norimichi Ukita. Motionaug: Augmentation with physical correction for human motion prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

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. *International Conference on Learning Representations (ICLR)*, 2026.

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

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

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

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

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022b.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Yuan Wang, Di Huang, Yaqi Zhang, Wanli Ouyang, Jile Jiao, Xuetao Feng, Yan Zhou, Pengfei Wan, Shixiang Tang, and Dan Xu. Motiongpt-2: A general-purpose motion-language model for motion generation and understanding. *arXiv preprint arXiv:2410.21747*, 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 (ICLR)*, 2025.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE TPAMI*, 2024a.

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

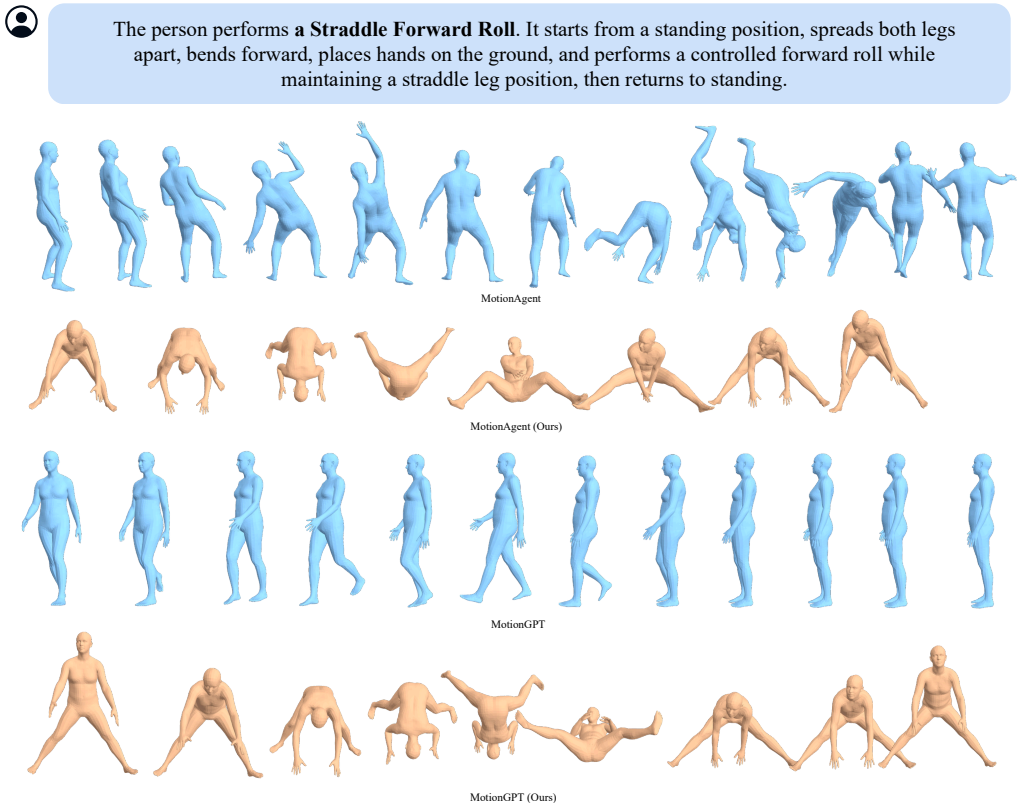
Yuhong Zhang, Jing Lin, Ailing Zeng, Guanlin Wu, Shunlin Lu, Yurong Fu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x++: A large-scale multimodal 3d whole-body human motion dataset. *arXiv preprint arXiv:2501.05098*, 2025.

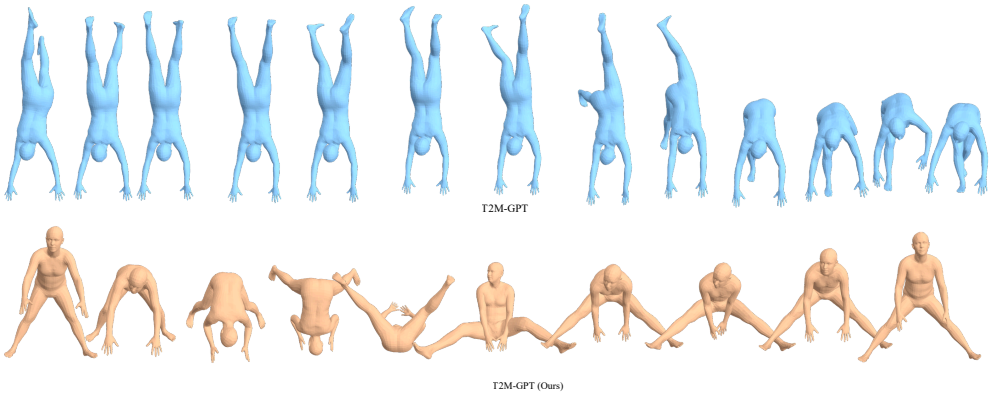
A Overview

In this appendix, we present:

