OMG: Omni-Modal Motion Generation for Generalist Humanoid Control

Author Names Omitted for Anonymous Review.

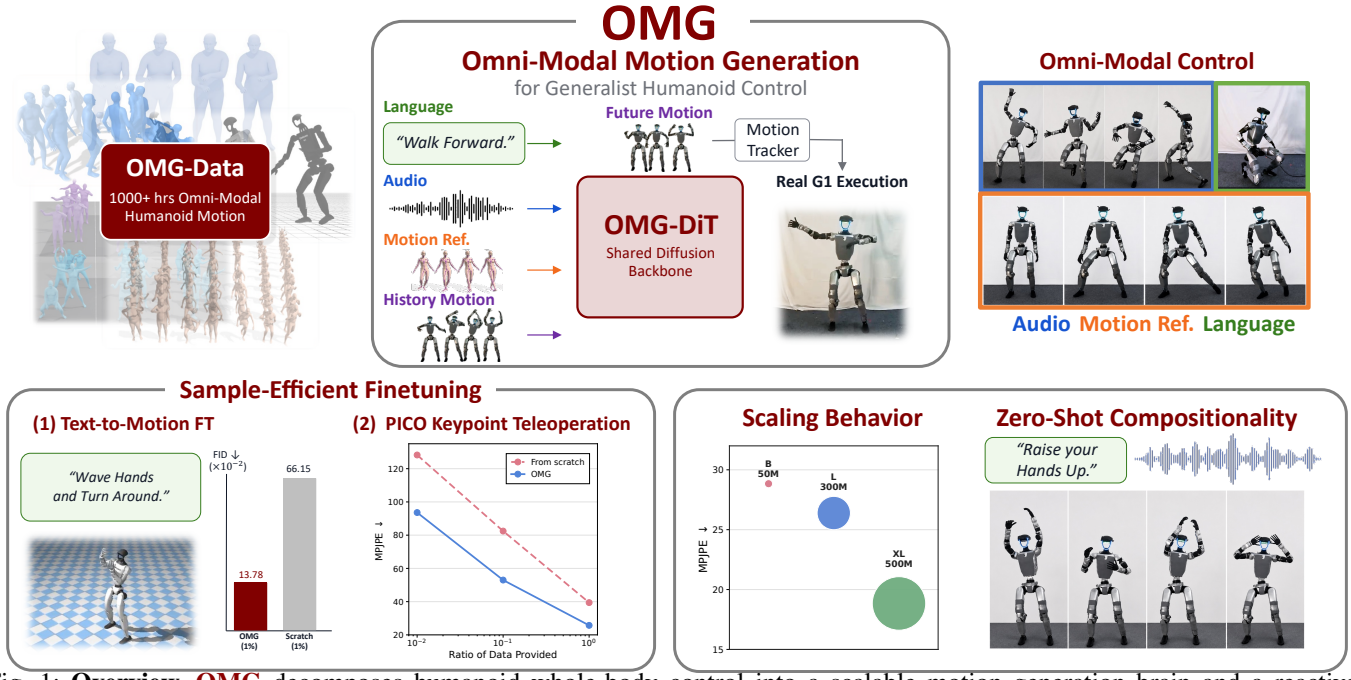


Fig. 1: **Overview.** **OMG** decomposes humanoid whole-body control into a scalable motion generation brain and a reactive motion tracking cerebellum. Built on **OMG-Data**, a curation of 1000+ hours omni-modal humanoid motion data, **OMG-DiT** maps language, audio, human reference, and their compositions into robot-executable future motions, which are deployed on a Unitree G1 in real time, paired with a pretrained motion tracker [4]. This unified generator-tracker hierarchy enables general-purpose omni-modal control and sample-efficient adaptation to new tasks and modalities.

***Abstract*—Humanoid whole-body control has made significant progress in recent years, yet existing approaches remain limited to few-skill policies with heavy reward engineering, or motion trackers that are difficult to extend to new input modalities. We argue that the key to general-purpose humanoid control is to build a scalable *brain*, a module capable of reasoning with diverse conditioning modalities, atop a reactive motion tracking *cerebellum*, mirroring the hierarchical structure of biological motor systems. Two challenges arise in realizing this vision: acquiring a vast amount of high-quality data to achieve general purpose control, and equipping the generator with the capability to condition on compositional, extensible multi-modal inputs. We present OMG, which addresses these challenges with a meticulous data curation, filtering and labeling pipeline, as well as a diffusion-based motion generation backbone that conditions on language, audio, and human reference motions. Extensive experiments validate OMG as an omni-modal whole-body controller exhibiting state-of-the-art performance, model scaling behavior and efficient adaptation to new distributions and modalities, marking a concrete step toward foundation models for humanoid robots.**

I. Introduction

Humanoid whole-body control has made rapid progress, enabling agile locomotion and dexterous loco-manipulation with reinforcement learning [74, 40, 68, 23]. However, most existing policies remain tied to specific skills and reward designs, making them difficult to scale. Motion tracking provides a more scalable alternative by learning to follow reference motions [32, 71, 24], but tracking alone largely replays given motions and offers limited autonomy under high-level, multi-modal human intent.

A promising abstraction is a generator-tracker hierarchy [45, 51, 57, 70, 56], where an upstream motion generator translates high-level conditions into future whole-body trajectories, and a downstream tracker executes them on the robot [4, 32]. Yet realizing this paradigm requires overcoming two key challenges. First, high-quality humanoid motion data is scarce, fragmented, and heterogeneous, unlike web-scale text or visual

data [43, 38]. Second, the motion generator must support diverse, composable, and extensible control modalities while remaining adaptable to new control interfaces.

To this end, we introduce **OMG**, an **O**mni-Modal **M**otion **G**eneration framework for generalist humanoid whole-body control. At the data level, we curate **OMG-Data**, a large-scale multi-modal humanoid motion corpus of **1000+** hours, by retargeting, filtering, annotating, and aligning heterogeneous motions into the Unitree G1 embodiment. At the model level, we instantiate **OMG-DiT**, a diffusion transformer backbone that maps language, audio, human-reference motions, and their compositions into robot-executable future trajectories. Importantly, new modalities can be incorporated through lightweight condition encoders while reusing the pretrained motion prior; unseen control signal combinations can be composed at inference through guidance.

Extensive experiments demonstrate **OMG**'s capabilities in producing high-quality, physically executable motions across diverse modalities. Furthermore, **OMG** exhibits foundation-model-like properties, including predictable scaling, sample-efficient adaptation, and zero-shot composition of control signals. These results suggest that generalist humanoid control can be advanced not only by stronger low-level controllers, but also by scaling the motion-generation brain that interfaces human intents with physical execution. In summary, our contributions are as follows:

- We introduce **OMG**, an omni-modal motion generation framework for generalist humanoid whole-body control, unifying diverse conditioning modalities under a shared backbone, mapping diverse human intents into physically executable robot trajectories.
- We curate **OMG-Data**, a large-scale omni-modal humanoid motion corpus of **1000+** hours. Through a unified pipeline of retargeting, filtering, and annotation, we align motions into a unified motion space, making supervised scaling of humanoid motion generation possible.
- We introduce **OMG-DiT**, a diffusion-based motion generation backbone that supports extensible and compositional conditioning from language, audio, and human reference motions, allowing new modalities to be incorporated through lightweight adaptation.
- Through extensive experiments, we validate **OMG** as an omni-modal whole-body controller, demonstrating foundation-model-like properties including model scaling behavior, few-shot adaptation, and zero-shot composition of control signals.

II. Related Works

a) Humanoid Whole-Body Control and Motion Tracking.: Recent advances in reinforcement learning have greatly expanded the capability of humanoid whole-body controllers [74, 40, 68, 23]. However, these systems are often specialized to particular task objectives and reward designs. Motion tracking [12, 11] offers a more scalable alternative, training a policy to execute human reference motions on the humanoid. Recent works [63, 32, 4, 71, 24] scale this paradigm with broader motion corpora, yielding stronger controllers that robustly track out-of-domain reference motions. These trackers form a powerful low-level execution layer for humanoid control, yet they assume availability of reference motions or low-level commands at inference time, leaving open the upstream problem of generating robot-executable motions from high-level and multi-modal conditions.

b) Interactive and Multi-Modal Motion Generation.: Motion generation [50, 67] addresses the complementary problem of translating high-level conditions into motion references. In graphics, modern generative architectures [58, 37] have substantially advanced motion generation systems, enabling expressive conditioning on text, audio, keypoints, and human references [27, 66, 17, 42], while benefiting from scaling data [8] and model parameters. However, most such models remain human-motion generators that operate in an offline manner, neglecting the need for *real-time interactive* control. Conversely, coupling motion generation with tracking systems has garnered increasing interest in the humanoid community [70, 24, 51, 45, 56], yet these systems are typically limited to a single command modality or narrow task domain. This highlights an important missing intersection: general-purpose, omni-modal motion generation for real-time, robot-executable humanoid control.

III. OMG-Data: Scaling Multi-Modal Data for Humanoid Motion Generation

A central obstacle to scaling humanoid whole-body motion generation is the lack of a unified, robot-executable, and multi-modal motion corpus. Existing motion datasets are highly fragmented, and datasets differ substantially in their available label modality, motion quality, and annotation granularity. To address this challenge, we curate **OMG-Data**, a large-scale omni-modal humanoid motion corpus unified in the Unitree G1 embodiment of **1174.66** hours, realized through a carefully designed pipeline of curation, preprocessing, retargeting, annotating and filtering.

a) Data Curation and Preprocessing.: To establish a comprehensive corpus, we aggregate a diverse set of publicly available motion datasets across graphics and humanoid domains [33, 10, 16, 34, 8, 9, 18, 27, 20, 69, 25, 14, 42, 3, 2, 15, 66]. The consolidated dataset spans multiple modalities, encompassing text labels, audios, and human reference motions. Upon curation, the files first go through a validation check: all corrupted files, samples with broken links, and instances suffering from severe missing frames or invalid joint attributes are purged. For multi-modal sequence blocks that contain synchronized audio inputs (e.g., dance or co-speech gesture datasets like AIST++ [20] and BEAT2 [27]), we additionally preprocess the raw acoustic streams and audio feature vectors to be frame-aligned and temporally synchronized with corresponding motion clip trajectories.

b) Motion Retargeting and Annotation.: Since human motion representations vary across datasets, unifying the topology of motion representation becomes essential. Using General Motion Retargeting (GMR, [62, 63, 1]), we retarget

