DyCoPart: Dynamic Body-Part Coordination for Text-to-Motion Generation

Anonymous authors
Paper under double-blind review

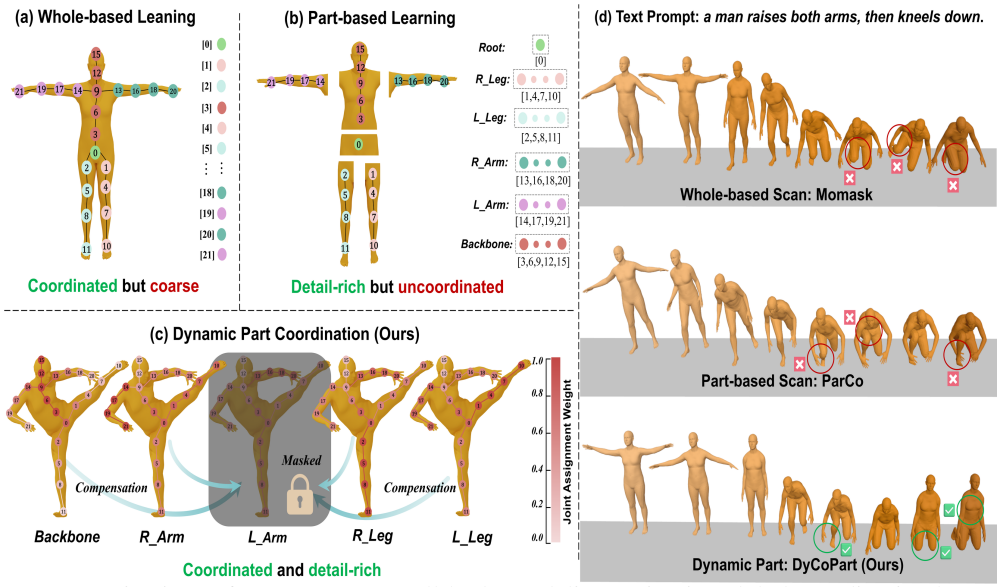

Figure 1: Motivations of DyCoPart. (a) Full-body modeling maintains global coordination but tends to generate coarse local motions. (b) Static part-wise modeling captures fine-grained details but can break whole-body coordination due to fixed anatomical partitions. (c) DyCoPart dynamically routes joints into motion-adaptive parts, where darker colors indicate larger routing weights of joints to this dynamic part, and further applies directed causal coordination among part slots to recover coherent motion under local perturbations. (d) Qualitative results show that our method produces more coherent and physically plausible motions.

Abstract

Text-to-motion (T2M) generation aims to synthesize realistic and physically plausible human motion sequences that are semantically consistent with natural-language descriptions. Despite substantial progress, existing T2M methods still face two key limitations: holistic models tend to miss fine-grained local articulations, while fixed part-based models struggle to capture motion-dependent body coordination due to predefined body-part partitions. To address these limitations, we propose **DyCoPart**, a dynamic coordinated part-aware framework for text-to-motion generation. DyCoPart first encodes joint-level kinematic dependencies with a lightweight graph module and softly routes joints into six dynamic parts under anatomical-prior guidance, producing part tokens that capture both fine-grained local dynamics and motion-dependent coordination. These part tokens are then processed by a shared spatial Part State Space Model, through which whole-body spatial structure is further modeled in the dynamic-part space. Based on the coordinated part representations, a directed causal coordination rectifier is introduced, where temporally causal cross-part messages are propagated among dynamic-part slots so that whole-body coherence can be recovered from corrupted or misaligned local dynamics. Experiments on HumanML3D and KIT-ML demonstrate strong quantitative performance, and coherence-level evaluations further show that DyCoPart generates more fine-grained, physically plausible, and globally coordinated whole-body motions.

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.

1 INTRODUCTION

Text-to-motion aims to generate realistic human motion sequences that are semantically consistent with natural-language descriptions. However, this task poses significant challenges, requiring a model to simultaneously account for intricate spatial-temporal dynamics, fine-grained body-part coordination, and physical plausibility constraints, while remaining faithful to rich semantic descriptions.

Early methods for text-to-motion generation mainly relied on holistic full-body modeling, as exemplified by representative approaches such as MDM Tevet et al. (2022b), MLD Chen et al. (2023), MotionDiffuse Zhang et al. (2024a), MoMask Guo et al. (2024), T2M Mamba Zhan et al. (2026), and Motion Mamba Zhang et al. (2024b). Although these methods have achieved strong overall generation quality, they predominantly learn text-motion correspondence at the global sequence level and therefore often struggle to faithfully capture fine-grained limb dynamics and localized motion semantics, as illustrated in Fig. 1 (a). To overcome this limitation, recent studies have shifted toward part-aware motion modeling. For example, ParCo Zou et al. (2024), AttT2M Zhong et al. (2023), LGTM Sun et al. (2024), and HiSTF Mamba Zhan et al. (2025) introduce explicit body decomposition to enhance the modeling of local motion details. However, as shown in Fig. 1 (b), these methods typically rely on predefined and fixed body partitions, which treat body parts as relatively independent generation units and only restore their dependency through subsequent fusion or coordination modules. As a result, they still lack an explicit mechanism to model time-varying information flow and coordination across body parts during generation, which may lead to locally plausible yet globally incoherent motions, especially for complex actions requiring strong whole-body coupling.

To address these limitations and jointly capture fine-grained local articulation and coordinated whole-body dynamics, we propose DyCoPart, a dynamic coordinated part-aware framework for text-to-motion generation. As illustrated in Fig. 1 (c), DyCoPart replaces fixed anatomical body parts with motion-adaptive Dynamic Parts, allowing each part representation to focus on action relevant joints while maintaining motion coordination. These dynamic-part tokens preserve fine-grained local articulation and provide a flexible representation space for modeling coordination among different body regions. The Spatial Part State Space Model further refines these tokens by capturing structured dependencies across Dynamic Parts, so that local details and whole body organization can be jointly represented. To improve temporal consistency, the directed causal coordination rectifier uses historical part states to provide corrective context for current part representations, helping the model recover from unreliable or inconsistent local dynamics. During training, counterfactual part perturbations encourage the remaining reliable parts to produce compensatory responses, which strengthens coordination robustness and leads to more coherent full body motion generation. As shown in of Fig. 1 (d), DyCoPart jointly improves spatial flexibility, temporal dependency modeling, and coordination robustness, leading to motions that are more semantically accurate, globally coherent, and physically plausible.

Our contributions are summarized as follows.

- We propose Dynamic Part Learning, a motion-adaptive dynamic-part representation module. Controlled ablations show that it improves local motion quality while mitigating the coordination degradation observed with fixed-part modeling.
- We introduce a Directed Causal Coordination Rectifier to explicitly model temporal causal dependencies among body parts and correct locally unreliable or uncoordinated dynamics through reliable part-level context.
- We validate DyCoPart on two standard benchmarks, HumanML3D and KIT-ML, where it achieves strong performance on widely used text-to-motion evaluation metrics.
- We propose an evaluation protocol for limb-level detail and whole-body coordination, under which DyCoPart achieves the best results and demonstrates stronger local expressiveness and global coordination quality.

2 Related Work

2.1 Whole-body text-to-motion generation

Whole-body methods generate a complete motion sequence directly from a textual description. Early studies progressed from direct language-to-pose mapping Ahuja & Morency (2019) to benchmark-driven full-body motion generation Guo et al. (2022a), with later works improving holistic motion representations through variational latent spaces Petrovich et al. (2022), CLIP-aligned embeddings Tevet et al. (2022a), temporal action composition Athanasiou et al. (2022), and reciprocal motion-text tokenization Guo et al. (2022b). Recent progress is mainly driven by diffusion-based and discrete-token modeling. Diffusion-based methods Tevet et al. (2022b); Chen et al. (2023); Zhang et al. (2024a; 2023b); Zhan et al. (2026); Zhang et al. (2024b) enhance realism, diversity, and sampling quality by denoising either motion sequences or latent representations. Discrete-token methods Zhang et al. (2023a); Jiang et al. (2023); Guo et al. (2024) formulate motion as quantized tokens and employ autoregressive, language-model, or masked prediction objectives for text-conditioned generation. Further extensions Zhang et al. (2023c); Chi et al. (2024); Barquero et al. (2024); Dai et al. (2024) explore long-motion composition, multi-motion generation, and fast controllable sampling. These methods serve as the main whole-body baselines and offer strong global sequence modeling for text-to-motion generation. However, their holistic representations often overlook fine local articulations and localized motion semantics, limiting detailed limb modeling.

2.2 Part-based text-to-motion generation

Part-based methods explicitly decompose motion representations according to semantic body regions, aiming to improve fine-grained limb generation. Early studies explored coarse upper- and lower-body decomposition, body-part attention, and fine-grained text-motion alignment to better handle compositional actions and localized motion semantics Ghosh et al. (2021); Zhong et al. (2023); Wang et al. (2023).

Recent works further move from coarse body decomposition to finer part-aware modeling. Along this line, existing methods strengthen part-level generation through LLM-refined body-part descriptions, fine-grained diffusion, part-specific narratives, independent part generators, discrete part codebooks, and part-guided fusion mechanisms Shi et al. (2024); Sun et al. (2024); Zou et al. (2024); Hong et al. (2026); Zhan et al. (2025); Heo et al. (2026). These methods form the main part-based baselines for text-to-motion generation. Compared with whole-body methods, they better capture localized articulations and multi-limb semantics, but still face challenges in maintaining coherent inter-part coordination and physically plausible full-body motion.

3 Method

3.1 Overview of DyCoPart

To address the issues of (1) insufficient modeling of fine grained local articulation dynamics and (2) weak coordination among body parts in whole body motion, we propose DyCoPart, as shown in Fig. 2, a dynamic coordinated part-aware framework for text-to-motion generation. DyCoPart first learns motion-aware joint dependencies with a Graph Neural Network (GNN) and dynamically routes joints into six semantic Dynamic-Parts under anatomical-prior guidance. Each dynamic-Parts is softly initilized according to the current motion state and skeletal context, enabling the part tokens to capture both fine-grained local dynamics and motion-dependent coordination. A shared spatial Part-SSM is then used to model the whole-body spatial structure in the Dynamic-Part space. Building on these dynamic-part representations, we further introduce a directed causal coordination rectifier to enhance motion consistency and recovery when local part dynamics become unreliable, perturbed, temporally desynchronized, or inconsistent with the global motion. Finally, the coordinated part representations are mapped back to the joint space and fused into a full body representation, which is further processed by a Temporal SSM. As a result, DyCoPart achieves fine grained local expressiveness while maintaining robust global coordination.

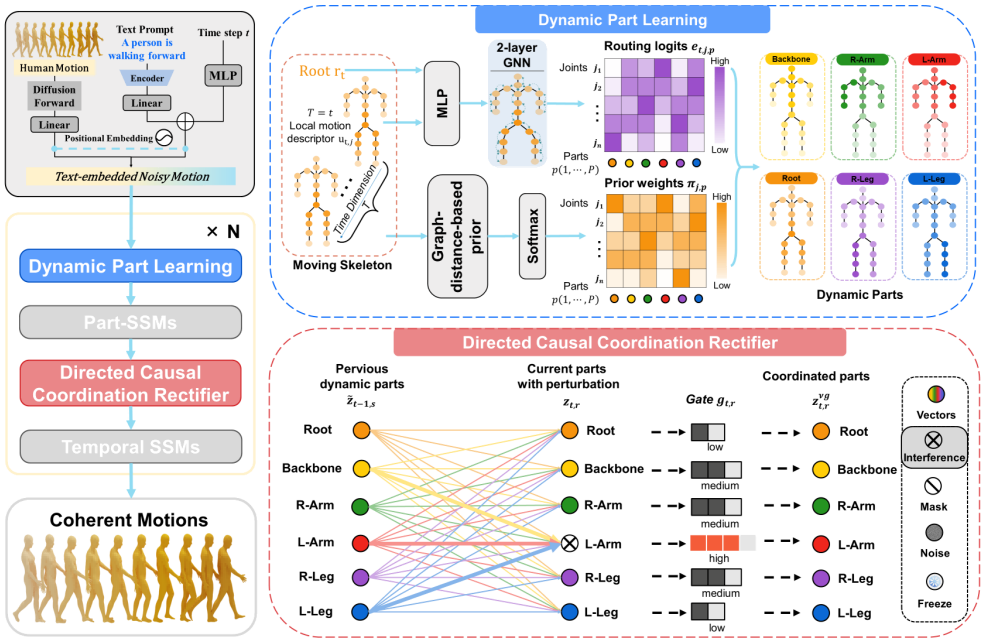

Figure 2: Overview of the proposed **DyCoPart** framework. Given a text prompt and diffusion timestep, the model processes text conditioned noisy motion with stacked DyCoPart layers, each composed of Dynamic Part Learning, Part SSMs, the Directed Causal Coordination Rectifier, and Temporal SSMs. Dynamic Part Learning adaptively assigns joints to time varying body parts through graph based priors and learned routing, while the Directed Causal Coordination Rectifier models directed inter part dependencies to recover coordinated body part dynamics under perturbations. The resulting coordinated part representations are fused to generate coherent full body motions.

