TPPMG: Temporal Planning-driven Progressive Motion Generation

Ruoyu Wang[1†], **Can Deng**[1†], **Xinyi Li**[1], **Zhuo Li**[1]
[1]Tsinghua University
dreamgirlwry@gmail.com, dengc25@mails.tsinghua.edu.cn, lixinyi20@mails.tsinghua.edu.cn, georgeliofbfls@outlook.com

Abstract

Most text-to-motion models treat motion duration as a fixed hyperparameter rather than a variable inferred from semantics, inducing two systematic failure modes: *duration trailing* on short prompts and *stage-wise semantic collapse* on long, multi-stage prompts. To address this issue, we propose **TPPMG** (Temporal Planning-driven Progressive Motion Generation), which factorizes generation as semantics $\rightarrow$ temporal structure $\rightarrow$ motion realization. TPPMG comprises two components: (1) a **Temporal Planner** that maps text to an explicit stage-duration script via an LLM with a lightweight duration calibrator; and (2) a **Progressive Diffusion Generator** that bridges fixed-length diffusion training and variable-length inference through heterogeneous noise scheduling and a sliding-window mechanism, enabling budget-driven variable-length motion synthesis. To evaluate both failure modes with verifiable ground truth, we construct three benchmarks: HumanML3D-Short, HumanML3D-Concat, and BABEL-Concat, along with a comprehensive evaluation suite. Experiments show that TPPMG suppresses duration trailing (UR: $0.52 \rightarrow 0.85$), substantially improves multi-stage structure recovery (PhaseRecall: $0.56 \rightarrow 0.90$, BoundaryF1: $0.28 \rightarrow 0.56$), and reduces boundary discontinuity at stage transitions.

1 Introduction

Human motion generation bridges high-level semantic understanding and low-level movement execution, with broad applications in virtual reality, game animation, and robotics [Yan *et al.*, 2019; Zhao *et al.*, 2020; Zhang *et al.*, 2020; Raab *et al.*, 2023]. In recent years, diffusion models [Sohl-Dickstein *et al.*, 2015; Song and Ermon, 2020] have become the dominant paradigm for text-to-motion (T2M) generation [Petrovich *et al.*, 2022; Kim *et al.*, 2023; Tevet *et al.*, 2023], substantially improving realism and diversity [Kong *et al.*, 2020; Ho *et al.*, 2022a; Saharia *et al.*, 2022; Ho *et al.*, 2022b]. However, most existing methods adopt a fixed-length sampling paradigm: motion sequences are padded or cropped to a preset length, and the denoiser is trained and decoded on a fixed window.

Key issue: systematic temporal mismatch between text and generated motion. This paradigm induces a systematic temporal mismatch between text semantics and generated motion, consistently manifesting as two phenomena:

(1) Duration trailing for short texts. Short instructions typically finish within 0.5–1 s, yet fixed-length sampling is forced to generate many more frames. Consequently, semantic content concentrates early, while the remaining frames become a low-energy tail with little new intent—often manifested as near-static jitter, tempo flattening, or small "action echoes"(Fig. 1, bottom).

(2) Stage-wise semantic collapse for long texts. Under a fixed-length budget,multi-stage instructions (e.g.,"walk to the table, wave, then turn around and leave") often collapse into a motion stream with unclear boundaries, leading to missing stages, semantic entanglement, and blurred transitions(Fig. 1, top)—essentially a distortion in temporal budget allocation.

These two phenomena share the same root cause: **duration is fixed as a training hyperparameter rather than inferred from semantics as a decision variable**. Under fixed-window training/sampling, diffusion models optimize

$$p_\theta(x_{1:T_{\text{fix}}} \mid c), \quad x_{1:T_{\text{fix}}} \in \mathbb{R}^{T_{\text{fix}} \times d}, \tag{1}$$

where T_{fix} is set prior to training. The text condition c specifies *what* to generate, but the generative space has no variable for *how long* to generate or how to allocate stages. Consequently, the model cannot adapt motion length to semantics: when the required duration is less than T_{fix}, it exhibits "passive filling"; when longer, it exhibits "passive compression."

Our perspective: semantics $\rightarrow$ temporal structure $\rightarrow$ motion realization. We argue that temporal structure is semantic information that can be inferred from text and modeled explicitly. Ideally, generation should be factorized as

$$p(x_{1:T}, T \mid c) = p(T \mid c) \cdot p(x_{1:T} \mid c, T), \tag{2}$$

and, for long texts, further introduce a stage script S to instantiate $p(T \mid c)$, yielding an explicit mapping from "semantics" to "(stages, durations)."

Motivated by this, we propose **TPPMG** (Temporal Planning-driven Progressive Motion Generation), which consists of two complementary components:

