PARTIGEN: PART-AWARE REPRESENTATIONS FOR LANGUAGE-CONDITIONED HUMANOID MOTION GENERATION

Anonymous authors
Paper under double-blind review

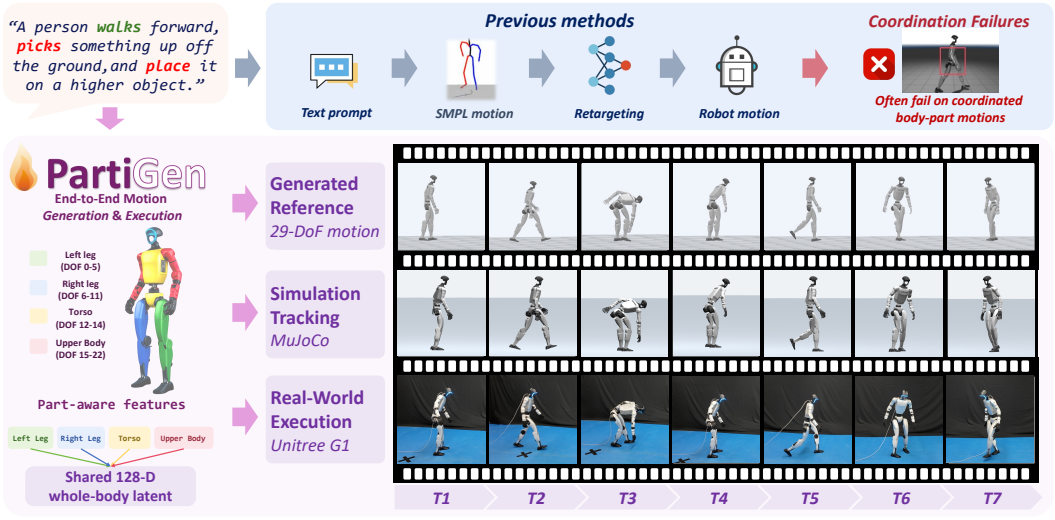

Figure 1: **The PartiGen system: from language to motion and execution.** Our motion generator produces references; the execution component uses a pretrained, frozen GRIT controller (mrzuang, 2026) for tracking in MuJoCo and on the Unitree G1. Hardware examples are selected after simulation screening.

ABSTRACT

Translating language into humanoid motion requires coordinating body parts, determining motion duration, and producing references suitable for robot execution. However, existing text-to-motion methods have limitations in whole-body coordination and duration adaptation, while the physical constraints of humanoid robots are not always explicitly considered. We introduce **PartiGen**, a part-aware latent diffusion framework that incorporates anatomical structure into a shared whole-body motion representation. Part-specific attention processes anatomically masked frame features, and its outputs are fused to combine local limb information with whole-body context. Conditioned on text and motion history, the model generates motion primitives autoregressively. Forward-kinematics supervision encourages kinematic consistency, while a learned end-of-sequence predictor selects a frame-level endpoint after generation. On HumanML3D retargeted to the Unitree G1, PartiGen reduces segment-level FID by 16.8% over TextOp and improves duration accuracy over a fixed-length cutoff. With a pretrained, frozen tracking controller, we evaluate generated references in MuJoCo and demonstrate execution of selected motions on the real G1 after simulation screening and manual selection. Code: `https://anonymous.4open.science/r/PartiGen-review/`.

Keywords: Motion Generation, Embodied AI, Humanoid Robotics, Structured Representations

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

Natural language offers an intuitive interface for specifying whole-body humanoid behaviors. Related research on humanoid behavior spans motion retargeting (Araujo et al., 2025), whole-body teleoperation (Ze et al., 2025), and motion-skill learning (Cao et al., 2025). Text-to-motion generation (Guo et al., 2022a; Tevet et al., 2022; Guo et al., 2024) supports their synthesis, but visually plausible motion may violate robot joint limits, dynamics, actuator capabilities, or contact constraints. End-effector tracking (Cao et al., 2026) addresses the complementary task of following motion references on humanoid robots. We study a system in which a language-conditioned generator specifies a whole-body reference and its temporal extent, while a tracking controller uses robot-state feedback to produce joint-level control targets. The challenge is to generate references that express the requested behavior while accounting for the robot's motion constraints and the task's temporal organization.

An instruction such as "walk while waving" illustrates the challenge: the arm gesture and locomotion must coexist within a single robot trajectory, with consistent root motion, joint configurations, and timing. A reference generator therefore needs to preserve the distinct roles of body parts while representing their relationships within a shared whole-body context. Part-aware modeling (Zhong et al., 2023) and physics-guided generation (Yuan et al., 2023) provide relevant foundations for addressing these requirements. We investigate how an anatomical prior can guide local feature processing within a shared robot-motion representation, together with kinematic supervision of the resulting trajectory. Its temporal extent must also reflect the instruction, with an endpoint that need not coincide with the boundary of a fixed-length motion primitive. These considerations connect body-part structure, kinematic consistency, and endpoint selection within the same robot reference-generation problem.

We propose **PartiGen**, integrating language-conditioned humanoid motion generation and execution. Part-aware attention builds a shared whole-body representation from anatomically masked inputs. Text- and history-conditioned latent diffusion generates motion primitives under forward-kinematics supervision, followed by frame-level endpoint selection on the complete candidate trajectory. A pretrained, frozen GRIT controller (mrzuang, 2026) tracks the references using robot-state feedback in MuJoCo and on the Unitree G1.

Our contributions are:

1. **An anatomical input-grouping mechanism for robot motion representation learning.** Fixed masks select joint-angle channels before independent Q/K/V projections, while shared motion features and a single whole-body latent retain access to global context.
2. **A language-conditioned motion generation framework with frame-level endpoint prediction.** Conditioned on text and motion history, the framework autoregressively generates motion primitives in the shared latent space under forward-kinematics supervision. A learned end-of-sequence predictor then selects an endpoint within the complete candidate trajectory.
3. **Simulation and real-world validation.** Integrated with a frozen GRIT controller (Figure 1), PartiGen achieves an 81.8% posture-screen pass rate over 500 simulated references. Following simulation screening and manual selection, 26 of 27 motions attempted on a physical Unitree G1 are judged successful by the authors.

2 RELATED WORK

Text-to-motion generation. Large-scale motion datasets (Guo et al., 2022a; Mahmood et al., 2019; Punnakkal et al., 2021) provide a foundation for data-driven generation. Common modeling paradigms include variational motion models (Guo et al., 2020; Petrovich et al., 2021; 2022), diffusion-based generation (Tevet et al., 2022; Zhang et al., 2022; 2023), and discrete motion tokenization (Guo et al., 2022b; Jiang et al., 2023). Motion representation learning (Zhu et al., 2023) and controllable diffusion priors (Karunratanakul et al., 2023; Shafir et al., 2023) offer complementary approaches to encoding and manipulating motion. DART and TextOp (Zhao et al., 2025; Xie et al., 2026) further address autoregressive generation and interactive control. Physics-guided approaches address physical plausibility in human motion and human-object interaction genera-

tion (Yuan et al., 2023; Xu et al., 2023). Robot-oriented work further explores whole-body motion generation from language descriptions (Jiang et al., 2024) and language-conditioned humanoid control through motion latent guidance (Li et al., 2026). Building on autoregressive primitive generation, PartiGen incorporates anatomical input masks and EOS-based endpoint prediction.

Structure-aware modeling. Part-aware attention (Zhong et al., 2023) and skeleton-aware latent diffusion (Hong et al., 2025) incorporate anatomical structure into motion representations. Hierarchical models address fine-grained semantic control (Jin et al., 2023) and human-object interaction (Pi et al., 2023), while cross-skeleton models accommodate varying skeletal structures (Khani et al., 2025; Xu et al., 2026; Gat et al., 2025). Skeleton conditioning has also been explored for human image generation (Ju et al., 2023). PartiGen places its anatomical prior at the attention input through fixed masks on raw frame features before part-specific Q/K/V projections. Each head receives its selected joint angles together with shared root/contact features and all joint-angle increments. Head outputs are fused into a shared stream, retaining a single motion latent rather than independent part latents.

Duration-Aware Motion Generation. Sequence continuation and termination are distinct design choices. Related temporal generation frameworks include latent audio diffusion (Liu et al., 2023), neural-codec speech modeling (Wang et al., 2023), and autoregressive video diffusion (Henschel et al., 2025). For motion, PartiGen selects an endpoint at frame resolution. It generates a complete candidate sequence before applying an EOS probability threshold to select its endpoint; this selection does not stop generation online.

3 METHOD

3.1 PROBLEM FORMULATION

Given a natural-language instruction $\mathbf{c}_{\text{text}}$ and a motion history $\mathbf{H} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_H}\}$, we aim to generate a future whole-body robot motion $\mathbf{F} = \{\mathbf{x}_{n_H+1}, \ldots, \mathbf{x}_{n_H+n_F}\}$ and determine its length n_F. The generated motion should express the requested behavior, continue coherently from the observed history, and maintain kinematic consistency across body parts.

We represent each motion frame using root motion, foot-contact states, joint configurations, and their inter-frame changes:

$$\mathbf{x}_t = \left[\mathbf{o}_t,\, \mathbf{c}_t,\, \Delta\mathbf{r}_t^{\text{local}},\, h_t,\, \mathbf{q}_t,\, \Delta\mathbf{q}_t\right] \in \mathbb{R}^{d_x}, \tag{1}$$

where d_x denotes the motion feature dimension. Here, $\mathbf{o}_t$ encodes root orientation, $\mathbf{c}_t$ contains foot-contact indicators, $\Delta\mathbf{r}_t^{\text{local}}$ is the root displacement in the local frame, and h_t is the root height. The joint angles $\mathbf{q}_t$ describe the articulated pose, while $\Delta\mathbf{q}_t$ captures its change between adjacent frames. Root displacement and joint-angle increments are represented without time normalization.

This representation combines whole-body context with local joint information for subsequent structure-aware motion modeling. Feature encoding conventions and the complete channel layout are provided in Appendix S2.

3.2 SYSTEM OVERVIEW: GENERATION AND EXECUTION

PartiGen comprises two complementary components: motion generation, which synthesizes whole-body reference trajectories from language instructions, and motion execution, which tracks these references through closed-loop robot control.

Motion generation. The generator is trained in two stages (Figure 2). BP-MVAE maps motion primitives into a compact shared latent space and learns to reconstruct whole-body motion from latent codes and motion history. DA-LDM then performs diffusion-based generation in this space, conditioned on language and motion history (Rombach et al., 2022). The compact representation reduces the dimensionality of the denoising problem, while text conditioning guides generation toward the requested behavior. At inference, latent codes are sampled autoregressively and decoded into a complete candidate reference trajectory, after which EOS probabilities select its endpoint.

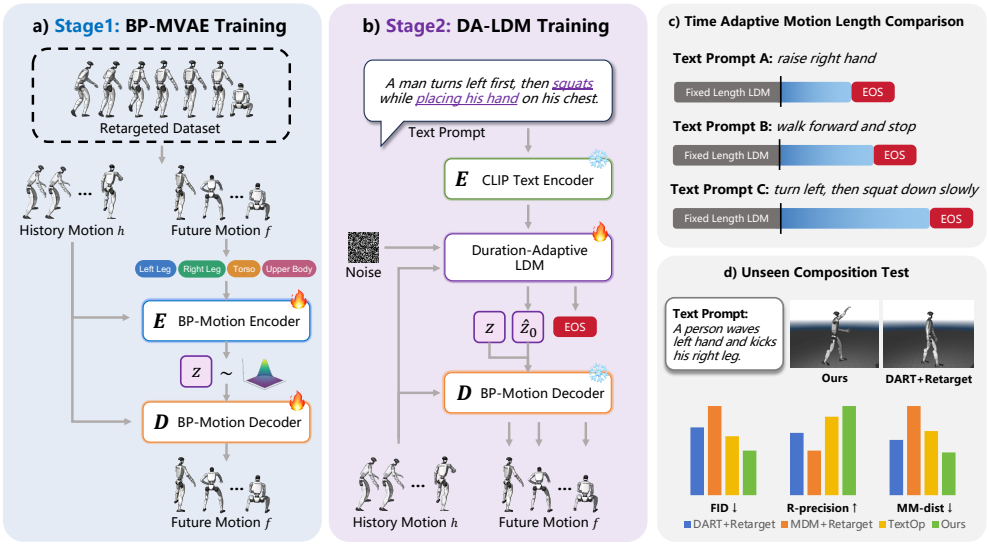

Figure 2: **Architecture and training of the PartiGen motion generator.** (a) BP-MVAE fuses part-aware features into a shared 128-dimensional latent for whole-body reconstruction. (b) Following latent diffusion (Rombach et al., 2022), DA-LDM predicts the clean motion latent (x_0) from a noisy latent, conditioned on CLIP text embeddings (Radford et al., 2021), motion history, and diffusion timestep. The decoder reconstructs future motion from the predicted latent. (c) EOS-based endpoint selection reduces duration error compared with a fixed-length cutoff. (d) PartiGen achieves better text-motion alignment and lower FID than previous methods on the diagnostic composition test.