3.2 Dynamic Part Learning

Our goal is to learn six semantically meaningful yet motion-adaptive body parts that capture both global coordination and fine-grained local dynamics. Instead of using a fixed hard partition of joints, we predefine six coarse dynamic-part slots

$$\mathcal{P} = \{\text{Root}, \text{R_Leg}, \text{L_Leg}, \text{Backbone}, \text{R_Arm}, \text{L_Arm}\}, \tag{1}$$

and softly route each joint to these slots according to its state of motion and skeletal context. Specifically, we first encode non-root joints with local motion cues and propagate information over the skeleton graph to obtain motion-aware joint features. We then predict motion-conditioned joint-to-part routing scores under an anatomical prior, enabling adaptive yet semantically consistent assignments. Finally, the routed joint features are aggregated into dynamic-part tokens and processed by a shared spatial Part-SSM, producing part representations that encode both local articulation details and coordinated whole body semantics.

Kinematic relation learning. Given a motion sequence $\mathbf{X} \in \mathbb{R}^{B\times T\times D}$, where B, T, and D denote the batch size, sequence length, and motion dimension, we construct a local descriptor $\mathbf{u}_{t,j}$ for each non root joint from its relative position, rotation, and local velocity. The root state $\mathbf{r}_t$ serves as a global reference to ensure consistent joint encoding. We then project the descriptor into the latent space and inject the root context into each joint feature:

$$\mathbf{h}^{(0)}_{t,j} = \text{GELU}\left(\mathbf{W}_2\,\text{GELU}\left(\mathbf{W}_1\mathbf{u}_{t,j}\right)\right) + \text{GELU}\left(\mathbf{W}_r\mathbf{r}_t\right). \tag{2}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_G\times d_u}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_G\times d_G}$ are learnable projection matrices for the local joint descriptor, and $\mathbf{W}_r \in \mathbb{R}^{d_G\times d_r}$ projects the root state into the same graph latent space. We then propagate information along the skeleton graph using a two-layer graph convolution to encode kinematic dependencies among joints. Let $\hat{\mathbf{A}}$ denote the normalized adjacency matrix of the skeleton graph and let J_{nr} denote the number of non-root joints. The ℓ-th propagation layer is defined as

$$\mathbf{h}^{(\ell+1)}_{t,j} = \text{GELU}\left(\mathbf{W}^{(\ell)}_{\text{self}}\mathbf{h}^{(\ell)}_{t,j} + \mathbf{W}^{(\ell)}_{\text{nbr}}\sum_{i=1}^{J_{\text{nr}}}\hat{\mathbf{A}}_{ji}\mathbf{h}^{(\ell)}_{t,i}\right), \quad \ell \in \{0,1\}. \tag{3}$$

where $\mathbf{W}_{\text{self}}^{(\ell)} \in \mathbb{R}^{d_G \times d_G}$ and $\mathbf{W}_{\text{nbr}}^{(\ell)} \in \mathbb{R}^{d_G \times d_G}$ are learnable transformations for the self feature and the aggregated neighboring feature, respectively. After propagation, we obtain the joint representation $\mathbf{h}_{t,j} = \mathbf{h}_{t,j}^{(2)}$, which integrates local motion cues, global root context, and skeletal topology. Based on this feature, we predict motion-conditioned routing logits as

$$\mathbf{e}_{t,j} = \mathbf{W}_a \mathbf{h}_{t,j} + \mathbf{d} = [e_{t,j,1}, e_{t,j,2}, \dots, e_{t,j,P}]^\top \in \mathbb{R}^P, \tag{4}$$

where $\mathbf{W}_a \in \mathbb{R}^{P \times d_G}$ is the learnable routing projection and $\mathbf{d} \in \mathbb{R}^P$ is a learnable bias vector. Each element $e_{t,j,p}$ denotes the routing score of joint j to part slot p at time step t.

Equivalently, the score for assigning joint j to part p is

$$e_{t,j,p} = \mathbf{w}_{a,p}^\top \mathbf{h}_{t,j} + d_p, \tag{5}$$

where $\mathbf{w}_{a,p}^\top$ is the p-th row of $\mathbf{W}_a$ and d_p is the p-th element of $\mathbf{d}$. These routing logits already reflect motion patterns and joint dependencies, but by themselves do not guarantee anatomically meaningful part assignments.

Anatomical-prior-guided dynamic routing. To make the learned part assignments semantically consistent while still preserving cross-part coordination, we introduce an anatomical prior over parts. For each part $p \in \mathcal{P}$, we define an anchor set $\mathcal{A}_p$ on the kinematic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The prior logit and its normalized prior weight are defined as

$$\begin{cases} b_{j,p} = -\alpha\, d_{\mathcal{G}}(j, \mathcal{A}_p) + c_p + \Delta b_{j,p}, \\ \pi_{j,p} = \dfrac{\exp(b_{j,p})}{\sum\limits_{p' \in \mathcal{P}} \exp(b_{j,p'})}, \end{cases} \qquad p \in \mathcal{P}. \tag{6}$$

Here, $\alpha > 0$ controls how quickly the prior decays as joint j moves farther away from the anchor set $\mathcal{A}_p$. c_p is a part-specific bias, and $\Delta b_{j,p}$ is a learnable residual term that allows moderate adaptation to different datasets and motion patterns. In this way, each joint is encouraged to favor its anatomically matched part, while the softmax normalization still preserves nonzero probability mass on other parts, which helps maintain cross-part coordination. We then combine the motion-conditioned routing logits with the anatomical prior to obtain the final joint-to-part assignment weights:

$$m_{t,j,p} = \frac{\pi_{j,p} \exp(e_{t,j,p})}{\sum\limits_{p' \in \mathcal{P}} \pi_{j,p'} \exp(e_{t,j,p'})}, \qquad p \in \mathcal{P}. \tag{7}$$

The weight $m_{t,j,p}$ measures how much joint j contributes to part p at time step t.

Dynamic-part modeling. Using the final assignment weights $m_{t,j,p}$, we aggregate joint features into part-level dynamic tokens and further refine them with a shared spatial Part-SSM, so that each part representation preserves fine-grained local details while capturing coordinated dependencies across body parts.

$$\begin{cases} \mathbf{z}_{t,p} = \sum\limits_{j=1}^{J_{\text{nr}}} m_{t,j,p}\, \mathbf{h}_{t,j}, \\ \tilde{\mathbf{z}}_{1:T,p} = \mathrm{PartSSM}(\mathbf{z}_{1:T,p}), \end{cases} \qquad p \in \mathcal{P}. \tag{8}$$

Here, $\mathbf{z}_{t,p}$ is the token of part p at time step t, obtained by weighted aggregation of all joint features according to their final assignment weights. The sequence $\mathbf{z}_{1:T,p}$ is then processed by a shared $\mathrm{PartSSM}$ to capture long-range temporal dependencies within that part, yielding the temporally enhanced representation $\tilde{\mathbf{z}}_{1:T,p}$. Consequently, at each time step t, the set $\{\tilde{\mathbf{z}}_{t,p}\}_{p \in \mathcal{P}}$ forms six dynamic-part tokens, which are subsequently used as nodes for directed causal vector-graph reasoning.

3.3 Directed Causal Coordination Rectifier

The dynamic-part learning module produces six motion adaptive dynamic-part tokens $\{\tilde{\mathbf{z}}_{t,p}\}_{p \in \mathcal{P}}$, capturing fine-grained local dynamics and motion-dependent coordination. To further maintain coherent whole body coordination under unreliable, perturbed, or temporally desynchronized local dynamics, we introduce a *directed causal coordination rectifier*. It treats each part token as a node and uses directed edges to model how a source part provides corrective coordination information to a

target part. Through vector valued causal message passing, the rectifier leverages reliable body part context to compensate for inconsistent part motions and recover coordinated whole body dynamics.

For simplicity, let $\mathbf{z}_{t,s} \in \mathbb{R}^{d_z}$ denote the Part-SSM-enhanced token of part s at time step t, where $s \in \{1, \dots, S\}$ and $S = 6$. We treat each part token as a graph node, with a directed edge $s \to r$ encoding the influence from source part s to target part r. To enforce temporal causality, each source node is constructed from the previous slot state, i.e., $\bar{\mathbf{z}}_{t,s} = \mathbf{z}_{t-1,s}$ for $t > 1$, with the first slot used as initialization, while target nodes remain in the current slot state. The resulting directed graph models how historical part states provide corrective context for current inter-part coordination.

Efficient factorized directed-edge scoring. A direct way to compute directed edge scores is to build pairwise features for all ordered slot pairs (s, r) and feed them into an MLP. However, this requires expanding source, target, condition, and phase features over all pairs, causing unnecessary memory and computation. We instead adopt a factorized query key scorer, which computes slot projections once and forms pairwise logits through lightweight dot products with condition and phase modulations, while preserving pair specific directed interactions. Specifically, for each ordered edge $s \to r$, we compute the directed compatibility logit as

$$e_{t,s,r} = \frac{q(\bar{\mathbf{z}}_{t,s})^\top k(\mathbf{z}_{t,r})}{\sqrt{d_z}} + \Psi_c(\mathbf{c}_t)_{s,r} + \Psi_p(\mathbf{p}_t)_{s,r}. \tag{9}$$

Here, $q(\cdot)$ and $k(\cdot)$ are linear projections for source and target slots, respectively. The first term measures the content-dependent compatibility between the previous source part and the current target part. The vector $\mathbf{c}_t$ denotes the whole-body condition extracted from the whole-motion branch, and $\Psi_c(\cdot)$ maps it to an $S \times S$ edge-wise modulation matrix. The vector $\mathbf{p}_t$ is a motion phase representation encoded from the root state and foot-contact features, and $\Psi_p(\cdot)$ provides phase-aware edge modulation. To avoid a trivial identity path, self-loops are masked before normalization:

$$\hat{e}_{t,s,r} = \begin{cases} e_{t,s,r}, & s \neq r, \\ -\infty, & s = r. \end{cases} \tag{10}$$

We then apply a standard softmax over the source dimension for each target node:

$$A_{t,s,r} = \frac{\exp(\hat{e}_{t,s,r})}{\sum_{s'=1}^{S} \exp(\hat{e}_{t,s',r})}, \qquad \sum_{s=1}^{S} A_{t,s,r} = 1. \tag{11}$$

The resulting matrix $\mathbf{A}_t \in \mathbb{R}^{S \times S}$ is a directed incoming coordination graph, where $A_{t,s,r}$ measures how much the previous state of source part s contributes to the current update of target part r.

Vector-valued directed message passing. Unlike a scalar relation graph that only estimates edge strengths, our graph propagates vector-valued messages along directed edges. Each source slot first generates a value vector through a message projection $v(\cdot)$. The incoming message for target part r is then obtained by aggregating all directed source messages:

$$\mathbf{m}_{t,r} = \sum_{s=1}^{S} A_{t,s,r}\, v(\bar{\mathbf{z}}_{t,s}). \tag{12}$$

This message contains coordinated motion information delivered from other parts to the target part. Since the edge weight $A_{t,s,r}$ is directional and normalized for each target, the graph can represent asymmetric dependencies such as legs driving the root or the root stabilizing arms, rather than forcing all inter-part relations to be symmetric. To adaptively control the strength of graph-based coordination, we introduce a target-specific gate conditioned on target slot, whole-body state, and the motion phase:

$$g_{t,r} = \sigma\left(\phi_z(\mathbf{z}_{t,r}) + \phi_c(\mathbf{c}_t) + \phi_p(\mathbf{p}_t)\right), \tag{13}$$

where ϕ_z, ϕ_c, and ϕ_p are lightweight linear projections and $\sigma(\cdot)$ denotes the sigmoid function. The gate is further enhanced at motion event frames, such as foot-contact transitions. Let $\epsilon_t \in \{0, 1\}$ denote the event indicator and let η be a learnable event gain. The final graph update is defined as

$$\mathbf{z}_{t,r}^{\mathrm{vg}} = \mathrm{LN}\left(\mathbf{z}_{t,r} + (1 + \eta\epsilon_t) g_{t,r} \mathbf{m}_{t,r}\right). \tag{14}$$

Here, $\mathrm{LN}(\cdot)$ denotes layer normalization. The residual connection preserves the target representation, while the gated directed message injects cross-part coordination only when it useful. Thus, the graph refines each dynamic-part token using temporally causal and direction-aware vector messages.