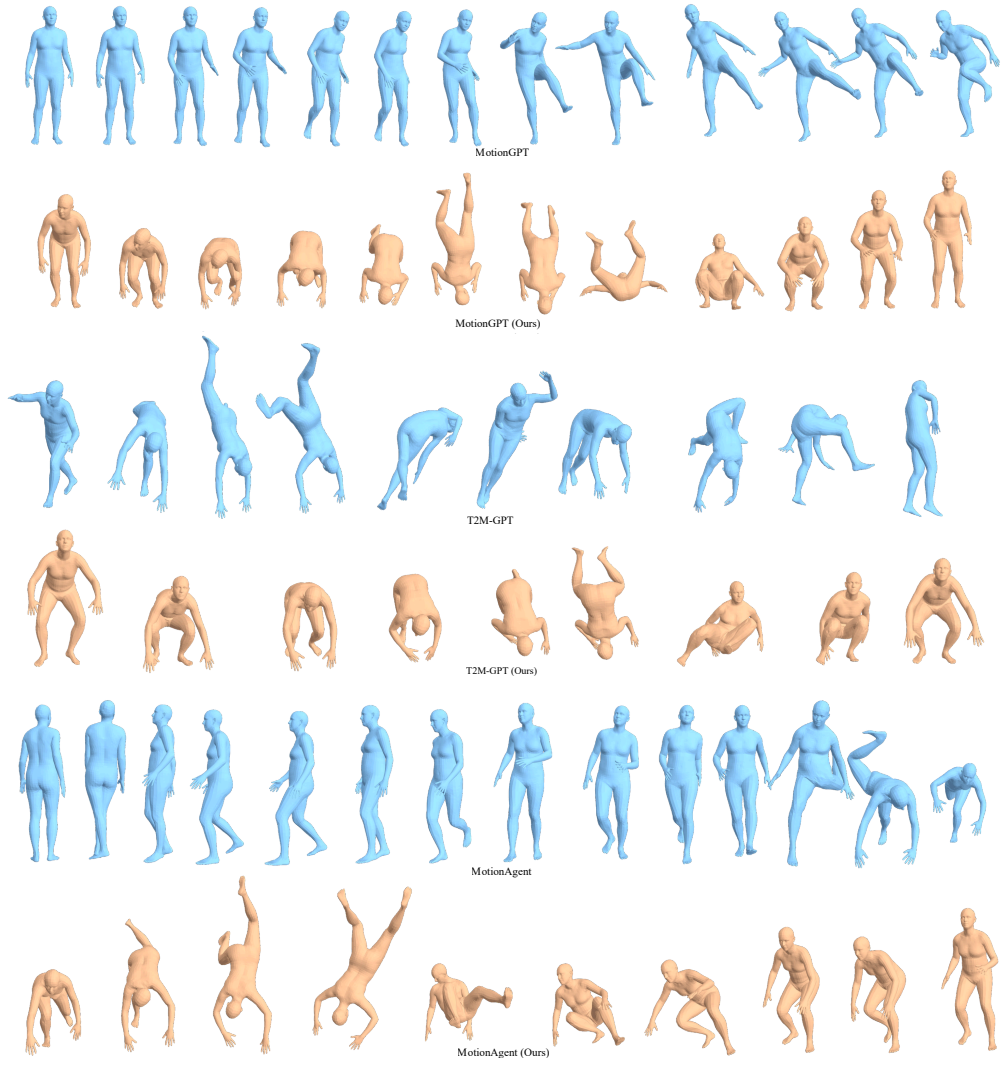
- Section B: more qualitative results.
- Section C:Experiment on 100 out-of-distribution long-tail motion.
- Section D: per-bone coverage statistics that quantify how the synthetic data expands beyond the original HumanML3D support.
- Section E: temporal and physics-related evaluation of synthetic motions.
- Section F: ablation study of the codebook size.
- Section G: ablation study of the mixing ratio between original and synthetic data.
- Section H: implementation details of the data generation, trajectory model training, tokenizer training and finetuning.
- Section **??**: additional details on the evaluation metrics and the motion representation.
- Section J: full per-subset out-of-distribution reconstruction results on Motion-X++.

B More OOD and Qualitative Results

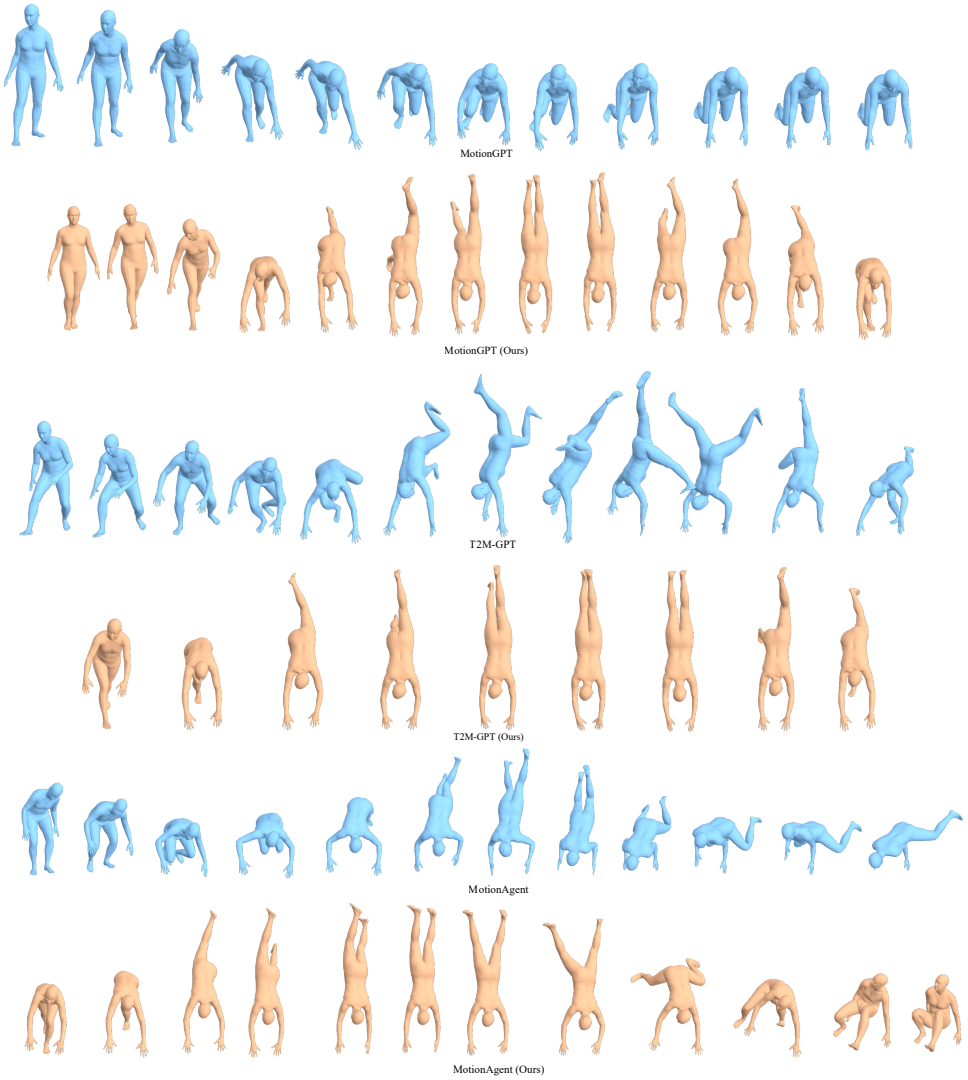




A person performs a **Front Flip**, jumping forward and tucking the body to rotate in the air before landing on both feet.