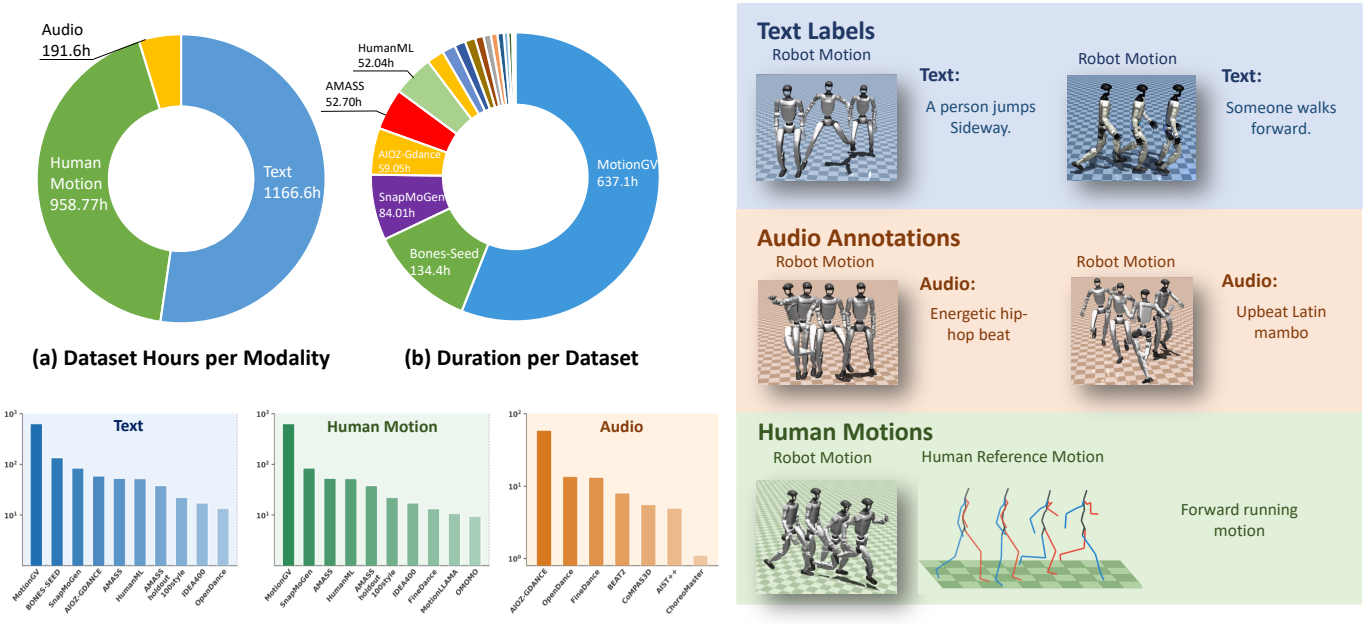


Fig. 2: **Dataset Statistics of OMG-Data**. We curate a large-scale omni-modal humanoid motion corpus by aggregating heterogeneous datasets and unifying them into the Unitree G1 motion space. Left: processed data statistics across conditioning modalities and source datasets. Right: representative conditioning modalities, including language, audio, and human reference motions.

all motion representations to the motion space of Unitree G1 humanoids. To enrich datasets that lack native language descriptions or require granular cross-modal semantics, we render the retargeted motion sequences in simulation and label them with `Seed-1.8` [44], using multi-view rendered videos or representative keyframes as visual inputs. We then segment motions into training clips according to language annotation boundaries, audio phrase cuts, uniform windows, or sliding windows for long episodes.

c) Simulation-In-The-Loop Filtering.: To prevent kinematically invalid, self-colliding, or physically impossible joint configurations from contaminating the training mixture, we apply a filtering step via tracker execution in simulation. Concretely, given a sample, we roll out this motion sequence in the MuJoCo rigid-body simulator [53] using a joint-space PD tracking controller. We design the fall detection heuristics as follows: Let h_t denote root height and θ_t denote root tilt angle. We mark a frame as a fall frame if $h_t < 0.20$, $\theta_t > 85°$, or $(h_t < 0.35 \wedge \theta_t > 60°)$. If the fall frame condition persists for over 10 consecutive frames, the trajectory is rejected. This physics-in-the-loop screening ensures that the processed dataset contains dynamically feasible and safe trajectories for humanoid deployment. Further details are provided in the Supplementary Material.

IV. OMG-DiT: Unified Diffusion Backbone for Motion Generation

Equipped with this large-scale multi-modal motion corpus, we instantiate **OMG-DiT**, a unified diffusion backbone that maps heterogeneous control modalities into robot-executable whole-body trajectories in real time. The key design principle is to decouple the *motion prior* from the *condition interface*: a shared denoising backbone maps the distribution of feasible motions, while modality-specific encoders translate high-level intents into steerable conditions over the shared manifold.

A. Problem Formulation

We consider the problem of controlling a humanoid robot to perform whole-body motions specified by multi-modal conditions. Given a set of conditioning variables $\mathcal{C} = \{c^{(1)}, c^{(2)}, \ldots, c^{(N)}\}$, which may include language, audio and other modalities, our goal is to learn a policy π that produces physically realizable whole-body actions $\mathbf{a}_t$ conditioned on $\mathcal{C}$ and history observations $\mathbf{o}_{\leq t}$. We decompose this policy learning problem into a cascaded generation-and-tracking formulation, or more commonly termed planning and inverse dynamics models in manipulation [6, 7]:

$$\pi(\mathbf{a}_{t:t+H}, \mathbf{o}_{t+1:t+H+1} \mid \mathbf{o}_{\leq t}, \mathcal{C}) \approx \underbrace{\pi_\phi(\mathbf{o}_{t+1:t+H+1} \mid \mathbf{o}_{t-L:t}, \mathcal{C})}_{\text{Motion Generation/Planning}} \cdot \underbrace{\pi_\psi(\mathbf{a}_{t:t+H} \mid \mathbf{o}_{t-L:t}, \mathbf{o}_{t+1:t+H+1})}_{\text{Motion Tracking/IDM}}.$$

where π_ϕ is a high-level motion generator that predicts future whole-body reference observations from multi-modal conditions, and π_ψ is a low-level motion tracker that converts these references into executable robot actions. In this work, we focus on motion generation, and leverage pre-existing general-purpose trackers (HoloMotion [4]) as our low-level tracker.

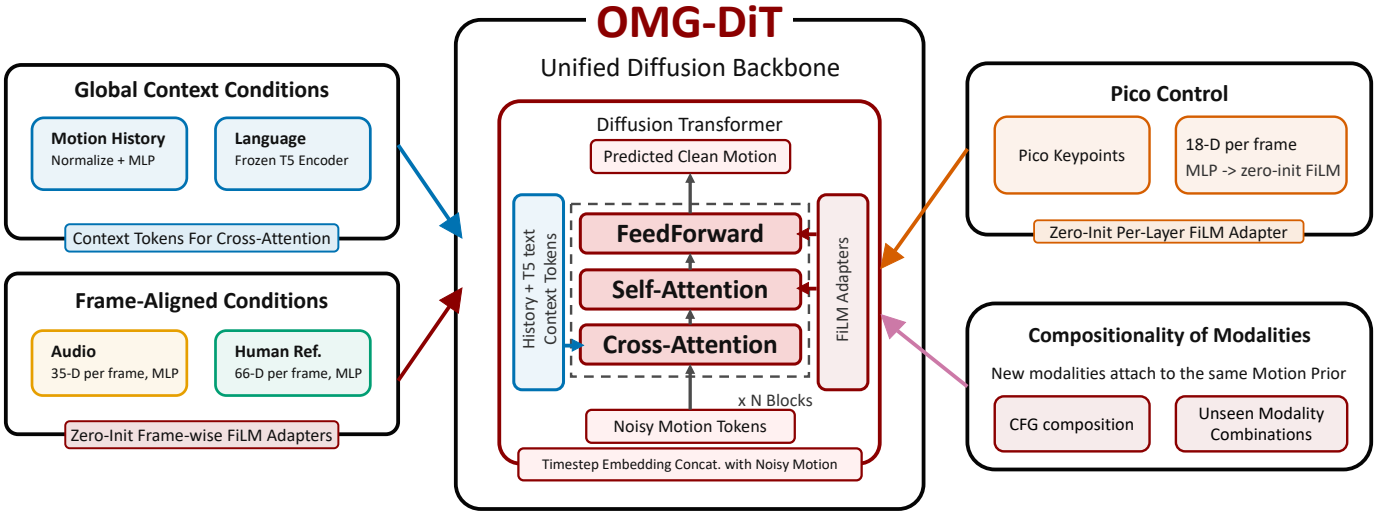


Fig. 3: **OMG-DiT** learns a shared diffusion backbone while enabling conditioning with modality-specific encoders. History motion and language are injected as global context tokens via cross-attention, whereas frame-aligned signals (i.e., audio and human reference motions) are injected through FiLM [39] adapters. New modalities are attached non-invasively through zero-initialized adapters, and multiple conditions can be composed at inference via classifier-free guidance.

B. Diffusion Transformer Backbone

a) Motion Representation.: We represent each frame in a canonical root-centric coordinate frame defined by the last observed state $\mathbf{x}_t = [\tilde{\mathbf{p}}_t, \tilde{\mathbf{r}}_t, \boldsymbol{\theta}_t, \tilde{\mathbf{j}}_t] \in \mathbb{R}^{125}$, where $\tilde{\mathbf{p}}_t$ and $\tilde{\mathbf{r}}_t$ denote the canonicalized root position and rotation, $\boldsymbol{\theta}_t$ denotes joint angles, and $\tilde{\mathbf{j}}_t$ denotes body-link positions. This root-centric representation removes global translation and heading ambiguity while preserving the full-body geometric structure required for motion generation and downstream tracking.

b) Model Architecture.: **OMG-DiT** is instantiated as a Diffusion Transformer (DiT [37]) backbone trained with an $\mathbf{x}$-prediction objective [21]. Drawing inspiration from recent advances in pixel-space diffusion [21, 30], we generate directly in motion space, avoiding training of motion encoders. History motion is injected via cross-attention, whereas the denoising timesteps are injected via channel-wise concatenating with the noisy motion features before input projection.

c) Training Objective.: We train **OMG-DiT** as a conditional diffusion model generating future motion trajectories $\mathbf{x}_{t+1:t+H}$ conditioned on recent history $\mathbf{x}_{t-L:t}$ and a subset of available conditioning modalities $\mathcal{C}_s \subseteq \mathcal{C}$. Given a diffusion timestep τ, we corrupt the future segment as

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

To enable classifier-free guidance and improve robustness, we randomly drop each available conditioning modality during training with probability p_{drop}. The resulting training objective is:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}, \mathcal{C}_s, \tau, \boldsymbol{\epsilon}} \left[\| \hat{\mathbf{x}}_{t+1:t+H} - \mathbf{x}_{t+1:t+H} \|_2^2 \right],$$
$$\text{where } \hat{\mathbf{x}}_{t+1:t+H} = \mathbf{x}_\theta\left(\mathbf{x}^{\tau}_{t+1:t+H}, \tau, \mathbf{x}_{t-L:t}, \mathcal{C}_s\right).$$

This objective trains a single denoising backbone to model the feasible Unitree G1 motion manifold, while allowing different condition encoders to steer generation through a shared motion prior.