Learning to rectify inconsistent motion. The resulting matrix $\mathbf{A}_t \in \mathbb{R}^{S\times S}$ is a directed incoming coordination graph, where $A_{t,s,r}$ measures how much the previous state of source part s contributes to the current update of target part r. To endow the model with coordination rectification ability, we introduce training-only counterfactual part-level perturbations, including masking, freezing, and noise injection, each with probability $\rho = 0.1$. These perturbations intentionally create incomplete or temporally desynchronized part representations, forcing the rectifier to recover coherent whole-body coordination from the remaining directed context. We further use three auxiliary coordination losses to ensure that the perturbed slot produces an effective intervention response, the remaining slots provide compensatory support, and the final whole-body output remains stable. These losses serve only as training signals for learning motion rectification, as detailed in Appendix A.

4 Experiments

Table 1: Comparison of results on the HumanML3D and KIT-ML test sets. The right arrow (→) indicates that values closer to real motions correspond to better performance. Each evaluation was run 20 times, and "±" denotes the 95% confidence interval. Bold text highlights the best scores, while underlined entries mark the second-best scores.

Methods	R Precision↑			FID↓	MM Dist↓	Diversity→	MModality↑
	Top 1	Top 2	Top 3				
On the HumanML3D	*0.511*$^{\pm.003}$	*0.703*$^{\pm.003}$	*0.797*$^{\pm.002}$	*0.002*$^{\pm.000}$	*2.974*$^{\pm.008}$	*9.503*$^{\pm.065}$	–
MDM Tevet et al. (2022b)	0.320$^{\pm.005}$	0.498$^{\pm.004}$	0.611$^{\pm.007}$	0.544$^{\pm.044}$	5.566$^{\pm.027}$	**9.559**$^{\pm.086}$	**2.799**$^{\pm.072}$
MotionDiffuse Zhang et al. (2024a)	0.491$^{\pm.001}$	0.681$^{\pm.001}$	0.782$^{\pm.001}$	0.630$^{\pm.001}$	3.113$^{\pm.001}$	9.410$^{\pm.049}$	1.553$^{\pm.042}$
MLD Chen et al. (2023)	0.481$^{\pm.003}$	0.673$^{\pm.003}$	0.772$^{\pm.002}$	0.473$^{\pm.013}$	3.196$^{\pm.010}$	9.724$^{\pm.082}$	2.413$^{\pm.079}$
T2M Mamba Zhan et al. (2026)	0.506$^{\pm.002}$	0.696$^{\pm.002}$	0.793$^{\pm.002}$	0.068$^{\pm.004}$	3.034$^{\pm.007}$	9.497$^{\pm.010}$	2.310$^{\pm.065}$
T2M-GPT Zhang et al. (2023a)	0.492$^{\pm.003}$	0.679$^{\pm.002}$	0.775$^{\pm.002}$	0.141$^{\pm.005}$	3.121$^{\pm.009}$	9.722$^{\pm.082}$	1.831$^{\pm.048}$
MoMask Guo et al. (2024)	0.521$^{\pm.002}$	0.713$^{\pm.002}$	0.807$^{\pm.002}$	0.045$^{\pm.002}$	2.958$^{\pm.008}$	–	1.241$^{\pm.040}$
MMM Pinyoanuntapong et al. (2024)	0.504$^{\pm.003}$	0.696$^{\pm.003}$	0.794$^{\pm.002}$	0.080$^{\pm.003}$	2.998$^{\pm.007}$	9.411$^{\pm.058}$	1.164$^{\pm.041}$
AttT2M Zhong et al. (2023)	0.499$^{\pm.003}$	0.690$^{\pm.002}$	0.786$^{\pm.002}$	0.112$^{\pm.006}$	3.038$^{\pm.007}$	9.700$^{\pm.090}$	2.452$^{\pm.051}$
ParCo Zou et al. (2024)	0.515$^{\pm.003}$	0.706$^{\pm.003}$	0.801$^{\pm.002}$	0.109$^{\pm.005}$	2.927$^{\pm.008}$	9.576$^{\pm.088}$	1.382$^{\pm.060}$
DyCoPart (Ours)	**0.565**$^{\pm.002}$	**0.761**$^{\pm.002}$	**0.848**$^{\pm.002}$	**0.041**$^{\pm.001}$	**2.682**$^{\pm.006}$	9.661$^{\pm.078}$	2.529$^{\pm.044}$
On the KIT-ML	*0.424*$^{\pm.005}$	*0.649*$^{\pm.006}$	*0.779*$^{\pm.006}$	*0.031*$^{\pm.004}$	*2.788*$^{\pm.012}$	*11.08*$^{\pm.097}$	–
MDM Tevet et al. (2022b)	0.164$^{\pm.004}$	0.291$^{\pm.004}$	0.396$^{\pm.004}$	0.497$^{\pm.021}$	9.191$^{\pm.022}$	10.85$^{\pm.109}$	1.907$^{\pm.214}$
MotionDiffuse Zhang et al. (2024a)	0.417$^{\pm.004}$	0.621$^{\pm.004}$	0.739$^{\pm.004}$	1.954$^{\pm.062}$	2.958$^{\pm.005}$	11.10$^{\pm.143}$	0.730$^{\pm.013}$
MLD Chen et al. (2023)	0.390$^{\pm.008}$	0.609$^{\pm.008}$	0.734$^{\pm.007}$	0.404$^{\pm.027}$	3.204$^{\pm.027}$	10.80$^{\pm.117}$	2.192$^{\pm.071}$
T2M Mamba Zhan et al. (2026)	0.437$^{\pm.006}$	0.659$^{\pm.006}$	0.784$^{\pm.006}$	0.195$^{\pm.021}$	2.756$^{\pm.018}$	10.92$^{\pm.096}$	1.512$^{\pm.088}$
T2M-GPT Zhang et al. (2023a)	0.416$^{\pm.006}$	0.627$^{\pm.006}$	0.745$^{\pm.006}$	0.514$^{\pm.029}$	3.007$^{\pm.023}$	10.921$^{\pm.108}$	1.570$^{\pm.039}$
MoMask Guo et al. (2024)	0.433$^{\pm.007}$	0.656$^{\pm.005}$	0.781$^{\pm.005}$	0.204$^{\pm.011}$	2.779$^{\pm.022}$	–	1.131$^{\pm.043}$
MMM Pinyoanuntapong et al. (2024)	0.404$^{\pm.005}$	0.621$^{\pm.005}$	0.744$^{\pm.004}$	0.316$^{\pm.028}$	2.977$^{\pm.019}$	10.910$^{\pm.101}$	1.232$^{\pm.039}$
AttT2M Zhong et al. (2023)	0.413$^{\pm.006}$	0.632$^{\pm.006}$	0.751$^{\pm.006}$	0.870$^{\pm.039}$	3.039$^{\pm.021}$	10.960$^{\pm.123}$	**2.281**$^{\pm.047}$
ParCo Zou et al. (2024)	0.430$^{\pm.004}$	0.649$^{\pm.007}$	0.772$^{\pm.006}$	0.453$^{\pm.027}$	2.820$^{\pm.028}$	10.950$^{\pm.094}$	1.245$^{\pm.022}$
DyCoPart (Ours)	**0.438**$^{\pm.008}$	**0.667**$^{\pm.007}$	**0.797**$^{\pm.005}$	**0.185**$^{\pm.007}$	**2.719**$^{\pm.016}$	**11.073**$^{\pm.187}$	0.962$^{\pm.034}$

4.1 Datasets and Metrics

We evaluate on HumanML3D Guo et al. (2022a) and KIT-ML Plappert et al. (2016) using the standard T2M protocol. Holistic quality is measured by R-Precision, FID, MM-Dist, Diversity, and Multimodality. We assess inter-part coordination using TC and SC, inspired by ParTY Heo et al. (2026), and physical artifacts using Penetrate, Float, Skate Ratio, and BLV, following Shape My Moves Liao et al. (2025). Our encoder-free part-level evaluation computes KFID, NPSS, and JerkGap from recovered 3D joints across six anatomical regions, measuring local dynamics, frequency structure, and jerk-energy consistency, respectively. Metric definitions and evaluation details are provided in Appendix B.1–B.3; dataset statistics are given in Appendix D.1.

4.2 Comparison with State-of-the-arts

To evaluate the effectiveness of DyCoPart, we compare it with recent state-of-the-art text-to-motion methods from three perspectives: holistic evaluation, coordination evaluation, and part-level evaluation. The holistic metrics assess text-motion alignment and overall motion quality on HumanML3D and KIT-ML, while the coordination and part-level metrics further measure whole-body coherence,

physical plausibility, and fine-grained local articulation. We also provide qualitative comparisons to visually demonstrate the advantages of DyCoPart in generating semantically accurate, well-coordinated, and physically plausible motions.

Holistic evaluation. We compare DyCoPart with state-of-the-art text-to-motion methods on HumanML3D Guo et al. (2022a) and KIT-ML Plappert et al. (2016), as reported in Table 1. Among the methods reported in Table 1, DyCoPart achieves the best R-Precision, FID, and MM-Dist on HumanML3D. Compared with the representative fixed part based method ParCo Zou et al. (2024), DyCoPart improves Top-1, Top-2, and Top-3 R-Precision by 9.7%, 7.8%, and 5.9%, respectively, while reducing FID by ***62.4%*** and MM-Dist by 8.4%. On KIT-ML, DyCoPart also obtains the best performance on R-Precision, FID, MM-Dist, and Diversity, with a 5.1% FID reduction and a 1.3% MM-Dist reduction over the strongest prior results. These consistent gains show that motion adaptive part routing and directed causal coordination help capture fine local semantics while preserving coherent whole body motion.

Table 2: Quantitative comparison of **coordination quality** on HumanML3D. We run each evaluation 20 times and report the average with 95% confidence intervals. Bold text highlights the best scores.

Method	TC↑	SC↑	Penetrate (cm)↓	Float (cm)↓	Skate (%)↓	BLV (cm^2)↓
Real	$0.671^{\pm.0001}$	$0.784^{\pm.0001}$	$0.007^{\pm.0011}$	$1.571^{\pm.0104}$	$2.562^{\pm.0840}$	$0.000^{\pm.0000}$
ParCo Zou et al. (2024)	$0.417^{\pm.0002}$	$0.532^{\pm.0005}$	$0.489^{\pm.0107}$	$5.751^{\pm.0341}$	$5.417^{\pm.1210}$	$50.687^{\pm.8750}$
LGTM Sun et al. (2024)	$0.404^{\pm.0002}$	$0.526^{\pm.0005}$	$0.541^{\pm.0127}$	$5.301^{\pm.0382}$	$4.873^{\pm.0974}$	$68.248^{\pm.9851}$
T2M-GPT Zhang et al. (2023a)	$0.607^{\pm.0001}$	$0.731^{\pm.0004}$	$0.098^{\pm.0069}$	$3.011^{\pm.0289}$	$4.781^{\pm.0981}$	$30.427^{\pm.8610}$
MoMask Guo et al. (2024)	$0.598^{\pm.0001}$	$0.754^{\pm.0004}$	$0.102^{\pm.0072}$	$3.241^{\pm.0251}$	$4.547^{\pm.0948}$	$28.377^{\pm.8755}$
DyCoPart (Ours)	$\mathbf{0.623}^{\pm.0002}$	$\mathbf{0.777}^{\pm.0004}$	$\mathbf{0.064}^{\pm.0065}$	$\mathbf{2.984}^{\pm.0242}$	$\mathbf{2.657}^{\pm.0805}$	$\mathbf{18.003}^{\pm.4033}$

Coordination evaluation. Table 2 evaluates whether generated motions are coordinated across body parts and physically plausible. TC and SC measure temporal rhythm consistency and spatial body structure, respectively, where DyCoPart achieves the best scores among all generated methods and approaches real motion in SC. The physical metrics further show that DyCoPart reduces ground penetration, floating, foot skating, and bone length variance, indicating more stable contact behavior and skeletal consistency. Compared with the strongest prior results, DyCoPart improves coherence while reducing skating by 41.6% and BLV by 36.6%, demonstrating that directed causal coordination effectively strengthens whole-body consistency.