A gymnast performs a **Press Handstand**. Beginning in a standing pike fold with hands flat on the mat near the feet, they slowly transfer body weight onto the hands and lift both legs together off the floor with control, pressing through straight arms into a fully extended vertical handstand.



C OOD LONG-TAIL MOTION QUANTITATIVE RESULTS

C.1 QUANTITATIVE RESULTS

To further evaluate the generalization of the learned motion representation beyond the HumanML3D training distribution, we construct a set of 100 OOD and long-tail text prompts covering rare, challenging, and sparsely represented human motions. We evaluate multiple generation backbones using their original tokenizer and our augmented tokenizer while keeping the generator architecture unchanged.

As shown in Table 4, our tokenizer consistently improves R-Precision across all three backbones, reduces MM-Dist, and substantially increases motion multimodality, while Top-3 remains largely comparable. These results indicate that the expanded motion representation generally improves text-motion alignment and generation diversity on OOD and long-tail prompts, with consistent gains across different generation backbones.

Method	Top-1 ↑	Top-2 ↑	Top-3 ↑	MM-Dist ↓	MModality ↑
T2M-GPT	0.413	0.667	**0.827**	3.724	1.292
T2M-GPT (Ours)	**0.502**	**0.708**	0.815	**3.551**	**1.983**
MotionGPT	0.398	0.641	0.798	3.892	1.436
MotionGPT (Ours)	**0.471**	**0.689**	**0.799**	**3.684**	**2.217**
MotionAgent	0.431	0.658	**0.811**	3.756	1.508
MotionAgent (Ours)	**0.486**	**0.697**	0.804	**3.602**	**2.126**

Table 4: OOD and long-tail motion generation on 100 prompts. Bold values indicate the better result within each baseline/ours pair.

C.2 100 PROMPTS

1. A person performs a Sit-Up, lifting the upper body from a lying position by engaging the core muscles.
2. The person performs a Breakdancing move called a Flare. It involves swinging the legs in a circular motion while supporting the body on the hands, alternating weight between the arms to maintain continuous rotation.
3. Someone is performing Shaolin Kung Fu Cartwheel training
4. A person performs the Tree Pose, standing on one leg while placing the sole of the other foot against the inner thigh, keeping the torso upright and hands pressed together above the head.
5. The person performs a Straddle Forward Roll. It starts from a standing position, spreads both legs apart, bends forward, places hands on the ground, and performs a controlled forward roll while maintaining a straddle leg position, then returns to standing.
6. A person performs a Front Flip, jumping forward and tucking the body to rotate in the air before landing on both feet.
7. A gymnast performs a Press Handstand. Beginning in a standing pike fold with hands flat on the mat near the feet, they slowly transfer body weight onto the hands and lift both legs together off the floor with control, pressing through straight arms into a fully extended vertical handstand.
8. A person Picks Up an object, bending forward at the hips and reaching down with one hand before lifting it and standing upright.
9. A dancer performs a Salsa Step, shifting weight quickly between the feet while coordinating hip rotation and rhythmic arm movement.
10. A dancer performs a Tango Step, stepping forward with precise foot placement and sharp, controlled upper body posture.
11. A person performs Child's Pose, kneeling and sitting the hips toward the heels while folding the torso forward and stretching both arms along the floor.
12. A person performs Triangle Pose, standing with the feet wide apart, hinging sideways over one straight leg, and reaching one hand downward while extending the other arm upward.
13. A gymnast performs a Backward Roll, sitting back from a squat, rolling over the rounded back, and pressing through the hands beside the shoulders to return to the feet.
14. A person performs Dancer Pose, balancing on one leg, holding the opposite foot behind the body, and lifting it backward while reaching the free arm forward.
15. A person performs Skater Hops, jumping sideways from one foot to the other while sweeping the free leg behind the supporting leg and swinging the arms across the body.
16. A person performs Mountain Climbers, holding a high plank and alternately driving each knee toward the chest while keeping both hands planted.
17. An athlete performs a Shot Put, holding the shot beside the neck, driving upward from bent legs, and rotating the torso while extending one arm to push the shot forward.
18. A person performs Bridge Pose, lying on the back with knees bent and feet planted, then lifting the pelvis while keeping the shoulders and arms on the floor.