C. Omni-Modal Condition Encoding

a) Pretraining Condition Encoders.: During pretraining, **OMG-DiT** supports conditioning on language, audio, and human-reference motion. Language annotations are encoded by a frozen T5 encoder [41] and injected into each DiT block through cross-attention, alongside encoded history frames. Frame-aligned modalities, including audio and human-reference motion, are projected through MLPs and injected into corresponding frames through per-layer FiLM modulation [39].

b) Task-specific Finetuning Encoders.: To enable sample-efficient adaptation to downstream tasks with new modalities, we adopt non-invasive injection methods with zero-initialization. For example, while fine-tuning for Pico keypoint-conditioned teleoperation, Pico keypoints are represented as 18D frame-aligned features, and injected through zero-initialized FiLM [39] adapters.

V. Experiments

Our experiments aim to answer the following questions: (1) How does **OMG** compare against previous hierarchical humanoid control frameworks, such as graphics-based generation plus retargeting; (2) Does pretraining lead to more efficient adaptation to new modalities/datasets compared to training from scratch; and (3) Does the shared diffusion backbone exhibit foundation-model-like properties, including data scaling and zero-shot composition of control signals?

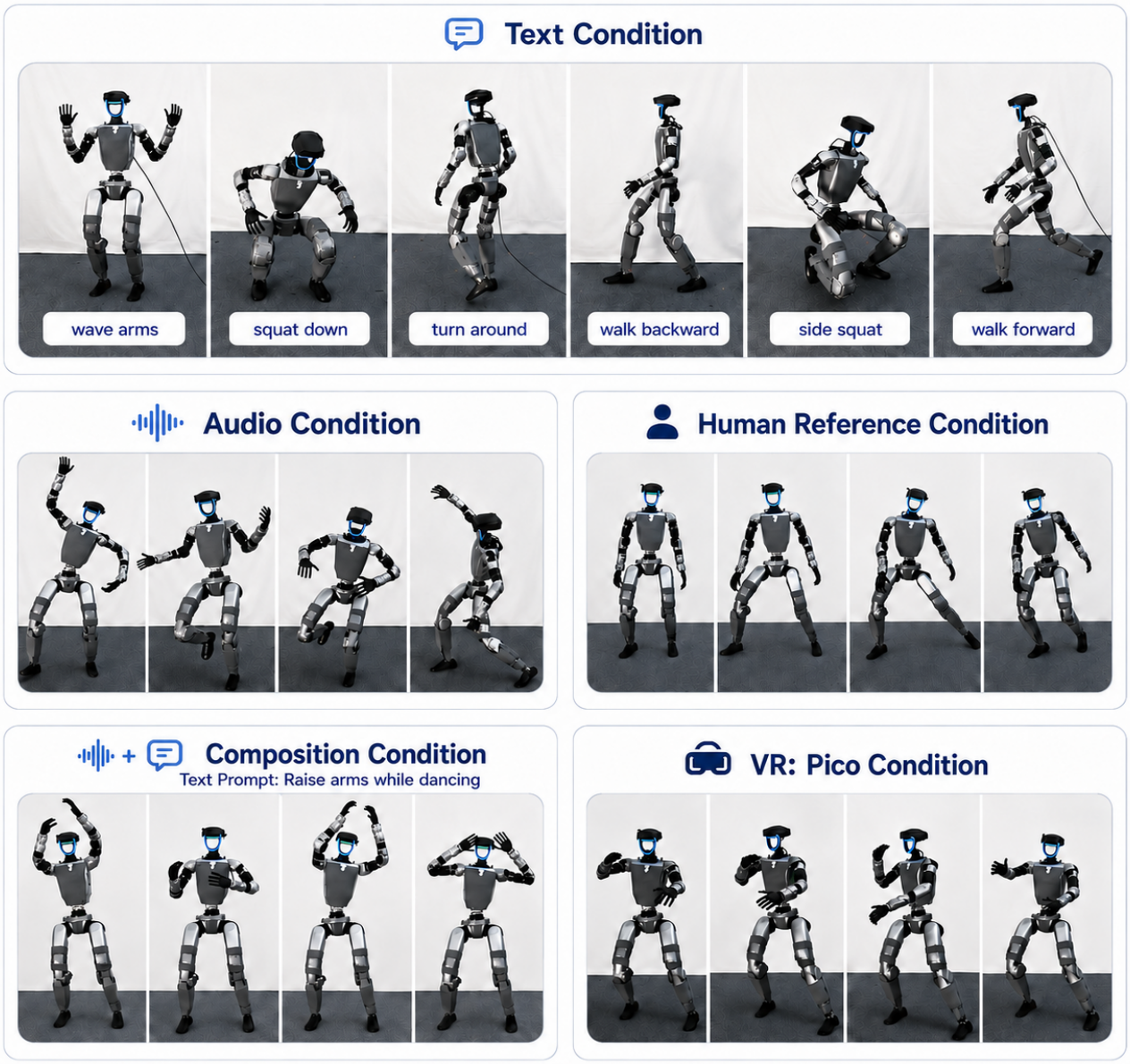


Fig. 4: **Real-World Omni-Modal Control.** **OMG** generates diverse Unitree G1 motions across various conditioning modalities in real time, executable in the real world.

A. Experiment Setup

a) Evaluation Protocol.: We evaluate **OMG** in two regimes: pretrained omni-modal motion generation and downstream finetuning. For pretraining, we consider language-, audio-, and human-reference-conditioned motion generation, where the model predicts future Unitree G1 whole-body trajectories from recent motion history and the corresponding control signal. For finetuning, we evaluate two downstream settings: text-conditioned generation on unseen data, and Pico keypoint-conditioned teleoperation. Unless otherwise specified, generated trajectories are executed by a pretrained HoloMotion [4] tracker. All evaluations use validation motions unseen during training.

b) Metrics.: We evaluate **OMG** along two axes: motion generation quality and tracking fidelity. For generation quality, we use modality-specific metrics, including Matching Score, R-Precision, FID, and Diversity [65] for language; BeatAlign, FID_k, FID_g, and PFC [20, 54] for audio; and MPJPE, global MPJPE, end-effector error, velocity error, and acceleration error [13, 56] for human-reference generation. For tracking fidelity, we report contact sliding, body jerk, tracker MPJPE, tracker global MPJPE, tracker velocity error and acceleration error, fall rate, and joint-limit violation rate [60, 5]. For downstream finetuning, we additionally report task-specific metrics, including keypoint tracking error [35] for Pico teleoperation. More details are provided in the supplementary material.

TABLE I: **Text-conditioned motion benchmark.** For better display, R@K, Fall, and J-Limit are reported in percentage. FID and C-Slide are scaled by 10^{-2} and 10^{-1}, respectively. Lower is better for metrics marked with ↓, and higher is better for metrics marked with ↑.

Methods	Motion Generation Quality								Tracking Fidelity and Execution Failures					
	FID ↓	Matching ↓	R@1 ↑	R@2 ↑	R@3 ↑	Diversity ↑	C-Slide ↓	Jerk ↓	MPJPE ↓	g-MPJPE ↓	E_vel ↓	E_acc ↓	Fall ↓	J-Limit ↓
GENMO	43.78	1.33	12.99	23.83	30.86	1.28	**2.66**	173.54	50.35	150.71	6.91	2.51	1.46	29.10
HYMotion	24.68	1.30	31.84	48.73	57.71	1.29	12.20	112.56	69.05	287.71	11.94	3.24	4.98	55.57
Kimodo-SMPLX-RP	38.44	1.31	26.27	39.36	46.48	1.25	4.52	189.39	43.51	198.96	7.80	2.64	2.64	23.63
Kimodo-G1-RP	46.36	1.32	25.59	36.13	44.63	1.21	3.92	81.44	**37.25**	125.04	5.72	1.68	1.46	19.14
Kimodo-G1-SEED	50.22	1.33	22.46	33.69	41.11	1.20	4.30	99.47	37.26	161.63	6.38	1.84	1.56	20.70
OMG-L	9.55	1.14	58.50	74.90	81.45	**1.39**	3.68	56.03	39.06	**98.37**	**5.11**	**1.47**	**0.29**	**14.94**
OMG-XL	**6.03**	**1.11**	**65.43**	**80.37**	**86.91**	**1.39**	3.98	**52.31**	42.68	111.41	5.48	1.53	0.78	19.34

TABLE II: **Audio-conditioned motion benchmark.** For better display, Fall and J-Limit are reported in percentage. Lower is better for metrics with ↓, and higher is better for metrics with ↑.

Methods	Audio Alignment and Target Motion Quality				Target Physical Quality and Tracker Fidelity							
	BeatAlign ↑	FID_k ↓	FID_g ↓	PFC ↓	C-Slide ↓	Jerk ↓	MPJPE ↓	g-MPJPE ↓	E_vel ↓	E_acc ↓	Fall ↓	J-Limit ↓
GENMO	0.57	506.07	9.42	**0.02**	**0.03**	**72.56**	**35.73**	111.69	**3.87**	**1.22**	0.39	12.50
LODGE	**0.58**	390.06	11.98	1.41	0.65	471.45	149.19	363.24	18.89	6.96	39.45	66.80
Bailando	0.56	134.90	15.71	0.39	0.25	117.45	38.30	106.39	5.55	1.99	**0.00**	**1.76**
OMG-L	0.49	59.20	0.74	0.57	0.41	95.73	41.07	**95.10**	6.10	1.90	**0.00**	14.26
OMG-XL	0.51	**40.46**	**0.63**	0.64	0.43	106.06	41.47	97.60	6.03	1.94	**0.00**	15.43

TABLE III: **Human-reference-conditioned motion benchmark.** For better display, Fall and J-Limit are reported in percentage. Lower / higher is better for metrics marked with ↓ / ↑.

Methods	Target Motion and Physical Quality							Tracker-Executed Target Fidelity and Failures					
	MPJPE ↓	g-MPJPE ↓	EE Error ↓	E_vel ↓	E_acc ↓	C-Slide ↓	Jerk ↓	Target MPJPE ↓	Target g-MPJPE ↓	Target E_vel ↓	Target E_acc ↓	Fall ↓	J-Limit ↓
GMR	81.95	223.69	219.23	11.72	10.41	0.45	404.80	90.54	249.47	12.52	5.22	11.33	68.75
NMR	173.77	390.82	477.08	14.87	7.43	0.77	97.75	102.28	299.27	11.26	3.32	14.84	87.89
PHC	51.44	153.44	152.18	5.06	3.97	0.45	131.38	47.80	115.92	5.86	1.99	3.71	39.45
OmniRetarget	56.90	119.71	131.14	5.55	4.87	**0.31**	188.92	44.34	111.58	5.94	2.35	0.98	24.22
OMG-L	26.40	64.64	72.39	6.16	3.91	0.35	**61.01**	**42.81**	**110.66**	5.63	**1.62**	**0.20**	**22.66**
OMG-XL	**18.84**	**47.83**	**52.26**	**5.03**	**3.69**	0.32	64.04	43.07	112.60	**5.55**	1.63	0.39	23.05

B. Pretraining Experiments

a) Text to Motion.: We compare OMG against recent human motion generation models, including GENMO [17], HYMotion [55], and Kimodo [42], where generated human motions are retargeted to the Unitree G1 when necessary via GMR [62]. As shown in Table I, **OMG** achieves substantial gains in motion generation quality, while retaining high tracking fidelity, showcasing its capability to generate high-quality and robot-executable motions conditioned on text.

b) Audio to Motion.: We next evaluate audio-conditioned motion generation, comparing against both generalist and dedicated audio-to-motion baselines, including GENMO [17], LODGE [19], and Bailando [46]. As shown in Table II, **OMG** maintains superior performance in audio alignment and motion quality, while achieving comparable performance in tracking fidelity.

c) Human Reference to Motion.: Finally, we evaluate human-reference-conditioned motion generation. We compare against both optimization and learning based retargeting methods, including GMR [62], NMR [72], PHC [31], and OmniRetarget [59]. As shown in Table III, **OMG** substantially outperforms these baselines in both motion quality and tracking fidelity, demonstrating that the learned generator can serve as an implicit yet highly-effective retargeting module for humanoid control.