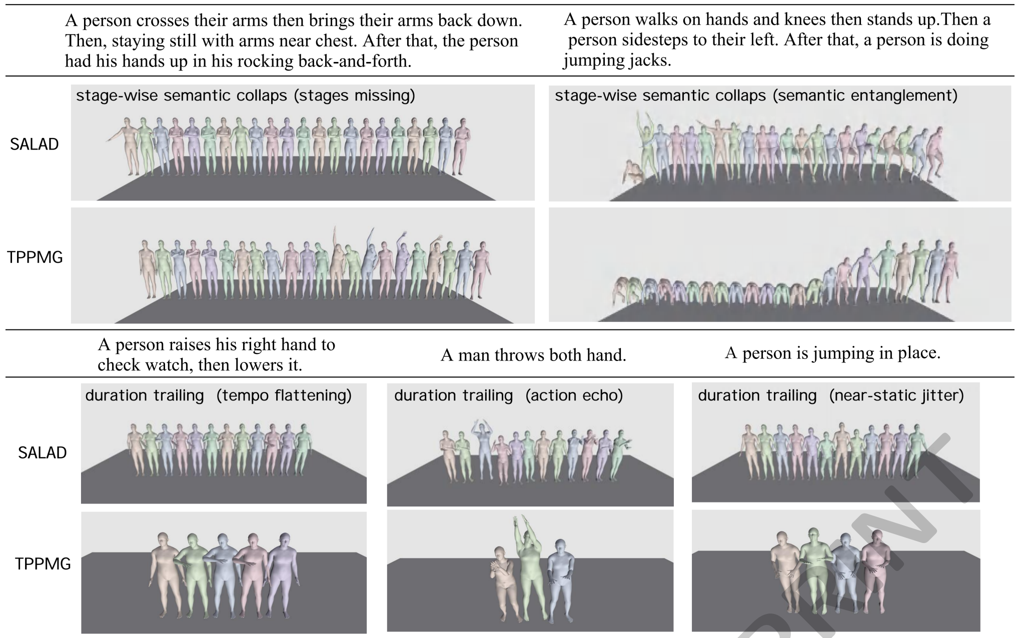

Figure 1: Qualitative evidence of duration trailing on short texts and stage-wise semantic collapse on long, multi-stage texts. Avatars are sampled with a fixed frame stride (same across panels)

(1) Temporal Planner. It uses an LLM[Raffel *et al.*, 2020] with a duration calibrator to infer stages and durations from text, establishing an explicit mapping from semantics to temporal structure.

(2) Progressive Diffusion Generator. Given a variable-length stage budget from the planner, the generator faces a key mismatch: the denoiser is trained on a fixed window length F, yet must produce variable-length sequences at inference. We therefore adopt a sliding-window procedure (*denoise→emit→shift→append*) that turns variable-length generation into budget-driven chunk emission, and introduce heterogeneous noise scheduling to improve cross-window stability and boundary quality. This design bridges fixed-length training and variable-length inference, resulting in motion sequences that match the planned temporal structure.

To evaluate our method, we construct three derived evaluation benchmarks: HumanML3D-Short, filtered by text length; HumanML3D-Concat and BABEL-Concat, multi-stage benchmarks built by concatenating single-stage samples with external ground-truth boundaries and segment lengths. We develop a comprehensive evaluation suite measuring effective-frame proportion, stage-structure recovery, boundary alignment, and segment-level semantic consistency.

Contributions

1. We systematically characterize duration trailing on short texts and stage-wise semantic collapse on long texts under the fixed-length paradigm, and identify the root cause as treating duration as a hyperparameter instead of a semantic variable.
2. We propose an LLM-based Temporal Planner with a duration calibrator to explicitly model the mapping semantics → (stages, durations).
3. We introduce heterogeneous noise scheduling and a sliding-window mechanism to resolve the structural mismatch between fixed-length training and variable-length sampling, enabling stage-aware progressive generation.
4. We construct three derived evaluation benchmarks and validate that TPPMG suppresses trailing (UR: $0.52 \rightarrow 0.85$), recovers temporal structure (PhaseRecall: $0.56 \rightarrow 0.90$, BoundaryF1: $0.28 \rightarrow 0.56$).

2 Related Work

2.1 Diffusion-based T2M and Fixed-Length Bias

Diffusion models have become dominant for T2M [Zhang *et al.*, 2024a], but jointly denoising the full sequence requires a fixed window length at training time, leading to semantic trailing for short prompts and phase blurring for long prompts [Tevet *et al.*, 2023]. Existing works incorporate keyframe/trajectory constraints [Cohan *et al.*, 2024; Karunratanakul *et al.*, 2023] or masked completion [Chen *et al.*, 2023a; Zhang *et al.*, 2024b], but lack mechanisms to infer *phased duration structures* and *drive generation budgets*.

2.2 Duration Modeling and Variable-Length Generation

Autoregressive methods support variable-length generation but cannot infer stage durations from text. Pose-level continuous AR [Guo *et al.*, 2020] enables flexible stopping but suffers from error accumulation; action-level AR (TEACH [Athanasiou *et al.*, 2022]) reduces accumulation via clip-wise generation, yet stage durations remain externally specified. Discrete token-based methods (T2M-GPT, MoMask [Zhang *et al.*, 2023; Guo *et al.*, 2024]) apply language-model-style generation but rely on heuristic token budgets