Motion execution. For execution, the generated trajectories are mapped from 23 DoF at 30 Hz to 29 DoF at 50 Hz, with an interpolated transition into the first reference frame. A pretrained, frozen GRIT controller (mrzuang, 2026) tracks these references using robot-state feedback to produce joint-level control targets. Feedback remains within the tracking loop; generator training uses neither controller gradients nor a joint generation-control objective. Deployment details are given in Appendix S6.

3.3 Body-Part-Aware Motion VAE (BP-MVAE)

BP-MVAE learns a shared latent representation of whole-body motion through part-aware feature processing and kinematically supervised reconstruction. It combines local joint information with whole-body context to capture motion patterns within body regions and dependencies across them.

Part-aware feature encoding. Anatomical structure is introduced through fixed feature masks applied before attention projections. For a motion primitive $\mathbf{X}$, the mask $\mathbf{m}_k$ retains the joint-angle channels of body region k, together with root-motion features, foot-contact states, and all joint-angle increments. The mask is broadcast across frames:

$$\mathbf{X}_k = \mathbf{X} \odot \mathbf{m}_k. \tag{2}$$

Each branch uses independent query, key, and value projections:

$$\mathbf{P}_k = \mathrm{Linear}_k^P(\mathbf{X}_k), \qquad P \in \{Q, K, V\}. \tag{3}$$

Temporal attention then aggregates information across motion frames:

$$\mathbf{O}_k = \mathrm{softmax}\left(\frac{\mathbf{Q}_k \mathbf{K}_k^\top}{\sqrt{d_h}}\right)\mathbf{V}_k, \tag{4}$$

where d_h is the projection dimension. The branch outputs are concatenated and projected into the shared Transformer stream:

$$\mathbf{O} = \mathrm{Linear}_{\mathrm{out}}(\mathrm{Concat}_k(\mathbf{O}_k)). \tag{5}$$

Shared input channels and output fusion provide access to whole-body information while incorporating anatomically selected joint features.

Shared motion latent. The encoder represents each motion primitive using a shared latent vector $\mathbf{z} \in \mathbb{R}^{d_z}$, where d_z denotes the latent dimension. Root and joint motion are encoded jointly, with the latent coordinates shared across all body regions. Conditioned on motion history, the decoder reconstructs the complete motion primitive from this code. This shared representation also provides the motion space used by the subsequent diffusion model.

Kinematic regularization. Errors in joint configurations can propagate along kinematic chains and produce spatial deviations that feature-space reconstruction losses do not directly capture. We therefore supervise reconstructed motion through the robot's forward kinematics and apply temporal smoothness regularization. The kinematic objective is summarized as

$$\mathcal{L}_{\text{kin}} = \rho(\Phi(\hat{\mathbf{x}}) - \Phi(\mathbf{x})) + \rho\big(\Delta^2 \hat{\mathbf{q}}\big)\,, \tag{6}$$

where Φ maps motion features to kinematic quantities, including joint positions and rotations, Δ^2 denotes the second-order temporal difference, and ρ is the Huber loss. The overall training objective combines motion reconstruction, latent regularization, and kinematic supervision:

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{rec}} + \beta\,\mathcal{L}_{\text{KL}} + \lambda_{\text{kin}}\,\mathcal{L}_{\text{kin}}, \tag{7}$$

where $\mathcal{L}_{\text{rec}}$ measures motion reconstruction error, $\mathcal{L}_{\text{KL}}$ regularizes the latent distribution, and β and λ_{kin} balance the corresponding terms. Architectural configurations and detailed training settings are provided in Appendix S2.

3.4 Duration-Adaptive Latent Motion Diffusion (DA-LDM)

DA-LDM applies latent diffusion (Rombach et al., 2022) in the shared motion space learned by BP-MVAE. Conditioned on language and motion history, it generates successive motion codes that are decoded into a whole-body reference trajectory. Frame-level endpoint selection is applied after complete candidate generation.

Shared-Latent Denoiser Architecture. We use a Transformer denoiser with direct clean-latent prediction. Let $\mathbf{x}_0$ denote a clean BP-MVAE motion code and $\mathbf{z}_\tau$ its noisy version at diffusion timestep τ. The noisy code is projected into the hidden space and combined with a timestep embedding:

$$\mathbf{u}_\tau = \mathbf{W}_z \mathbf{z}_\tau + \mathbf{b}_z + \mathbf{E}_{\text{time}}(\tau). \tag{8}$$

Here, $\mathbf{W}_z$ and $\mathbf{b}_z$ are learned projection parameters, and $\mathbf{E}_{\text{time}}$ encodes the diffusion timestep. The denoiser processes a shared motion code that jointly represents the whole body.

Multi-Modal Conditioning. Motion history conditions the denoiser through cross-attention, while CLIP text features (Radford et al., 2021) convey the requested behavior. We use classifier-free guidance (Ho & Salimans, 2022), randomly replacing the text condition with zero during training while retaining motion history. During sampling, predictions with the text condition present and zeroed are combined as

$$\begin{aligned}
\hat{\mathbf{x}}_0^{\text{text}} &= f_\theta\big(\mathbf{z}_\tau, \tau, \mathbf{c}_{\text{text}}, \mathbf{H}\big), \\
\hat{\mathbf{x}}_0^{\text{zero}} &= f_\theta\big(\mathbf{z}_\tau, \tau, \mathbf{0}, \mathbf{H}\big), \\
\hat{\mathbf{x}}_0^{\text{CFG}} &= \hat{\mathbf{x}}_0^{\text{zero}} + s\,\big(\hat{\mathbf{x}}_0^{\text{text}} - \hat{\mathbf{x}}_0^{\text{zero}}\big)\,,
\end{aligned} \tag{9}$$

where s controls guidance strength. Both predictions use the same noisy latent, timestep, and motion history.

Clean-Latent Training and Sampling. Training uses ground-truth motion history and uniformly sampled diffusion timesteps. The forward process perturbs the clean motion code with Gaussian noise:

$$\mathbf{z}_\tau = \sqrt{\bar{\alpha}_\tau}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_\tau}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{10}$$

where $\bar{\alpha}_\tau$ is the cumulative signal coefficient of the noise schedule. The predicted clean code is decoded into a future motion primitive:

$$\hat{\mathbf{f}} = D\big(\hat{\mathbf{x}}_0, \mathbf{H}\big). \tag{11}$$

The motion objective combines latent reconstruction, decoded motion reconstruction, and geometric supervision:

$$\mathcal{L}_{\text{motion}} = \mathbb{E}\Big[w_\tau\,\Big(\lambda_z \rho\big(\hat{\mathbf{x}}_0 - \mathbf{x}_0\big) + \lambda_f \rho\big(\hat{\mathbf{f}} - \mathbf{f}\big) + \lambda_g \mathcal{L}_{\text{geom}}\Big)\Big]\,. \tag{12}$$

Here, $\mathbf{f}$ is the target future primitive, ρ denotes the mean Huber loss, and w_τ weights the sampled diffusion timestep. The coefficients λ_z, λ_f, and λ_g balance the objective terms. Geometric supervision aggregates weighted kinematic reconstruction errors and trajectory-drift penalties evaluated from the decoded motion after denormalization.

Sampling starts from Gaussian noise and follows DDPM reverse updates using the guided clean-latent prediction:

$$\mathbf{z}_{\tau-1} = \tilde{\boldsymbol{\mu}}_\tau\big(\mathbf{z}_\tau, \hat{\mathbf{x}}_0^{\mathrm{CFG}}\big) + \sqrt{\tilde{\beta}_\tau}\boldsymbol{\eta}, \qquad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{13}$$

where $\tilde{\boldsymbol{\mu}}_\tau$ and $\tilde{\beta}_\tau$ are the forward-process posterior mean and variance. The final reverse step returns the posterior mean without additional noise, and the resulting code is decoded into future motion.

Duration Prediction. A requested behavior may end within a fixed-length motion primitive. To model its temporal extent, DA-LDM incorporates an end-of-sequence (EOS) head that predicts a frame-level termination probability. Implemented as an MLP, this head complements clean-latent prediction by identifying an endpoint for the generated reference. EOS supervision uses positive-weighted binary cross-entropy:

$$\mathcal{L}_{\mathrm{EOS}} = -\frac{1}{|\mathcal{V}|}\sum_{t\in\mathcal{V}}\left[\alpha y_t \log \hat{s}_t + (1-y_t)\log(1-\hat{s}_t)\right], \tag{14}$$

where $\mathcal{V}$ is the set of valid frame targets, y_t is the binary EOS label, $\hat{s}_t$ is the predicted probability, and α weights the positive class. The overall DA-LDM objective combines motion generation and EOS prediction:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{motion}} + \lambda_{\mathrm{EOS}}\mathcal{L}_{\mathrm{EOS}}, \tag{15}$$

where λ_{EOS} controls the contribution of EOS supervision.

At inference, DA-LDM first generates a complete candidate trajectory of T_{cand} frames. Given an EOS threshold γ, the retained motion length is

$$T_{\mathrm{out}} = \begin{cases} \min\left\{t \in \{1,\dots,T_{\mathrm{cand}}\} : \hat{s}_t > \gamma\right\}, & \text{if a threshold crossing exists,} \\ T_{\mathrm{cand}}, & \text{otherwise.} \end{cases} \tag{16}$$

The output retains frames through T_{out}, including the threshold-crossing frame. Its endpoint can therefore lie within a primitive, allowing the final duration to vary at frame resolution. Endpoint selection is performed after complete candidate generation. Architectural configurations, detailed loss terms, and sampling and endpoint-selection settings are provided in Appendix S2.

4 Experiments

We first describe the datasets, baselines, and evaluation metrics. We then evaluate reference motion quality on HumanML3D-G1, followed by a compositional stress test of the ability to follow multi-part instructions. Next, we assess closed-loop tracking in simulation and demonstrate selected motions on the physical G1 using a frozen controller. Finally, ablation studies examine the contributions of anatomical grouping and part-aware modeling, while the duration study evaluates EOS-based endpoint selection. "Ours" and "Full" denote the PartiGen generator. Bold values indicate the best results among the evaluated methods; underlining marks the second-best results in Table 1.

4.1 Experimental Setup

4.1.1 Datasets and Protocols

We evaluate on HumanML3D-G1, obtained by retargeting HumanML3D motions to the Unitree G1 using General Motion Retargeting (GMR) (Araujo et al., 2025). The generator and G1-specific text–motion evaluator share a motion-disjoint training and test split. All methods use the same frozen evaluator and preprocessing, with FID and Diversity computed over the complete test set. A separate compositional test pairs multi-part instructions with approximate references assembled from part-level trajectories. Execution is evaluated in MuJoCo using the pretrained, frozen GRIT controller, followed by selected physical G1 demonstrations after simulation screening and manual review.