C. Finetuning Experiments

a) Few-shot Text-to-Motion Finetuning.: To evaluate whether motion-generation pretraining provides positive transfer to unseen data distributions, we finetune **OMG** on AMASS-CMU, held out during pretraining. As shown in Table IV, the pretrained model consistently outperforms the from-scratch counterpart across all finetuning fractions. Notably, with only 1% of the target data, finetuning already yields comparable motion quality to training from scratch with the full data budget.

b) Pico Keypoint-Based Teleoperation.: To test if positive transfer extends to novel *modalities*, we adapt **OMG** to

TABLE IV: **Text-to-Motion Finetuning.**

Finetuning Data %	FID ↓	R@1 ↑	R@2 ↑	R@3 ↑
From Scratch				
1%	0.6615	0.0430	0.0762	0.1025
10%	0.1763	0.1162	0.2080	0.2998
100%	0.1001	0.1689	0.2764	0.3779
OMG *Finetuned*				
1%	0.1378	0.1533	0.2637	0.3428
10%	0.0886	0.1816	0.3096	0.3994
100%	**0.0827**	**0.2021**	**0.3115**	**0.4141**

TABLE V: **Pico Keypoint-Based Teleoperation.**

Finetuning Data %	MPJPE ↓	g-MPJPE ↓	E_vel ↓	E_acc ↓
From Scratch				
1%	128.24	448.09	24.34	17.13
10%	82.46	279.21	14.90	3.92
100%	39.42	134.85	8.03	2.95
OMG *Finetuned*				
1%	93.61	286.84	16.11	3.74
10%	52.99	164.91	9.91	3.09
100%	**25.73**	**77.01**	**5.70**	**2.62**

condition on Pico keypoints, enabling teleoperation. As shown in Table V, finetuned models consistently outperform training from scratch under the same data budgets, suggesting that the pretrained model serves as a reusable prior when incorporating a new control interface.

D. Analysis

a) Zero-shot Compositionality.: A hallmark of foundation models is *compositional generalization*, i.e. the ability to combine capabilities learned independently. We test this by composing novel language and audio conditions at inference. As shown in Figure 4, the humanoid successfully follows both the language instruction and the audio rhythm simultaneously, producing motions that are qualitatively distinct from either condition applied alone.

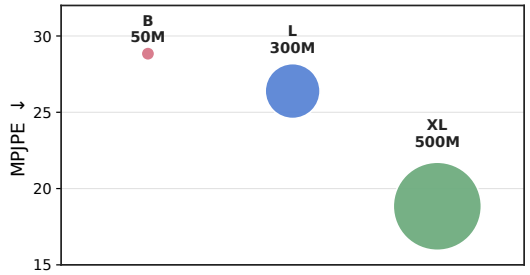


Fig. 5: **Scaling OMG-DiT.**

b) Scaling Behavior.: Finally, we ask whether motion generation is a scalable objective: do larger diffusion backbones yield better humanoid motion quality, given the same data and evaluation protocol? To answer this, we pretrain three **OMG-DiT** variants with increasing numbers of parameters. As shown in Figure 5, performance improves consistently with model size, suggesting that humanoid motion generation benefits from increased model capacity.

VI. Conclusion

We present **OMG**, an omni-modal motion generation framework for generalist humanoid whole-body control. **OMG** combines **OMG-Data**, a large-scale corpus unified in Unitree G1 motion space, with **OMG-DiT**, a shared diffusion backbone that maps language, audio, human references, and their compositions into robot-executable future motions. Experiments show strong motion quality, physical executability, sample-efficient adaptation, scaling behavior, and zero-shot compositionality, positioning scaling motion generation as a promising path toward humanoid foundation models.

VII. Limitations

This work has several limitations. First, our training data mainly cover flat-ground motions; extending **OMG** to uneven and in-the-wild terrain remains challenging. Second, **OMG** uses a modular generator-tracker hierarchy, however, we focus only on motion generation for simplicity. We believe jointly adapting the generator and tracker, or incorporating execution feedback into generation, is an important direction for improving performance of the system, which we leave for future work.

References

[1] 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.

[2] Bermet Burkanova, Payam Jome Yazdian, Chuxuan Zhang, Trinity Evans, Paige Tuttösí, and Angelica Lim. Salsa as a nonverbal embodied language – the compas3d dataset and benchmarks, 2025. URL https://arxiv.org/abs/2507.19684.

[3] Kang Chen, Zhipeng Tan, Jin Lei, Song-Hai Zhang, Yuan-Chen Guo, Weidong Zhang, and Shi-Min Hu. Choreomaster: choreography-oriented music-driven dance synthesis. *ACM Trans. Graph.*, 40(4), July 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459932. URL https://doi.org/10.1145/3450626.3459932.

[4] Maiyue Chen, Kaihui Wang, Bo Zhang, Xihan Ma, Zhiyuan Yang, Yi Ren, Qijun Huang, Zihao Zhu, Yucheng Wang, and Zhizhong Su. Holomotion-1 technical report, 2026. URL https://arxiv.org/abs/2605.15336.

[5] Hanbyel Cho, Sang-Hun Kim, Jeonguk Kang, and Donghan Koo. Safeflow: Real-time text-driven humanoid whole-body control via physics-guided rectified flow and selective safety gating. *arXiv preprint arXiv:2603.23983*, 2026.

[6] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

[7] Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, Leslie Kaelbling, et al. Video language planning. In *International Conference on Learning Representations*, volume 2024, pages 31138–31155, 2024.

[8] Ke Fan, Shunlin Lu, Minyue Dai, Runyi Yu, Lixing Xiao, Zhiyang Dou, Junting Dong, Lizhuang Ma, and Jingbo Wang. Go to zero: Towards zero-shot motion generation with million-scale data, 2025. URL https://arxiv.org/abs/2507.07095.

[9] Chuan Guo, Inwoo Hwang, Jian Wang, and Bing Zhou. Snapmogen: Human motion generation from expressive texts, 2025. URL https://arxiv.org/abs/2507.09122.

[10] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *arXiv:2102.04942*, 39(4), 2020.

[11] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[12] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. *arXiv preprint arXiv:2403.04436*, 2024.

[13] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. Avatarposer: Articulated full-body pose tracking from sparse motion sensing. In *European conference on computer vision*, pages 443–460. Springer, 2022.

[14] Boeun Kim, Hea In Jeong, JungHoon Sung, Yihua Cheng, Jeongmin Lee, Ju Yong Chang, Sang-Il Choi, Younggeun Choi, Saim Shin, Jungho Kim, and Hyung Jin Chang. Personabooth: Personalized text-to-motion generation. *arXiv preprint arXiv:2503.07390*, 2025.

[15] Nhat Le, Thang Pham, Tuong Do, Erman Tjiputra, Quang D. Tran, and Anh Nguyen. Music-driven group choreography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[16] Jiaman Li, Jiajun Wu, and C Karen Liu. Object motion guided human motion synthesis. *ACM Trans. Graph.*, 42 (6), 2023.

[17] Jiefeng Li, Jinkun Cao, Haotian Zhang, Davis Rempe, Jan Kautz, Umar Iqbal, and Ye Yuan. Genmo: A generalist model for human motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[18] Ronghui Li, Junfan Zhao, Yachao Zhang, Mingyang Su, Zeping Ren, Han Zhang, Yansong Tang, and Xiu Li. Finedance: A fine-grained choreography dataset for 3d full body dance generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10234–10243, 2023.

[19] Ronghui Li, YuXiang Zhang, Yachao Zhang, Hongwen Zhang, Jie Guo, Yan Zhang, Yebin Liu, and Xiu Li. Lodge: A coarse to fine diffusion network for long dance generation guided by the characteristic dance primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1524–1534, 2024.

[20] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation, 2021.

[21] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

[22] Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, et al. Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning. *arXiv preprint arXiv:2511.04131*, 2025.

[23] Yitang Li, Yuanhang Zhang, Wenli Xiao, Chaoyi Pan, Haoyang Weng, Guanqi He, Tairan He, and Guanya Shi. Hold my beer: Learning gentle humanoid locomotion and end-effector stabilization control. *arXiv preprint arXiv:2505.24198*, 2025.

[24] Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *arXiv preprint arXiv:2508.08241*, 2025.

[25] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 2023.

[26] Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7353–7363, 2025.

[27] Haiyang Liu, Zihao Zhu, Giorgio Becherini, Yichen Peng, Mingyang Su, You Zhou, Xuefei Zhe, Naoya Iwamoto, Bo Zheng, and Michael J. Black. Emage: Towards unified holistic co-speech gesture generation via expressive masked audio gesture modeling, 2024. URL https://arxiv.org/abs/2401.00374.

[28] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

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

2017.

[30] Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158*, 2026.

[31] Zhengyi Luo, Jinkun Cao, Alexander W. Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *International Conference on Computer Vision (ICCV)*, 2023.

[32] Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castaneda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.

[33] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[34] Ian Mason, Sebastian Starke, and Taku Komura. Real-time style modelling of human locomotion via feature-wise transformations and local motion phases. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 5(1):1–18, 2022.

[35] Ruiqian Nai, Boyuan Zheng, Junming Zhao, Haodong Zhu, Sicong Dai, Zunhao Chen, Yihang Hu, Yingdong Hu, Tong Zhang, Chuan Wen, et al. Humanoid manipulation interface: Humanoid whole-body manipulation from robot-free demonstrations. *arXiv preprint arXiv:2602.06643*, 2026.

[36] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[38] Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

[39] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[40] Ilija Radosavovic, Sarthak Kamat, Trevor Darrell, and Jitendra Malik. Learning humanoid locomotion over challenging terrain. *arXiv preprint arXiv:2410.03654*, 2024.

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

[42] Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, Jiefeng Li, Chen Tessler, Edy Lim, Eugene Jeong, Sam Wu, Ehsan Hassani, Michael Huang, Jin-Bey Yu, Chaeyeon Chung, Lina Song, Olivier Dionne, Jan Kautz, Simon Yuen, and Sanja Fidler. Kimodo: Scaling controllable human motion generation. *arXiv:2603.15546*, 2026.

[43] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

[44] Bytedance Seed. Seed1. 8 model card: Towards generalized real-world agency. *arXiv preprint arXiv:2603.20633*, 2026.