Table 3: Quantitative comparison of **part-level** motion quality on HumanML3D. We run each evaluation 20 times and report the average with 95% confidence intervals. Bold text highlights the best scores.

Method	Part	KFID↓	NPSS↓	JerkGap↓	Part	KFID↓	NPSS↓	JerkGap↓
ParCo Zou et al. (2024)	Left Arm	$0.315^{\pm.021}$	$6.918^{\pm.027}$	$0.854^{\pm.008}$	All Body Average	$0.278^{\pm.012}$	$7.691^{\pm.019}$	$0.789^{\pm.005}$
LGTM Sun et al. (2024)		$0.342^{\pm.024}$	$6.874^{\pm.031}$	$0.839^{\pm.009}$		$0.291^{\pm.014}$	$7.642^{\pm.021}$	$0.776^{\pm.006}$
T2M-GPT Zhang et al. (2023a)		$0.471^{\pm.029}$	$7.238^{\pm.036}$	$0.931^{\pm.011}$		$0.399^{\pm.019}$	$7.918^{\pm.026}$	$0.846^{\pm.008}$
MoMask Guo et al. (2024)		$0.286^{\pm.019}$	$6.762^{\pm.024}$	$0.817^{\pm.007}$		$0.238^{\pm.010}$	$7.501^{\pm.017}$	$0.745^{\pm.005}$
DyCoPart (Ours)		$\mathbf{0.228}^{\pm.016}$	$\mathbf{6.626}^{\pm.018}$	$\mathbf{0.782}^{\pm.004}$		$\mathbf{0.191}^{\pm.007}$	$\mathbf{7.366}^{\pm.014}$	$\mathbf{0.716}^{\pm.004}$

Part-level evaluation. Table 3 reports representative part-level results for the left arm and the average over all body parts, while the complete results for all body parts are provided in Appendix C.1. DyCoPart achieves the best scores across all displayed parts and metrics. Compared with MoMask, our method consistently reduces KFID, NPSS, and JerkGap, indicating more realistic local motion distributions, better temporal frequency structures, and smoother part dynamics.

Qualitative comparisons. We compare our method with two holistic methods, MDM and MoMask, and one part-based method, ParCo, under two text prompts. As shown in Fig. 3, MDM and MoMask often suffer from weak limb dynamics. ParCo improves part-level modeling, yet still lacks reliable inter-part coordination. In contrast, our method better follows the text descriptions and produces more natural and coherent whole-body motions. More visualizations are provided in Appendix C.5. The supplementary video provides prompt-matched comparisons with MDM, MoMask, ParCo, and SALAD to support inspection of motion continuity, local articulation, and inter-part coordination.

4.3 Ablation Studies

We conduct ablation studies on key modules and DyCoPart layers, as shown in Table 4. The baseline is a text-conditioned diffusion model with two Spatial–Temporal Mamba blocks, excluding the proposed part-modeling and coordination modules; its configuration is detailed in Appendix D.3.

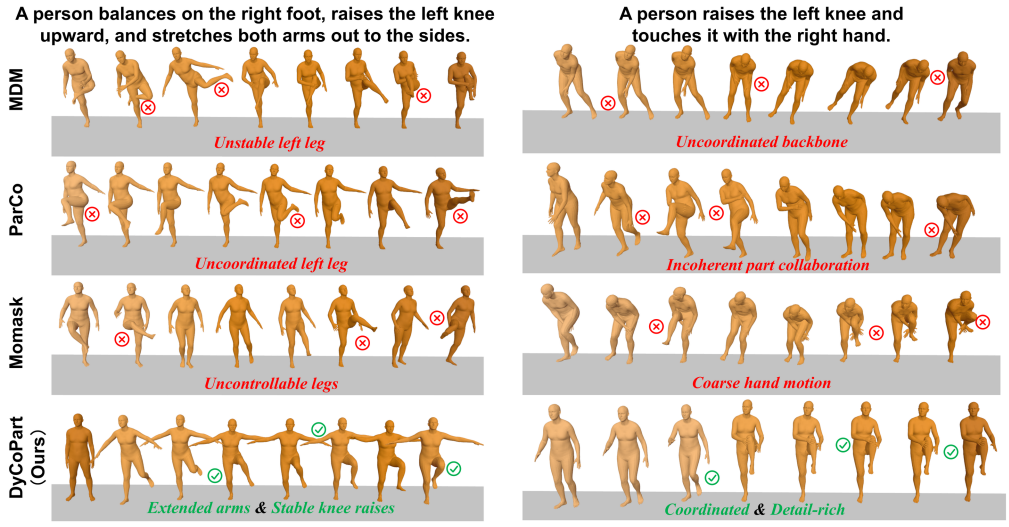

Figure 3: Qualitative comparison on HumanML3D.

Table 4: Ablation studies on key modules and DyCoPart depth. **(a)** Key module ablation, where FP denotes fixed-part modeling, DPL denotes Dynamic Part Learning, and DCCR denotes the Directed Causal Coordination Rectifier. **(b)** Layer ablation of DyCoPart. Bold text highlights the best scores.

Model	R Top 3↑	FID↓	TC↑	SC↑
Baseline	0.832	0.085	0.603	0.742
Base+FP	0.835	0.069	0.433	0.591
Base+DPL	0.844	0.058	0.614	0.769
Base+FP+DCCR	0.841	0.052	0.607	0.762
DyCoPart (Ours)	**0.848**	**0.041**	**0.623**	**0.777**

(a) Key module ablation.

Layer	R Top 1↑	R Top 3↑	FID↓	MM Dist↓
1	0.551	0.839	0.059	2.744
2	**0.565**	**0.848**	**0.041**	**2.682**
3	0.561	0.841	0.048	2.722
4	0.559	0.843	0.052	2.713
5	0.553	0.836	0.067	2.788

(b) Layer ablation of DyCoPart.

Joint local–global ablations and transfer experiments on MDM are reported in Appendices C.2 and C.3, respectively.

Ablation of key modules. Adding Fixed Part Modeling improves FID from 0.085 to 0.069, but decreases TC and SC from 0.603 and 0.742 to 0.433 and 0.591, indicating that fixed part decomposition may weaken inter-part coordination. With the Directed Causal Coordination Rectifier, TC and SC are recovered to 0.607 and 0.762, while FID is further reduced to 0.052. Dynamic Part Learning improves FID to 0.058 without coordination degradation, achieving TC and SC of 0.614 and 0.769. These results verify the effectiveness of Dynamic Part Learning and the Directed Causal Coordination Rectifier. With all components, our full model achieves the best performance.

Layer ablation of DyCoPart. The two-layer setting achieves the best overall results, with the highest R Top 1 and R Top 3, as well as the lowest FID and MM Dist. Increasing the number of layers does not bring further improvement. Therefore, we adopt two DyCoPart layers in our final model.

5 Conclusion

In this paper, we presented DyCoPart, a dynamic body part coordination framework for text to motion generation. By softly routing joints into motion adaptive parts under anatomical prior guidance, DyCoPart captures fine grained local articulations while preserving whole body structure. We further introduced a directed causal coordination rectifier to propagate temporally causal messages among dynamic-part slots, enabling the model to recover coherent motion from unreliable local dynamics. Experiments on HumanML3D and KIT-ML show that DyCoPart achieves strong performance in text motion alignment, motion quality, physical plausibility, and part-level detail. These results demonstrate that dynamic-part representation and causal cross part coordination provide an effective direction for generating more realistic, expressive, and globally coherent human motions.

AI USE STATEMENT

Generative AI tools assisted with literature retrieval, manuscript polishing, and drafting selected passages. The authors take responsibility for the final content.

REFERENCES

Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *2019 International conference on 3D vision (3DV)*, pp. 719–728. IEEE, 2019.

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)*, pp. 414–423. IEEE, 2022.

German Barquero, Sergio Escalera, and Cristina Palmero. Seamless human motion composition with blended positional encodings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 457–469, 2024.

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*, pp. 18000–18010, 2023.

Seunggeun Chi, Hyung-gun Chi, Hengbo Ma, Nakul Agarwal, Faizan Siddiqui, Karthik Ramani, and Kwonjoon Lee. M2d2m: Multi-motion generation from text with discrete diffusion models. In *European conference on computer vision*, pp. 18–36. Springer, 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*, pp. 390–408. Springer, 2024.

Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1396–1406, 2021.

Anand Gopalakrishnan, Ankur Mali, Dan Kifer, Lee Giles, and Alexander G Ororbia. A neural temporal model for human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12116–12125, 2019.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5152–5161, 2022a.

Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, pp. 580–597. Springer, 2022b.

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

KunHo Heo, SuYeon Kim, Yonghyun Gwon, Youngbin Kim, and MyeongAh Cho. Party: Part-guidance for expressive text-to-motion synthesis. *arXiv preprint arXiv:2603.09611*, 2026.

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

Seong-Eun Hong, Soobin Lim, Juyeong Hwang, Minwook Chang, and Hyeongyeop Kang. Bipo: Bidirectional partial occlusion network for text-to-motion synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 32–42, 2026.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.

Taras Kucherenko*, Pieter Wolfert*, Youngwoo Yoon*, Carla Viegas, Teodor Nikolov, Mihail Tsakov, and Gustav Eje Henter. Evaluating gesture generation in a large-scale open challenge: The genea challenge 2022. *ACM Transactions on Graphics*, 43(3):1–28, 2024.

Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 13401–13412, 2021.

Ting-Hsuan Liao, Yi Zhou, Yu Shen, Chun-Hao Paul Huang, Saayan Mitra, Jia-Bin Huang, and Uttaran Bhattacharya. Shape my moves: Text-driven shape-aware synthesis of human motions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1917–1928, 2025.

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

Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *European conference on computer vision*, pp. 480–497. Springer, 2022.

Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. Mmm: Generative masked motion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1546–1555, 2024.

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

Xu Shi, Wei Yao, Chuanchen Luo, Junran Peng, Hongwen Zhang, and Yunlian Sun. Fg-mdm: Towards zero-shot human motion generation via chatgpt-refined descriptions. In *International Conference on Pattern Recognition*, pp. 446–461. Springer, 2024.

Haowen Sun, Ruikun Zheng, Haibin Huang, Chongyang Ma, Hui Huang, and Ruizhen Hu. Lgtm: Local-to-global text-driven human motion diffusion model. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–9, 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*, pp. 358–374. Springer, 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.

Yin Wang, Zhiying Leng, Frederick WB Li, Shun-Cheng Wu, and Xiaohui Liang. Fg-t2m: Fine-grained text-driven human motion generation via diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 22035–22044, 2023.

Xingzu Zhan, Chen Xie, Honghang Chen, Haoran Sun, and Xiaochun Mai. Multi-granular body modeling with redundancy-free spatiotemporal fusion for text-driven motion generation. *arXiv preprint arXiv:2503.06897*, 2025.

Xingzu Zhan, Chen Xie, Honghang Chen, Yixun Lin, and Xiaochun Mai. T2m mamba: Motion periodicity-saliency coupling approach for stable text-driven motion generation. *arXiv preprint arXiv:2602.01352*, 2026.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14730–14740, 2023a.

Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 364–373, 2023b.

Mingyuan Zhang, Huirong Li, Zhongang Cai, Jiawei Ren, Lei Yang, and Ziwei Liu. Finemogen: Fine-grained spatio-temporal motion generation and editing. *Advances in Neural Information Processing Systems*, 36: 13981–13992, 2023c.

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, 2024a.

Zeyu Zhang, Akide Liu, Ian Reid, Richard Hartley, Bohan Zhuang, and Hao Tang. Motion mamba: Efficient and long sequence motion generation. In *European Conference on Computer Vision*, pp. 265–282. Springer, 2024b.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36: 49842–49869, 2023.

Chongyang Zhong, Lei Hu, Zihao Zhang, and Shihong Xia. Attt2m: Text-driven human motion generation with multi-perspective attention mechanism. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 509–519, 2023.

Qiran Zou, Shangyuan Yuan, Shian Du, Yu Wang, Chang Liu, Yi Xu, Jie Chen, and Xiangyang Ji. Parco: Part-coordinating text-to-motion synthesis. In *European Conference on Computer Vision*, pp. 126–143. Springer, 2024.