4.1.2 BASELINES AND VARIANTS

TextOp provides a baseline trained in the G1 motion domain. DART (Zhao et al., 2025), MDM (Tevet et al., 2022), and AttT2M (Zhong et al., 2023) generate human motions subsequently retargeted through GMR; these comparisons assess the complete generation-and-retargeting pipelines. Ground-truth references (GT) provide an additional execution comparison.

Full denotes the complete PartiGen model. Random-mask replaces anatomical assignments with random joint groups while preserving group sizes and raw-feature projections. The w/o Part Attention variant replaces the part-aware module with standard attention over shared frame embeddings, while w/o FK removes forward-kinematics supervision. For duration evaluation, Fixed-Length applies a dataset-mean cutoff after generation, while Oracle Length supplies annotated lengths during generation as a reference-length comparator. Data and optimization settings are held fixed within each ablation, and duration variants share the same primitive length.

4.1.3 EVALUATION METRICS

Reference quality is measured before tracking. R-Precision and multimodal distance assess text–motion alignment (Guo et al., 2022a). FID and Diversity characterize motion feature distributions at segment and transition levels, with Diversity evaluated by proximity to the reference value. Position Jerk measures temporal smoothness, and Foot Contact Accuracy measures agreement with reference contact states derived through forward kinematics. Duration Error is the absolute difference between predicted and annotated sequence lengths.

Execution is assessed using Global and root-relative MPJPE against the supplied trajectories; root-relative evaluation removes pelvis translation. Mean and median errors include all valid runs, including unstable executions. The posture-screen pass rate measures compliance with pelvis-height and body-inclination criteria over all requested trials. Hardware completion is assessed by the authors through instruction fulfillment and consistency with simulation. Detailed implementation settings and statistical protocols are provided in the supplementary material.

4.2 QUANTITATIVE EVALUATION ON HUMANML3D-G1

PartiGen leads most segment-level retrieval and distributional comparisons, achieving the lowest FID and multimodal distance while ranking among the top two methods across all retrieval metrics (Table 1). For motion transitions, it achieves diversity closest to the reference dataset and ranks second in both FID and jerk. Together, these results demonstrate a favorable balance of text–motion alignment, distributional quality, and temporal smoothness.

Figure 3 illustrates these capabilities in generated references. For "walk while waving," PartiGen expresses the arm gesture more clearly while maintaining locomotion. For the sequential throwing-and-walking instruction, EOS selects a frame-level endpoint after candidate generation, allowing the reference to end within a motion primitive. These examples illustrate the expression of concurrent actions and flexible endpoint selection.

Table 1: HumanML3D-G1 reference quality: segment (top) and transition (bottom).

Method	R@1 ↑	R@2 ↑	R@3 ↑	FID ↓	Div →	mm-dist ↓
Dataset	0.391±0.004	0.475±0.005	0.703±0.009	0±0.000	8.157±0.0035	3.814±0.013
DART+GMR	0.230±0.007	0.372±0.020	0.486±0.018	4.837±0.103	**8.012**±0.069	5.746±0.026
MDM+GMR	0.186±0.004	0.357±0.017	0.473±0.018	8.038±0.082	7.883±0.046	6.025±0.014
AttT2M+GMR	0.157±0.003	0.302±0.009	0.438±0.013	6.712±0.135	8.323±0.071	5.561±0.023
TextOp	0.287±0.005	**0.398**±0.009	0.587±0.012	3.802±0.145	8.764±0.087	4.464±0.059
Ours	**0.301**±0.007	0.386±0.014	**0.611**±0.015	**3.165**±0.068	7.736±0.129	**4.308**±0.086

Method	FID ↓	Div →	PJ ↓
Dataset	0±0.000	6.223±0.024	0.005±0.000
DART+GMR	**3.925**±0.114	6.360±0.056	**0.009**±0.000
MDM+GMR	6.273±0.181	5.991±0.042	0.612±0.013
AttT2M+GMR	6.633±0.102	7.017±0.135	0.357±0.004
TextOp	4.273±0.125	6.772±0.073	0.023±0.001
Ours	3.982±0.103	**6.266**±0.052	0.019±0.001

Table 2: Generated-reference results on the compositional stress test (100 multi-part instructions).

Method	R@1 ↑	R@2 ↑	R@3 ↑	FID ↓	Div →
Compositional Dataset	0.331±0.006	0.439±0.009	0.634±0.013	0.000±0.000	6.102±0.127
DART+GMR	0.102±0.005	0.148±0.012	0.220±0.009	9.340±0.125	15.120±0.392
TextOp	0.129±0.007	0.143±0.018	0.241±0.011	9.951±0.174	9.121±0.298
Ours	**0.211**±0.008	**0.297**±0.016	**0.413**±0.021	**6.531**±0.119	**8.693**±0.167

4.3 Compositional Stress Test

When instructions combine actions across body regions, PartiGen ranks first across all reported metrics (Table 2). Its stronger retrieval performance indicates better alignment with the combined instructions, while its motion diversity and feature distribution more closely match the synthesized references. These gains extend the reference-generation improvements observed on HumanML3D-G1 to compositional instructions that require multiple body-part actions to be expressed together.

Locomotion combined with large-amplitude arm gestures remains challenging across methods. The approximate reference trajectories make this a diagnostic evaluation of semantic alignment and distributional agreement, limiting conclusions about coordinated whole-body behavior.

4.4 System Evaluation: Closed-Loop Execution

Under the frozen GRIT controller, PartiGen achieves the lowest Global MPJPE and the second-highest posture-screen pass rate in Table 3. Its root-relative tracking errors remain close to those of GT and Random-mask. Correcting initial translation offsets narrows the global-error gap to GT, indicating that this advantage partly reflects initialization.

Table 3: Frozen-GRIT tracking in MuJoCo. MPJPE: mean / median (mm), including valid unstable runs. Stability proxy: passes / requests.

Method	Requests	Valid	Excluded	Stability passes	Stability rate (↑)	Global MPJPE (mm, ↓)	Root-relative MPJPE (mm, ↓)
PartiGen (Ours)	500	496	4	409	81.8%	**252.402 / 108.871**	57.995 / 39.765
GT	500	499	1	427	**85.4%**	399.78 / 183.22	**56.24 / 37.94**
DART+GMR	500	494	6	395	79.0%	288.35 / 241.03	58.32 / 40.07
TextOp	500	492	8	396	79.2%	295.85 / 259.38	59.24 / 41.08
Random-mask	500	494	6	401	80.2%	283.90 / 157.29	56.38 / 38.25

The FK ablation shows lower global and root-relative tracking errors with kinematic supervision, despite a lower posture-screen pass rate. This supports the role of kinematic supervision in producing references that the controller can follow more accurately, while highlighting the need to assess tracking alongside posture stability.

The hardware demonstrations illustrate execution of generated references on the physical G1 (Figure 4). Most attempted motions in the screened and manually selected set were judged successful by the authors, fulfilling the instructions with behavior consistent with simulation. Detailed execution analyses and hardware outcomes are provided in Appendix S6.

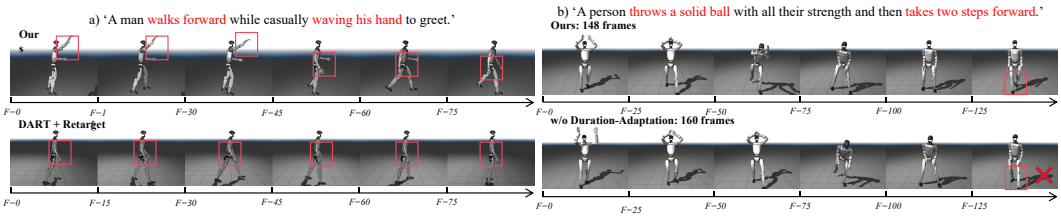

Figure 3: **Qualitative comparison.** (a) For "walk while waving," the illustrated DART output shows a less pronounced arm gesture than PartiGen. (b) For "throw then take two steps," PartiGen produces 148 frames after EOS-based endpoint selection, while the variant without duration adaptation uses a fixed length of 160 frames. These examples illustrate generated reference motions.

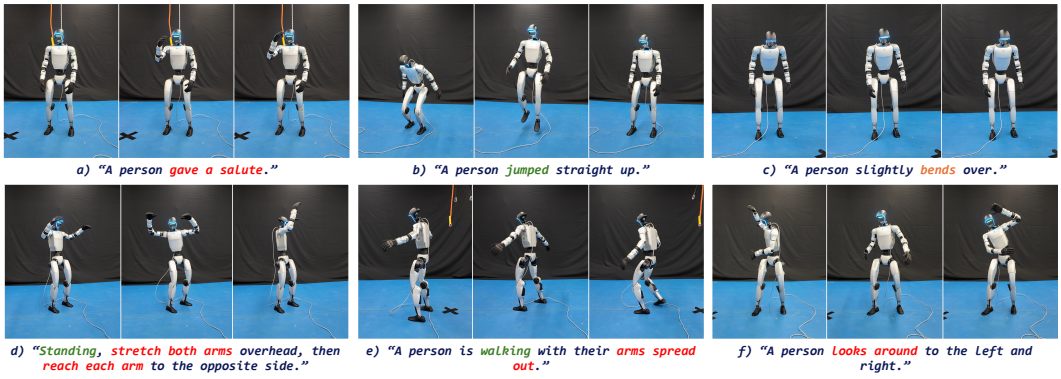

Figure 4: G1 executions selected after simulation screening and manual review.

Table 4: Generator ablations on HumanML3D-G1. Random-mask summarizes three different partition/run combinations (mean ± sample SD). Contact Acc: foot-contact agreement.

Variant	R@1 ↑	R@2 ↑	R@3 ↑	FID ↓	Div →	PJ ↓	Contact Acc ↑
w/o Part Attention	0.274±0.005	0.356±0.009	0.575±0.012	3.811±0.094	**7.791**±0.108	0.035±0.001	0.861±0.007
Random-mask	0.281±0.017	0.362±0.009	0.586±0.016	3.533±0.218	7.771±0.043	0.028±0.009	0.880±0.016
Full	**0.301**±0.007	**0.386**±0.014	**0.611**±0.015	**3.165**±0.068	7.736±0.129	**0.019**±0.001	**0.914**±0.005

Table 5: Effect of duration modeling on generated references on HumanML3D-G1.

Variant	Duration Error ↓	R@3 ↑	FID ↓	Div →
Oracle Length	**0.000**±0.000	**0.618**±0.002	**2.980**±0.015	6.900±0.045
Fixed-Length	12.471±3.596	0.584±0.005	4.317±0.021	6.760±0.038
Full (16-frame primitives)	3.080±0.385	0.599±0.013	3.207±0.033	**7.836**±0.048

4.5 Ablation Study

Full leads most reported metrics, combining stronger text–motion alignment with better distributional fidelity, smoother motion, and higher foot-contact agreement (Table 4). Its advantage over Random-mask supports the value of anatomically meaningful feature grouping. Replacing the part-aware module with standard attention also degrades these aspects of reference quality. Together, these comparisons support the part-aware design, with Random-mask providing more direct evidence for the contribution of anatomical assignments.

4.6 Duration Modeling (EOS) Study

Compared with Fixed-Length, Full reduces duration error while improving text–motion alignment and distributional fidelity (Table 5). It ranks second to Oracle Length on these metrics, using EOS to select endpoints without requiring annotated durations at inference. Duration error reflects agreement with dataset annotations, and a given instruction may admit multiple valid motion durations.

5 Conclusion

PartiGen is a part-aware latent diffusion framework for language-conditioned humanoid motion generation. It introduces anatomical structure into feature processing while maintaining a shared whole-body motion representation. Kinematic supervision encourages geometric consistency, and EOS-based endpoint selection adapts motion duration at frame resolution. Experiments on HumanML3D-G1 and the compositional stress test show improved text–motion alignment and distributional quality, while ablations support the contribution of anatomical grouping and part-aware processing. Duration evaluation further shows improved agreement with annotated lengths over a fixed cutoff. A pretrained, frozen GRIT controller tracks the generated references in simulation and selected G1 hardware trials. These findings highlight the value of incorporating anatomical structure and temporal extent into the generation of robot motion references (see Appendix S4 for limitations).