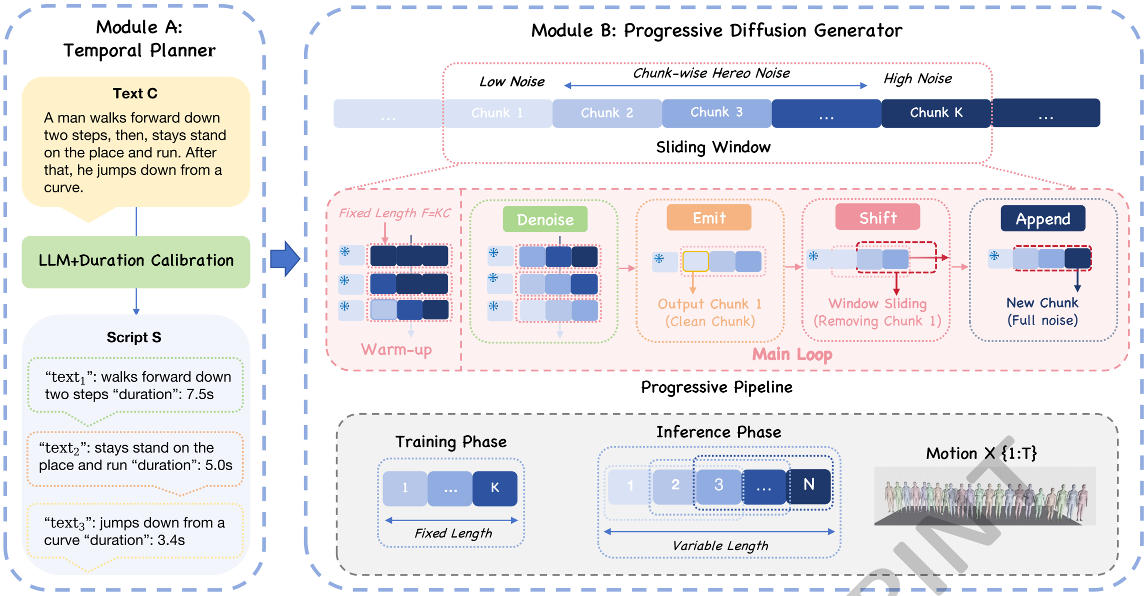

Figure 2: Overview of TPPMG. The Temporal Planner predicts a stage script, and the Progressive Diffusion Generator synthesizes variable-length motion under the planned budgets.

without stage-level planning. VAE-based methods (TEMOS, ACTOR [Petrovich *et al.*, 2022; Petrovich *et al.*, 2021]) condition on duration as external input. Latent diffusion [Chen *et al.*, 2023b; Dai *et al.*, 2024], Flow Matching [Hu *et al.*, 2023], and INR [Cervantes *et al.*, 2022] offer efficient sampling or continuous-time representation, but none decompose total duration into text-inferred stage-level structures.

2.3 Summary

Existing methods lack unified mechanisms to infer phased duration structures from text, achieve stable variable-length inference, and ensure inter-segment continuity—gaps TPPMG addresses via two components.

3 Method

Problem Setup Given text c, our goal is to generate a motion sequence x with an explicit temporal structure. We model generation via a latent stage script $S = \{(\text{text}_i, d_i)\}_{i=1}^{N}$, where text_i is the i-th sub-instruction and $d_i > 0$ its duration (s); total length is induced by S via fps. Accordingly,

$$p(x \mid c) = \sum_{S} p(S \mid c) \cdot p(x \mid S), \tag{3}$$

which decomposes the problem into (i) semantics $\to$ temporal structure ($p(S \mid c)$) and (ii) temporal structure $\to$ motion realization ($p(x \mid S)$).

Method Overview As illustrated in Fig. 2, TPPMG is a plan-then-generate pipeline: (i) **Temporal Planner** outputs $\{(\text{text}_i, \hat{d}_i)\}$ with a lightweight calibrator for duration bias; (ii) **Progressive Diffusion Generator** bridges fixed-window diffusion and variable-length budgets via sliding-window sampling, with heterogeneous noise and overlapped hard-conditioning to stabilize boundaries.

3.1 Temporal Planner

Schema-Constrained Script Generation

Given instruction c, the planner outputs a stage script $S = \{(\text{text}_i, d_i)\}_{i=1}^{N}$ in a fixed JSON schema. We enforce schema validation (and log decoding hyperparameters) to ensure reproducible and parseable plans. The script is constrained by

$$S = \{(\text{text}_i, d_i)\}_{i=1}^{N},\ d_i \in [d_{\min}, d_{\max}],\ \sum_{i=1}^{N} d_i \le d_{\text{tot-max}}. \tag{4}$$

These bounds prevent degenerate estimates and cap total length.

Duration Calibration

Raw LLM durations can be systematically biased across action types. We thus apply a lightweight calibrator h_ψ to obtain the final budget

$$\hat{d}_i = \Pi_{[d_{\min}, d_{\max}]}\left(\text{softplus}\big(h_\psi([\phi(\text{text}_i),\, d_i])\big)\right), \tag{5}$$

where $\phi(\cdot)$ is a frozen text encoder, and softplus with projection enforces positivity and feasibility. We train h_ψ with stage-level supervision:

$$\mathcal{L}_{\text{dur}} = \mathbb{E}\left[\sum_{i=1}^{N} \left|\hat{d}_i - d_i^\star\right|\right], \tag{6}$$

where $d_i^\star$ comes from single-stage real samples or Concat benchmarks with external ground-truth segment lengths, avoiding circular targets.

3.2 Progressive Diffusion Generator

Goal and Challenges

Given a stage script produced by the planner, $S = \{(\text{text}_i, T_i)\}_{i=1}^{N}$, our goal is budget-driven variable-length generation using a fixed-window denoiser. Naively generating each stage independently and stitching can match durations, but introduces tempo distortion, boundary discontinuities, and error accumulation.

To adapt the planner's variable-length temporal structure without changing the fixed-length training paradigm, we introduce three core designs:

(1) **Budget-Driven Sliding-Window Sampling**, which converts variable-length generation into a controllable chunk emission process;

(2) **Heterogeneous Noise Scheduling** within a fixed window, which assigns lower noise to previously generated frames to suppress repeated perturbations, thereby improving cross-window stability and boundary quality;

(3) **Cross-Stage Continuity Interfaces**, which combine overlapped hard-conditioning and end-of-stage cropping to ensure boundary continuity and exact length alignment.

Setup: Fixed-Window EDM Training

We train a text-conditioned denoiser on a fixed window length F, using EDM-style continuous noise parameterization and the standard v-*parameterization* (v-pred). To minimize train–test mismatch, we assign a per-frame noise scale $\{\sigma_n\}_{n=1}^F$ and apply the forward perturbation

$$x[n] = x_0[n] + \sigma_n\,\epsilon[n], \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{7}$$

The network v_θ takes as input the noisy motion x, a noise embedding $\gamma(\sigma_n)$ (e.g., a log-σ MLP), and the text condition $\mathrm{cond}(c)$, and minimizes a frame-wise MSE under v-prediction:

$$\begin{aligned} \mathcal{L}_{\text{diff}} = \mathbb{E}_{x_0, c, \{\sigma_n\}, \epsilon}\Big[\frac{1}{F}\sum_{n=1}^{F} \\ \big\| v_\theta(x, \gamma(\sigma_n), \mathrm{cond}(c))[n] - v[n] \big\|_2^2\Big], \end{aligned} \tag{8}$$

where v follows the standard EDM v-parameterization (details in Appendix to avoid notation ambiguity). If a training sequence is shorter than F, we use a length mask and compute the loss only on valid frames.