19. A martial artist performs a Horse-Stance Punch, lowering into a wide squat and driving one fist straight forward while drawing the opposite fist back to the waist.
20. A person performs an Inchworm, bending forward from standing to place the hands on the floor, walking the hands into a plank, and then stepping the feet toward the hands.
21. A player performs a Tennis Backhand, turning the shoulders sideways, bringing both hands across the body, and swinging the racket forward while shifting weight onto the front foot.
22. A dancer performs a Rumba Side Step, slowly shifting weight from one leg to the other and allowing the hips to settle over each supporting foot while extending the arms outward.
23. A person performs a Single-Leg Hop and Hold, hopping forward from one foot, landing on the same foot with the knee bent, and extending both arms sideways to hold a steady balance.
24. A person performs Warrior I, stepping one foot backward into a long stance, bending the front knee, and raising both arms overhead while keeping the chest facing forward.
25. A martial artist performs a Side Kick, turning the body sideways, drawing one knee inward, and driving the heel outward before retracting the leg into a balanced stance.
26. A gymnast performs a Round-Off, taking a short approach, placing the hands on the floor in succession, and bringing the legs together overhead before landing on both feet facing backward.
27. A gymnast performs a Front Walkover, stepping into a split-leg handstand and lowering one foot beyond the hands, then transferring weight forward to rise onto both feet.
28. A dancer performs a Flamenco Phrase, stamping the feet in a quick rhythmic pattern while lifting the chest and circling the arms from waist level to overhead.
29. A dancer performs a Contemporary Floor Spiral, lowering from standing onto one knee, rotating the torso toward the floor, and sweeping one leg around before rising again.
30. A dancer performs a Cha-Cha Sequence, alternating a forward and backward weight shift with three quick side steps while keeping the hips moving rhythmically.
31. A person performs a Golf Swing, rotating the torso into a backswing, then shifting weight forward and sweeping both arms through a broad arc into a high follow-through.
32. A person performs Downward-Facing Dog, pressing the hands and feet into the floor and lifting the hips upward to form an inverted V with the body.
33. A player performs a Soccer Instep Kick, planting one foot beside the ball and swinging the opposite leg forward while rotating the hips and spreading the arms for balance.
34. A dancer performs a Locking Sequence, swinging the arms freely before stopping them abruptly at bent-elbow angles, then adding a knee lift and a small rebound.
35. A martial artist performs a Crane-Stance Balance, standing on one leg with the other knee raised high while opening both arms outward and holding the torso upright.
36. A gymnast performs a Split Leap, taking a short run and jumping from one foot while extending one leg forward and the other backward, then landing on the leading foot.
37. A person performs a Squat-to-Calf-Raise Sequence, lowering into a squat, standing up, and continuing onto the balls of both feet while reaching the arms overhead.
38. A martial artist performs a Low Sweeping Kick, dropping into a crouch and rotating on one bent leg while the opposite leg sweeps close to the ground in a wide circle.
39. A gymnast performs an L-Sit, placing both hands on the floor beside the hips and lifting the pelvis while holding both legs straight forward above the ground.
40. A person performs Warrior III, shifting onto one leg and hinging forward at the hips while extending the other leg backward and reaching both arms forward.
41. A dancer performs a Hip-Hop Body Wave, passing a smooth ripple from the head through the chest and abdomen to the hips, then returning to an upright stance.
42. A gymnast performs a Back Handspring, bending the knees and swinging the arms before jumping backward onto the hands, then pushing through the shoulders to land on both feet.

43. A person performs Chair Pose, bending both knees and moving the hips backward as if sitting, while lifting both arms overhead and keeping the feet close together.
44. An athlete performs a Hurdle Clearance, driving one leg forward over the hurdle while folding the trailing leg sideways, then landing on the lead foot and continuing to run.
45. A martial artist performs an Outside Crescent Kick, swinging a straight leg upward across the front of the body and outward in a broad arc while extending the arms for balance.
46. A person performs Tai Chi Cloud Hands, stepping slowly sideways while shifting weight between the legs and circling the hands alternately across the front of the torso.
47. A goalkeeper performs a Low Diving Save, pushing sideways off one foot and extending both arms toward a low ball before landing on the side of the body.
48. A person performs a Pistol Squat, balancing on one foot and extending the other leg forward while lowering into a deep single-leg squat, then rising without placing the raised foot down.
49. A person performs a Tai Chi Brush-Knee Movement, stepping into a forward stance while one hand sweeps past the front knee and the other palm extends forward at chest height.
50. A boxer performs a Jab-Cross-Duck Combination, throwing two alternating straight punches, bending the knees to lower the head beneath an imagined strike, and returning to guard.
51. A martial artist performs a Roundhouse Kick, pivoting on the supporting foot and swinging the raised leg horizontally across the body before returning to a guarded stance.
52. A martial artist performs a Jumping Front Kick, driving one knee upward, pushing off the supporting foot, and extending a leg forward in midair before landing in a guarded stance.
53. A person performs a Bear Crawl, moving forward on both hands and the balls of the feet with the knees hovering above the floor, alternating opposite hand and foot steps.
54. An athlete performs a Sprint Start, pushing out of a crouched position and driving the arms alternately while taking short, powerful accelerating steps.
55. A player performs a Basketball Layup, taking two quick steps toward the basket, driving one knee upward, and jumping off the opposite foot while reaching one arm toward the hoop.
56. A gymnast performs a Tucked Forward Roll, squatting down, placing both hands on the floor, and rolling forward over the rounded back before returning to a squat.
57. A dancer performs a Samba Bounce Step, alternating small forward and backward steps while repeatedly bending and straightening the knees to create a rhythmic rise and fall.
58. A dancer performs a Running Man Step, repeatedly lifting one knee while sliding the opposite foot backward, switching legs in a continuous bouncing rhythm.
59. A person performs Boat Pose, balancing on the sitting bones, lifting both feet from the floor with the knees bent, and reaching the arms forward beside the shins.
60. A dancer performs a Waltz Box Step, moving forward, sideways, and together, then backward, sideways, and together while maintaining an upright torso and rounded arms.
61. A gymnast performs a Tuck Jump, springing upward from both feet and drawing both knees toward the chest before extending the legs to land in the same place.
62. A martial artist performs a Bow-Stance Punch, stepping into a deep forward lunge and rotating the hips as the rear fist extends straight ahead.
63. A gymnast performs a Back Walkover, bending backward from standing onto both hands and passing the legs overhead one at a time before returning to standing.
64. A gymnast performs a Handstand Walk, kicking up onto both hands and taking several short alternating hand steps while holding the legs extended overhead.
65. A person performs Seated Forward Fold, sitting with both legs extended together and hinging forward from the hips while reaching both hands toward the feet.
66. A person performs a Burpee, lowering from standing into a squat, placing both hands down, jumping the feet backward into a plank, and returning to a vertical jump.
67. A person performs Alternating Reverse Lunges, stepping one foot backward and lowering both knees, then pushing through the front foot to stand and switching sides.