AI USE STATEMENT

We used generative AI tools to polish the language and improve the clarity and readability of the manuscript. The authors take responsibility for all final content of this work.

REFERENCES

Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C. Karen Liu. Retargeting matters: General motion retargeting for humanoid motion tracking. *arXiv preprint arXiv:2510.02252*, 2025.

Zhanxiang Cao, Yang Zhang, Buqing Nie, Huangxuan Lin, Haoyang Li, and Yue Gao. Learning motion skills with adaptive assistive curriculum force in humanoid robots. *arXiv preprint arXiv:2506.23125*, 2025. URL https://arxiv.org/abs/2506.23125.

Zhanxiang Cao, Liyun Yan, Yang Zhang, Sirui Chen, Jianming Ma, Tianyue Zhan, Shengcheng Fu, Yufei Jia, Cewu Lu, and Yue Gao. HiWET: Hierarchical world-frame end-effector tracking for long-horizon humanoid loco-manipulation. *arXiv preprint arXiv:2602.06341*, 2026.

Inbar Gat, Sigal Raab, Guy Tevet, Yuval Reshef, Amit Haim Bermano, and Daniel Cohen-Or. AnyTop: Character animation diffusion with any topology. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2025.

Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2Motion: Conditioned generation of 3D human motions. In *Proceedings of the 28th ACM international conference on multimedia*, pp. 2021–2029, 2020.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3D human motions from text. In *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.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. StreamingT2V: Consistent, dynamic, and extendable long video generation from text. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2568–2577, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

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.

Zhenyu Jiang, Yuqi Xie, Jinhan Li, Ye Yuan, Yifeng Zhu, and Yuke Zhu. Harmon: Whole-body motion generation of humanoid robots from language descriptions. *arXiv preprint arXiv:2410.12773*, 2024.

Peng Jin, Yang Wu, Yanbo Fan, Zhongqian Sun, Wei Yang, and Li Yuan. Act as you wish: Fine-grained control of motion diffusion model with hierarchical semantic graphs. *Advances in Neural Information Processing Systems*, 36:15497–15518, 2023.

Xuan Ju, Ailing Zeng, Chenchen Zhao, Jianan Wang, Lei Zhang, and Qiang Xu. HumanSD: A native skeleton-guided diffusion model for human image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15988–15998, 2023.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2151–2162, 2023.

Aliasghar Khani, Arianna Rampini, Evan Atherton, and Bruno Roy. UniMoGen: Universal motion generation. *arXiv preprint arXiv:2505.21837*, 2025.

Zhe Li, Yangyang Wei, Boan Zhu, Yibo Peng, Tao Huang, Pengwei Wang, Zhongyuan Wang, Cheng Chi, Chang Xu, and Shanghang Zhang. From language to locomotion: Retargeting-free humanoid control via motion latent guidance. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=k3Cyx3Uets.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pp. 5442–5451, October 2019.

mrzuang. GRIT: Unitree G1 deployment code. GitHub repository, 2026. URL https://github.com/mrzuang/GRIT_teleop_deploy. Accessed September 18, 2026.

Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10985–10995, 2021.

Mathis Petrovich, Michael J. Black, and Gül Varol. TEMOS: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision (ECCV)*, pp. 480–497, 2022.

Huaijin Pi, Sida Peng, Minghui Yang, Xiaowei Zhou, and Hujun Bao. Hierarchical generation of human-object interactions with diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15061–15073, October 2023.

Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 722–731, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. *arXiv preprint arXiv:2303.01418*, 2023.

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

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

Weiji Xie, Jiakun Zheng, Jinrui Han, Jiyuan Shi, Weinan Zhang, Chenjia Bai, and Xuelong Li. TextOp: Real-time interactive text-driven humanoid robot motion generation and control. *arXiv preprint*, 2026.

Mingxi Xu, Qi Wang, Zhengyu Wen, Phong Dao Thien, Zhengyu Li, Ning Zhang, Xiaoyu He, Wei Zhao, Kehong Gong, and Mingyuan Zhang. NECromancer: Breathing life into skeletons via bvh animation. *arXiv preprint arXiv:2602.06548*, 2026.

Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. InterDiff: Generating 3D human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14928–14940, 2023.

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. PhysDiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16010–16021, 2023.

Yanjie Ze, Zixuan Chen, João Pedro Araújo, Zi ang Cao, Xue Bin Peng, Jiajun Wu, and C. Karen Liu. TWIST: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. MotionDiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022.

Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. ReMoDiffuse: Retrieval-augmented motion diffusion model. *arXiv preprint arXiv:2304.01116*, 2023.

Kaifeng Zhao, Gen Li, and Siyu Tang. DartControl: A diffusion-based autoregressive motion model for real-time text-driven motion control. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

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.

Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. MotionBERT: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15085–15099, 2023.

APPENDIX: SUPPLEMENTARY MATERIAL

APPENDIX CONTENTS

S1 Overview 13
S2 Detailed Architecture 13
S3 Dataset and Evaluation Details 20
S4 Failure Cases and Limitations 22
S5 Supplementary Video 22
S6 GRIT Execution Component and Deployment on Unitree G1 22
S7 Extended Related Work 26

S1 OVERVIEW

This appendix details the PartiGen motion generator, datasets and evaluation, limitations, and adopted GRIT execution component. Section S2 describes the proposed generator; Section S6 describes its interface to the pretrained, frozen tracker and the deployment protocol. An extended literature review is also included.

S2 DETAILED ARCHITECTURE

We provide the internal configurations of the motion-generator networks below. These configurations describe BP-MVAE and DA-LDM, not the adopted GRIT policy.

S2.1 BP-MVAE ENCODER AND DECODER

The BP-MVAE encoder and decoder use Transformer stacks with 9 layers each. The shared hidden dimension is 512, with 4 attention heads, GELU activations, and Layer Normalization. The part-aware modification concerns the construction of Q/K/V from masked motion features; the shared frame embedding, residual connections, and feedforward blocks retain their standard form. The decoder reconstructs future motion from the shared latent and history. In both encoder and decoder, the nine-layer stack repeats two masked-feature attention layers followed by one standard attention layer three times. Each masked Q/K/V branch reads the 57-dimensional frame features; this input dimension is distinct from the 512-dimensional hidden residual stream.

S2.2 HIDDEN LAYER DIMENSIONS AND BLOCK ARCHITECTURE

Following the generator's YAML specifications, we define its core dimensional parameters as follows:

Model Dimensions: We employ a consistent hidden dimension of $h_{\text{dim}} = 512$ and a feedforward network size of $ff_{\text{size}} = 1024$.

- *Hidden Dimension ($h_{dim} = 512$):* This serves as the uniform width for all Transformer layers, including the total dimension for Query, Key, and Value matrices, as well as the dimensionality for Layer Normalization and residual connections.
- *Feedforward Size ($ff_{size} = 1024$):* This represents the intermediate dimension of the Feedforward Network (FFN). The FFN follows a linear $512 \rightarrow 1024$ projection, GELU, dropout, and a linear $1024 \rightarrow 512$ projection, maintaining a $2\times$ expansion ratio.

Structure of the Transformer Block: The shared backbone retains standard attention, normalization, residual connections, and feedforward sublayers. In the part-aware module, the residual stream is 512-dimensional, while the Q/K/V branch reads the original 57-dimensional frame features through fixed part masks. The concatenated head outputs pass through the output projection

and return to this shared stream. No additional modification of the residual or feedforward blocks is introduced.

S2.3 Multi-Head Attention Configuration and Body-Part Mapping

The Multi-Head Attention (MHA) mechanism uses body-part information during feature processing. The resulting features are fused into a shared motion latent.

Head Allocation and Dimensionality: The model utilizes $N_{\text{heads}} = 4$ attention heads, ensuring a one-to-one mapping between heads and the four predefined semantic body parts.

- *Dimension per Head (d_k):* Calculated as $h_{\text{dim}}/N_{\text{heads}} = 512/4 = 128$.
- *Head-to-Part Mapping:* Each head has a fixed mask selecting its part's joint angles. Root/contact features and all joint-angle increments are shared across heads.

Kinematic Grouping Table: Figure S1 illustrates the four body groups, and Table S1 lists their head assignments. All DoF indices are zero-based within the 23-dimensional joint-angle vector, rather than the full 57-dimensional frame representation: joint index j corresponds to feature channel $11 + j$. Thus, the left leg, right leg, torso, and upper-body groups select joint-angle channels 11–16, 17–22, 23–25, and 26–33, respectively. The label "Upper Body" in the figure and table denotes both arms only; the torso is a separate group.

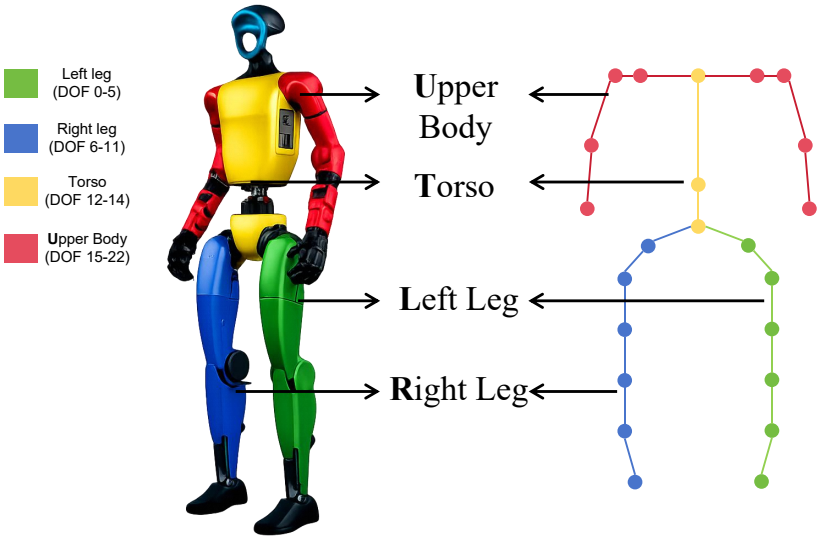

Figure S1: Body-part grouping of the G1 robot. Colors identify the left leg (DoFs 0–5), right leg (6–11), torso (12–14), and both arms (15–22, labeled "Upper Body"). The skeleton is schematic; its nodes do not correspond one-to-one to individual DoFs. Grouping applies only to the joint-angle channels: root orientation, foot contacts, local root displacement, and root height (channels 0–10), together with all joint-angle increments (34–56), are shared across all heads.

Table S1: Mapping of Attention Heads to Semantic Body Parts and DOF Indices.

Head Index	Body Part	DOF Indices	DOF Count
Head 0	Left Leg	[0, 1, 2, 3, 4, 5]	6
Head 1	Right Leg	[6, 7, 8, 9, 10, 11]	6
Head 2	Torso	[12, 13, 14]	3
Head 3	Upper Body	[15, 16, 17, 18, 19, 20, 21, 22]	8

Fixed Feature Masks and Independent Q/K/V Projections. Let $\mathcal{J}_k$ be the zero-based DoF index set for head k in Table S1. Its fixed mask $\mathbf{m}_k \in \{0, 1\}^{57}$ is constructed from the body-part

configuration:

$$m_{k,c} = \begin{cases} 1, & c \in \{0, \ldots, 10\} \cup \{34, \ldots, 56\}, \\ 1, & c = 11 + j \text{ for } j \in \mathcal{J}_k, \\ 0, & \text{otherwise.} \end{cases} \tag{S1}$$

Thus, only the 23 joint-angle channels are gated by part. The 11 root/contact features and all 23 joint-angle increments remain visible to every head. For $\mathbf{X} \in \mathbb{R}^{L \times 57}$, masks broadcast over frames to give $\mathbf{X}_k = \mathbf{X} \odot \mathbf{m}_k$. Each head uses independent linear projections:

$$\mathbf{Q}_k = \mathrm{Linear}_k^Q(\mathbf{X}_k), \quad \mathbf{K}_k = \mathrm{Linear}_k^K(\mathbf{X}_k), \quad \mathbf{V}_k = \mathrm{Linear}_k^V(\mathbf{X}_k), \tag{S2}$$

where each projection maps 57 input features to a head width of 128. The head output is $\mathbf{O}_k = \mathrm{softmax}(\mathbf{Q}_k \mathbf{K}_k^\top / \sqrt{128})\mathbf{V}_k$. Concatenation yields $[\mathbf{O}_0; \mathbf{O}_1; \mathbf{O}_2; \mathbf{O}_3] \in \mathbb{R}^{L \times 512}$ along the feature axis, and the standard output projection returns this tensor to the shared residual stream. Including batch size B, Q/K/V have shape $[B, L, 128]$ per head and the attention weights have shape $[B, 4, L, L]$. The feature masks act before projection; they are not masks over temporal attention scores. Shared increment channels and output fusion preserve access to whole-body information, so the design does not enforce independent body-part representations.