Budget-Driven Sliding-Window Sampling

The planner outputs a duration script in seconds. To interface it with a fixed-window denoiser, we first convert the calibrated duration of each stage $\hat{d}_i$ into a target number of frames:

$$T_i = \mathrm{round}(\hat{d}_i \cdot \mathrm{fps}), \qquad T = \sum_{i=1}^{N} T_i. \tag{9}$$

We then discretize the stage length into the number of chunks to *emit*:

$$M_i = \mathrm{ceil}(T_i / C), \tag{10}$$

where C is the chunk length (in frames) and the denoiser window length is fixed to $F = KC$. This yields a key equivalence: **variable-length inference reduces to budget control over the emission count** M_i. The denoiser always operates on an input of fixed length F; by emitting exactly M_i chunks we obtain a sequence that matches the planned stage length, without any post-hoc truncation/interpolation that retimes the motion.

To make chunk-by-chunk emission stable, we structure sampling into two phases: a warm-up phase that establishes a stable conditional prefix, followed by a main loop that emits one chunk per iteration while sliding the window, until M_i chunks are produced.

(1) Warm-up. We initialize $W \sim \mathcal{N}(0, \mathbf{I})$. If the previous stage provides a tail chunk $Tail$, we copy it into the first chunk of the window as an optional hard condition and keep it frozen throughout the reverse process; the remaining positions are filled with noise placeholders. We then run a fixed number of reverse updates L_{warm}, allowing the prefix to converge into a stable context while preserving the suffix as noise placeholders for subsequent expansion.

(2) Main loop. After warm-up, we repeatedly perform *denoise–emit–shift–append*. In each iteration, we run L_{step} reverse updates within the fixed-length window W (denoise), output the first chunk (emit), shift the window left by one chunk (shift), and append an all-noise chunk at the end as a placeholder for future generation (append). We repeat this loop until M_i chunks have been emitted.

Heterogeneous Noise Scheduling

Under sliding-window sampling, after each shift the already-generated prefix remains inside the window and is updated again in subsequent reverse steps. If this prefix is repeatedly perturbed at a comparable noise level, drift accumulates and stage boundaries blur. To mitigate this, we introduce heterogeneous noise scheduling within a fixed window.

We partition a window into K chunks of length C, and assign a monotonically increasing noise scale to chunk k:

$$\begin{aligned} \sigma_n = \sigma_k, \quad n \in [kC, (k+1)C - 1], \\ k = 0, \dots, K-1, \quad \sigma_0 < \cdots < \sigma_{K-1}. \end{aligned} \tag{11}$$

Here σ_n is the per-frame noise scale and $\{\sigma_k\}$ is derived from a standard EDM/Karras schedule (approximately uniform in log-space) and broadcast within each chunk. Intuitively, the prefix chunk is placed in a low-noise region (σ_0), so it is only mildly updated across iterations, while the suffix stays at higher noise for new content. We use the same organization during training to reduce train–test shift.

To improve robustness to noise-scale allocations, we apply a log-space offset augmentation:

$$\begin{aligned} &\log \tilde{\sigma}_k = \mathrm{clip}\big(\log \sigma_k + \delta_k,\ \log \sigma_{\min},\ \log \sigma_{\max}\big), \\ &\delta_k \sim \mathrm{Unif}[-\Delta, \Delta]. \end{aligned} \tag{12}$$

where $\tilde{\sigma}_k$ is the augmented noise scale, $[\sigma_{\min}, \sigma_{\max}]$ is the EDM noise range, and Δ controls the offset magnitude. This log-space perturbation is equivalent to multiplicative noise-scale jitter and improves generalization to different schedules.

Cross-Stage Continuity

Budgeted sliding-window sampling resolves *within-stage* variable-length generation, but *between-stage* discontinuities may still occur at boundaries. We thus introduce an overlapped hard-conditioning interface.

Let the last chunk of stage i be

$$Tail^{(i)} = [x^{(i)}_{T_i-C+1}, \dots, x^{(i)}_{T_i}] \in \mathbb{R}^{C\times d}. \tag{13}$$

When generating stage $i+1$, we place $Tail^{(i)}$ into the first chunk of the new window and hard-freeze it throughout the entire reverse process:

$$W_{1:C} \leftarrow Tail^{(i)},\ \forall \text{ reverse steps: enforce } W_{1:C} = Tail^{(i)}. \tag{14}$$

The remaining $K-1$ chunks are initialized as noise and denoised under the heterogeneous schedule. This provides an immutable overlapped prefix for the new stage, substantially reducing boundary jumps and improving continuity.

Finally, if T_i is not divisible by C, we apply *end-of-stage cropping* only to achieve exact length alignment: let $r = T_i - (M_i - 1)C$, and keep only the first r frames of the last emitted chunk (discard the rest). This cropping is performed after stage generation completes and does not introduce interpolation or re-timing, avoiding tempo distortion.

4 Experiments

4.1 Core Questions and Evaluation Setup

TPPMG targets the **semantic–temporal mismatch** induced by fixed-length sampling. Accordingly, our experiments are organized around two core questions, each paired with a dedicated benchmark and evaluation suite.