68. A person performs Cobra Pose, lying face down with the palms beside the chest and slowly lifting the chest while keeping the pelvis and legs on the floor.
69. A person performs a Side-Plank Rotation, balancing on one hand and the outer edge of one foot, reaching the free arm upward, and then threading it beneath the torso.
70. A dancer performs a Popping Sequence, sharply contracting and releasing the arms and chest in short pulses while stepping sideways in time with the rhythm.
71. A martial artist performs a High Block and Counterpunch, raising one forearm above the forehead, shifting weight forward, and driving the opposite fist straight ahead.
72. A dancer performs a Breakdance Six-Step, supporting the body with the hands while stepping the feet around in a low circular pattern and returning to the starting position.
73. A player performs a Badminton Overhead Smash, stepping backward beneath the shuttle, raising the racket arm, and rotating the torso into a sharp downward swing.
74. A martial artist performs a Palm-Strike Combination, stepping forward twice while alternately extending an open palm and retracting the other hand beside the ribs.
75. A martial artist performs a Front Snap Kick, lifting one knee toward the chest, quickly extending the lower leg forward, and retracting it before placing the foot down.
76. A martial artist performs a Spinning Back Kick, turning the head and shoulders backward, pivoting on one foot, and extending the opposite leg behind the body before regaining the guard.
77. A dancer performs a Moonwalk, alternating pressure between the toes and heels so that one foot slides backward while the upper body remains upright and appears to move forward.
78. A person performs Extended Side Angle, bending one knee into a wide side-facing stance, resting the same-side forearm on the thigh, and reaching the opposite arm diagonally overhead.
79. A dancer performs a Ballet Pirouette, bending the knees briefly before rising onto one supporting foot, drawing the other foot toward the knee, and completing a controlled turn.
80. A person performs a Kneeling-to-Standing Transition, starting on both knees, planting one foot in front, and pressing upward to stand before reversing the movement to kneel again.
81. A person performs a Lateral Lunge, taking a wide step to one side and bending that knee while keeping the opposite leg straight, then pushing back to the center.
82. A gymnast performs a Straddle Jump, jumping vertically and opening both legs sideways in midair while reaching the arms outward, then bringing the feet together for landing.
83. A dancer performs a Charleston Step, kicking one foot forward and then stepping it backward while swinging the arms in opposition to the legs.
84. A dancer performs a Ballet Arabesque, balancing on one leg while extending the other leg straight backward and reaching the arms forward and sideways.
85. A person performs a Crab Walk, supporting the body on the hands and feet with the chest facing upward, lifting the hips, and taking alternating steps backward.
86. A player performs a Volleyball Spike, taking a short approach, jumping off both feet, and swinging one arm overhead to strike the ball before landing with bent knees.
87. A person performs Cat-Cow, starting on hands and knees and alternating between rounding the back upward and lowering the abdomen while lifting the chest and head.
88. A person performs Eagle Pose, bending the supporting knee, wrapping the opposite leg around it, and crossing the arms in front of the chest while maintaining balance.
89. A person performs a Push-Up with Shoulder Taps, lowering the chest toward the floor, pressing back up, and alternately lifting each hand to touch the opposite shoulder.
90. A player performs a Volleyball Forearm Pass, lowering into a wide stance, joining the forearms in front of the waist, and extending the legs to redirect the incoming ball.
91. A player performs a Tennis Serve, tossing the ball upward with one hand, drawing the racket arm behind the head, and extending upward into an overhead swing.

92. A gymnast performs an Aerial Cartwheel, stepping sideways into a takeoff and rotating the body laterally with the legs separated, passing through an inverted position without hand support.
93. A gymnast performs a Half-Turn Jump, swinging the arms into a vertical jump and rotating the body by half a turn before landing with both feet together.
94. An athlete performs a Baseball Pitch, lifting the lead knee, striding forward, and rotating the hips and shoulders as the throwing arm whips forward across the body.
95. A gymnast performs a Back Tuck, jumping vertically from both feet, drawing the knees toward the chest during a backward rotation, and extending the legs before landing.
96. A player performs a Basketball Jump Shot, bending both knees before jumping vertically and extending both arms overhead to release the ball, then landing on both feet.
97. A gymnast performs a Front Handspring, stepping forward into a handstand-like inversion and pushing off both hands to bring the feet down beyond the body.
98. An athlete performs a Standing Long Jump, swinging both arms backward, bending the knees, and launching forward off both feet before landing with the feet together.
99. A person performs Warrior II, opening into a wide stance with the front knee bent and extending both arms horizontally while turning the head toward the front hand.
100. A person performs a Single-Leg Hip Hinge, balancing on one foot and tipping the torso forward while extending the other leg backward, then returning to standing without touching that foot down.

D PER-BONE COVERAGE STATISTICS

#	Bone	JS div.	L1 TV	Expansion
1	Pelvis → L_Hip	1.03%	11.18%	12.94%
2	L_Hip → L_Knee	2.88%	17.48%	10.92%
3	L_Knee → L_Ankle	5.82%	27.00%	13.30%
4	L_Ankle → L_Foot	8.10%	30.81%	7.71%
5	Pelvis → R_Hip	1.13%	11.79%	13.46%
6	R_Hip → R_Knee	2.68%	17.27%	10.12%
7	R_Knee → R_Ankle	5.55%	26.10%	12.95%
8	R_Ankle → R_Foot	7.86%	30.48%	6.06%
9	Pelvis → Spine1	1.24%	12.11%	15.05%
10	Spine1 → Spine2	3.34%	22.32%	3.87%
11	Spine2 → Spine3	2.35%	17.83%	3.09%
12	Spine3 → Neck	1.75%	16.20%	7.30%
13	Neck → Head	2.32%	18.04%	1.31%
14	Spine3 → L_Collar	1.58%	14.46%	8.54%
15	L_Collar → L_Shoulder	1.75%	12.14%	1.24%
16	L_Shoulder → L_Elbow	2.55%	19.25%	1.91%
17	**L_Elbow → L_Wrist**	**3.81%**	**22.44%**	**0.14%**
18	Spine3 → R_Collar	1.53%	13.90%	8.36%
19	R_Collar → R_Shoulder	1.45%	11.31%	0.67%
20	R_Shoulder → R_Elbow	1.77%	14.99%	1.98%
21	R_Elbow → R_Wrist	3.28%	21.20%	0.31%
	Mean	**3.04%**	**18.49%**	**6.72%**

Table 5: **Per-bone distributional comparison across all 21 SMPL bones.** JS divergence and L1 total variation measure the overall distributional shift between HumanML3D and our synthetic data; expansion ratio measures the fraction of (θ, ϕ) region newly introduced by the synthetic data, normalized by the original HumanML3D support. All 21 bones exhibit non-zero expansion. The bone visualized in Fig. 3 (**L_Elbow → L_Wrist**, shown in bold) is highlighted and corresponds to one of the smallest expansions, indicating that Fig. 3 is a conservative example.