Random-mask Control (Preliminary Results). The control randomly permutes the 23 joint-angle indices and partitions them into groups of sizes 6, 6, 3, and 8, with each angle assigned to exactly one head. Within each run, the partition is fixed across all masked-feature layers in the encoder and decoder, throughout training and inference. Only the assignment of angle channels 11–33 changes; root/contact channels 0–10 and all angle-increment channels 34–56 remain shared. The 57-to-128 Q/K/V projections, head count, output projection, shared latent, and remaining architecture are unchanged. The VAE and its corresponding LDM use the Full model's 8-frame training and evaluation settings. The realized partitions and their random seeds will be recorded with the checkpoints. Table S2 lists the three runs with different partitions; the main-paper ablation table reports their mean $\pm$ sample SD. With one reported run per partition, the results do not isolate partition effects from training-run variation.

Table S2: Individual Random-mask results. A, B, and C use different fixed partitions and separate training runs; each row is a point estimate.

Partition	R@1 ↑	R@2 ↑	R@3 ↑	FID ↓	Div →	PJ ↓	Contact Acc ↑
A	0.279	0.363	0.588	3.701	7.726	0.038	0.870
B	0.299	0.370	0.601	3.287	7.812	0.021	0.898
C	0.265	0.353	0.569	3.612	7.776	0.024	0.871

The aggregate SD uses denominator $n - 1 = 2$ and describes variation across these three partition/run combinations, not fixed-partition training-seed variation or a confidence interval. Full's reported mean has higher retrieval and contact accuracy and lower FID and PJ than each random run; Random B and C have Diversity closer to the dataset reference. These descriptive comparisons do not establish statistical significance or general superiority over arbitrary partitions.

Shared 128-Dimensional Latent Representation

The PartiGen generator uses a single shared motion latent $\mathbf{z} \in \mathbb{R}^{128}$ for each encoded motion primitive. Body-part information enters feature processing, while the latent represents root and articulated-body motion jointly.

Concept of the Unified Latent Space: All body regions share the same 128 coordinates. There are no dedicated body-part subvectors and no separate global, arm, torso, or leg latent branches. The 128-dimensional latent size is distinct from the 128-dimensional attention head width; these are separate architectural quantities.

Design Rationale: The shared latent provides a compact common representation for the following aspects of motion:

- *Whole-Body Context:* Locomotion and gestures involve dependencies across limbs and the torso. Their features are represented jointly in the shared code.

- *Root and Articulation:* Global trajectory information and local joint motion use the same latent representation without separate global and local code branches.
- *Shared Capacity:* No subset of coordinates is reserved for a particular body region, so the full 128-dimensional representation is available for the encoded motion.

Role of Body-Part Attention in a Shared Space: Body-part attention is used during feature processing:

- **In the Encoder:** Attention-derived motion features are fused into the shared latent vector $\mathbf{z}$.
- **In the Decoder:** Whole-body motion is reconstructed from the shared code together with motion history. Individual body parts do not receive independent latent codes.

Input Feature Representation and Dimensionality
Each motion frame $\mathbf{x}_t$ contains $n_{\text{feats}} = 57$ features describing root orientation, foot contacts, local root displacement, root height, joint angles, and joint-angle increments.

Feature Decomposition: The ordered layout is given in Table S3, using zero-based inclusive channel indices. The dimensions sum to $5 + 2 + 3 + 1 + 23 + 23 = 57$.

Table S3: Ordered layout of the 57-dimensional motion feature vector. Indices are zero-based and inclusive.

Channels	Dimension	Feature
0	1	$\sin(\text{roll})$
1	1	$\cos(\text{roll}) - 1$
2	1	$\sin(\text{pitch})$
3	1	$\cos(\text{pitch}) - 1$
4	1	Δyaw: inter-frame yaw increment
5–6	2	Left and right foot contact indicators
7–9	3	$\Delta\mathbf{r}^{\text{local}}$: local-frame root displacement
10	1	Root height
11–33	23	$\mathbf{q}$: joint angles (DoF)
34–56	23	$\Delta\mathbf{q}$: inter-frame joint-angle increments
Total	**57**	

Feature Conventions:

- *Orientation Features:* Roll and pitch are represented by sine and shifted-cosine pairs; yaw is represented by an inter-frame increment. These five channels are not a 6D rotation representation or a quaternion.
- *Position Features:* Channels 7–9 contain displacement in the local frame, while channel 10 stores root height. They are not an absolute world-frame translation vector.
- *Joint Features:* Joint angles occupy channels 11–33, and their inter-frame increments occupy channels 34–56. The increment channels are not joint angular velocities; conversion to velocity requires accounting for the frame interval.
- *Exported Trajectories:* The root positions and quaternions used by the deployment pipeline describe reconstructed reference trajectories. They are distinct from the root feature channels used by the generative model.

Feature normalization. History and future frames share feature-wise Z-score normalization, $x'_d = (x_d - \mu_d)/\sigma_d$. The statistics used in the experiments are cached in `meanstd.pkl` and originate from the training split. If no cache exists, the loader computes statistics only from training data; validation reuses the cache. Statistics are estimated from sampled training motion clips by pooling their sample and time dimensions, with $\mu = E[x]$ and $\sigma = \sqrt{E[x^2] - E[x]^2}$. This is neither per-sequence normalization nor an exhaustive average over every training frame.

S2.4 Training Configuration and Loss Decomposition

We provide the detailed training schedules and the granular decomposition of the loss functions used for the BP-MVAE stage.

Optimization and Learning Rate Schedule: Both stages are trained on a single NVIDIA RTX 4090 GPU with batch size 512 and optimized using AdamW with an initial learning rate of 5.0×10^{-5}. Evaluation uses batch size 128. The BP-MVAE uses cosine learning-rate annealing, whereas DA-LDM uses linear learning-rate annealing. The gradient norm is clipped at 1.0. BP-MVAE uses $\beta = 10^{-4}$ and $\lambda_{\text{kin}} = 0.8$; DA-LDM uses an overall EOS loss weight of $\lambda_{\text{EOS}} = 0.5$.

Multi-Stage Training Schedule: The BP-MVAE is trained for 200,000 steps, divided into four strategic stages:

- *Stage 0 (0–30k):* Initial feature and joint position reconstruction.
- *Stage 1 (30k–80k):* Introduction of latent space regularization ($\mathcal{L}_{\text{KL}}$).
- *Stage 2 (80k–140k):* Emphasis on temporal continuity and root trajectory stability.
- *Stage 3 (140k–200k):* Final fine-tuning of kinematic smoothness and foot contact.

Granular Loss Weight Breakdown:
In the BP-MVAE section of the main paper, we summarized the objective as $\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{KL}} + \lambda_{\text{kin}}\mathcal{L}_{\text{kin}}$. For implementation, the aggregate kinematic objective $\mathcal{L}_{\text{kin}}$ is defined as a weighted sum of specific geometric and temporal constraints: $\mathcal{L}_{\text{kin}} = \sum \omega_i \mathcal{L}_i$, where ω_i denotes the sub-weight for each component. The global coefficient $\lambda_{\text{kin}} = 0.8$ serves as a scaling factor to balance the total kinematic regularizers against the primary reconstruction loss. The detailed breakdown of these sub-weights is provided in Table S4.

Table S4: Comprehensive Decomposition of Loss Weights for BP-MVAE Training.

Aggregate Term (Main Paper)	Internal Component	Description	Sub-weight (ω)
$\mathcal{L}_{\text{rec}}$	`rec`	Primary feature reconstruction	1.0
$\beta\mathcal{L}_{\text{KL}}$	`kl`	Variational bottleneck regularization	1.0×10^{-4}
$\lambda_{\text{kin}}\mathcal{L}_{\text{kin}}$ ($\lambda_{\text{kin}} = 0.8$)	`body_trans`	FK-based body position error	0.05
	`body_rot`	FK-based body rotation error	0.01
	`dof_pos`	Joint angle precision	0.03
	`dof_vel`	Temporal joint velocity smoothness	1.0×10^{-5}
	`foot_con`	GT-contact-masked foot position reconstruction	0.01
	`delta_joints`	First-order joint position change	1.0
	`delta_trans`	Temporal change in root translation	1.0
	`delta_orient`	Temporal change in root orientation	1.0
	`delta_dof`	First-order joint angle change	1.0
	`smooth`	Second-order Laplacian smoothness	0.1
	`drift_yaw`	Global heading (Yaw) drift penalty	0.3
	`drift_xy`	Planar (XY) position drift penalty	0.3

This hierarchical weighting scheme allows for fine-grained control over physical plausibility while maintaining the high-level balance of the optimization objective. Quantization weights (`quantize_rot/trans`) were set to 0.0 in the final version.

S2.5 Duration-Adaptive Latent Diffusion Model (DA-LDM) Specifications

The second stage, DA-LDM, performs conditional motion synthesis in the shared latent space $\mathcal{Z} = \mathbb{R}^{128}$. The Transformer denoiser projects the shared noisy code into its 512-dimensional hidden space, conditioned on text, motion history, and diffusion timestep. It does not use separate body-part latent branches.

Denoiser Architecture: The core of the DA-LDM is a *DenoiserTransformer* consisting of 8 transformer layers. Notably, this is one layer shallower than the BP-MVAE (9 layers), optimizing the

balance between generative capacity and inference speed. The internal dimensions ($h_{\text{dim}} = 512$, $ff_{\text{size}} = 1024$) are kept consistent with the VAE to ensure seamless feature transition.

Multi-Modal Input and Conditioning: The denoiser processes three distinct input streams to generate kinematically grounded motion:

- *Noisy Latent ($\mathbf{z}_\tau$):* A single latent token per primitive, with batched shape $[B, 1, 128]$, representing the noised clean motion code at diffusion timestep τ.
- *Motion History ($\mathbf{H}$):* A temporal window of 2 historical frames ($[2, 57]$), providing the local kinematic context for transition stability.
- *Textual Conditioning:* Global semantic intent extracted from a pre-trained CLIP (ViT-B/32) model with a fixed embedding dimension of 512.

Detailed Hyperparameters: Table S5 summarizes the structural and training configurations for the DA-LDM stage.

Table S5: Detailed Specifications for the DA-LDM Denoiser.

Category	Parameter	Value
Transformer Backbone	Number of Layers	8
	Hidden Dimension (h_{dim})	512
	FFN Intermediate Dimension (ff_{size})	1024
	Activation Function	GELU
Attention Mechanism	Number of Heads (N_{heads})	4
	Dimension per Head (d_k)	128
	Dropout Rate	0.1
Conditioning & Noise	CLIP Text Embedding Dim	512
	History Window Size	2 frames
	CFG Masking Probability (p)	0.1

DA-LDM Training Schedule. DA-LDM is trained for 300,000 optimization steps with teacher forcing: the history condition is taken from ground-truth motion throughout training. Each example samples one diffusion timestep and evaluates the corresponding single-step denoising objective. The learning rate is annealed linearly over training. Although the training implementation contains stage-dependent options for autoregressive rollout and static perturbations of the history, these options are disabled in the experimental configuration. The reported training therefore uses teacher forcing throughout, without a rollout or history-perturbation curriculum.