Q1 (Short texts): Duration trailing. Under fixed-length generation, short instructions finish their semantic content early and exhibit *passive filling* in remaining frames. To evaluate this, we construct HumanML3D-Short:

- **HumanML3D-Short:** a short-text subset filtered from HumanML3D test split by word count ($\leq$ 6 or $\leq$ 10 words). Note that word count is used solely as a filtering proxy for dataset construction. A pilot study confirmed 98% of $\leq$ 6-word samples correspond to short-duration motions under HumanML3D's distribution.

On HumanML3D-Short, we measure the Utilization Ratio (UR) (↑), defined as the proportion of effective frames from velocity/acceleration energy; higher UR indicates fewer trailing frames and more compact motion.

Q2 (Long texts): Structure, semantics, and continuity. The challenge is not merely generating more frames, but allocating time by semantics and producing stable boundaries. Since natural long-text datasets lack verifiable ground-truth boundaries, using planner output as supervision creates circular evaluation. We thus build two oracle Concat benchmarks with external ground-truth boundaries:

- **HumanML3D-Concat (oracle):** we sample $m \in \{2, 3, 4\}$ single-stage samples from the HumanML3D[Guo *et al.*, 2022] test split and concatenate both motions and texts. Concatenation points provide ground-truth boundaries $\mathcal{B}^\star$ and ground-truth segment lengths $\{T^\star_j\}$.
- **BABEL-Concat (oracle):** we concatenate $m \in \{2, 3, 4, 5\}$ atomic segments from BABEL and describe the sequence via templated text (e.g., `'do ... then ...'`) to reduce linguistic noise. Motions are converted to HumanML3D's representation with fps $= 20$.

Method	UR@$\leq$ 6 words (↑)	UR@$\leq$ 10 words (↑)
MDM	0.52 ± 0.11	0.61 ± 0.12
MotionDiffuse	0.53 ± 0.11	0.63 ± 0.12
SALAD	0.55 ± 0.12	0.65 ± 0.12
T2M-GPT	0.68 ± 0.16	0.74 ± 0.15
MotionGPT	0.69 ± 0.16	0.77 ± 0.15
TPPMG	0.85 ± 0.12	0.90 ± 0.11

Table 1: UR on HumanML3D-Short (↑). Mean±std over prompts.

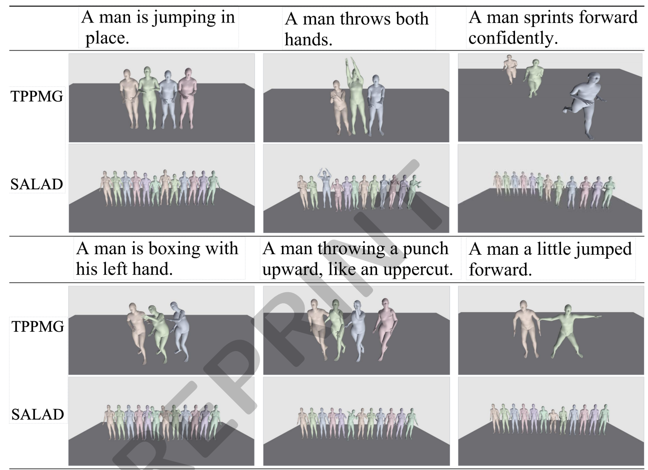

Figure 3: Trailing on short texts. Baselines over-generate redundant tails, while TPPMGterminates promptly. Avatars are sampled with a fixed frame stride.

On the oracle Concat benchmarks, we answer Q2 using three complementary metrics: (i) **structure recovery** via $\text{PhaseRecall}_{\text{cap}}$ (↑), measuring whether the predicted number of stages reaches a sufficient level; Boundary-F1 (↑), measuring how well predicted boundaries align with $\mathcal{B}^\star$; (ii) **segment-level semantic consistency** via PSC@1 (↑), measuring whether the generated i-th segment best matches the ground-truth i-th segment; and (iii) **boundary continuity** via RootJump/JerkSpike (↓) computed at the **oracle boundaries** $\mathcal{B}^\star$, measuring abrupt root-position jumps and jerk spikes at stage transitions.

4.2 Baselines

TPPMG is designed to improve temporal control *without changing the diffusion backbone*. We therefore compare against representative diffusion-based T2M methods under a unified setting. We also report a small number of variable-length (token-based / autoregressive) approaches as reference lines to contextualize performance under natively variable-length paradigms.

4.3 Protocol and Implementation Details

We report mean±std over the evaluation set.

All methods use the HumanML3D motion representation with fps $= 20$. Following common practice, diffusion baselines are sampled at the community-standard fixed length

Method	Structure / Semantics (↑)				Continuity (↓)	
	PhaseRecall$_{\text{cap}}$	BoundaryF1$_A$	BoundaryF1$_B$	PSC@1	RootJump	JerkSpike
HUMANML3D-CONCAT						
MDM	0.56±0.08	0.28±0.06	0.25±0.06	0.36±0.09	0.14±0.05	0.58±0.12
MotionDiffuse	0.58±0.08	0.30±0.06	0.27±0.06	0.38±0.09	0.13±0.05	0.56±0.11
SALAD	0.64±0.07	0.35±0.06	0.32±0.06	0.44±0.09	0.12±0.04	0.52±0.10
T2M-GPT	0.72±0.07	0.40±0.06	0.37±0.06	0.50±0.09	0.11±0.04	0.49±0.10
MotionGPT	0.78±0.06	0.44±0.06	0.41±0.06	0.54±0.08	0.10±0.04	0.46±0.09
TPPMG	**0.90±0.05**	**0.56±0.06**	**0.53±0.06**	**0.72±0.07**	**0.06±0.03**	**0.35±0.08**
BABEL-SIMPLECONCAT						
MDM	0.65±0.07	0.32±0.06	0.29±0.06	0.43±0.09	0.13±0.05	0.55±0.11
MotionDiffuse	0.67±0.07	0.34±0.06	0.31±0.06	0.45±0.09	0.12±0.05	0.53±0.10
SALAD	0.74±0.07	0.40±0.06	0.37±0.06	0.52±0.09	0.11±0.04	0.50±0.10
T2M-GPT	0.81±0.06	0.45±0.06	0.42±0.06	0.58±0.08	0.10±0.04	0.47±0.09
MotionGPT	0.86±0.06	0.49±0.06	0.46±0.06	0.62±0.08	0.09±0.04	0.44±0.09
TPPMG	**0.94±0.04**	**0.62±0.06**	**0.58±0.06**	**0.80±0.06**	**0.05±0.03**	**0.32±0.07**