For each of the 21 bones in the SMPL skeleton, we compute the bone vector (child joint − parent joint) per frame, convert it to spherical coordinates (r, θ, ϕ), and discard r to retain only the direction

($\theta \in [-180°, 180°], \phi \in [-90°, 90°]$). We bin each bone's direction at $3° \times 3°$ resolution, yielding a 121×61 spherical histogram that describes the empirical distribution of directions the bone takes. All statistics in Tab. 5 are computed between two such histograms: one aggregated over HumanML3D, one over our synthetic data.

JS divergence and L1 total variation. JS divergence and L1 total variation jointly measure how far the synthetic distribution has shifted from the original. JS divergence is defined as

$$\mathrm{JS}(P\|Q) = \tfrac{1}{2}\mathrm{KL}(P\|M) + \tfrac{1}{2}\mathrm{KL}(Q\|M), \quad M = \tfrac{1}{2}(P+Q), \tag{8}$$

where P and Q are the normalized direction histograms of the synthetic and original data respectively. JS is symmetric and bounded by $\ln 2 \approx 0.693$ under natural log, making per-bone values directly comparable. L1 total variation is defined as $\frac{1}{2}\sum_i |P_i - Q_i|$, bounded by 1. The two are complementary: JS is sensitive to differences in low-probability regions, while L1 TV measures the total probability mass that must be moved to align the two distributions. We report both so that the conclusion does not depend on a single distance choice.

Expansion ratio. Expansion ratio measures coverage growth rather than distributional shift. Let $\mathcal{S}_{\text{orig}} = \{i : Q_i > 0\}$ denote the set of (θ, ϕ) cells covered by HumanML3D, and $\mathcal{S}_{\text{aug only}} = \{i : P_i > 0 \wedge Q_i = 0\}$ denote the cells reached only by the synthetic data. We define

$$\text{Expansion} \;=\; \frac{|\mathcal{S}_{\text{aug only}}|}{\max(|\mathcal{S}_{\text{orig}}|, 1)}. \tag{9}$$

Unlike JS divergence and L1 TV, which describe how the two distributions differ within their joint support, the expansion ratio directly answers whether the synthetic data enters new regions of direction space that HumanML3D never reaches. A non-zero value implies coverage growth rather than redistribution within the existing support.

Tab. 5 reports all three statistics for the 21 bones. The **terminal joints** (wrist, ankle) consistently show the largest JS and TV, while the **proximal joints** (pelvis-to-hip, pelvis-to-spine) show the largest expansion ratios, reflecting that synthetic motion both modifies extremity dynamics and opens up new torso configurations not present in MoCap.

E Validity Results

The tokenizer encodes local four-frame windows, making local pose validity and short-range dynamics particularly relevant to its training. To assess temporal artifacts beyond pose-level filtering, we additionally compare HumanML3D and synthetic motions using three sequence-level metrics: foot-sliding error, acceleration smoothness, and center-of-mass (COM) stability (Table 6). These assess foot sliding, temporal smoothness, and COM excursions, respectively.

Table 6: Temporal and physics-related metrics for real and synthetic motions. Sequence lengths and motion content differ between the two datasets, so lower values alone do not imply greater physical realism.

Metric	HumanML3D	Synthetic (Ours)
Foot-sliding error (mm)	0.418	0.548
Acceleration smoothness (mm)	6.477	0.804
COM stability (m)	0.241	0.108

Synthetic motions exhibit slightly higher foot-sliding error, with an absolute increase of 0.130 mm. Their lower acceleration value is consistent with interpolation-based temporal smoothing. The lower COM value should not be interpreted as greater dynamic stability: synthetic sequences contain 64 frames, compared with an average of 141 frames in HumanML3D, and mainly represent transitions between two poses with smaller spatial displacement. It indicates limited COM excursions in these sequences, rather than establishing physically balanced motion. These measurements provide complementary evidence of temporal plausibility for tokenizer training.

codebook	FID ↓	Top-1 ↑	Top-3 ↑	Matching ↓	Diversity
512	0.141 ± 0.001	0.490	0.777	3.106	9.666
1024	0.107 ± 0.001	0.490	0.777	3.106	9.666
2048	$\mathbf{0.046 \pm 0.001}$	**0.504**	**0.778**	3.105	9.728
4096	0.093 ± 0.001	0.488	0.778	**3.098**	**9.562**
8192	0.120 ± 0.001	0.488	0.778	3.116	9.600

Table 7: **Codebook size ablation.** Lower FID and MM-Dist are better; higher Top-1 and Top-3 are better. The row with the lowest FID is shown in bold. All experiments are conducted with a HumanML3D-to-synthetic-data ratio of 1:1.

F Codebook Size Ablation

In this section, we ablate the codebook size K, which controls the capacity of the discrete motion vocabulary. The results are provided in Tab. 7. We adopt $K{=}2048$ based on three observations. First, $K{=}2048$ achieves a reconstruction FID within 0.001 of the best $K{=}4096$ while strictly outperforming both $K{\leq}1024$ and $K{\geq}8192$, indicating it lies near the optimum of the FID curve. Second, $K{=}2048$ keeps approximately 76% of codes active, whereas codebooks with $K{\geq}4096$ enter an under-utilized regime where most codes are rarely accessed. Third, $K{=}2048$ is the largest size at which both reconstruction FID and codebook utilization continue to improve under our augmented data, making it the operating point our synthesis pipeline can effectively fill.

G Mixing Ratio Ablation

	HumanML3D (in-dist)		Motion-X++ MPJPE (OOD) ↓			
Tokenizer	FID ↓	Top-1 ↑	Overall	Daily	Sports	Dance
T2M-GPT	0.070	0.501	272.5	251.2	314.4	183.8
ratio=1:2	0.076	**0.504**	**247.3**	244.5	**267.3**	**159.2**
ratio=1:1	**0.046**	0.499	252.4	**232.0**	282.7	161.8
ratio=2:1	0.069	0.492	278.7	255.4	315.3	196.9

Table 8: Ablation on the mixing ratio between real HumanML3D data and synthetic data ($K{=}2048$, 500k iterations). Synthetic augmentation consistently improves in- distribution reconstruction over the original T2M-GPT tokenizer, while different ratios trade off reconstruction fidelity against out-of-distribution generalization on Motion- X++. The ratio 1:1 gives the best overall performance and is used in our final model.