Stochastic Regularization: During training, the text condition is set to zero with probability $p = 0.1$. History is always retained and is not masked. At each reverse step during inference, the model is evaluated with the text condition present and zeroed, while the noisy latent, timestep, and history are identical in both evaluations:

$$\begin{aligned} \hat{\mathbf{x}}_0^{\text{text}} &= f_\theta(\mathbf{z}_\tau, \tau, \mathbf{c}_{\text{text}}, \mathbf{H}), \\ \hat{\mathbf{x}}_0^{\text{zero}} &= f_\theta(\mathbf{z}_\tau, \tau, \mathbf{0}, \mathbf{H}), \\ \hat{\mathbf{x}}_0^{\text{CFG}} &= \hat{\mathbf{x}}_0^{\text{zero}} + s\big(\hat{\mathbf{x}}_0^{\text{text}} - \hat{\mathbf{x}}_0^{\text{zero}}\big), \qquad s = 5.0. \end{aligned} \tag{S3}$$

The text-zeroed prediction remains conditioned on history. Guidance is applied to the clean-latent prediction before the reverse sampling update.

End-of-Sequence (EOS) Prediction Head:
DA-LDM uses a dedicated EOS head for frame-level binary classification. At inference, EOS is evaluated after complete candidate generation and selects an endpoint at frame resolution, including within a generated primitive. The generation block length therefore does not determine the truncation resolution.

MLP Architecture: The EOS head is a four-layer MLP with GELU activations in the first three layers. The last hidden layer, immediately before the output layer, has width 128.

Semantic Termination Logic: Each frame has a binary yes/no EOS label. EOS training uses weighted binary cross-entropy with positive-class coefficient $\alpha = 0.3$ and overall loss coefficient

$\lambda_{\text{EOS}} = 0.5$. At inference, the first frame-level threshold crossing at $\gamma = 0.9$ determines the cutoff, with the threshold-crossing frame retained. If the threshold is never crossed, the complete candidate is retained.

Diffusion Parameterization and Training Objective. The denoiser uses direct clean-latent prediction (`START_X`). The clean target $\mathbf{x}_0 = \mathbf{z}$ has batched shape $[B, 1, 128]$. With $N = 5$, training samples an integer timestep $\tau \sim \text{Uniform}\{0, \ldots, N-1\}$ and applies the forward process

$$\mathbf{z}_\tau = \sqrt{\bar{\alpha}_\tau}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_\tau}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{S4}$$

Here $\bar{\alpha}_\tau = \prod_{j=0}^{\tau}(1-\beta_j)$ is the cumulative signal coefficient of the configured discrete cosine schedule. Timesteps are passed without rescaling. The symbol $\mathbf{x}_0$ always denotes the clean target; $\mathbf{z}_\tau$ denotes a noisy state using the implementation's zero-based timestep indices.

The output projection maps each hidden token from 512 to 128 dimensions and yields $\hat{\mathbf{x}}_0 = f_\theta(\mathbf{z}_\tau, \tau, \mathbf{c}_{\text{text}}, \mathbf{H})$. Latent reconstruction and normalized future-feature reconstruction each have weight 1.0. The geometry term has outer coefficient $\lambda_g = 1$ and contains individually weighted losses; its internal weights are not normalized to sum to one. All active terms use Huber loss with $\delta = 1$ and mean reduction over every element of the corresponding tensor. Weighted terms are summed without division by the number of terms or the sum of weights. The DA-LDM objective has no KL contribution. The implementation multiplies the final loss by the mean timestep-sampling weight, which is 1 under the uniform sampler used here.

DA-LDM geometry terms and units. For feature version 3, the active geometry terms are FK global body position (0.05), global body quaternion components (0.01), DoF joint angles (0.03), DoF angular velocity (10^{-5}), GT-contact-masked foot position (0.01), and final-frame root yaw and XY drift (1.0 each). Smoothness, quantization, and the additional delta/consistency terms are disabled in this stage. The per-term drift weights are increased from 0.3 in BP-MVAE to 1.0 in DA-LDM to strengthen these constraints in the second stage. Table S4 separately describes the BP-MVAE objective, including its outer kinematic coefficient.

Geometry is evaluated after feature denormalization, trajectory reconstruction, and FK, without further standardization. Positions are in meters, joint angles and yaw in radians, angular velocities in radians per second, and quaternion components are dimensionless. Angular velocity is $(q_{t+1} - q_t)/\Delta t$, with the FK implementation's fixed default $\Delta t = 1/30$ s for the 30 Hz motion data; the interval is not dynamically read from dataset metadata. This differs from the feature-channel angle increments. The numerical Huber threshold of 1 thus corresponds to different physical scales for different terms.

Contact-masked foot position reconstruction. The foot term is

$$\mathcal{L}_{\text{foot}} = \text{mean } h_1(c_{\text{GT}} \odot (p_{\text{pred}} - p_{\text{GT}})), \tag{S5}$$

where h_1 is element-wise Huber loss and the GT contact mask is broadcast over the XYZ coordinates of both feet. Masked zero elements remain in the mean denominator; the loss is not normalized by the number of active contacts. It compares foot positions, not contact-period velocities or adjacent-frame displacements, and is therefore not a direct foot-sliding penalty.

Quaternion-loss limitation. Body rotation loss applies Huber loss to quaternion components, rather than a geodesic rotation distance. The matrix-to-quaternion conversion selects a representation whose largest-magnitude component is positive, without enforcing $w \geq 0$ or aligning predicted and target quaternion signs. Although identical matrices give deterministic representations, nearby rotations can cross a representation boundary and produce an artificial loss increase. An implementation diagnostic with yaw angles differing by 2×10^{-5} rad produced a quaternion dot product near -1 and component Huber loss 0.4571, versus approximately 1.25×10^{-11} after sign alignment. The frequency and impact of this effect in training have not been quantified; this diagnostic does not measure a change in downstream performance.

Reverse Sampling. The default sampler is DDPM with all five reverse updates and `FIXED_SMALL` posterior variance. Initialize $\mathbf{z}_{N-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. At each step, use the guided prediction $\hat{\mathbf{x}}_0^{\text{CFG}}$ to evaluate the forward-process posterior mean. Writing this guided estimate as $\hat{\mathbf{x}}_0$ below, for $\tau > 0$ sample

$$\mathbf{z}_{\tau-1} = \tilde{\boldsymbol{\mu}}_\tau(\mathbf{z}_\tau, \hat{\mathbf{x}}_0) + \sqrt{\tilde{\beta}_\tau}\boldsymbol{\eta}, \qquad \tilde{\beta}_\tau = \beta_\tau \frac{1-\bar{\alpha}_{\tau-1}}{1-\bar{\alpha}_\tau}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{S6}$$

At $\tau = 0$, return the posterior mean without additional noise, then decode the resulting clean latent with the VAE. Predicted clean latents are not clipped (`clip_denoised=False`). The implementation also provides an optional DDIM sampler; the stochastic posterior update above describes the default DDPM path.

S2.6 Frame-Level EOS and Post-Generation Selection

The generator produces primitives, while EOS selects a frame-level endpoint. Primitive k contains $\mathbf{P}_k \in \mathbb{R}^{\ell \times 57}$ and has a single shared latent $\mathbf{z}^{[k]} \in \mathbb{R}^{128}$; the index k denotes a temporal block, not a body-part branch. Conditioning history is separate from the newly generated frames. The segment/transition comparison in Table 1 of the main paper and all generator ablations use $\ell = 8$. All three variants in the duration study use $\ell = 16$, with other settings held fixed within that study.

All experiments use the dataset maximum of 320 frames as the candidate horizon: $N_{\text{primitive}} = 40$ for $\ell = 8$ and $N_{\text{primitive}} = 20$ for $\ell = 16$. This common horizon is fixed rather than set separately to each test motion's length. The complete candidate is generated first. With one-based frame indices, the retained length is

$$T_{\text{out}} = \begin{cases} \min\{t \in \{1, \dots, 320\} : \hat{s}_t > 0.9\}, & \text{if a crossing exists,} \\ 320, & \text{otherwise.} \end{cases} \tag{S7}$$

Here $\hat{s}_t$ denotes the EOS probability associated with frame t; the output includes frames 1 through T_{out}. Thus the threshold-crossing frame is retained, and the final length need not be divisible by ℓ. In the qualitative example, the EOS-selected output has 148 frames and the fixed-length comparison has 160 frames; these output lengths are distinct from the 320-frame candidate horizon. This post-generation operation does not stop autoregressive generation online or save already incurred diffusion computation.

S3 Dataset and Evaluation Details

In this section, we provide further details regarding our data processing pipeline for humanoid retargeting and the construction of our custom evaluation protocol.

S3.1 Unified Retargeting Protocol Details

As introduced in Section 4.1, PartiGen and TextOp are trained in the G1 motion domain. DART, MDM, and AttT2M instead generate human-domain motions, which are converted to SMPL as needed and retargeted to G1 using the same GMR method used for the training data. The resulting DART+GMR, MDM+GMR, and AttT2M+GMR pipelines are evaluated on G1 references under the same metric protocol as the native-domain generators. This comparison assesses the complete pipelines; it does not isolate architecture from training-domain and conversion effects. TextOp provides the same-domain baseline. The GMR joint definitions and ordering are identical for the retargeted training data used by PartiGen and TextOp and for the retargeted human-motion outputs of the comparison baselines.

Step 1: SMPL-X to SMPL via SMPLify Optimization.
Since the General Motion Retargeting (GMR) framework expects standard SMPL body models, we first project the SMPL-X motions into the SMPL parameter space. We utilize an optimization-based fitting procedure inspired by SMPLify. For each frame, we optimize the SMPL pose parameters θ and shape parameters β to minimize the distance between the extracted 3D joint positions of the source SMPL-X motion (J_{src}) and the generated SMPL joints ($J_{smpl}(\theta, \beta)$). The objective function is defined as:

$$\mathcal{L}_{\text{fit}} = \mathcal{L}_{\text{joints}} + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}} \tag{S8}$$

where $\mathcal{L}_{\text{joints}} = \|J_{src} - J_{smpl}(\theta, \beta)\|_2^2$ is the primary data term. To prevent unnatural poses and temporal jittering, we apply a pose prior regularization $\mathcal{L}_{\text{pose}}$ and a first-order temporal smoothness penalty $\mathcal{L}_{\text{smooth}}$ across adjacent frames.

Step 2: SMPL to Unitree G1 via GMR.
Once the motion is accurately represented in the SMPL space, we feed the parameter sequences into

the GMR pipeline, which maps the human kinematic chain to the specific 23-DOF topology of the Unitree G1 robot. This mapping rigorously enforces the physical joint limits of the hardware.

S3.2 Evaluator and Statistical Protocol

The generator and G1-specific text–motion evaluator share the same motion-disjoint dataset split: approximately 25,000 training motions and 2,576 test motions. No test motion is used in training. All methods are scored with the same frozen evaluator checkpoint and identical text and motion preprocessing. The evaluator uses the GRU motion encoder, CLIP text encoder, and 512-dimensional alignment space described in the main paper. R-Precision and text–motion matching distance use the same evaluation implementation across methods; the HumanML3D-G1 reference-quality evaluation computes FID and Diversity from the designated evaluator features over the full 2,576-motion test set. The separate compositional stress test uses its 100-instruction evaluation set.

For learned methods, entries reported as mean $\pm$ standard deviation summarize three runs trained with different random seeds. The evaluator checkpoint remains fixed across these runs. These error bars describe variation across training seeds, not repeated evaluation of one trained checkpoint or confidence intervals. Tracking-table mean/median entries and stability counts retain their separately stated definitions and are not converted to this reporting format. The Random-mask row is an exception to the fixed-architecture training-seed convention: it aggregates three different random partitions and training runs using mean $\pm$ sample SD. Table S2 gives the individual point estimates. Its SD measures variation across partition/run combinations, not fixed-partition training-seed variation. For dataset-reference rows, mean $\pm$ standard deviation instead summarizes three evaluation runs on the fixed reference dataset using the same frozen evaluator. These standard deviations describe variation across evaluation runs, rather than across independently trained generators, and are not confidence intervals.