Table 2: Phase structure, semantic consistency, and boundary continuity on oracle Concat. PhaseRecall$_{\text{cap}}$/BoundaryF1/PSC@1 (↑) are better; RootJump/JerkSpike (↓) are better. BoundaryF1 reports Detector-A/B. Values are mean±std.

$T_{\text{fix}} = 196$. TPPMG inference uses a fixed window $F = 64$ and chunk size $C = 8$ ($K = 8$), with one-chunk overlap freezing for cross-stage continuity.

On the Concat benchmarks, oracle boundaries $\mathcal{B}^\star$ and segment lengths $\{T_j^\star\}$ are available. To avoid ambiguity caused by different generated total lengths, we first resample each generated sequence to the oracle total length $T^\star = \sum_j T_j^\star$, and then segment it by $\mathcal{B}^\star$ before computing structure, semantic-consistency, and boundary-continuity metrics.

4.4 Main Results

Q1: Trailing Suppression on Short Texts

We first evaluate whether TPPMG suppresses duration trailing on short instructions. We test on **HumanML3D-Short** and report the **Utilization Ratio (UR)** (↑).

HumanML3D-Short As shown in Tab. 1, fixed-length diffusion baselines (MDM[Tevet *et al.*, 2023], MotionDiffuse[Zhang *et al.*, 2024a], SALAD[Hong *et al.*, 2025]) exhibit substantial trailing on short texts. In the $\leq$ 6-word bucket, their UR remains low (0.52–0.55), indicating that only a small portion of frames are effective motion while the rest form low-energy tails. Token/AR baselines (T2M-GPT, MotionGPT[Jiang *et al.*, 2024]) alleviate trailing ($\approx$ 0.68–0.69), yet TPPMGachieves the highest UR (0.85 ± 0.12). The $\leq$ 10-word bucket shows the same pattern: MotionGPT reaches 0.77 ± 0.15, while TPPMGattains 0.90 ± 0.11, indicating consistently fewer trailing frames under budget-driven generation.

Qualitative evidence. Fig. 1 and Fig. 3 visualize the same effect: for short prompts, SALAD finishes the core action early but exhibits trailing artifacts (near-static jitter, tempo flattening, or action echo), whereas TPPMGterminates promptly, indicating effective trailing suppression.

Q2: Structure, Semantics, and Continuity on Oracle Concat

We evaluate multi-stage long-text generation on two oracle Concat benchmarks (Tab. 2). Across both datasets, TPPMGconsistently achieves the best *structure–semantics–continuity* trade-off: higher PhaseRecall$_{\text{cap}}$, Boundary-F1 (robust under Detector-A/B), and PSC@1, with lower RootJump/JerkSpike.

HumanML3D-Concat. Diffusion baselines (MDM, MotionDiffuse, SALAD) remain limited by fixed-length sampling, while token/AR models are stronger but still lag behind TPPMG. Against the strongest baseline MotionGPT, TPPMGimproves PhaseRecall$_{\text{cap}}$ from 0.78 ± 0.06 to 0.90 ± 0.05, BoundaryF1$_{A}/_{B}$ from $0.44 \pm 0.06/0.41 \pm 0.06$ to $0.56 \pm 0.06/0.53 \pm 0.06$, and PSC@1 from 0.54 ± 0.08 to 0.72 ± 0.07, while reducing RootJump/JerkSpike from $0.10 \pm 0.04/0.46 \pm 0.09$ to $0.06 \pm 0.03/0.35 \pm 0.08$.

BABEL-SimpleConcat. The same advantage holds under cleaner concatenation: TPPMGoutperforms MotionGPT in PhaseRecall$_{\text{cap}}$/PSC@1 ($0.94 \pm 0.04/0.80 \pm 0.06$ vs. $0.86 \pm 0.06/0.62 \pm 0.08$), raises BoundaryF1$_{A}/_{B}$ ($0.62 \pm 0.06/0.58 \pm 0.06$ vs. $0.49 \pm 0.06/0.46 \pm 0.06$), and lowers RootJump/JerkSpike ($0.05 \pm 0.03/0.32 \pm 0.07$ vs. $0.09 \pm 0.04/0.44 \pm 0.09$).

Qualitative evidence. Fig. 1 and Fig. 4 reflects the same trend: under a fixed-length budget, baselines often collapse multi-stage texts into a stream with blurred oracle boundaries, causing stage omission/repetition and semantic entanglement, with visible discontinuities near splice points; in contrast, TPPMGyields clearer stage progression and smoother boundary transitions, consistent with the gains in boundary alignment, continuity, and segment correspondence.

Generation Quality. Beyond temporal metrics, TPPMG also achieves competitive generation quality on our benchmarks, attaining the lowest FID and highest R-Precision@1 among all compared methods, confirming that temporal planning improves rather than trades off against overall motion quality.