Tab. 8 highlights the trade-off between fidelity and coverage when mixing real and synthetic training data. Original MoCap data, HumanML3D, is indispensable for preserving accurate in-distribution reconstruction, as it provides precise MoCap supervision and anchors the tokenizer to realistic motion structure. At the same time, synthetic data also significantly improves in-distribution reconstruction quality over the original tokenizer baseline, showing that its benefit is not limited to out-of-distribution generalization. This suggests that the synthetic set does not merely add diversity, but also helps the tokenizer better cover underrepresented yet still valid motion patterns within the HumanML3D distribution.

Synthetic data, in contrast, primarily contributes coverage: as its proportion increases, out-of-distribution performance on Motion-X++ improves, indicating that the tokenizer benefits from exposure to a broader motion support than HumanML3D alone can offer.

These results support our central design choice: the most effective tokenizer is obtained not by replacing real data with synthetic data, but by balancing scarce high-quality MoCap data with large-scale synthetic augmentation.

H Data Generation Implementation Details

H.1 Hierarchical Representation and Limb Replacement

Our evolutionary pipeline operates on single HumanML3D poses in the 22×3 joint representation. The skeleton is modeled as a rooted kinematic tree with pelvis as the root, followed by two leg chains, one spine chain, and two arm chains. Rather than exchanging an arbitrary set of joints in Cartesian space, we perform crossover on a hierarchical representation of *non-torso* bone directions. Concretely, we extract 13 exchangeable limb bones (head, collar–shoulder, shoulder–elbow, elbow–wrist, hip–knee, knee–ankle, and ankle/foot bones) and represent each bone in a local coordinate system defined by the torso and its parent limb.

To perform limb replacement, we first choose a *subtree root* uniformly from a fixed set of joints whose descendants correspond to a replaceable limb segment: pelvis, left/right hip, left/right knee, upper spine, left/right collar, left/right shoulder, and left/right elbow. Each selected root is mapped to a predefined subtree of limb bones. For example, choosing the pelvis replaces both lower-body chains, choosing a collar replaces the corresponding arm branch, and choosing an elbow replaces only the forearm and wrist segment. We then swap the local bone directions in that subtree between the two parent poses. This hierarchical replacement preserves the kinematic structure of the motion and avoids implausible Cartesian mixing across unrelated body parts.

H.2 Crossover and Mutation

Before applying crossover, each pose is converted from the stored joint order to the coordinate order expected by the angle-limit code and then mapped into the local bone representation. Given a pair of parents, crossover exchanges the selected subtree of local bone directions between them, producing two candidate children. Each child is reconstructed on the other parent's limb lengths, so that the swapped branch keeps a coherent morphology after replacement.

Mutation is applied *only* to the bones affected by crossover. For each swapped bone, with probability `MRL` we sample a random rotation axis from $\{x, y, z\}$ and a rotation angle from a zero-mean Gaussian with standard deviation `SDL` degrees, and rotate the local bone vector accordingly. This differs from the main-text description in two ways: first, mutation is not applied to the whole body but only to the replaced subtree; second, the perturbation is defined in the local coordinate system, so it changes relative articulation instead of global orientation. After reconstruction, we optionally apply a global rigid rotation, but this branch is disabled in our final pipeline. Finally, each accepted pose is translated vertically so that its lowest joint lies on a sampled ground plane.

H.3 Pose Prior and Validity Filtering

We use the pose prior of Akhter and Black Akhter & Black (2015) as a filter on synthesized poses. In the original prior, anatomically valid bone directions are defined through pose-conditioned joint-angle limits. For first-level limb bones directly attached to the torso, the prior stores occupancy masks over discretized spherical angles. For second-level bones, validity is conditioned on the orientation of the parent limb through a separating plane and a bounded region in a local 2D coordinate system. Following the released implementation, the spherical angles are discretized at 3° resolution, with $\theta \in [-180^\circ, 180^\circ]$ and $\phi \in [-90^\circ, 90^\circ]$.

To use this prior on HumanML3D, we build three torso-centered local frames: one for the upper body, one for the lower body, and one for the collar extension introduced by the 22-joint skeleton. Each non-torso bone is then expressed in the appropriate local frame and checked against the original angular constraints. The resulting validity score is simply the number of valid bones among the 13 exchangeable ones. In our filter, a synthesized child is discarded only if its score is both below a threshold `Th` and lower than its parent's score; equivalently, we keep a child whenever it is at least as valid as its parent, or already exceeds the threshold. In the final pipeline we use `Th`$=13$, under which 67.33% of generated poses are retained.

H.4 Evolution Hyperparameters

Our released code exposes the full evolutionary process through a small set of hyperparameters. The number of generations is controlled by `G`, parent selection by `F`, local mutation by `MRL` and `SDL`, global mutation by `MRG` and `SDG`, and validity filtering by `Th`. When `Mer=True`, newly synthesized poses are merged back into the current population instead of replacing it.

In the configuration used for our main data-generation pipeline, we set `G`=11, `F`=0.2, `MRL`=0.8, `SDL`=50°, `MG=False`, `Th`=13, and `seed`=111. We run the pipeline in the `largefiles` mode, where data are grouped across source files and evolved independently. With `F`=0.2, each generation selects 20% of the current population as fathers and another 20% as mothers, yielding up to $0.4N$ candidate children before filtering for a population of size N. Because `Mer=True`, these survivors are appended to the current population, so the dataset grows over generations instead of being refreshed from scratch.