[45] Agon Serifi, Ruben Grandia, Espen Knoop, Markus Gross, and Moritz Bächer. Robot motion diffusion model: Motion generation for robotic characters. In *SIGGRAPH asia 2024 conference papers*, pages 1–9, 2024.

[46] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11050–11059, 2022.

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[48] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[49] Zelin Tao, Zeran Su, Peiran Liu, Jingkai Sun, Wenqiang Que, Jiahao Ma, Jialin Yu, Jiahang Cao, Pihai Sun, Hao Liang, et al. Heracles: Bridging precise tracking and generative synthesis for general humanoid control. *arXiv preprint arXiv:2603.27756*, 2026.

[50] 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.

[51] Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit Bermano, and Michiel Van de Panne. Closd: Closing the loop between simulation and diffusion for multi-task character control. In *International Conference on Learning Representations*, volume 2025, pages 46506–46520, 2025.

[52] Andrea Tirinzoni, Ahmed Touati, Jesse Farebrother, Mateusz Guzek, Anssi Kanervisto, Yingchen Xu, Alessandro

Lazaric, and Matteo Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models. *arXiv preprint arXiv:2504.11054*, 2025.

[53] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

[54] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 448–458, 2023.

[55] Yuxin Wen, Qing Shuai, Di Kang, Jing Li, Cheng Wen, Yue Qian, Ningxin Jiao, Changhai Chen, Weijie Chen, Yiran Wang, et al. Hy-motion 1.0: Scaling flow matching models for text-to-motion generation. *arXiv preprint arXiv:2512.23464*, 2025.

[56] 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 arXiv:2602.07439*, 2026.

[57] Michael Xu, Yi Shi, KangKang Yin, and Xue Bin Peng. Parc: Physics-based augmentation with reinforcement learning for character controllers. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025.

[58] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4):1–39, 2023.

[59] Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C Karen Liu, Rocky Duan, and Guanya Shi. Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction. *arXiv preprint arXiv:2509.26633*, 2025.

[60] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13167–13178, 2022.

[61] Mingqi Yuan, Tao Yu, Wenqi Ge, Xiuyong Yao, Dapeng Li, Huijiang Wang, Jiayu Chen, Bo Li, Wei Zhang, Wenjun Zeng, et al. A survey of behavior foundation model: Next-generation whole-body control system of humanoid robots. *IEEE transactions on pattern analysis and machine intelligence*, 2025.

[62] Yanjie Ze, João Pedro Araújo, Jiajun Wu, and C. Karen Liu. Gmr: General motion retargeting, 2025. URL https://github.com/YanjieZe/GMR. GitHub repository.

[63] 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.

[64] Weishuai Zeng, Shunlin Lu, Kangning Yin, Xiaojie Niu, Minyue Dai, Jingbo Wang, and Jiangmiao Pang. Behavior foundation model for humanoid robots. *arXiv preprint arXiv:2509.13780*, 2025.

[65] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14730–14740, 2023.

[66] Jinlu Zhang, Zixi Kang, Libin Liu, Jianlong Chang, Qi Tian, Feng Gao, and Yizhou Wang. Opendance: Multimodal controllable 3d dance generation with large-scale internet data, 2025. URL https://arxiv.org/abs/2506.07565.

[67] 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.

[68] Yuanhang Zhang, Yifu Yuan, Prajwal Gurunath, Ishita Gupta, Shayegan Omidshafiei, Ali-akbar Agha-mohammadi, Marcell Vazquez-Chanlatte, Liam Pedersen, Tairan He, and Guanya Shi. Falcon: Learning force-adaptive humanoid loco-manipulation. *arXiv preprint arXiv:2505.06776*, 2025.

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

[70] Zewei Zhang, Kehan Wen, Michael Xu, Junzhe He, Chenhao Li, Takahiro Miki, Clemens Schwarke, Chong Zhang, Xue Bin Peng, and Marco Hutter. Learning whole-body humanoid locomotion via motion generation and motion tracking. *arXiv preprint arXiv:2604.17335*, 2026.

[71] Zhikai Zhang, Jun Guo, Chao Chen, Jilong Wang, Chenghuai Lin, Yunrui Lian, Han Xue, Zhenrong Wang, Maoqi Liu, Jiangran Lyu, et al. Track any motions under any disturbances. *arXiv preprint arXiv:2509.13833*, 2025.

[72] Qingrui Zhao, Kaiyue Yang, Xiyu Wang, Shiqi Zhao, Yi Lu, Xinfang Zhang, Qiu Shen, Xiao-Xiao Long, and Xun Cao. Make tracking easy: Neural motion retargeting for humanoid whole-body control. *arXiv preprint arXiv:2603.22201*, 2026.

[73] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[74] Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024.

APPENDIX A
EXTENDED RELATED WORK

a) Behavior Foundation Models for Humanoid Robots.: Going beyond isolated skills, recent works have started to explore systems that capture broad, reusable behavioral knowledge for humanoid robots, often referred to as *Behavior Foundation Models* [64, 61]. Using forward-backward representations, Meta Motivo [52] and BFM-Zero [22] learn promptable latent-conditioned policies for zero-shot behavior selection, enabling multiple objectives such as motion tracking and goal reaching. Other representative works learn structured generative behavior priors with masked control interfaces for multiple low-level control modes [64], or introduce a generative middleware that rewrites reference commands for robust recovery [49]. Together, these works show that general humanoid behavior benefits from reusable priors, but their priors are primarily instantiated as motion primitives that operate close to low-level execution. In contrast, **OMG** focuses on the upstream motion-generation interface, addressing how to translate heterogeneous *human* commands, such as language, audio, or human reference motions, into robot-executable motion references.

APPENDIX B
DETAILS ON OMG-DATA

A. Dataset Composition

We report statistics for **OMG-Data**, as shown in Table B.1. All motion sequences are represented at a temporal resolution of 30 Hz. AMASS-CMU and Weizmann are held out during pretraining.

TABLE B.1: **Statistics of the Pretraining Dataset by Conditioning Modality.** We curate publicly available motion datasets from graphics and humanoid domains and then apply careful filtering, annotation, and augmentation, yielding 1174.66 hours of data in total. ⌨, 📡, and 🧍 denote text descriptions, paired audio, and human reference motion, respectively.

Dataset	Label	Original			Processed		
		# Samples	Avg. Length	Total Hours	# Samples	Avg. Length	Total Hours
AMASS [33]	⌨, 🧍	17.9K	1.4K	94.8	51.1K	111	52.7
AMASS (holdout CMU & WEIZMANN) [33]	⌨, 🧍	13.65K	1.4K	44.9	39.0K	105	38
LAFAN1 [10]	⌨	40	6.6K	2.5	977	240	2.2
OMOMO [16]	⌨, 🧍	5.9K	179	10	15K	67	9.4
100style [34]	⌨, 🧍	0.8K	2.9K	22.12	10.7K	223	22.10
HumanML [8]	⌨, 🧍	26.8K	219	54.5	25.7K	218	52.04
Kungfu [8]	🧍	1.0K	453	4.3	–	–	–
MotionGV [8]	⌨, 🧍	56K	140	727	53.8K	128	637.14
fitness [8]	⌨, 🧍	262	610	1.48	337	244	0.76
MotionLLAMA (subset) [8]	⌨, 🧍	4.5K	286	12	4.9K	236	10.7
SnapMoGen [9]	⌨, 🧍	50K	407	188.7	41.0K	222	84.01
FineDance [18]	⌨, 📡, 🧍	0.2K	4.0K	14.6	6.0K	240	13.26
BEAT2 [27]	📡, 🧍	1.7K	3.4K	60	417	2.0K	8.06
AIST++ [20]	⌨, 📡, 🧍	1.4K	400	5.2	2.2K	240	4.98
IDEA400 [69, 25]	⌨, 🧍	12.5K	176	20	10K	177	17.24
PerMo [14]	⌨, 🧍	6.6K	140	8.6	6.5K	139	8.38
BONES-SEED [42]	⌨	71K	219	144	79K	182	134.35
ChoreoMaster [3]	⌨, 📡	2.7K	46.7	1.2	2.5K	47	1.11
CoMPAS3D [2]	⌨, 📡	4.8K	138	6.2	4.66K	129	5.57
AIOZ-GDANCE [15]	⌨, 📡	6K	1.1K	60.8	5.8K	1091	59.05
OpenDance [66]	⌨, 📡	5.0K	300	14.0	4.9K	300	13.61
Total	⌨, 📡, 🧍	**288.8K**	**411**	**1496.9**	**364.5K**	**177**	**1174.66**

B. Details on Retargeting

For motion represented in SMPL [28] or SMPL-X [36] format, body parameters are directly translated and mapped onto the skeletal frame of the G1 robot via geometric optimization in GMR. For datasets derived from raw videos, we employ the GENMO framework [17] to recover 3D human body meshes, which are subsequently passed into GMR to compute G1 joint trajectories. For motions represented in the FBX format, we convert them using standardized joint-mapping heuristics or GMR, depending on the structural complexity of the source skeleton hierarchy.

C. Details on Labeling

We provide the prompt used for Seed-1.8 VLM annotation below:

Prompt B.1: Prompt used for VLM-based temporal motion annotation.

```
You are a humanoid robot motion labeling model (VLM). Your task is to perform structured motion labeling for the input
video, rather than generating descriptive text.

[Most Important Rule]
You must segment when there is a change in "motion atomic action". If the specific action changes, even if the overall
style remains consistent, you still need to create a new segment.

[Core Requirement]
Please divide the input video into multiple consecutive time segments, and generate a structured action label for each
segment.

The output of each segment must include:
1. start time
2. end time
3. style
4. action

[Segmentation Criteria]
You should decide whether to split mainly based on:
- whether the motion style changes
- whether a finer motion style / type can be judged to have changed
- whether the main action changes

If the motion remains continuous and no obvious change occurs, do not over-segment.

[Action Requirements]
1. action must be written in English
2. action should summarize the concrete motion performed in this segment
3. action should describe the motion itself as much as possible, and should not write overly abstract content
4. action should be concise, and no more than 25 words
5. action should be helpful for training a text-to-motion generation model

[Style Requirements]
1. style must be written in English
2. style should summarize the motion style of this segment, such as smooth, intense, rhythmic, explosive, etc.
3. style should be concise

[Output Format]
Please output in JSON format:
{
  "video_summary": "overall summary of the video",
  "segments": [
    {
      "start_time": 0.0,
      "end_time": 1.5,
      "style": "smooth",
      "action": "raise both arms and step slowly to the left"
    },
    {
      "start_time": 1.5,
      "end_time": 3.0,
      "style": "rhythmic",
      "action": "turn torso and swing arms in alternating beats"
    }
  ]
}

[Output Constraints]
1. Output valid JSON only
2. Do not output any explanatory text
3. Time must be in seconds
4. Keep all start_time and end_time values to one decimal place
5. Segments must be consecutive and non-overlapping
6. start_time must be less than end_time
7. video_summary should be a brief summary of the whole video

[Goal]
Your output should help train a motion generation model, so the labels should preserve motion content, motion style,
and temporal changes as much as possible.
```

D. Segmentation Details

For text-conditioned sequences, motions are sliced according to the temporal boundaries of their language annotations. For audio-paired datasets, motion and audio sequences are segmented according to existing music phrase cuts or partitioned into uniform window lengths. For datasets with prohibitively long motion sequences, e.g., LAFAN1, 100style, FineDance, and AMASS, we further decompose each sequence into fixed-length sub-clips using a sliding-window strategy.