4.5 Ablation Studies

We ablate each component under identical denoiser weights, sampling budgets, and evaluation protocol (Sec. 4.3). Tab. 3 reports UR on HumanML3D-Short and Concat metrics.

w/o Heterogeneous Noise Scheduling. Replacing per-chunk monotonic noise scales with uniform ones (warm-up and overlap freezing unchanged) causes the

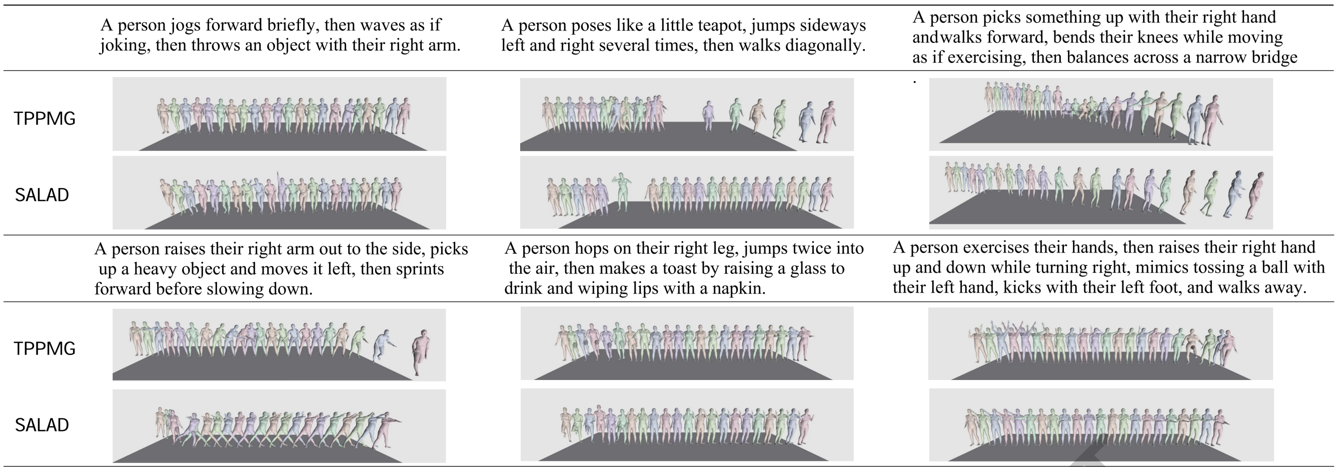

Figure 4: Stage-wise semantic collapse on long texts. Baselines blur/entangle stages, while TPPMGpreserves stage progression. Avatars are sampled with a fixed frame stride

Variant	HumanML3D-Short (UR ↑)		HumanML3D-Concat						BABEL-SimpleConcat					
	UR@$\le$ 6	UR@$\le$ 10	PR_{cap}↑	BF1_A↑	BF1_B↑	PSC@1↑	RJ↓	JS↓	PR_{cap}↑	BF1_A↑	BF1_B↑	PSC@1↑	RJ↓	JS↓
Full (TPPMG)	**0.85**±0.02	**0.90**±0.02	**0.90**±0.02	**0.56**±0.03	**0.53**±0.03	**0.72**±0.03	**0.06**±0.01	**0.35**±0.03	**0.94**±0.02	**0.62**±0.03	**0.58**±0.03	**0.80**±0.03	**0.05**±0.01	**0.32**±0.03
w/o Hetero Noise	0.84±0.02	0.89±0.02	0.79±0.03	0.46±0.03	0.43±0.03	0.58±0.04	0.07±0.01	0.39±0.03	0.86±0.03	0.53±0.03	0.50±0.03	0.68±0.04	0.06±0.01	0.36±0.03
w/o Overlap	0.85±0.02	0.90±0.02	0.88±0.02	0.55±0.03	0.52±0.03	0.71±0.03	0.11±0.02	0.49±0.04	0.93±0.02	0.61±0.03	0.57±0.03	0.79±0.03	0.09±0.02	0.45±0.04
w/o Calibration	0.77±0.03	0.82±0.03	0.87±0.02	0.52±0.03	0.49±0.03	0.67±0.04	0.06±0.01	0.35±0.03	0.91±0.02	0.58±0.03	0.55±0.03	0.74±0.04	0.05±0.01	0.33±0.03

Table 3: Ablations (full pipeline). UR (↑) on HumanML3D-Short; Concat metrics: PhaseRecall(PR_{cap}), Boundary-F1($\text{BF1}_{A/B}$), PSC@1; RootJump(RJ), JerkSpike(JS) at oracle $\mathcal{B}^{\star}$.

largest drops in structure/semantics: on HumanML3D-Concat, $\text{PhaseRecall}_{\text{cap}}$ $0.90{\to}0.79$, PSC@1 $0.72{\to}0.58$, $\text{BoundaryF1}_{A/B}$ $0.56/0.53{\to}0.46/0.43$, with minor continuity changes (RootJump/JerkSpike $0.06/0.35{\to}0.07/0.39$); BABEL-SimpleConcat follows the same trend. This confirms heterogeneous scheduling as key to stage separability and segment correspondence.

w/o Overlap Freezing. Removing overlap freezing selectively hurts continuity: RootJump/JerkSpike rises $0.06/0.35{\to}0.11/0.49$ on HumanML3D-Concat, while structure/semantics change marginally ($\le$ 0.02); BABEL-SimpleConcat shows the same pattern. This supports overlap freezing as the primary mechanism for smooth cross-stage transitions.

w/o Duration Calibration. Removing calibration degrades alignment on both Concat datasets: HumanML3D-Concat PSC@1 $0.72{\to}0.67$, $\text{BoundaryF1}_{A/B}$ $0.56/0.53{\to}0.52/0.49$; BABEL-SimpleConcat PSC@1 $0.80{\to}0.74$, $\text{BoundaryF1}_{A/B}$ $0.62/0.58{\to}0.58/0.55$; continuity is near-unchanged. UR decreases on HumanML3D-Short, indicating weaker duration matching.