Appendix

A Coordination Graph Regularization

B Evaluation metrics

B.1 Holistic evaluation metrics

B.2 Coordination evaluation metrics

B.3 Part-level evaluation metrics

C Additional Results and Visualizations

C.1 Full Part-level Evaluation

C.2 Joint Evaluation of Local Articulation and Global Coordination

C.3 Cross-Backbone Evaluation on MDM

C.4 Matched Depth Analysis

C.5 Additional Qualitative Results

C.6 Temporal Changes in Dynamic Part Routing

D Implementation Details

D.1 Datasets

D.2 Routing Notation and Anatomical Anchoring

D.3 Baseline Architecture and Experimental Controls

D.4 Computational Overhead

E Limitations and Future Work

A Coordination Graph Regularization

In the main text, the dynamic-part learning module transforms joint-level motion features into six motion-adaptive part tokens. These tokens are then used as nodes in the directed causal vector graph, where each directed edge $s \to r$ represents how the previous state of source part s contributes to the current update of target part r. Therefore, the learned adjacency matrix $\mathbf{A}_{b,t} \in \mathbb{R}^{S\times S}$ is not merely an auxiliary attention map: it directly controls the vector-valued cross-part message delivered to each dynamic part token. Since the subsequent message passing depends on this graph to refine whole-body coordination, degenerate graph structures can harm motion generation. For example, rapidly oscillating edges may introduce temporal jitter, overly dense edges may blur part-specific dependencies, and nearly symmetric reciprocal edges may weaken the intended directional causal interpretation.

To improve the quality of the learned coordination graph, we complement the stochastic training perturbations described in the main text, including masking, freezing, and noise injection, with three lightweight auxiliary regularizers. The perturbations encourage the graph module to recover coherent coordination from incomplete or disturbed part representations, while the following regularizers directly shape the adjacency matrices $\{\mathbf{A}_{b,t}\}$ to be temporally stable, selective, and direction-aware. These losses are used only during training and therefore introduce no additional computational cost at inference time.

Let

$$\Omega = \{(s, r) \mid s, r \in \{1, \ldots, S\},\ s \neq r\} \tag{15}$$

denote the set of valid directed off-diagonal edges. Since self-loops are masked before the source-wise softmax in the directed graph module, we have $A_{b,t,r,r} = 0$ and $\sum_{s\neq r} A_{b,t,s,r} = 1$ for each target part r.

Temporal smoothness. Human motion usually evolves continuously over time, and the coordination pattern among body parts should also vary smoothly in most frames. If the adjacency matrix changes abruptly between two consecutive frames without a corresponding motion transition, the graph may overfit to local feature noise and produce unstable cross-part communication. We therefore impose a temporal variation loss on the learned directed adjacency matrices:

$$\mathcal{L}_{\mathrm{tv}} = \lambda_{\mathrm{tv}} \frac{1}{B(T-1)|\Omega|} \sum_{b=1}^{B} \sum_{t=2}^{T} (1 - \epsilon_{b,t}) \sum_{(s,r)\in\Omega} \left| A_{b,t,s,r} - A_{b,t-1,s,r} \right|. \tag{16}$$

Here, $\epsilon_{b,t} \in \{0, 1\}$ is the motion-event indicator used in the graph update, such as foot-contact changes or other transition frames. When $\epsilon_{b,t} = 0$, the regularizer encourages the incoming coordination pattern of each target part to evolve smoothly. When $\epsilon_{b,t} = 1$, the constraint is relaxed, allowing the graph to reconfigure rapidly around event frames. This is consistent with the event-enhanced graph update in the main text, where the message strength is increased by the factor $(1 + \eta\epsilon_t)$ during motion events. In other words, event frames are treated as moments at which stronger and possibly more abrupt inter-part coordination changes are expected rather than penalized.

Sparse incoming coordination. The adjacency matrix $\mathbf{A}_{b,t}$ is normalized over incoming source parts for each target part. Without additional constraints, a target part may attend almost uniformly to all other parts. Such dense averaging weakens the semantic meaning of directed coordination, because the target update then becomes a generic mixture of all sources rather than a selective message from the most relevant body parts. To encourage each target part to rely on a small number of dominant source parts, we minimize the entropy of each incoming edge distribution:

$$\mathcal{L}_{\mathrm{sparse}} = \lambda_{\mathrm{sparse}} \frac{1}{BTS} \sum_{b=1}^{B} \sum_{t=1}^{T} \sum_{r=1}^{S} \left(- \sum_{\substack{s=1 \\ s\neq r}}^{S} A_{b,t,s,r} \log \left(A_{b,t,s,r} + \varepsilon_{\log} \right) \right), \tag{17}$$

where $\varepsilon_{\log}$ is a small numerical constant for stability. Since $\sum_{s\neq r} A_{b,t,s,r} = 1$, the entropy is high when the target part receives nearly uniform messages from all source parts, and low when the message is concentrated on a few important sources. Minimizing this term makes the learned graph more selective and interpretable. For instance, during locomotion, the root or backbone may mainly receive coordination messages from the supporting leg, while arm motion may be more strongly conditioned on the backbone or the opposite arm depending on the current motion phase. Importantly, this regularizer does not impose a fixed sparse topology; the selected source parts can still change across samples, time steps, and motion phases.

Direction-aware reciprocity suppression. The directed causal graph is designed to represent asymmetric inter-part influence. Although some physical interactions are naturally reciprocal, assigning high weights to both $s \to r$ and $r \to s$ at the same frame may make the graph behave similarly to an undirected relation map and weaken the directional interpretation of message passing. We therefore discourage simultaneously strong reciprocal edges by penalizing their product:

$$\mathcal{L}_{\mathrm{bidir}} = \lambda_{\mathrm{bidir}} \frac{1}{BT|\Omega|} \sum_{b=1}^{B} \sum_{t=1}^{T} \sum_{(s,r)\in\Omega} A_{b,t,s,r} A_{b,t,r,s}. \tag{18}$$

This term is small when, for a pair of parts, at most one direction dominates at a given time step. It does not enforce a fixed causal ordering between body parts, nor does it forbid the preferred direction from switching over time. Instead, it reduces ambiguous bidirectional coupling and encourages the graph to express temporally causal, direction-specific coordination patterns, such as the supporting leg driving the root during stance or the backbone stabilizing arm motion during turning.

The final coordination-graph regularizer is

$$\mathcal{L}_{\mathrm{coord}} = \mathcal{L}_{\mathrm{tv}} + \mathcal{L}_{\mathrm{sparse}} + \mathcal{L}_{\mathrm{bidir}}. \tag{19}$$

It is added to the diffusion denoising objective with small coefficients, so that the main reconstruction loss remains dominant while the learned graph is biased toward stable, selective, and direction-aware coordination. During inference, all regularization terms are removed and only the learned directed graph module is used.

B EVALUATION METRICS.

B.1 HOLISTIC EVALUATION METRICS.

Following the standard text-to-motion evaluation protocol of T2M Guo et al. (2022a), we evaluate generated motions from four aspects: text-motion alignment, motion realism, motion diversity, and multimodality. Let $\phi_T(\cdot)$ and $\phi_M(\cdot)$ denote the text and motion encoders used for evaluation. Given a test set $\{(x_i, m_i, \hat{m}_i)\}_{i=1}^{N}$, where x_i is the text description, m_i is the real motion, and $\hat{m}_i$ is the generated motion, we define

$$\mathbf{t}_i = \phi_T(x_i), \quad \mathbf{r}_i = \phi_M(m_i), \quad \mathbf{g}_i = \phi_M(\hat{m}_i). \tag{20}$$

R-Precision measures whether a generated motion can retrieve its paired text from a candidate text pool. For each generated motion $\hat{m}_i$, we construct a text set $\mathcal{C}_i$ containing its matched text x_i and several mismatched texts. The rank of the matched text is computed according to the Euclidean distance in the joint embedding space:

$$\text{rank}_i = 1 + \sum_{x \in \mathcal{C}_i \setminus \{x_i\}} \mathbb{I}\left[\|\mathbf{g}_i - \phi_T(x)\|_2 < \|\mathbf{g}_i - \mathbf{t}_i\|_2\right]. \tag{21}$$

The Top-k R-Precision is then defined as

$$\text{R@}k = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left[\text{rank}_i \le k\right], \quad k \in \{1, 2, 3\}. \tag{22}$$

A higher R-Precision indicates better semantic consistency between the generated motion and the input text.

Frechet Inception Distance evaluates the distributional gap between generated and real motions in the motion feature space. Let $(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ and $(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ be the mean and covariance of generated and real motion features, respectively. FID is computed as

$$\text{FID} = \left\|\boldsymbol{\mu}_g - \boldsymbol{\mu}_r\right\|_2^2 + \text{Tr}\left(\boldsymbol{\Sigma}_g + \boldsymbol{\Sigma}_r - 2(\boldsymbol{\Sigma}_g\boldsymbol{\Sigma}_r)^{\frac{1}{2}}\right). \tag{23}$$

A lower FID suggests that the generated motions are closer to real motion distributions.

MultiModal Distance measures the feature distance between each generated motion and its corresponding text description:

$$\text{MMDist} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{g}_i - \mathbf{t}_i\|_2. \tag{24}$$

A smaller value indicates stronger text-motion alignment.

Diversity measures the overall variation among generated motions. Given Q randomly sampled pairs of generated motions, denoted by (a_q, b_q) with $a_q \neq b_q$, Diversity is computed as

$$\text{Div} = \frac{1}{Q}\sum_{q=1}^{Q} \left\|\mathbf{g}_{a_q} - \mathbf{g}_{b_q}\right\|_2. \tag{25}$$

This metric reflects whether the generated set covers diverse motion patterns, and a value closer to real motions is preferred.

Multimodality evaluates whether the model can generate different plausible motions for the same text condition. For each text x_i, we sample K motions and denote their motion features as $\{\mathbf{g}_i^k\}_{k=1}^{K}$. The multimodality score is defined as

$$\text{MMod} = \frac{1}{N}\sum_{i=1}^{N} \frac{2}{K(K-1)} \sum_{1 \le a < b \le K} \left\|\mathbf{g}_i^a - \mathbf{g}_i^b\right\|_2. \tag{26}$$

A higher multimodality score indicates that the model can produce more diverse motion variants under the same textual description.

B.2 COORDINATION EVALUATION METRICS.

TC and SC measure temporal and spatial coherence among body parts, we further evaluate physical coordination using four complementary metrics, namely Penetration, Floating, Skate Ratio, and Bone Length Variance, following the physical plausibility evaluation protocol of shape-aware motion generation Liao et al. (2025). All metrics are computed from recovered global 3D joint positions on HumanML3D, and lower values indicate better physical plausibility.

Temporal Coherence. Temporal Coherence measures the temporal coordination among body parts. Given the recovered 3D joint positions $\hat{\mathbf{p}}_j(t)$, we divide the body into five anatomical parts: backbone, left arm, right arm, left leg, and right leg. For each part g, we compute its frame-wise RMS velocity as

$$x_g(t) = \sqrt{\frac{1}{n_g} \sum_{j \in g} \|\hat{\mathbf{p}}_j(t) - \hat{\mathbf{p}}_j(t-1)\|_2^2}, \tag{27}$$

where n_g denotes the number of joints in part g. The resulting velocity signals are locally z-normalized within sliding windows, and the normalized signals of each body-part pair (g, h) are compared by cross-correlation over temporal lags. For each window w, we aggregate the lag-wise correlations with a softmax-weighted expectation and penalize excessive temporal offsets:

$$\widetilde{R}_{g,h}^{(w)} = \max\left(0, \mathbb{E}_\tau\left[r_{g,h}^{(w)}(\tau)\right]\right) \exp\left(-\frac{\mathbb{E}_\tau[|\tau|]}{\kappa}\right), \tag{28}$$

where $r_{g,h}^{(w)}(\tau)$ is the cross-correlation at lag τ, and κ controls the lag penalty. The final TC score is obtained by averaging $\widetilde{R}_{g,h}^{(w)}$ over all sliding windows and all body-part pairs. A higher TC score indicates stronger rhythmic coordination across body parts.