E. Simulation-in-the-Loop Filtering Details

For each generated motion, we first execute the sequence in the MuJoCo rigid-body simulator using the same tracker runtime used for deployment. The tracker predicts joint-space actions, which are clipped by `action_clip`=10.0 and converted to desired joint positions. At each control step, we apply a joint-space PD executor for `control_substeps`=10 simulation substeps:

$$\tau = K_p(q_{\text{des}} - q) - K_d\dot{q}, \tag{B.1}$$

where q_{des} is the desired joint configuration, q and $\dot{q}$ are the simulated joint position and velocity, and K_p, K_d are the actuator gains from the robot model.

After rollout, filtering is performed on the executed trajectory rather than on the original kinematic reference. We compute the root height h_t and root tilt angle θ_t from the simulated root pose. A trajectory is rejected if non-finite states occur, or if a fall condition persists for at least 10 consecutive frames. Following the main text, a frame is considered a fall frame when

$$h_t < 0.20 \quad \text{or} \quad \theta_t > 85^\circ \quad \text{or} \quad (h_t < 0.35 \wedge \theta_t > 60^\circ). \tag{B.2}$$

For datasets where strict robot feasibility is required, we additionally discard samples whose executed joint positions exceed the G1 joint limits. The remaining samples are kept as tracker-executed training data.

APPENDIX C
DETAILS OF OMG-DIT

A. Motion Representation and Canonicalization

a) Motion Representation.: **OMG-DiT** operates directly in the Unitree G1 motion space, without a learned motion tokenizer or latent autoencoder. Each motion frame is represented by a 125-dimensional feature vector

$$\mathbf{x}_i = [\tilde{\mathbf{p}}_i, \tilde{\mathbf{r}}_i, \boldsymbol{\theta}_i, \tilde{\mathbf{j}}_i], \qquad \mathbf{x}_i \in \mathbb{R}^{125}. \tag{C.1}$$

Here $\tilde{\mathbf{p}}_i \in \mathbb{R}^3$ is the root position in a canonical local coordinate frame, $\tilde{\mathbf{r}}_i \in \mathbb{R}^6$ is the root orientation represented by the continuous 6D rotation representation [73], $\boldsymbol{\theta}_i \in \mathbb{R}^{29}$ contains the G1 joint degrees of freedom, and $\tilde{\mathbf{j}}_i \in \mathbb{R}^{29\times 3}$ contains local positions of non-root body links. This yields a $3 + 6 + 29 + 29 \times 3 = 125$-dimensional motion vector per frame.

b) Root Rotation Parameterization.: Concretely, a quaternion represents orientation as $\mathbf{q} = (q_w, q_x, q_y, q_z) \in \mathbb{R}^4$ with the unit-norm constraint $\|\mathbf{q}\|_2 = 1$. An axis-angle (rotation-vector) representation uses $\mathbf{r} = \alpha\mathbf{u} \in \mathbb{R}^3$, where $\mathbf{u}$ is a unit rotation axis and α is the rotation angle. The 6D representation stores the first two columns of a rotation matrix, $\mathbf{R}_{6D} = [\mathbf{r}_1, \mathbf{r}_2] \in \mathbb{R}^6$, and recovers a valid rotation by orthonormalizing these two vectors. We use 6D root rotation features by default. Compared with directly regressing quaternions, the 6D representation avoids unit-norm and sign discontinuities in the network output space; compared with axis-angle vectors, it provides a smooth over-parameterized representation that was empirically more stable during large-scale pretraining.

c) Canonicalization.: For every training sample, the model observes a history window of $L = 10$ frames and predicts a future horizon of $H = 60$ frames at 30 FPS. We use the *last history frame* as the canonical anchor. Concretely, let $(\mathbf{p}_i, \mathbf{q}_i)$ denote the world-frame root position and quaternion orientation at motion frame i, and let $(\mathbf{p}_a, \mathbf{q}_a)$ be the anchor root state. We extract the yaw-only heading quaternion $\mathbf{h}_a$ from $\mathbf{q}_a$ and canonicalize the root trajectory as

$$\tilde{\mathbf{p}}_i = \mathbf{h}_a^{-1} \otimes (\mathbf{p}_i - \mathbf{p}_a), \qquad \tilde{\mathbf{q}}_i = \mathbf{h}_a^{-1} \otimes \mathbf{q}_i, \tag{C.2}$$

where $\otimes$ denotes quaternion rotation or composition depending on whether it is applied to a vector or a quaternion. The canonicalized quaternion $\tilde{\mathbf{q}}_i$ is then converted to the 6D root rotation feature $\tilde{\mathbf{r}}_i$. For each non-root body-link position $\mathbf{j}_{i,\ell}$, we similarly compute

$$\tilde{\mathbf{j}}_{i,\ell} = \mathbf{h}_a^{-1} \otimes (\mathbf{j}_{i,\ell} - \mathbf{p}_a). \tag{C.3}$$

The resulting canonical features remove arbitrary world-frame placement and heading, while retaining temporal motion trends from the history frames and the full G1 pose information required by the downstream tracker. At inference time, generated local features are decoded back to world-frame G1 states by composing them with the current anchor state.

As illustrated in Figure C.1, before canonicalization, root trajectories are widely scattered in the world frame: their coordinates are dominated by arbitrary global positions and headings, even when the underlying local motion patterns are similar. This creates a highly multi-modal input distribution that forces the model to explain nuisance variation unrelated to the commanded behavior. This motivates us to apply canonicalization. After canonicalization, trajectories collapse into a compact anchor-centered distribution, where remaining variation primarily reflects the robot's local motion dynamics rather than its absolute placement in the scene. This reduces the modeling burden of the diffusion backbone while preserving the temporal motion trend and full-body pose information required by the downstream tracker.

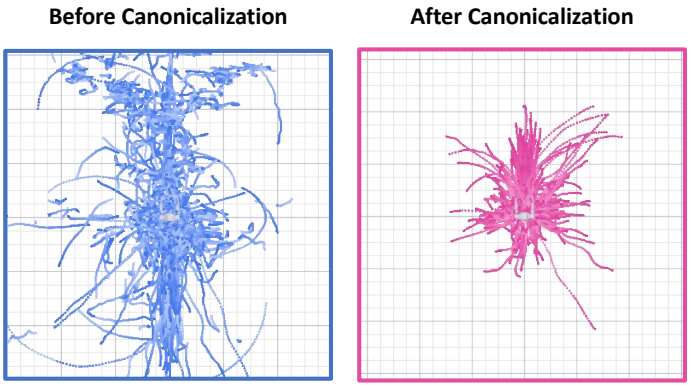


Fig. C.1: **Canonicalization.**

TABLE C.1: **Architecture variants of OMG-DiT.** To understand the scalability of motion generation, we introduce three variants of the denoising backbone randing from 50M to 500M parameters (B / L / XL), differing in transformer width, depth, and number of attention heads.

Model	Params	Hidden Dim.	Layers	Heads
OMG-DiT-B	50M	512	10	8
OMG-DiT-L	300M	1152	14	18
OMG-DiT-XL	500M	1536	13	24

d) Normalization.: After canonicalization, every feature channel is normalized using training-set statistics:

$$\bar{\mathbf{x}}_i = \frac{\mathbf{x}_i - \boldsymbol{\mu}}{\boldsymbol{\sigma}}, \tag{C.4}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are computed from the training split. We clamp $\boldsymbol{\sigma}$ by a small positive minimum value for numerical stability. The diffusion model is trained and sampled in this normalized feature space, and outputs are denormalized before decoding to G1 joint states.

B. Denoising Backbone

The generator is implemented as a Diffusion Transformer trained with the $\mathbf{x}$-prediction objective. Here $\mathbf{x}$ denotes the normalized canonical motion feature defined above. Given a corrupted future motion segment $\mathbf{x}^{\tau}_{t+1:t+H} \in \mathbb{R}^{H\times 125}$, diffusion timestep τ, recent motion history $\mathbf{x}_{t-L:t}$, and a subset of available conditioning modalities $\mathcal{C}_s \subseteq \mathcal{C}$, the denoiser predicts the clean future motion as

$$\hat{\mathbf{x}}_{t+1:t+H} = \mathbf{x}_\theta\left(\mathbf{x}^{\tau}_{t+1:t+H}, \tau, \mathbf{x}_{t-L:t}, \mathcal{C}_s\right), \qquad \hat{\mathbf{x}}_{t+1:t+H} \in \mathbb{R}^{H\times 125}. \tag{C.5}$$

a) Architecture.: The denoiser first projects the corrupted future motion features into the Transformer hidden dimension. Each DiT block applies bidirectional temporal self-attention over the H future frames. The recent history $\mathbf{x}_{t-L:t}$ and the condition subset $\mathcal{C}_s$ are encoded into context tokens and injected through cross-attention, while temporally aligned modalities such as audio or human reference motion can additionally be incorporated through frame-wise modulation. We use rotary positional embeddings (RoPE, [48]) for temporal self-attention, sinusoidal embeddings for the diffusion timestep τ, and feed-forward blocks with GELU activations. The final prediction is a sequence of normalized clean motion features in the same 125D motion representation in Unitree G1 motion space. We list the architecture specifications for **OMG-DiT-B / L / XL** in Table C.1. All variants share the same motion representation, diffusion objective, and conditioning interface architecture; they differ only in Transformer width, depth, and number of attention heads.

b) Denoising Objective and Sampling.: We train the model to predict the clean future motion $\mathbf{x}_0$ from a noisy future trajectory. We use discrete diffusion timesteps with a cosine noise schedule. Given a clean normalized future sequence $\mathbf{x}_0$, timestep τ, and Gaussian noise $\boldsymbol{\epsilon}$, the noisy input is

$$\mathbf{x}_\tau = \sqrt{\bar{\alpha}_\tau}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_\tau}\boldsymbol{\epsilon}. \tag{C.6}$$

At inference time, we use DDIM [47] sampling with 50 denoising steps and $\eta = 0$. The default global guidance scale is 2.5 when a single guidance scale is used. For composed modalities, modality-specific scales can be set separately as described in Section C-E.