4.6 Inference Overhead

Despite stage-wise sampling, TPPMG achieves comparable wall-clock time and lower peak GPU memory than fixed-length baselines. The smaller window (F=64 vs. F=196) reduces self-attention cost quadratically, partially offsetting multi-stage overhead; heterogeneous scheduling limits per-slide denoising to a few steps. Stage-wise generation retains only conditioning frames between stages, reducing peak memory.

5 Conclusion

This paper addresses temporal misalignment in text-driven human motion generation by proposing TPPMG. We reveal that treating duration as a hyperparameter rather than a semantic variable causes duration trailing on short texts and stage-wise semantic collapse on long texts. TPPMG follows "semantics → temporal structure → motion realization": the Temporal Planner models semantics-to-structure mapping, and the Progressive Diffusion Generator resolves the fixed-length training / variable-length inference mismatch. Experiments on three derived evaluation benchmarks (HumanML3D-Short, HumanML3D-Concat, BABEL-Concat) show that TPPMG improves effective duration utilization, stage recovery, boundary alignment, and semantic consistency.

Limitations and Future Work. On HumanML3D-Concat, the Temporal Planner achieves 88.2% stage segmentation accuracy and 0.48s mean duration error, confirming overall reliability. However, two failure modes remain: on ambiguous prompts (e.g., "do some warm-up exercises"), underspecified semantics lead to unstable stage decomposition; on prompts with many unevenly paced stages, brief actions tend to be over-extended while longer stages are compressed, causing systematic boundary shifts. End-to-end optimization of the planner with boundary alignment as a reward signal is a promising direction for addressing these limitations.

Acknowledgements

The authors would like to thank Prof. Qianchuan Zhao for his valuable guidance and insightful contributions throughout this work.

This work was supported in part by the National Natural Science Foundation of China under Grant 62192751, in part by the Key R&D Project of China under Grant 2017YFC0704100, in part by the 111 International Collaboration Program of China under Grant B25027, in part by BNRist Program under Grant BNR2019TD01009, in part by the National Innovation Center of High Speed Train R&D Project under Grant CX/KJ-2020-0006, in part by the InnoHK Initiative, The Government of HKSAR, and in part by the Laboratory for AI-Powered Financial Technologies.

Contribution Statement

Ruoyu Wang and Can Deng contributed equally to this work. Ruoyu Wang is the corresponding author.

References

[Athanasiou *et al.*, 2022] Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. Teach: Temporal action composition for 3d humans. In *2022 International Conference on 3D Vision (3DV)*, pages 414–423. IEEE, 2022.

[Cervantes *et al.*, 2022] Pablo Cervantes, Yusuke Sekikawa, Ikuro Sato, and Koichi Shinoda. Implicit neural representations for variable length human motion generation. In *European Conference on Computer Vision (ECCV)*, pages 356–372. Springer, 2022.

[Chen *et al.*, 2023a] Ling-Hao Chen, Jiawei Zhang, Yewen Li, Yiren Pang, Xiaobo Xia, and Tongliang Liu. Humanmac: Masked motion completion for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9544–9555, 2023.

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

[Cohan *et al.*, 2024] Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–9, 2024.

[Dai *et al.*, 2024] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. In *European Conference on Computer Vision (ECCV)*, pages 387–404. Springer, 2024.

[Guo *et al.*, 2020] 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 *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020.

[Guo *et al.*, 2022] 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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5142–5151, 2022.

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

[Ho *et al.*, 2022a] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[Ho *et al.*, 2022b] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[Hong *et al.*, 2025] Seokhyeon Hong, Chaelin Kim, Serin Yoon, Junghyun Nam, Sihun Cha, and Junyong Noh. Salad: Skeleton-aware latent diffusion for text-driven motion generation and editing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7158–7168, 2025.

[Hu *et al.*, 2023] Vincent Tao Hu, Wenzhe Yin, Pingchuan Ma, Yunlu Chen, Basura Fernando, Yuki M Asano, Efstratios Gavves, Pascal Mettes, Bjorn Ommer, and Cees G. M. Snoek. Motion flow matching for human motion synthesis and editing, 2023.

[Jiang *et al.*, 2024] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36, 2024.

[Karunratanakul *et al.*, 2023] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2151–2162, 2023.

[Kim *et al.*, 2023] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. Flame: Free-form language-based motion synthesis & editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8255–8263, 2023.

[Kong *et al.*, 2020] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

[Petrovich *et al.*, 2021] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021.

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

[Raab *et al.*, 2023] Sigal Raab, Inbal Leibovitch, Peizhuo Li, Kfir Aberman, Olga Sorkine-Hornung, and Daniel Cohen-Or. Modi: Unconditional motion synthesis from diverse data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13873–13883, 2023.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[Saharia *et al.*, 2022] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

[Sohl-Dickstein *et al.*, 2015] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[Song and Ermon, 2020] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[Tevet *et al.*, 2023] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[Yan *et al.*, 2019] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4394–4402, 2019.

[Zhang *et al.*, 2020] Yan Zhang, Michael J Black, and Siyu Tang. Perpetual motion: Generating unbounded human motion. *arXiv preprint arXiv:2007.13886*, 2020.

[Zhang *et al.*, 2023] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14141–14150, 2023.

[Zhang *et al.*, 2024a] 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 Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4115–4128, 2024.

[Zhang *et al.*, 2024b] Siwei Zhang, Bharat Lal Bhatnagar, Yuanlu Xu, Alexander Winkler, Petr Kadlecek, Siyu Tang, and Federica Bogo. Rohm: Robust human motion reconstruction via diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1151–1161, 2024.

[Zhao *et al.*, 2020] Rui Zhao, Hui Su, and Qiang Ji. Bayesian adversarial human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6225–6234, 2020.