Spatial Coherence. Spatial Coherence evaluates whether each generated frame preserves plausible inter-part geometry. For each body part g, we first compute its centroid $\mathbf{c}_g(t) = \frac{1}{n_g} \sum_{j \in g} \hat{\mathbf{p}}_j(t)$. We then measure two types of spatial relations: inter-part distances $d_{g,h}(t) = \|\mathbf{c}_g(t) - \mathbf{c}_h(t)\|_2$ and limb-to-torso articulation angles $\theta_g(t)$. To define the reference distribution of natural human geometry, we compute the mean and standard deviation of these distances and angles from real motions in the training split. During evaluation, each generated frame is normalized by these reference statistics and converted into Gaussian consistency scores:

$$s_{g,h}^d(t) = \exp\left(-\frac{(z_{g,h}^d(t))^2}{\beta_d^2}\right), \qquad s_g^\theta(t) = \exp\left(-\frac{(z_g^\theta(t))^2}{\beta_\theta^2}\right), \tag{29}$$

where $z_{g,h}^d(t)$ and $z_g^\theta(t)$ are the z-scores of the distance and angle measurements, respectively. The final SC score is computed by averaging these consistency scores over all frames, all body-part pairs, and all limb-to-torso angle relations:

$$S_{\mathrm{SC}} = \frac{1}{T} \sum_{t=1}^{T} \frac{\sum_{(g,h) \in \mathcal{P}} s_{g,h}^d(t) + \sum_{g \in \mathcal{Q}} s_g^\theta(t)}{|\mathcal{P}| + |\mathcal{Q}|}, \tag{30}$$

where $\mathcal{P}$ denotes all body-part pairs and $\mathcal{Q}$ denotes the set of limbs. A higher SC score indicates that the generated motion better follows the empirical spatial structure of real human motions.

Penetration. Penetration evaluates whether the generated body moves below the estimated ground plane. Since the absolute vertical origin of a recovered motion sequence may not always correspond to the true floor level, we follow the implementation and estimate a sequence-specific ground height from the valid frames. Specifically, let $\mathcal{F}$ denote the dataset-specific set of foot joints, and let $m_t \in \{0, 1\}$ indicate whether frame t is valid. The ground height is estimated as the lowest foot-joint height observed in the valid part of the sequence:

$$y_g = \min_{t: m_t = 1,\ f \in \mathcal{F}} \hat{\mathbf{p}}_{t,f}^y. \tag{31}$$

We then compute the lowest height among all body joints at each frame,

$$h_t^{\min} = \min_{j \in \mathcal{J}} \hat{\mathbf{p}}_{t,j}^y, \tag{32}$$

where $\mathcal{J}$ is the set of body joints. If $h_t^{\min}$ is lower than y_g, the body is considered to penetrate the ground at that frame. The final score averages this below-ground depth over valid frames:

$$\text{Penetrate} = \frac{100}{T_v} \sum_{t=1}^{T} m_t \, \max(0, y_g - h_t^{\min}), \quad T_v = \sum_{t=1}^{T} m_t. \tag{33}$$

The factor 100 converts the value from meters to centimeters, matching the implementation. A smaller value indicates that fewer body joints pass through the ground plane, and therefore the generated motion has fewer ground-intersection artifacts.

Floating. Floating measures the opposite type of ground-contact error: instead of going below the floor, the body remains unnaturally above it. Using the same estimated ground height y_g, we measure the vertical gap between the lowest body joint and the ground plane when the entire body is above the ground:

$$\text{Float} = \frac{100}{T_v} \sum_{t=1}^{T} m_t \, \max(0, h_t^{\min} - y_g). \tag{34}$$

This metric is high when no body joint, typically no foot joint, is close enough to the ground for a sustained portion of the sequence. It is therefore useful for detecting motions where the character appears to hover, lose support, or fail to make plausible contact with the floor. In the implementation, Penetration, Floating, and Skate Ratio are computed only on motions regarded as ground-based. A sequence is filtered out when the lowest body joint remains higher than $y_g + \delta_{\text{air}}$ for at least N_{air} consecutive valid frames, which helps avoid penalizing naturally airborne motions such as jumping or climbing. The default values are $\delta_{\text{air}} = 0.25$ m and $N_{\text{air}} = 5$.

Skate Ratio. Skate Ratio evaluates foot sliding during estimated ground contact. For each foot joint $f \in \mathcal{F}$, a contact label is first inferred from its height relative to the estimated ground:

$$c_{t,f} = m_t \, \mathbb{I}\left(\hat{\mathbf{p}}_{t,f}^{y} < y_g + \delta_h\right), \tag{35}$$

where δ_h is the contact-height threshold. In the implementation, $\delta_h = 0.03$ m by default. For frames after the first one, we compute the horizontal velocity of the foot on the ground plane:

$$v_{t,f}^{xz} = \left\| \hat{\mathbf{p}}_{t,f}^{xz} - \hat{\mathbf{p}}_{t-1,f}^{xz} \right\|_2 \cdot \text{fps}, \quad t = 2, \ldots, T. \tag{36}$$

The skate ratio is the percentage of contact frames in which the horizontal foot velocity is larger than a velocity threshold δ_v:

$$\text{Skate} = 100 \times \frac{\sum_{t=2}^{T} \sum_{f \in \mathcal{F}} c_{t,f} \, \mathbb{I}\left(v_{t,f}^{xz} > \delta_v\right)}{\max\left(\sum_{t=2}^{T} \sum_{f \in \mathcal{F}} c_{t,f}, 1\right)}. \tag{37}$$

The default implementation uses $\delta_v = 0.50$ m/s and $\text{fps} = 20$. A high Skate Ratio means that the model often predicts a foot as being close to the ground while the same foot is still moving horizontally too fast. This usually corresponds to visible foot sliding and indicates inconsistency among foot contact, root motion, and leg articulation.

Bone Length Variance. Bone Length Variance measures whether the generated skeleton preserves stable limb proportions over time. For a physically consistent human motion, the lengths of limb segments should not fluctuate across frames after the 3-D joints are recovered. The implementation evaluates four limb chains, including the left leg, right leg, left arm, and right arm. For a limb chain $g \in \mathcal{G}$ with ordered joints $(j_1, \ldots, j_K)$, its total chain length at frame t is computed as

$$L_g(t) = 100 \sum_{r=1}^{K-1} \left\| \hat{\mathbf{p}}_{t,j_r} - \hat{\mathbf{p}}_{t,j_{r+1}} \right\|_2. \tag{38}$$

Here the factor 100 converts the length from meters to centimeters before computing the variance. The BLV score is then obtained by averaging the temporal variance of these limb-chain lengths:

$$\text{BLV} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \text{Var}\left(\{L_g(t) \mid m_t = 1\}\right). \tag{39}$$

For the full evaluation set, this value is further averaged across sequences. Unlike Penetration, Floating, and Skate Ratio, BLV is computed for all valid motions because it measures skeletal consistency rather than ground-contact behavior. A lower BLV indicates that the generated motion better maintains fixed human limb lengths, while a higher value suggests temporal distortion of the recovered skeleton.

B.3 PART-LEVEL EVALUATION METRICS

We compute part-level local detail metrics from recovered global 3D joint positions. For each motion sequence, let $\mathbf{p}_{t,j} \in \mathbb{R}^3$ denote the global 3D position of joint j at frame t, where $t = 1, \ldots, T$ and T is the valid sequence length. Padding frames are excluded according to the motion length. We divide the skeleton into six semantic parts,

$$\mathcal{G} = \{\text{Root}, \text{R_Leg}, \text{L_Leg}, \text{Backbone}, \text{R_Arm}, \text{L_Arm}\}. \tag{40}$$

For each part $g \in \mathcal{G}$, let $\mathcal{J}_g$ be its joint set and $|\mathcal{J}_g|$ be the number of joints in this part. The concrete joint sets follow the implementation. For HumanML3D, we use

$$\begin{aligned} &\text{Root} = \{0, 1, 2, 3\}, \quad \text{R_Leg} = \{0, 2, 5, 8, 11\}, \quad \text{L_Leg} = \{0, 1, 4, 7, 10\}, \\ &\text{Backbone} = \{0, 3, 6, 9, 12, 15\}, \quad \text{R_Arm} = \{9, 14, 17, 19, 21\}, \quad \text{L_Arm} = \{9, 13, 16, 18, 20\}. \end{aligned} \tag{41}$$

The KIT split uses the corresponding KIT skeleton indices. Root is represented by multiple root-neighboring joints rather than a single root joint, since geometric and kinetic statistics require local structure.

For each part g, we use its first joint as the anchor a_g. The local part position is defined by subtracting this anchor from all joints in the same part:

$$\mathbf{p}_{t,j}^{g,\text{rel}} = \mathbf{p}_{t,j} - \mathbf{p}_{t,a_g}, \quad j \in \mathcal{J}_g. \tag{42}$$

This local coordinate form removes global translation and focuses the metric on the articulation and dynamics within the part. Let f_{fps} denote the frame rate. We compute velocity, acceleration, and jerk by finite differences:

$$\mathbf{v}_{t,j}^g = f_{\text{fps}} \left(\mathbf{p}_{t+1,j}^{g,\text{rel}} - \mathbf{p}_{t,j}^{g,\text{rel}} \right), \tag{43}$$

$$\mathbf{a}_{t,j}^g = f_{\text{fps}} \left(\mathbf{v}_{t+1,j}^g - \mathbf{v}_{t,j}^g \right), \quad \mathbf{r}_{t,j}^g = f_{\text{fps}} \left(\mathbf{a}_{t+1,j}^g - \mathbf{a}_{t,j}^g \right). \tag{44}$$

Here, $\mathbf{v}_{t,j}^g$, $\mathbf{a}_{t,j}^g$, and $\mathbf{r}_{t,j}^g$ denote local velocity, acceleration, and jerk, respectively. All reported part-level metrics are lower-is-better.

Part-level KFID. Kinetic Fréchet Distance measures whether the generated part dynamics follow the distribution of real part dynamics. It is inspired by the kinetic-feature Fréchet evaluation used in AIST++ and AI Choreographer Li et al. (2021). For each sequence and part g, we construct a kinetic descriptor ϕ_K^g from the temporal statistics of speed, acceleration magnitude, and jerk magnitude:

$$s_{t,j}^g = \|\mathbf{v}_{t,j}^g\|_2, \quad u_{t,j}^g = \|\mathbf{a}_{t,j}^g\|_2, \quad q_{t,j}^g = \|\mathbf{r}_{t,j}^g\|_2. \tag{45}$$

The descriptor contains the mean and standard deviation of $s_{t,j}^g$, $u_{t,j}^g$, and $q_{t,j}^g$ over time for every joint in $\mathcal{J}_g$:

$$\phi_K^g = [\mu_t(s_{:,j}^g), \sigma_t(s_{:,j}^g), \mu_t(u_{:,j}^g), \sigma_t(u_{:,j}^g), \mu_t(q_{:,j}^g), \sigma_t(q_{:,j}^g)]_{j \in \mathcal{J}_g}. \tag{46}$$

Here, $\mu_t(\cdot)$ and $\sigma_t(\cdot)$ denote temporal mean and standard deviation. Given generated kinetic features with mean and covariance $(\mu_{K,\text{gen}}^g, \Sigma_{K,\text{gen}}^g)$, and real kinetic features with $(\mu_{K,\text{real}}^g, \Sigma_{K,\text{real}}^g)$, Part-level KFID is

$$\text{KFID}^g = \|\mu_{K,\text{gen}}^g - \mu_{K,\text{real}}^g\|_2^2 + \text{Tr}\left(\Sigma_{K,\text{gen}}^g + \Sigma_{K,\text{real}}^g - 2 \left(\Sigma_{K,\text{gen}}^g \Sigma_{K,\text{real}}^g \right)^{1/2} \right). \tag{47}$$

A lower KFID^g indicates that the generated part has more realistic local motion energy, velocity variation, acceleration patterns, and jerk statistics.