H.5 Interpolation

After pose synthesis, we densify the generated set by interpolation in joint-rotation space rather than by linear interpolation on joint coordinates. For each input file, we split the poses into two halves and pair them in order; if the number of poses is odd, we pad the smaller half with the first pose so that all pairs are matched. Each pose is converted to per-joint quaternions by inverse kinematics, and intermediate poses are generated by spherical linear interpolation (SLERP) over joint rotations. In our implementation, the root translation is kept fixed to the first pose in the pair, while bone lengths are inherited from that first pose during forward kinematics reconstruction. We use T=65 interpolated frames for each pair.

H.6 Trajectory Recovery Model

Training objective. We train the trajectory-recovery model on the HumanML3D training split using a weighted MSE over normalized root features:

$$\mathcal{L}_{\text{traj}} = \frac{1}{|\mathcal{V}|} \sum_{t \in \mathcal{V}} \sum_{d=1}^{4} w_d \left(\hat{\tilde{r}}_{t,d} - \tilde{r}_{t,d} \right)^2, \qquad \mathbf{w} = (1.0, 1.0, 1.0, 0.5), \tag{10}$$

where $\mathcal{V}$ denotes the set of valid frames, and $\hat{\tilde{r}}_{t,d}$ and $\tilde{r}_{t,d}$ denote the predicted and ground-truth root features in normalized space, ordered as in the main text.

Quantitative evaluation. On the HumanML3D test split (Table 9), the model achieves a $17.6\times$ MPJPE reduction over the zero-trajectory baseline. The larger final-frame root-position error is consistent with error accumulation during temporal integration. These in-distribution results do not directly establish trajectory quality on synthetic motions.

Table 9: Trajectory recovery on the HumanML3D test split. Lower values are better for all metrics. Dashes indicate unreported baseline measurements.

Metric	Zero trajectory	Trajectory recovery
MPJPE (mm)	843.099	**47.795**
Mean root-position L2 error (m)	—	0.047
Final-frame root-position L2 error (m)	—	0.098

H.7 Training Details

For motion tokenization, we set a temporal downsampling rate $l = 4$, hidden dimension 512, code dimension 512 and three residual blocks. Due to the introduction of large-scale synthetic motion data, we increase the codebook size from the original $K = 512$ to $K = 2048$ to accommodate the expanded motion diversity. To balance the original HumanML3D data and the synthetic data during training, we apply the weighted sampling strategy with an effective mixing ratio of 1:1. We evaluate the effectiveness of the augmented tokenizer on downstream motion generation models, including

T2M-GPT Zhang et al. (2023), MotionGPT Jiang et al. (2024a), and MotionAgent Wu et al. (2025). For all models, we replace the original motion tokenizer with our augmented tokenizer while keeping the rest of the architecture unchanged. During training, motion sequences are first encoded into discrete tokens using the learned codebook then we follow the original training protocols of each baseline model for fair comparison. Both tokenizer training and downstream model finetuning are conducted on a single NVIDIA RTX 6000 Pro GPU.

I Analysis of the Text–Motion Alignment Trade-off

We investigate the decline in R-Precision and increase in MM-Dist through two diagnostic analyses: token usage and evaluator behavior under distribution shift.

Token usage. We define the learned-code rate as the fraction of codebook entries activated by text-paired training motions, and the learned-code hit rate as the fraction of generated token occurrences belonging to this set. Table 10a shows that our tokenizer increases the number of codes observed during text-conditioned training from 512 to 1756, despite a lower percentage of the enlarged codebook being covered. Only 0.17% of generated tokens fall outside this learned set. Thus, generation rarely uses codes unseen in text-paired training; however, this does not establish that every generated code is correctly aligned with the conditioning text.

Method	Learned-code rate	Learned-code hit rate
T2M-GPT	100% (512/512)	100%
Ours	85.7% (1756/2048)	99.83%

(a) Token usage analysis

Data	OOV	Top-1 ↑	MM-Dist ↓
HumanML3D test set	—	0.516	2.939
Motion-X++	4.76%	0.133	5.194
idea400	3.60%	0.150	4.209
humman	3.68%	0.072	4.862

(b) Evaluator distribution bias

Table 10: Diagnostic analyses of the text–motion alignment trade-off. (a) Codebook coverage by text-paired training motions and token usage during text-conditioned generation. (b) Ground-truth motion–caption pairs evaluated using the HumanML3D evaluator. Top-1 denotes R-Precision; the dash indicates an unreported OOV rate.

Evaluator distribution bias. We evaluate ground-truth motion–caption pairs from Motion-X++ using the same HumanML3D evaluator. Although these pairs contain real motions and their corresponding captions, their alignment scores degrade substantially (Table 10b), showing that low scores can arise without motion-generation errors. We further examine idea400 and humman, the two Motion-X++ subsets whose captions have the lowest out-of-vocabulary (OOV) rates relative to HumanML3D. Their low R-Precision suggests that lexical novelty alone does not explain the degradation, although OOV rates do not capture all differences in caption semantics.

Together, these analyses suggest that evaluator distribution bias is a substantial contributor to the observed trade-off. They do not quantify its contribution relative to weaker semantic controllability, which cannot be ruled out.

J Per-Subset Reconstruction on Motion-X++

Tab. 11 reports per-frame MPJPE (mm) on all eight Motion-X++ subsets, comparing T2M-GPT and our tokenizer trained with a 1:1 real-to-synthetic ratio. Our final tokenizer improves over T2M-GPT on **7 of 8 subsets**, with the only regression on *perform* ($+2.8\%$, 923 samples, 3.6% of total).

		MPJPE (mm) ↓	
Group	n	T2M-GPT	**Ours**
animation	559	352.7	344.1 (−2.5%)
haa500	6,944	322.7	**282.7** (−14.1%)
humman	971	372.7	**358.3** (−4.0%)
idea400	12,042	244.7	**232.0** (−5.5%)
kungfu	1,032	438.4	**419.0** (−4.6%)
music	3,394	180.8	**161.8** (−11.7%)
perform	923	262.3	269.7 (+2.7%)
Overall	**25,865**	**272.7**	**252.4** (−8.1%)

Table 11: **Per-frame MPJPE (mm) on all Motion-X++ subsets.** The Ours column reports MPJPE alongside the relative change Δ over T2M-GPT (green: improvement, red: regression). ratio=$1:2$ is our final tokenizer. Lower MPJPE is better.