S3.3 Compositional Stress Test Prompts

In Section 4.3 of the main paper, we introduced a custom stress test comprising 100 complex multi-part instructions. This dataset probes coordinated multi-part motion generation using the shared latent representation.

Due to space constraints, we provide a representative subset of 15 prompts in Table S6. As demonstrated, these instructions demand the simultaneous execution of distinct lower-body dynamics (e.g., locomotion, jumping, or postural shifts) and upper-body semantics (e.g., waving, clapping, or pointing). Monolithic models typically struggle with such prompts, often exhibiting severe kinematic interference.

Table S6: Representative Prompts from the Compositional Stress Test. The instructions are systematically categorized based on the primary lower-body/root dynamics.

Kinematic Category	Instruction Prompt
Continuous Locomotion	A person is walking and pointing forward.
	A person is walking sideways and crossing arms.
	A person is walking backward and clapping hands.
	A person is jogging and putting hands on hips.
	A person is running and waving hands.
	A person is running and touching the head.
Jumping & Bounding	A person is jumping and clapping hands.
	A person is jumping and waving one hand.
	A person is hopping and swinging arms.
	A person is skipping and waving both hands.
In-place & Postural	A person is kicking and raising one hand.
	A person is squatting and stretching arms.
	A person is stepping forward and saluting.
	A person is turning around and raising both arms.
	A person is lunging forward and pointing upward.

S4 Failure Cases and Limitations

PartiGen has limitations in its manually specified body-part structure, dependence on an adopted tracker, and evaluation scope.

- **Body-Part Grouping and High-Intensity Motion:** The body-part grouping used in feature processing is manually specified. Although all body regions share a 128-dimensional latent vector, this design does not guarantee accurate modeling of high-intensity, whole-body maneuvers such as gymnastics or parkour, including their global momentum and Center of Mass (CoM) dynamics. The effect of alternative groupings and the representation of global coordination require further study. The shared latent also does not provide independently addressable latent controls for individual limbs. These limitations should not be interpreted as evidence of independent latent codes or an explicit penalty on cross-limb coupling.
- **Cross-Embodiment Transfer:** Our evaluation is primarily restricted to a single humanoid platform (Unitree G1). We have not empirically verified the cross-morphology robustness of the learned representation or its zero-shot transferability to other hardware configurations. The input layout and fixed joint-angle masks are defined for the G1's 23-DoF representation; applying the model to another morphology would require adapting these definitions and evaluating the resulting model. The amount of retraining needed for such adaptation has not been established, and generalizing part-aware feature learning to heterogeneous skeletons remains an open challenge.
- **Dependence on the Execution Component:** GRIT is pretrained independently and kept frozen. Execution quality depends on both the generated reference and this controller, including its tracking limits and deployment settings. Generator metrics computed before tracking do not measure realized robot motion or isolate the source of execution failures. We evaluate one tracker and selected hardware demonstrations; these do not establish controller-independent tracking improvements or a hardware success rate over the full test set. Robot-state feedback is confined to tracking, and the generator does not replan in response to execution errors.

S5 Supplementary Video

The accompanying supplementary video introduces PartiGen and presents representative motion examples comparing MuJoCo tracking with execution on the Unitree G1. These demonstrations illustrate the system's behavior in simulation and on the physical robot using the frozen GRIT execution component described in Section S6.1. The hardware examples are selected after simulation screening. The video provides qualitative evidence for these selected motions; aggregate tracking accuracy and hardware completion rates require the quantitative evaluation described in Section S6.3.

S6 GRIT Execution Component and Deployment on Unitree G1

S6.1 Component Interface and Deployment Pipeline

The execution component adopts the pretrained GRIT tracking policy through its public deployment repository (mrzuang, 2026). GRIT is existing software, not a controller proposed or trained in this work. Its weights remain frozen, and robot-state feedback is used by the tracker rather than the motion generator. The generator supplies reference trajectories; GRIT produces joint-level control targets; the deployment stack handles state feedback, policy inference, and command transmission.

To assess execution, we use the protocol *trajectory preprocessing* $\rightarrow$ *closed-loop MuJoCo evaluation* $\rightarrow$ *candidate selection* $\rightarrow$ *real-world execution*. Simulation screening and candidate selection define the hardware evaluation protocol, rather than additional learned components.

Trajectory preprocessing. For generated motions, the deployment pipeline receives reconstructed reference trajectories at 30 Hz, with root position, root orientation quaternion, and 23 joint angles per frame. This exported trajectory format differs from the model's 57-dimensional feature layout in Table S3. All original frames are retained, and trajectory dimensionality, numerical validity,

and quaternion legality are verified. An interpolated transition from the robot's current posture to the first reference frame precedes motion playback. This transition addresses the initial posture mismatch; it does not remove discontinuities between frames within the generated reference. Root quaternions are converted from *xyzw* to *wxyz* order; root positions and joint angles are resampled by linear interpolation and root rotations by spherical linear interpolation (SLERP), yielding 50 Hz references at the original playback speed. No source frames are trimmed (`trimmed_frames=0`). The 23-DoF trajectory is mapped onto the 29-DoF model by joint name, with the six missing wrist joints set to zero. Finally, reference states of all body links are obtained via forward kinematics and velocity computation, and stored in the NPZ format readable by GRIT.

Closed-loop evaluation in MuJoCo. Preprocessed trajectories are executed by the GRIT policy in MuJoCo in closed loop, rather than by kinematic replay through direct joint-position setting. Each trajectory is initialized independently, and the test procedure covers, in sequence: the default posture, a smooth transition to the first motion frame, full motion execution, return to the default posture, and a post-motion stability observation. Evaluation metrics include minimum pelvis height, maximum body inclination, joint reference tracking error, peak joint velocity, and the actuator torque saturation ratio. A trial is classified as *simulation-stable* if the minimum pelvis height is no less than 0.60 m and the maximum inclination does not exceed 35°; as *suspected fall* if the height drops below 0.45 m or the inclination exceeds 60°; and as *marginally stable* otherwise. Trials with abnormal termination, incomplete testing, or communication failures are logged separately and are not counted as passes.

Candidate selection. Passing the simulation stability check does not directly imply suitability for real-world execution. We therefore apply a conservative selection stage with stricter thresholds on posture, reference tracking error, joint velocity, and torque saturation, and grade the candidate motions according to lower-limb motion amplitude and root displacement. Motions that pass the automatic screening are further reviewed by manual replay. We emphasize that this screening is a risk-control step prior to hardware trials and does not constitute a safety guarantee for the robot.

Real-world deployment. The real-robot setup reuses the selected reference trajectories and the same frozen GRIT tracking policy as simulation. A Python controller receives robot states and computes control targets at 50 Hz, communicating with a C++ bridge program over local UDP; the bridge connects to the G1 via the Unitree SDK/DDS and publishes low-level control commands at 500 Hz, reusing the most recent policy output for the high-frequency loop. The operating procedure decouples standing-up from motion initiation: after pressing START on the gamepad, the robot transitions from its currently measured joint configuration to the default posture via a 5-second minimum-jerk interpolation; once the posture stabilizes, pressing the A button triggers another smooth reference transition into the first motion frame and starts playback. The SELECT button triggers a latching damping stop that blocks subsequent policy commands. Note that the damping stop does not guarantee that the robot remains standing and is not a substitute for independent physical safety measures.

S6.2 Position Tracking Metrics

Body points and temporal association. We evaluate $K = 19$ body-frame origins, including the pelvis (Table S7). Actual positions are obtained from MuJoCo's `MjData.xpos`, rather than the centers of mass in `xipos`. Actual and execution-reference positions use the same point definitions and ordering. The references are resampled from 30 to 50 Hz at their original speed; the metric is evaluated on the execution timeline, not directly on the original 30 Hz grid. Reference frames are paired with the actual states read by the controller using exact state IDs, without post-hoc time shifts or best-pose matching.

Metric definitions. For a motion with T evaluated frames, let $\mathbf{p}_{t,k}$ and $\hat{\mathbf{p}}_{t,k}$ be the actual and execution-reference positions in meters. We compute

$$E_{\text{global}} = \frac{1000}{TK} \sum_{t=1}^{T} \sum_{k=1}^{K} \left\| \mathbf{p}_{t,k} - \hat{\mathbf{p}}_{t,k} \right\|_2 , \tag{S9}$$

$$E_{\text{root-relative}} = \frac{1000}{TK} \sum_{t=1}^{T} \sum_{k=1}^{K} \left\| (\mathbf{p}_{t,k} - \mathbf{p}_{t,\text{pelvis}}) - (\hat{\mathbf{p}}_{t,k} - \hat{\mathbf{p}}_{t,\text{pelvis}}) \right\|_2 . \tag{S10}$$

Table S7: The 19 body origins used for tracking metrics. Concatenating rows from top to bottom gives the point order. The pelvis is included in both metrics.

Region	MuJoCo body names
Root	`pelvis`
Left leg	`left_hip_pitch_link`, `left_knee_link`, `left_ankle_roll_link`, `left_toe_link`
Right leg	`right_hip_pitch_link`, `right_knee_link`, `right_ankle_roll_link`, `right_toe_link`
Torso/head	`torso_link`, `head_mimic`
Left arm	`left_shoulder_pitch_link`, `left_elbow_link`, `left_wrist_roll_link`, `left_hand_mimic`
Right arm	`right_shoulder_pitch_link`, `right_elbow_link`, `right_wrist_roll_link`, `right_hand_mimic`

Both scores are in millimeters. Root-relative MPJPE subtracts each trajectory's own pelvis translation, retaining orientation errors. It uses no root-rotation, heading, scale, or Procrustes alignment. The pelvis contributes zero to this score but remains in the denominator $K = 19$. The reference is the motion supplied for execution, rather than a paired dataset target or the controller's motor targets.

Evaluation interval and aggregation. MPJPE includes only frames with `evaluation_phase = "motion"`; entry interpolation, return to the default posture, and final stabilization are excluded. Stability screening still covers the complete prescribed procedure. Each motion's score averages over its body points and playback frames. Dataset mean and median are computed over the valid per-motion scores with equal motion weights. Fully recorded unstable motions remain included, while ERRORs are excluded from MPJPE aggregation and counted separately.

Fixed initial XYZ translation correction. For this diagnostic, use zero-based playback indices and let $t = 0$ be the first motion frame after entry interpolation. For motion i, define its fixed pelvis offset and corrected global error as

$$\mathbf{d}_i = \mathbf{p}^{(i)}_{0,\text{pelvis}} - \hat{\mathbf{p}}^{(i)}_{0,\text{pelvis}} \in \mathbb{R}^3,$$
$$E^{(i)}_{\text{corrected}} = \frac{1000}{19T_i} \sum_{t=0}^{T_i-1} \sum_{j=1}^{19} \left\| \left(\mathbf{p}^{(i)}_{t,j} - \hat{\mathbf{p}}^{(i)}_{t,j}\right) - \mathbf{d}_i \right\|_2 . \tag{S11}$$

The group mean is $N^{-1} \sum_{i=1}^{N} E^{(i)}_{\text{corrected}}$, giving each motion equal weight. The correction includes all three XYZ components, including height, and subtracts the same $\mathbf{d}_i$ at every time and every body point within motion i. The first frame remains in the metric. No per-frame translation alignment, rotation, scale, or temporal alignment is applied, so subsequent drift remains measured. This differs from root-relative MPJPE and need not reduce the error. It supplements, rather than replaces, raw Global MPJPE. Over all valid motions, without stability filtering or upper-tail trimming, corrected Global means are 308.89 mm for GT ($N = 499$) and 256.79 mm for PartiGen ($N = 496$).

S6.3 Results and Statistics

Generated-motion evaluation. The main-paper tracking table reports 500 references randomly sampled from the 2,576 generated test motions. Four records logged as ERRORs are excluded, leaving 496 valid MPJPE records. Stability checks are passed by 409 motions, or 81.8% of all requests. Excluded records are not counted as passes and are not automatically classified as physical failures. Mean and median MPJPE include fully recorded unstable runs; they are not statistics of the stability-passing subset alone.