Part-level NPSS. Normalized Power Spectrum Distance measures whether the generated part preserves the temporal frequency structure of real local motion. It follows the normalized power spectrum similarity used in human motion prediction Gopalakrishnan et al. (2019). For each part g, we use the local velocity signal and flatten it into

$$\mathbf{x}^g \in \mathbb{R}^{(T-1) \times C_g}, \quad C_g = 3|\mathcal{J}_g|, \tag{48}$$

where C_g is the number of velocity channels. The signal is centered over time, and its discrete Fourier transform is computed for each channel c:

$$P_c^g(f) = |\mathcal{F}(\mathbf{x}_{:,c}^g)(f)|^2, \tag{49}$$

where $P_c^g(f)$ is the power at frequency f. The power spectrum is normalized over frequencies:

$$\bar{P}_c^g(f) = \frac{P_c^g(f)}{\sum_{f'} P_c^g(f') + \epsilon}. \tag{50}$$

We then compute the cumulative normalized spectrum:

$$C_c^g(f) = \sum_{f' \le f} \bar{P}_c^g(f'). \tag{51}$$

For each channel, the spectral distance between generated and real motion is the L_1 distance between cumulative spectra:

$$D_c^g = \sum_f \left| C_{c,\text{gen}}^g(f) - C_{c,\text{real}}^g(f) \right|. \tag{52}$$

Following the implementation, channels are weighted by the total real-motion power:

$$w_c^g = \frac{\sum_f P_{c,\text{real}}^g(f)}{\sum_{c'} \sum_f P_{c',\text{real}}^g(f) + \epsilon}. \tag{53}$$

The part-level NPSS is

$$\text{NPSS}^g = \sum_c w_c^g D_c^g. \tag{54}$$

Sequences shorter than the minimum valid length are skipped. A lower NPSS^g indicates that the generated part better preserves the frequency distribution of real local motion. This is useful for detecting missing motion details, since over-smoothed motions tend to lose high-frequency components, while jittery motions introduce abnormal spectral energy.

Part-level JerkGap. JerkGap measures whether the local smoothness of a generated part is close to that of real motion. It is motivated by motion smoothness evaluation in gesture generation, where jerk is the third temporal derivative of joint positions Kucherenko* et al. (2024). For each part g, we compute jerk energy as

$$E_{\text{jerk}}^g = \frac{1}{(T-3)|\mathcal{J}_g|} \sum_{t=1}^{T-3} \sum_{j \in \mathcal{J}_g} \|\mathbf{r}_{t,j}^g\|_2. \tag{55}$$

Let $E_{\text{jerk,gen}}^g$ and $E_{\text{jerk,real}}^g$ denote the jerk energy of generated and real motion. JerkGap is defined as

$$\text{JerkGap}^g = \frac{|E_{\text{jerk,gen}}^g - E_{\text{jerk,real}}^g|}{E_{\text{jerk,real}}^g + \epsilon}. \tag{56}$$

A lower JerkGap^g means that the generated part has a smoothness level closer to real motion. Unlike directly minimizing jerk, this reference-based metric does not favor frozen or overly smooth motion. It penalizes both excessive high-frequency jitter and insufficient local motion variation.

C Additional Results and Visualizations

C.1 Full Part-level Evaluation

Table 5 provides the complete part-level evaluation on HumanML3D. Unlike the main paper, which reports representative results on the left arm, left leg, and backbone due to space limits, this appendix includes all six semantic parts, namely Root, left and right legs, Backbone, and left and right arms. DyCoPart consistently achieves the best performance across all parts and all reported metrics, including KFID, NPSS, and JerkGap. Averaged over the six parts, DyCoPart improves over the strongest baseline MoMask by reducing KFID from 0.238 to 0.191, NPSS from 7.501 to 7.366, and JerkGap from 0.745 to 0.716.

The improvements are most evident on highly articulated limbs. For the arms, DyCoPart reduces KFID from 0.286 to 0.228 on the left arm and from 0.442 to 0.346 on the right arm, showing that dynamic part routing better captures fine-grained local articulation. For the legs, DyCoPart reduces KFID from 0.312 to 0.263 on the left leg and from 0.313 to 0.246 on the right leg, while also

Table 5: Quantitative comparison of **part-level** motion quality on HumanML3D. We run each evaluation 20 times and report the average with 95% confidence intervals. Bold text highlights the best scores.

Method	Part	KFID↓	NPSS↓	JerkGap↓	Part	KFID↓	NPSS↓	JerkGap↓
ParCo Zou et al. (2024)	Left Arm	$0.315^{\pm.021}$	$6.918^{\pm.027}$	$0.854^{\pm.008}$	Right Arm	$0.529^{\pm.028}$	$7.043^{\pm.025}$	$0.949^{\pm.008}$
LGTM Sun et al. (2024)		$0.342^{\pm.024}$	$6.874^{\pm.031}$	$0.839^{\pm.009}$		$0.547^{\pm.032}$	$7.000^{\pm.029}$	$0.935^{\pm.009}$
T2M-GPT Zhang et al. (2023a)		$0.471^{\pm.029}$	$7.238^{\pm.036}$	$0.931^{\pm.011}$		$0.752^{\pm.039}$	$7.240^{\pm.033}$	$1.017^{\pm.011}$
MoMask Guo et al. (2024)		$0.286^{\pm.019}$	$6.762^{\pm.024}$	$0.817^{\pm.007}$		$0.442^{\pm.025}$	$6.843^{\pm.022}$	$0.886^{\pm.007}$
Ours		$\mathbf{0.228}^{\pm.016}$	$\mathbf{6.626}^{\pm.018}$	$\mathbf{0.782}^{\pm.004}$		$\mathbf{0.346}^{\pm.021}$	$\mathbf{6.723}^{\pm.016}$	$\mathbf{0.853}^{\pm.004}$
ParCo Zou et al. (2024)	Left Leg	$0.351^{\pm.017}$	$7.386^{\pm.031}$	$0.737^{\pm.006}$	Right Leg	$0.375^{\pm.015}$	$7.247^{\pm.029}$	$0.716^{\pm.005}$
LGTM Sun et al. (2024)		$0.378^{\pm.020}$	$7.314^{\pm.028}$	$0.721^{\pm.007}$		$0.389^{\pm.018}$	$7.203^{\pm.026}$	$0.706^{\pm.006}$
T2M-GPT Zhang et al. (2023a)		$0.502^{\pm.026}$	$7.642^{\pm.041}$	$0.798^{\pm.010}$		$0.535^{\pm.023}$	$7.451^{\pm.038}$	$0.769^{\pm.008}$
MoMask Guo et al. (2024)		$0.312^{\pm.014}$	$7.188^{\pm.026}$	$0.692^{\pm.005}$		$0.313^{\pm.013}$	$7.042^{\pm.024}$	$0.670^{\pm.004}$
Ours		$\mathbf{0.263}^{\pm.010}$	$\mathbf{7.033}^{\pm.019}$	$\mathbf{0.658}^{\pm.004}$		$\mathbf{0.246}^{\pm.009}$	$\mathbf{6.918}^{\pm.018}$	$\mathbf{0.645}^{\pm.004}$
ParCo Zou et al. (2024)	Backbone	$0.087^{\pm.004}$	$8.046^{\pm.022}$	$0.864^{\pm.006}$	Root	$0.011^{\pm.001}$	$9.506^{\pm.023}$	$0.614^{\pm.006}$
LGTM Sun et al. (2024)		$0.079^{\pm.005}$	$8.012^{\pm.020}$	$0.849^{\pm.005}$		$0.011^{\pm.001}$	$9.449^{\pm.021}$	$0.606^{\pm.005}$
T2M-GPT Zhang et al. (2023a)		$0.118^{\pm.007}$	$8.163^{\pm.027}$	$0.902^{\pm.008}$		$0.016^{\pm.002}$	$9.774^{\pm.029}$	$0.659^{\pm.009}$
MoMask Guo et al. (2024)		$0.066^{\pm.003}$	$7.934^{\pm.018}$	$0.831^{\pm.005}$		$0.009^{\pm.001}$	$9.237^{\pm.019}$	$0.574^{\pm.005}$
Ours		$\mathbf{0.054}^{\pm.002}$	$\mathbf{7.819}^{\pm.019}$	$\mathbf{0.802}^{\pm.004}$		$\mathbf{0.007}^{\pm.0005}$	$\mathbf{9.075}^{\pm.021}$	$\mathbf{0.553}^{\pm.004}$

lowering JerkGap from 0.692 to 0.658 and from 0.670 to 0.645, respectively. These results indicate more realistic leg dynamics and smoother contact-related motion. The backbone and root also show consistent gains, suggesting that the improved local expressiveness does not sacrifice central body stability. Overall, the full part-level results further support our design: motion-adaptive dynamic parts improve local motion quality, while directed causal coordination helps preserve coherent whole-body dynamics.

C.2 Joint Evaluation of Local Articulation and Global Coordination

We isolate the contributions of fixed-part modeling (FP), Dynamic Part Learning (DPL), and the Directed Causal Coordination Rectifier (DCCR) using a common Spatial–Temporal Mamba backbone and training protocol. Appendix D.3 specifies the baseline. Table 6 combines holistic and part-level quality with coordination and physical-plausibility metrics, allowing improvements in local articulation to be assessed alongside their effects on the whole body.

Table 6: Same-backbone module ablation on HumanML3D. Each added component is relative to the same baseline. Part-level metrics are averaged over six parts. All entries are point estimates.

(a) Holistic and part-level motion quality

Variant	R Top-3 ↑	FID ↓	KFID ↓	NPSS ↓	JerkGap ↓
Baseline	0.832	0.085	0.334	7.846	0.832
+ FP	0.835	0.069	0.267	7.632	0.779
+ DPL	0.844	0.058	0.218	7.441	0.732
+ FP + DCCR	0.841	0.052	0.258	7.587	0.767
DyCoPart (Full)	**0.848**	**0.041**	**0.191**	**7.366**	**0.716**

(b) Whole-body coordination and physical plausibility

Variant	TC ↑	SC ↑	Penetrate (cm) ↓	Float (cm) ↓	Skate (%) ↓	BLV (cm^2) ↓
Baseline	0.603	0.742	0.116	3.286	4.618	31.274
+ FP	0.433	0.591	0.451	5.516	5.183	47.916
+ DPL	0.614	0.769	0.081	3.037	3.691	23.548
+ FP + DCCR	0.607	0.762	0.094	3.128	3.844	25.763
DyCoPart (Full)	**0.623**	**0.777**	**0.064**	**2.984**	**2.657**	**18.003**

FP improves all three part-level metrics over the baseline but reduces TC from 0.603 to 0.433 and SC from 0.742 to 0.591, while increasing every reported physical-artifact metric. This is the specific local–global tension observed in this controlled setting. DPL lowers KFID from 0.334 to 0.218 while improving both TC and SC. Keeping the fixed-part representation and adding DCCR restores TC/SC to 0.607/0.762 and reduces Skate from 5.183% to 3.844%. Their combination obtains the best

point estimates across the reported metrics. These comparisons identify the modules' incremental effects beyond the strong backbone, without attributing the baseline's absolute performance to DPL or DCCR.

C.3 Cross-Backbone Evaluation on MDM

We integrate the proposed modules into MDM Tevet et al. (2022b) to test their effects on a different backbone. All variants in Table 7 use the same 15-step UniPC sampler and identical evaluation settings. This controlled experiment uses a different sampling budget from the 20-step DyCoPart configuration in Appendix D and from the original 1,000-step MDM sampler. The reproduced MDM row therefore serves as the reference for comparisons within this table.

Table 7: Cross-backbone module evaluation on HumanML3D using a common 15-step UniPC sampler. Part-level metrics are six-part averages. These are point estimates under the controlled protocol described in Appendix C.3.

(a) Holistic and part-level motion quality

Variant	R Top-3 ↑	FID ↓	KFID ↓	NPSS ↓	JerkGap ↓
MDM	0.768	0.289	0.267	7.592	0.773
MDM + DPL	0.796	0.093	0.219	7.422	0.727
MDM + FP + DCCR	0.813	0.078	0.244	7.508	0.748
MDM + DPL + DCCR	**0.837**	**0.046**	**0.203**	**7.389**	**0.721**

(b) Whole-body coordination and physical plausibility

Variant	TC ↑	SC ↑	Penetrate (cm) ↓	Float (cm) ↓	Skate (%) ↓	BLV (cm^2) ↓
MDM	0.592	0.746	0.116	3.386	4.783	30.462
MDM + DPL	0.605	0.766	0.086	3.131	3.824	22.964
MDM + FP + DCCR	0.607	0.764	0.094	3.168	3.706	24.318
MDM + DPL + DCCR	**0.620**	**0.775**	**0.068**	**3.026**	**2.741**	**18.526**

Adding DPL improves R Top-3 from 0.768 to 0.796 and reduces FID from 0.289 to 0.093. Adding DCCR to DPL further improves them to 0.837 and 0.046. Relative to the reproduced MDM baseline, the full variant reduces KFID by 24.0%, Skate by 42.7%, and BLV by 39.2%, while increasing TC/SC from 0.592/0.746 to 0.620/0.775. Together with Appendix C.2, these results support improvements from DPL and DCCR on both evaluated backbones under their respective controlled protocols.