C. Omni-Modal Condition Encoders

We provide details on condition encoders for different modalities in **OMG-DiT**.

a) Motion History.: The 10-frame motion history is first canonicalized and normalized using the same feature statistics as the future target. A two-layer MLP maps each history frame to the Transformer hidden dimension. The resulting history tokens are appended to the global context token sequence and interact with generated motion latents through per-block cross-attention layers.

b) Language.: Text annotations are encoded by a frozen T5-Base [41] encoder with maximum length 50. The T5 language embeddings are projected into the transformer hidden dimension and injected as global context tokens alongside motion history through cross-attention. During training, text conditioning is randomly dropped with probability 0.3.

c) Audio.: Audio is encoded as frame-aligned 35D acoustic features. Given a raw waveform, we first average stereo channels to mono and peak-normalize the signal. For audio sample rate s and motion frame rate $f = 30$ Hz, the hop size is

$$h = \mathrm{round}(s/f), \tag{C.7}$$

and each audio frame uses a Hann window of length

$$w = \max\{h, \mathrm{round}(0.05s)\}, \tag{C.8}$$

corresponding to at least a 50 ms analysis window. The feature vector at motion frame i is extracted from the waveform segment starting at sample ih and is defined as

$$\mathbf{a}_i = \left[\mathbf{e}_i^{1:32},\ \mathrm{rms}_i,\ \mathrm{zcr}_i,\ \mathrm{flux}_i + \mathrm{centroid}_i\right] \in \mathbb{R}^{35}. \tag{C.9}$$

Here $\mathbf{e}_i^{1:32}$ are 32 log FFT band energies computed by splitting the magnitude spectrum into 32 bands and normalizing by the maximum band energy within the same frame, rms_i is root-mean-square energy, zcr_i is zero-crossing rate, flux_i is spectral flux from the previous audio frame, and $\mathrm{centroid}_i$ is the normalized spectral centroid. Audio features are trimmed or zero-padded to the motion horizon, with a validity mask for padded frames. During training, these features are stored offline as `.npy` arrays; at inference time, the same extractor can also be applied to raw WAV input before planning. The 35D features are passed through a LayerNorm and projected by an MLP into the denoiser hidden dimension. Since audio is temporally aligned with the future motion horizon, we inject it as a frame-wise condition using per-layer FiLM modulation. Audio conditioning is randomly dropped with probability 0.1 during training.

d) Human Reference Motion.: We represent human reference motions as 66D frame-aligned features, corresponding to 22 human joints in 3D space. As with audio, the human reference signal is projected by an MLP and injected through per-layer FiLM modulation at the corresponding motion frame. Human-reference conditioning is randomly dropped with probability 0.5 during training.

For samples where a modality is unavailable, we set its availability mask to false. The mask gates the corresponding condition encoder and modulation, so the denoiser receives no conditioning signal from that modality. The same mechanism is used for modality dropout during training, allowing one shared model to train on heterogeneous datasets with different subsets of available conditions.

D. Adapting to New Modalities

OMG-DiT is designed so that new control modalities can be included while preserving as much of the pretrained motion prior as possible. During adaptation, we reuse the pretrained denoising backbone and add new modalities with lightweight encoders.

For global conditioning signals, such as visual features for perceptive locomotion, the encoded tokens are appended to the cross-attention context, following the same interface as language and history tokens. For frame-aligned signals, such as Pico sparse keypoints for teleoperation, the encoded features are injected into each denoising block through lightweight adapters. In our implementation, these adapters can be per-layer FiLM modules or AdaLN-style modulation modules. The final linear layer of each newly added adapter is initialized to zero, so the adapter initially outputs no modulation and the pretrained generator's function is preserved at the start of finetuning.

This non-invasive initialization makes sample-efficient finetuning practical. The default adaptation setting trains the new modality encoder and its adapters while reusing the pretrained motion backbone. In our experiments, we use full finetuning across all backbone parameters, but parameter-efficient finetuning may also be feasible for simple tasks.

E. Classifier-Free Guidance and Composition

a) Classifier-Free Guidance and Multi-Modal Composition.: We train with modality dropout and use classifier-free guidance at inference time. For a single condition, the denoiser prediction is guided by comparing a conditional branch with a null-condition branch. For composed conditions, we additionally support modality-specific guidance branches. Given text, audio, and human-reference conditions, the guided prediction can be written as

$$\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_0^{\varnothing} + \omega_{\text{text}}(\hat{\mathbf{x}}_0^{\text{text}} - \hat{\mathbf{x}}_0^{\varnothing}) + \omega_{\text{audio}}(\hat{\mathbf{x}}_0^{\text{audio}} - \hat{\mathbf{x}}_0^{\varnothing}) + \omega_{\text{human}}(\hat{\mathbf{x}}_0^{\text{human}} - \hat{\mathbf{x}}_0^{\varnothing}), \tag{C.10}$$

where $\hat{\mathbf{x}}_0^{\varnothing}$ is the null-condition prediction and ω_{text}, ω_{audio}, and ω_{human} are modality-specific guidance scales. By composing the guidance directions from different modalities, **OMG-DiT** enables zero-shot composition of command combinations previously unseen during training.

b) Practical Implementation.: We instantiate classifier-free guidance in two ways during inference. If only a single global scale ω is specified, we use a full-condition branch:

$$\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_0^{\varnothing} + \omega(\hat{\mathbf{x}}_0^{\text{all}} - \hat{\mathbf{x}}_0^{\varnothing}), \tag{C.11}$$

where $\hat{\mathbf{x}}_0^{\text{all}}$ is predicted with all available external modalities unmasked. If modality-specific scales are specified, we use separate guidance branches instead of a full-condition branch.

Unless otherwise specified, we use global scale $\omega = 2.5$ for single-modality motion generation. For composed audio-language generation, we use $\omega_{\text{text}} = 3.0$ and $\omega_{\text{audio}} = 1.5$, giving text a stronger semantic prior while retaining audio rhythm. For human-reference conditioning, we use $\omega_{\text{human}} = 2.0$. Intuitively, increasing ω_{text} favors semantic instruction following, whereas increasing ω_{audio} favors rhythmic and dance-style adherence. We provide further visualizations in Appendix E-D.

c) Extending Classifier-Free Guidance to New Modalities.: Classifier-free guidance extends naturally to newly added modalities. We construct a null branch and a modality-specific conditional branch for the new signal, and use

$$\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_0^{\varnothing} + \omega_{\text{new}}(\hat{\mathbf{x}}_0^{\text{new}} - \hat{\mathbf{x}}_0^{\varnothing}), \tag{C.12}$$

where ω_{new} is the guidance scale for the new modality. This branch can be combined with the existing modalities, enabling new control interfaces to be participate in compositional generation.

F. Real-Time Deployment

To enable real-time inference, we use a combination of existing runtime acceleration methods, including ONNX/TensorRT export, FP16 precision, and DiT caching [26]. We provide runtime analysis and additional details in Appendix E-A. At deployment, the motion tracker runs onboard on the NVIDIA Orin chip, while the motion generator runs on an off-board NVIDIA RTX 4090 workstation connected to the robot through a wired Ethernet link.

G. Training Hyperparameters

We summarize key training hyperparameters for pretraining in Table C.2. We train in mixed bfloat16 precision with the AdamW optimizer [29], and apply standard techniques including gradient clipping, a linear-warmup cosine-decay learning-rate schedule, and weight decay. Training is completed on 8 NVIDIA A800 GPUs in under 10 hours.

TABLE C.2: **Hyperparameters for OMG-DiT Pretraining.** Unless otherwise specified, evaluation is conducted on the **-L** (300M) version of the model.

Hyperparameter	**Value**
Motion Setup	
Prediction / history length	60 / 10 frames
Motion frame rate	30 FPS
Motion representation	125D G1 features with 6D root rotation
Optimization	
Optimizer	AdamW
Learning rate	6.0×10^{-5}
Weight decay	0.01
LR schedule	Linear warmup, cosine decay
Warmup steps	2,000
Minimum LR	1.0×10^{-6}
Gradient clipping	Global norm 1.0
Training	
Max training steps	100,000
Precision	bf16 mixed precision
GPUs	8
Per-GPU batch size	128
Global batch size	1,024
Inference	
Method	DDIM
NFE calls	50

APPENDIX D
EXPERIMENT DETAILS

A. Evaluation Protocols

We provide details on evaluation protocols shared across different experiments, including the evaluated tasks, motion format, and data splits. Unless otherwise stated, all experiments follow these defaults.

1) Evaluated Tasks and Motion Format:

a) Evaluated Tasks.: We evaluate six tasks: text-to-motion, audio-to-motion, human reference-to-motion, Pico keypoint-based teleoperation, text-to-motion finetuning, and perceptive locomotion. Although the conditions differ, all methods output Unitree G1 motion in the same G1 qpos space, making different methods and baselines directly comparable.

b) Motion Format.: Each robot frame is represented by a 36D qpos vector in G1 motion space:

$$\mathbf{q}_t = \left[\mathbf{x}_t^{\text{root}},\ \mathbf{r}_t^{\text{root}},\ \mathbf{q}_t^{\text{joint}}\right] \in \mathbb{R}^{36}, \tag{D.1}$$

where $\mathbf{x}_t^{\text{root}} \in \mathbb{R}^3$, $\mathbf{r}_t^{\text{root}} \in \mathbb{R}^4$, and $\mathbf{q}_t^{\text{joint}} \in \mathbb{R}^{29}$ denote root position, root quaternion, and G1 joint DOFs. A sequence of length T is $\mathbf{q} \in \mathbb{R}^{T\times 36}$. For body-level metrics, we use forward kinematics:

$$\mathbf{p} = \mathrm{FK}(\mathbf{q}) \in \mathbb{R}^{T\times 90}, \tag{D.2}$$

where the 90 dimensions are 3D positions of 30 G1 body/joint points. For baselines whose native outputs are not G1 qpos, such as SMPL-X, SMPL/SMPL-H motion, SMPL pickle files, or dance motion arrays, we first convert or retarget them to G1 qpos and then use the same evaluation pipeline.

2) Evaluation Data Settings:

a) Evaluation Data Split.: Evaluations are performed on validation splits. The default generated sequence has 60 frames at 30 FPS. Generated motions are stored as G1 qpos, and body-level metrics are computed from FK-derived 90D body positions.

TABLE D.1: **Evaluation data and sampling settings.**

Task	Evaluation data	# samples
Text-to-motion	Caption-bearing validation samples	1024
Audio-to-motion	Validation samples with audio/music features	512
Human reference-to-motion	Validation samples with human reference motion	512
Pico teleoperation	AMASS (CMU + WEIZMANN), with Pico keypoints	512
Text-to-motion finetuning	AMASS (CMU + WEIZMANN), with captions	1024

As shown in Table D.1, for text-to-motion, we sample 1024 caption-bearing validation conditions without replacement. R-precision uses batch size 32, giving 32 retrieval batches, and constructs candidates by dataset-stratified reordering. FID reference motions are sampled from real data; if the reference count is unspecified, it equals the generated sample count. AMASS CMU and WEIZMANN are used only for text-to-motion finetuning and Pico teleoperation from-scratch/finetuning experiments, and held out during pretraining, to avoid data leakage.

B. Evaluation Metrics

In this section, we provide an overview of metrics used during evaluation, including general metrics shared across experiments and specific metrics for text-to-motion and audio-to-motion experiments. We additionally provide details on evaluator training for text-to-motion evaluation.