GT reference evaluation. All 500 GT runs have been completed. One record (GT `006351`) is excluded because of an incomplete tracking record or a failed alignment check, leaving 499 valid records. Global MPJPE is 399.78 / 183.22 mm and root-relative MPJPE is 56.24 / 37.94 mm (mean / median), with equal motion weights and fully recorded unstable motions included. Of the valid runs, 427 pass the stability checks: 427/499 = 85.57%. For a common denominator across methods, the main-paper table reports passes divided by all requests, giving 427/500 = 85.4% for GT. The excluded record is not classified as a physical instability.

GT and PartiGen are evaluated with the same frozen GRIT controller and 29-DoF model, with the same six wrist joints set to zero in both reference groups.

DART+GMR reference evaluation. Of 500 requested runs, 494 have valid tracking records and six are excluded. Stability checks are passed by 395 motions, giving 395/500 = 79.0% over all requests and approximately 395/494 = 79.96% within valid records. Global MPJPE is 288.35 / 241.03 mm and root-relative MPJPE is 58.32 / 40.07 mm (mean / median).

Random-mask execution evaluation. The Random-mask execution evaluation contains 500 requests, with 494 valid records and six excluded records. Of all requests, 401 pass the posture screen (80.2%); the pass rate over valid records is 81.17%. Global MPJPE is 283.90 / 157.29 mm and root-relative MPJPE is 56.38 / 38.25 mm (mean / median). Full has a 1.6-percentage-point higher observed pass rate and lower Global MPJPE, whereas Random-mask has lower root-relative MPJPE. These aggregate results do not establish a uniform execution advantage or a statistically significant difference.

Execution without FK supervision. Table S8 compares Full and w/o FK using the same frozen GRIT controller and simulation stability criteria. Each group contains 500 requests. For w/o FK, 495 records are valid and five are excluded; 464 pass, yielding 92.8% over requests and 93.74% over valid records. The other 31 valid motions do not pass the stability proxy. Excluded records are not automatically classified as falls. MPJPE includes valid unstable runs; the stable-subset values are reported separately in Table S9.

Table S8: FK execution ablation under frozen GRIT. MPJPE: mean / median (mm), including valid unstable runs. The posture-screen rate is passes divided by all requests, rather than a measure of semantic completion.

Method	Requests	Valid	Excluded	Posture passes	Pass rate (↑)	Global MPJPE (mm, ↓)	Root-relative MPJPE (mm, ↓)
Full	500	496	4	409	81.8%	**252.402 / 108.871**	**57.995 / 39.765**
w/o FK	500	495	5	464	**92.8%**	275.61 / 112.37	60.01 / 41.82

Scope of the simulation pass rate. Full has a lower observed pass rate under the prescribed height/inclination screen (81.8% vs. 92.8%). This result does not support a claim that FK supervision improves that rate. However, the screen measures compliance with posture thresholds, not instruction completion or overall execution quality; its pass-rate ordering alone cannot establish that Full is inferior to w/o FK on those broader criteria.

Manual inspection of 15 w/o FK trials that passed the posture screen revealed cases in which the robot remained upright but failed to follow the reference motion. These observations illustrate that a higher posture-screen pass rate does not necessarily indicate better motion execution. The prevalence of such failures across the full evaluation set remains unquantified.

Full has lower all-valid Global and root-relative means and medians. These errors quantify tracking of each method's own references and do not validate their semantics or independently establish overall superiority. In the stable subsets, Full has lower mean errors, but w/o FK has a slightly lower root-relative median (37.53 vs. 37.98 mm). Stability-conditioned comparisons also involve different selected subsets. Evaluating broader execution quality requires assessing semantic completion and task-appropriate execution outcomes on the same instructions under the same controller.

Stability-conditioned tracking diagnostic. Table S9 reports raw, uncorrected MPJPE on valid runs satisfying the original simulation stability criteria: minimum pelvis height at least 0.60 m and maximum inclination at most 35°. For PartiGen and GT, filtering after ERROR exclusion exactly recovers the recorded PASS sets (409 and 427 motions, respectively); the DART+GMR and w/o FK stable subsets contain 395 and 464 motions, respectively. This is not the stricter hardware-candidate screen, and no upper-tail trimming is applied. Each motion averages over all playback frames and the same 19 body points; group means and medians give motions equal weight.

The remaining valid motions that do not pass the stability criteria have substantially higher mean errors: for PartiGen ($N = 87$), Global and root-relative MPJPE are 508.15 and 129.24 mm; for GT ($N = 72$), they are 1004.55 and 127.30 mm. These motions raise the all-valid means. Within the stable subsets, GT still has a higher Global mean, while the root-relative means are close. The

Table S9: Raw MPJPE within the simulation-stability-passing subsets. Values are mean / median (mm); no initial-translation correction or upper-tail trimming is applied.

Method	Stable motions	Global MPJPE	Root-relative MPJPE
PartiGen	409	198.00 / 97.08	42.84 / 37.98
GT	427	297.81 / 161.34	44.26 / 36.04
DART+GMR	395	247.33 / 145.29	44.37 / 36.64
w/o FK	464	252.47 / 140.92	45.20 / 37.53

height/inclination criteria do not rule out global position drift. Both groups use the same 29-DoF configuration with the same six wrist joints zeroed, but their samples are unpaired and their stability-passing subsets differ; these conditional differences do not establish generator superiority over GT. Non-passing motions are not automatically classified as falls.

The evaluation covers 500 requested runs per group; PartiGen statistics remain a subset of the 2,576 available references. Simulation stability, hardware candidacy, and real-robot completion are distinct outcomes.

S6.4 Selected Real-Robot Outcomes

From the 500 simulated PartiGen motions, tracking-error metrics including MPJPE and stability-related checks were used to select 100 hardware candidates. Instruction completion was not systematically established for this entire candidate pool. We then manually selected 30 distinct motions: ten upper-body, ten lower-body, five long-sequence, and five composite motions. Of these, 27 were executed on the G1; two large-amplitude jumps and one running motion were not attempted because the available venue was too small.

Table S10: Outcomes for manually selected hardware motions. Counts refer to distinct motions, not repeated executions. Success is assessed manually by the authors for instruction fulfillment and qualitative agreement with simulation. Unattempted motions are separate from execution failures.

Category	Selected	Attempted	Successful	Failed	Unattempted
Upper-body	10	10	10	0	0
Lower-body	10	7	7	0	3
Long-sequence	5	5	5	0	0
Composite	5	5	4	1	0
Total	30	27	26	1	3

The authors judged a motion successful when the robot fulfilled the text instruction and its execution was qualitatively consistent with simulation. Under this assessment, 26 of the 27 attempted motions succeeded (96.30%; Table S10). The single failure involved walking with both arms raised, during which GRIT failed to track the reference. Low foot clearance is a suspected contributor, but the cause has not been isolated. Some motions were repeated for video recording; these repetitions are not counted as additional motions. We report neither a first-attempt success rate nor a per-execution trial rate.

These results describe a screened, manually selected set under the adopted controller and venue conditions. They are not an estimate of hardware success over all 500 simulated motions, all 100 candidates, or the full test set, and the author assessment is not an independent blinded semantic evaluation.

S7 Extended Related Work

This section expands the literature review in the main paper, covering motion generation, anatomical structure, physical plausibility, and temporal modeling.

S7.1 TEXT-TO-MOTION GENERATION

Text-driven human motion generation has advanced with the emergence of large-scale datasets such as HumanML3D (Guo et al., 2022a), AMASS (Mahmood et al., 2019), and BABEL (Punnakkal et al., 2021), together with progress in deep generative modeling. Variational approaches model action-conditioned motion distributions (Guo et al., 2020; Petrovich et al., 2021), while TEMOS (Petrovich et al., 2022) addresses diverse motion generation from text. Diffusion-based methods, including MDM (Tevet et al., 2022), MotionDiffuse (Zhang et al., 2022), and ReMoDiffuse (Zhang et al., 2023), provide another route to text-conditioned synthesis. Parallel to continuous diffusion, discrete token-based formulations such as TM2T and MotionGPT (Guo et al., 2022b; Jiang et al., 2023) explore motion as a language-like representation.

Complementary efforts investigate unified motion representation learning (Zhu et al., 2023) and controllable diffusion priors (Karunratanakul et al., 2023; Shafir et al., 2023). DART and TextOp further explore autoregressive generation and interactive motion control (Zhao et al., 2025; Xie et al., 2026). Language-conditioned humanoid control has also been studied through motion latent guidance without retargeting (Li et al., 2026). Other robot-oriented work addresses general motion retargeting (Araujo et al., 2025) and whole-body teleoperation and imitation (Ze et al., 2025). These approaches concern different stages of the generation-to-execution pipeline; retargeting and tracking methods should be distinguished from text-conditioned motion generators.

These developments provide complementary choices of representation, conditioning, and temporal generation. Our focus is how body-part information enters feature processing within a shared motion representation. A whole-body representation does not itself imply the absence of anatomical structure, nor does its use alone establish interference between body parts.

S7.2 STRUCTURE-AWARE AND PHYSICALLY PLAUSIBLE MOTION GENERATION

Structure-aware motion modeling incorporates anatomical information into representations and feature processing. AttT2M (Zhong et al., 2023) explores part-aware attention, while SALAD (Hong et al., 2025) introduces skeleton-aware latent diffusion for motion generation and editing. Hierarchical approaches address fine-grained control through semantic graphs (Jin et al., 2023) and multi-stage generation of human-object interactions (Pi et al., 2023).

Recent efforts also address generation across heterogeneous skeletons, including UniMoGen (Khani et al., 2025), NECromancer (Xu et al., 2026), and AnyTop (Gat et al., 2025). These works broaden the range of skeletal structures supported by motion models. Related use of skeletal conditioning extends beyond motion: HumanSD (Ju et al., 2023) studies skeleton-guided human image generation. It therefore provides an example of structural conditioning in a different output domain.

Another direction focuses on physical and kinematic plausibility. PhysDiff (Yuan et al., 2023) incorporates physics guidance into human motion diffusion, while InterDiff (Xu et al., 2023) addresses human-object interaction with physics-informed diffusion. Harmon (Jiang et al., 2024) studies whole-body humanoid motion generation from language descriptions. Such work motivates examining both the quality of generated reference motions and their behavior under downstream control.

PartiGen introduces anatomical structure through fixed masks on raw frame features before independent part-specific Q/K/V projections. Joint-angle channels are selected by body part, while root/contact features and all joint-angle increments remain shared. Head outputs are fused into a shared stream and represented by a single motion latent. FK-based supervision is applied during generator training without a physical simulator or controller gradients. The system adopts pretrained GRIT (mrzuang, 2026) as its frozen execution component; reference-motion quality and closed-loop execution are evaluated at their respective stages.

S7.3 TEMPORAL MODELING IN MOTION GENERATION

Motion duration is relevant to instructions involving concurrent or sequential actions. Motion diffusion systems commonly specify a target horizon or length as part of generation (Tevet et al., 2022; Shafir et al., 2023). However, a single instruction can admit multiple valid durations, so tempo-

ral modeling must accommodate variation rather than assume that semantics uniquely determines sequence length.

Temporal generation is also studied in other modalities. AudioLDM (Liu et al., 2023) uses latent diffusion for text-to-audio synthesis, neural codec language models support text-to-speech generation (Wang et al., 2023), and StreamingT2V (Henschel et al., 2025) explores autoregressive extension of video sequences. These examples illustrate different representations and generation strategies for temporal data; they do not directly establish a motion-specific termination mechanism.

PartiGen generates motion primitives and selects the endpoint at frame resolution. A complete candidate sequence is generated before an EOS threshold is applied; the cutoff may lie within a primitive. This is a post-generation selection stage and does not stop autoregressive generation online.

Discussion. The **PartiGen** generator combines part-aware feature learning, a shared 128-dimensional latent representation, FK-based regularization, and EOS-based duration prediction. Body-part grouping guides feature processing without imposing independent part-specific latent variables. The shared representation provides a common space for modeling local movements and their whole-body context. The system pairs this generator with the adopted frozen GRIT controller for execution; its methodological contributions concern generation.