C.4 Matched Depth Analysis

We compare one to five layers of the Spatial–Temporal Mamba baseline and DyCoPart. All depth variants use the same learning rate and training hyperparameters, are trained for up to 300 epochs, and select their best checkpoints by validation performance. Table 8 reports the matched comparisons.

Table 8: Matched depth ablation on HumanML3D. Hyperparameters are shared across depths rather than separately optimized for each depth.

	R Top-3 ↑		FID ↓	
Layers	Baseline	DyCoPart	Baseline	DyCoPart
1	0.814	0.839	0.113	0.059
2	**0.832**	**0.848**	**0.085**	**0.041**
3	0.816	0.841	0.122	0.048
4	0.804	0.843	0.143	0.052
5	0.799	0.836	0.168	0.067

DyCoPart outperforms the matched baseline at every evaluated depth, but both models obtain their best results at two layers. Thus, two layers is the best observed setting under this training setup. The results establish module benefits at the tested depths, while the cause of the shared non-monotonic trend remains unresolved. They do not distinguish possible feature over-smoothing from dataset

scale or training-budget effects, nor establish improved scalability with network depth. Appendix E discusses this boundary alongside the generalization limits.

C.5 Additional Qualitative Results

Fig. 4 presents additional qualitative comparisons under two text prompts involving squat-jump transitions and in-place turning with raised arms. For the squat-and-jump motion, MDM captures the coarse action but produces detached body parts, while ParCo suffers from a collapsed backbone and MoMask shows incorrect joint configurations. In contrast, DyCoPart preserves coordinated joint movement and maintains a natural backbone during both squatting and jumping.

For the turning motion, MDM performs an incomplete turn, ParCo generates stiff limb motions, and MoMask loses fine action details halfway through the sequence. Our method better preserves the raised-arm posture throughout the turn and produces smoother whole-body coordination. These results further show that DyCoPart improves both local limb quality and global body consistency, especially for motions requiring strong inter-part coordination.

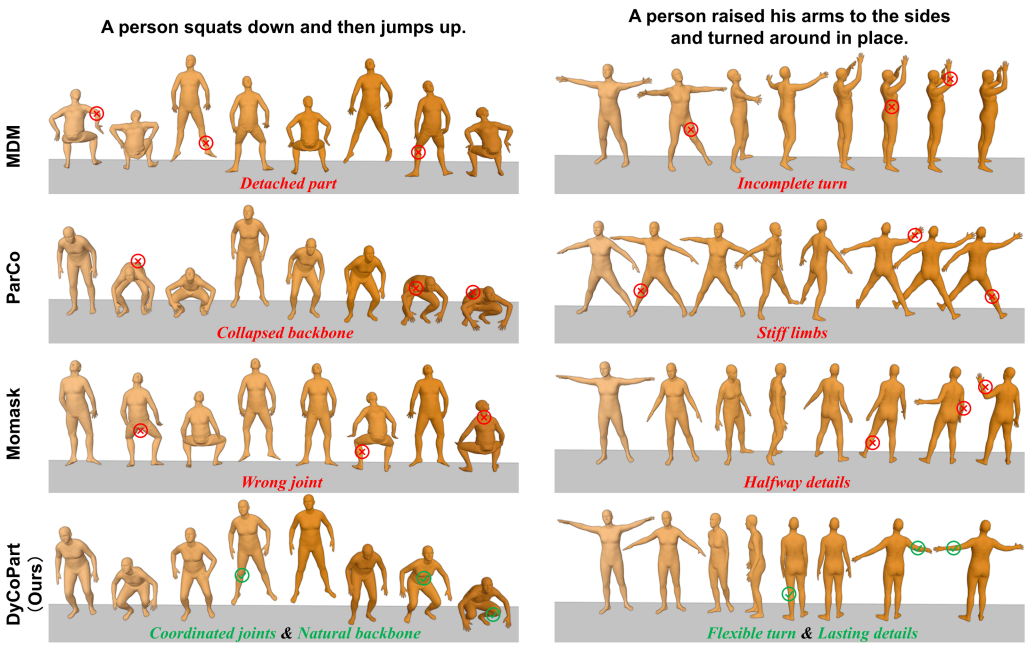

Figure 4: Qualitative comparison on HumanML3D.

C.6 Temporal Changes in Dynamic Part Routing

To complement the raised-arm turning example in Appendix C.5, we inspect the prompt "a person raises both arms to the sides and turns around in place." Table 9 reports routing statistics at frames 50 and 100. Each entry is the normalized sum of the joint-to-slot weights $m_{t,j,p}$ over joints belonging to the indicated anatomical region; "Others" collects the remaining destination slots.

At frame 50, most routing mass remains on the corresponding anatomical slots. At frame 100, the left-arm weight on Backbone increases from 0.21 to 0.29, and the left-leg weight on Root increases from 0.16 to 0.23. In this example, the same anatomical anchors support a changing distribution of joint contributions as the motion enters active turning. The statistics illustrate motion-dependent routing while retaining the semantic identity of the dominant slots.

Table 9: Routing changes during raised-arm turning. For each region and frame, the three destination weights sum to one.

		Aggregated routing weights		
Frame	Motion phase	Own slot	Central slot	Others
Left-arm region: own slot = L_Arm; central slot = Backbone				
50	Arms raised / turn begins	0.71	0.21	0.08
100	Active turning / foot pivot	0.55	0.29	0.16
Left-leg region: own slot = L_Leg; central slot = Root				
50	Arms raised / turn begins	0.77	0.16	0.07
100	Active turning / foot pivot	0.63	0.23	0.14

D Implementation Details

We implemented DyCoPart with a frozen CLIP text encoder with condition dropout probability 0.1. The diffusion process follows DDPM Ho et al. (2020) with 1000 training steps, a linear beta schedule from 1e-4 to 0.02. During inference, we adopt the UniPC Zhao et al. (2023) multistep sampler with solver order 2 and 20 denoising steps, together with classifier free guidance using a guidance scale of 5. The model is optimized by AdamW Loshchilov & Hutter (2017) with a learning rate of 1e-4, batch size 128. We train the model for 300 epochs on HumanML3D and 1000 epochs on KIT using two NVIDIA RTX 4090 GPUs, with an average training time of about 11 hours. The final model contains 17.6M parameters and average inference time is 121 ms.

D.1 Datasets

We evaluate our method on two widely used text-to-motion benchmarks, **HumanML3D** Guo et al. (2022a) and **KIT-ML** Plappert et al. (2016). HumanML3D contains 14,616 motion clips and 44,970 textual descriptions, covering diverse daily actions, sports, and locomotion patterns. KIT-ML includes 3,911 motion sequences and 6,278 language annotations collected from multiple motion capture sources.

D.2 Routing Notation and Anatomical Anchoring

The six semantic slots used in Dynamic Part Learning and the six nodes of DCCR refer to the same set of body regions, so the slot counts denoted by P and S in the main text satisfy $P = S = 6$. The anatomical distance is the minimum shortest-path distance from joint j to the anchor set $\mathcal{A}_p$ on the kinematic graph:

$$d_{\mathcal{G}}(j, \mathcal{A}_p) = \min_{a \in \mathcal{A}_p} \mathrm{dist}_{\mathcal{G}}(j, a). \tag{57}$$

The GNN adjacency and anatomical routing prior serve complementary roles. The adjacency supplies a kinematic-connectivity prior for propagating joint information along skeletal edges. The routing prior anchors each slot to a coarse semantic region, such as an arm or a leg. Skeletal connectivity therefore informs the learned features, while the anchor sets fix the semantic meaning of the slots. The final routing weights combine these features with the slot prior and vary with the motion state.

The token notation tracks three stages: $\mathbf{z}_{t,p}$ denotes the aggregated joint features in Dynamic Part Learning, $\tilde{\mathbf{z}}_{t,p}$ denotes their Part-SSM-enhanced representation, and $\mathbf{z}^{\mathrm{vg}}_{t,p}$ denotes the DCCR-rectified token. The DCCR subsection locally abbreviates its Part-SSM-enhanced input as $\mathbf{z}_{t,s}$; its historical source $\bar{\mathbf{z}}_{t,s}$ is this enhanced token shifted to the previous frame. Similarly, $e_{t,j,p}$ is a joint-to-slot routing logit, whereas $e_{t,s,r}$ is a directed slot-to-slot compatibility logit. Their indices identify the distinct operations.

The whole-body condition $\mathbf{c}_t$ supplies context from the whole-motion branch, and the phase feature $\mathbf{p}_t$ encodes the root state and foot-contact features. The maps Ψ_c and Ψ_p convert these conditions into $P \times P$ edge-modulation matrices; the gate projections modulate the target update. These contextual signals complement the part tokens used by DCCR and the training regularizers detailed in Appendix A.

D.3 Baseline Architecture and Experimental Controls

The baseline in Appendix C.2 is a text-conditioned diffusion denoiser with two stacked Spatial–Temporal Mamba blocks. Spatial Mamba models inter-joint dependencies within each frame, and Temporal Mamba models motion evolution across frames. It shares the full model's text encoder, diffusion objective, training protocol, sampler, and guidance setting. It excludes FP, DPL, Part-SSM, DCCR, part perturbations, and coordination regularization. The FP, DPL, and DCCR variants in Table 6 isolate the added modules under these shared controls. The separate MDM experiment uses the common 15-step sampling protocol specified in Appendix C.3.

D.4 Computational Overhead

Table 10 separates model parameters, denoiser computation, inference latency, and training cost for the same-backbone variants. The full model adds 0.41M parameters (2.4%), 5.94% GFLOPs, and 5.9% inference latency over the baseline. Training wall-clock time increases from 9.44 to 11.00 hours on two RTX 4090 GPUs, corresponding to 18.88 and 22.00 GPU-hours, respectively.

Table 10: Computational overhead of fixed and learned part designs. GFLOPs are per denoiser evaluation; inference latency and total training cost are separate measurements. Training uses two RTX 4090 GPUs.

Variant	Params (M)	GFLOPs / denoiser	Inference (ms)	Training (h)	GPU-hours
Baseline	17.22	11.818	114.3	9.44	18.88
+ FP	17.25	11.928	116.5	9.50	19.00
+ DPL	17.34	12.216	118.1	9.76	19.52
+ FP + DCCR	17.50	12.122	117.9	10.65	21.30
DyCoPart (Full)	17.63	12.520	121.0	11.00	22.00

For fixed feature widths, the additional joint-to-slot routing and cross-part coordination operations scale as $O(TJP)$ and $O(TP^2)$, respectively, where T is sequence length, J is joint count, and P is the number of part slots. With $P = 6$ and a fixed skeleton, these operations scale linearly with sequence length. The six coarse anatomical slots are shared across motions; motion-dependent routing and directed dependencies are learned without action-specific manual rules. The measured overhead and MDM transfer support practical integration across the evaluated backbones. Depth scaling is assessed separately in Appendix C.4.

E Limitations and Future Work

Although DyCoPart achieves strong performance in text to motion generation, it still has two limitations. First, our experiments are conducted on HumanML3D and KIT-ML, which are widely used benchmarks but mainly focus on short single-person motions. As a result, the current results do not fully demonstrate the generalization ability of DyCoPart in more complex scenarios, such as long-horizon motion generation, human-object interaction, scene-constrained motion, and multi-person coordination. Future work will extend dynamic body-part coordination to these challenging settings, where generated motions need to satisfy richer semantic, spatial, and physical constraints.

The matched-depth results in Appendix C.4 further bound the current evaluation: two layers is the best observed setting under a shared training configuration, rather than a universally optimized depth. Depth-specific learning rates, schedules, and training budgets were not explored. The present evidence does not establish the cause of the degradation beyond two layers or improved depth scalability.

Second, DyCoPart learns motion adaptive body parts from six coarse semantic part slots under anatomical prior guidance. This design provides greater flexibility than fixed part partitions, but the number and semantic granularity of part slots are still manually specified. Such a formulation may be suboptimal for actions that require finer hand, foot, head, or torso control, or for motions whose coordination structures vary significantly across action types. In future work, we will explore hierarchical and data-driven part discovery, enabling the model to adaptively determine the number, scale, and organization of body parts according to the input motion and text.

The anatomical anchors also depend on the skeleton definition. Applying the model to a substantially different skeleton may require remapping these anchors, even though the learned routing remains motion-dependent.