1) Shared Evaluation Metrics: We use the same set of metrics for evaluating physical plausibility, reconstruction error relative to reference motion, and tracker execution failures across experiments, as shown in Table D.2. Here $\mathbf{p}^{pred}$ and $\mathbf{p}^{ref}$ are predicted and reference G1 body positions, and $\tilde{\mathbf{p}}$ is the root-relative position. Velocity and acceleration are computed as first- and second-order finite differences of body positions.

$$\mathbf{v}_{b,t,j} = \mathbf{p}_{b,t+1,j} - \mathbf{p}_{b,t,j}, \quad \mathbf{a}_{b,t,j} = \mathbf{p}_{b,t+2,j} - 2\mathbf{p}_{b,t+1,j} + \mathbf{p}_{b,t,j}. \tag{D.3}$$

For c-slide, $\mathcal{C}$ is the set of valid foot-contact intervals and $\mathcal{S}_f$ contains sole proxy points for foot f. Fall Rate is determined by root height and tilt. For J-Limit, $e_{i,t,j}$ is the joint-limit violation magnitude and $\epsilon = 10^{-4}$.

TABLE D.2: **Shared Evaluation Metrics.**

Metric	Formula	Description
c-slide	$\frac{1}{\lvert\mathcal{C}\rvert}\sum_{(t,f)\in\mathcal{C}}\max_{p\in\mathcal{S}_f}\lVert\mathbf{s}^{xy}_{t+1,p}-\mathbf{s}^{xy}_{t,p}\rVert_2\cdot f_{\text{fps}}$	Horizontal foot sliding speed during contact
Jerk	$\frac{1}{\lvert\mathcal{V}\rvert}\sum_{(t,j)\in\mathcal{V}}\lVert\mathbf{p}_{t+3,j}-3\mathbf{p}_{t+2,j}+3\mathbf{p}_{t+1,j}-\mathbf{p}_{t,j}\rVert_2\cdot f_{\text{fps}}^3$	Third-order motion variation
g-MPJPE	$1000\cdot\frac{1}{BTJ}\sum_{b,t,j}\lVert\mathbf{p}^{pred}_{b,t,j}-\mathbf{p}^{ref}_{b,t,j}\rVert_2$	Global body position error in mm
MPJPE	$1000\cdot\frac{1}{BTJ}\sum_{b,t,j}\lVert\tilde{\mathbf{p}}^{pred}_{b,t,j}-\tilde{\mathbf{p}}^{ref}_{b,t,j}\rVert_2$	Root-relative body position error in mm
e-vel	$1000\cdot\frac{1}{B(T-1)J}\sum_{b,t,j}\lVert\mathbf{v}^{pred}_{b,t,j}-\mathbf{v}^{ref}_{b,t,j}\rVert_2$	Velocity error in mm/frame
e-acc	$1000\cdot\frac{1}{B(T-2)J}\sum_{b,t,j}\lVert\mathbf{a}^{pred}_{b,t,j}-\mathbf{a}^{ref}_{b,t,j}\rVert_2$	Acceleration error in mm/frame^2
Fall Rate	$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[\text{sequence}_i\text{ fallen}]$	Sequence-level fall rate
J-Limit	$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[\max_{t,j}e_{i,t,j}>\epsilon]$	Sequence-level joint-limit violation rate

TABLE D.3: **Text-to-motion metrics.**

Metric	Formula	Description
Matching Score	$\frac{1}{N}\sum_{i=1}^{N}\lVert\mathbf{m}_i-\mathbf{t}_i\rVert_2$	Average distance between paired text-motion embeddings
R-precision@K	$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[\text{rank}_i(i)\le K]$	Whether the ground-truth text is in the top-K retrieval results
FID	$\lVert\boldsymbol{\mu}_r-\boldsymbol{\mu}_g\rVert_2^2+\text{Tr}(\boldsymbol{\Sigma}_r+\boldsymbol{\Sigma}_g-2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{1/2})$	Fréchet distance between real and generated motion embeddings
Diversity	$\frac{1}{\lvert\mathcal{P}\rvert}\sum_{(a,b)\in\mathcal{P}}\lVert\mathbf{m}_a-\mathbf{m}_b\rVert_2$	Average embedding distance between generated samples

2) Text-to-Motion Metrics: For text-to-motion, we additionally evaluate text-motion alignment and the generated motion distribution. Each generated G1 qpos is converted via FK to 90D body positions, encoded as a motion embedding, and compared with the corresponding text embedding. We report Matching Score, R-precision, FID, and Diversity, as shown in Table D.3.

Here $\mathbf{m}_i$ and $\mathbf{t}_i$ are motion and text embeddings, and N is the number of samples. For R-precision, candidate texts are ranked by

$$D_{ij}=\lVert\mathbf{m}_i-\mathbf{t}_j\rVert_2^2, \tag{D.4}$$

and $\text{rank}_i(i)$ is the rank of the paired text. We report $R@1$, $R@2$, and $R@3$. For FID, $(\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_r)$ and $(\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)$ are the empirical mean and covariance of real and generated motion embeddings. For Diversity, $\mathcal{P}$ contains randomly sampled generated embedding pairs; we use 300 pairs by default.

3) Audio-to-Motion Metrics: Specific audio-to-motion metrics evaluate beat alignment and compare generated and real motions using kinetic and geometric statistics.

TABLE D.4: **Audio-to-motion metrics.**

Metric	Formula	Description
Beat Align	$\frac{1}{\lvert\mathcal{B}^a\rvert}\sum_i\exp(-d_i^2/(2\sigma^2))$	Temporal alignment between music and motion beats
FID-k	$\lVert\boldsymbol{\mu}_{k,r}-\boldsymbol{\mu}_{k,g}\rVert_2^2+\text{Tr}(\boldsymbol{\Sigma}_{k,r}+\boldsymbol{\Sigma}_{k,g}-2(\boldsymbol{\Sigma}_{k,r}\boldsymbol{\Sigma}_{k,g})^{1/2})$	Fréchet distance over kinetic features
FID-g	$\lVert\boldsymbol{\mu}_{geo,r}-\boldsymbol{\mu}_{geo,g}\rVert_2^2+\text{Tr}(\boldsymbol{\Sigma}_{geo,r}+\boldsymbol{\Sigma}_{geo,g}-2(\boldsymbol{\Sigma}_{geo,r}\boldsymbol{\Sigma}_{geo,g})^{1/2})$	Fréchet distance over geometric features
PFC	$10000\cdot\frac{1}{T-2}\sum_t L_tR_t\hat{A}_t$	Penalizes foot sliding during root upward acceleration

Here $\mathcal{B}^a$ is the music beat set and d_i is the distance from music beat i to the nearest motion beat. The Gaussian kernel width is $\sigma=3/f_{\text{fps}}$, with $f_{\text{fps}}=30$. FID-k uses kinetic features, FID-g uses geometric features, and PFC uses left/right foot sliding L_t,R_t with normalized root upward acceleration $\hat{A}_t$.

4) Details on Evaluator Training for Text-to-Motion Evaluation: Text-to-motion embedding metrics use a separately trained text-motion evaluation encoder, following a CLIP-style contrastive retrieval protocol with symmetric InfoNCE loss. The evaluator is used only for evaluation and is not part of the generator. It contains a motion encoder and a text encoder that map motions and texts into a shared embedding space. We provide architecture details and key hyperparameters for evaluator training in Table D.5 and Table D.6, respectively.

The text encoder consists of a pretrained frozen T5-3B encoder followed by a trainable linear projection to the shared embedding space, where T5-3B denotes a T5 encoder with approximately 3 billion parameters. For sample i, the motion

TABLE D.5: **Motion Evaluator architecture used for evaluation.**

Stage	Architecture	Output
Movement encoder	Linear + LayerNorm + GELU + Dropout	Per-frame feature
Hidden projection	Linear or Identity	Hidden sequence
Temporal encoder	RoPE Transformer	Temporal feature sequence
Pooling/projection	Mean pooling + MLP projection	Motion embedding
Normalization	L2 normalization	Unit-norm embedding

TABLE D.6: **Hyperparameters for Text-to-Motion Evaluator Training.**

Hyperparameter	Value
Motion input dim	90
Movement dim	512
Hidden dim	512
Output dim	512
Transformer layers	4
Attention heads	8
MLP ratio	4

encoder takes the FK-derived 90D body-position sequence and the text encoder takes the corresponding text:

$$\mathbf{m}_i = E_{\text{motion}}(\mathbf{p}_i), \quad \mathbf{t}_i = E_{\text{text}}(\mathbf{c}_i), \tag{D.5}$$

where $\mathbf{p}_i \in \mathbb{R}^{T\times 90}$, $\mathbf{c}_i$ is the text condition, and $\mathbf{m}_i, \mathbf{t}_i$ are motion and text embeddings.

The two encoders are trained with symmetric InfoNCE:

$$\mathcal{L} = \frac{1}{2}\left(\mathcal{L}_{m\to t} + \mathcal{L}_{t\to m}\right), \tag{D.6}$$

where

$$\mathcal{L}_{m\to t} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\text{sim}(\mathbf{m}_i, \mathbf{t}_i)/\tau)}{\sum_{j=1}^{B}\exp(\text{sim}(\mathbf{m}_i, \mathbf{t}_j)/\tau)}. \tag{D.7}$$

$\mathcal{L}_{t\to m}$ is symmetric. Here B is the batch size, τ is temperature, and $\text{sim}(\cdot,\cdot)$ is embedding similarity.

For standard text-to-motion evaluation, the encoder is trained on the full train split and evaluated on the validation split. For finetuning, it is trained and evaluated only on AMASS CMU and WEIZMANN, which are excluded from regular pretraining.

C. Details of Evaluation Baselines

We compare against a wide variety of baselines from both the graphics and humanoid communities. For motion generation baselines in graphics, we use GMR [1] to retarget the generated human poses into G1 motion space.

1) Text-to-Motion Baselines: We compare against state-of-the-art motion generation baselines, including GENMO [17], HY-Motion [55], and Kimodo [42]. For Kimodo, we compare against variants trained with different datasets, including Bones Seed and Bones Rigplay, as well as generation in SMPL-X space or G1 space. All final outputs are evaluated as G1 `qpos_36`. These baselines operate in a full-sequence manner, generating in advance instead of interactively in real time.

TABLE D.7: **Text-to-motion Baselines.**

Baseline	Original output space	Conversion to G1	Final artifact
GENMO [17] + GMR	SMPL-X motion	GMR retargeting	G1 `qpos_36`
HYMotion [55] + GMR	SMPL/SMPL-H style motion	Convert to SMPL-X, then GMR	G1 `qpos_36`
Kimodo-SMPLX-RP-v1 [42] + GMR	SMPL-X motion	GMR retargeting	G1 `qpos_36`
Kimodo-G1-RP-v1 [42]	G1 qpos	Direct use	G1 `qpos_36`
Kimodo-G1-SEED-v1 [42, 44]	G1 qpos	Direct use	G1 `qpos_36`

2) Audio-to-Motion Baselines: For audio-to-motion baselines, we compare against motion generation models that can condition on multiple modalities, i.e. GENMO [17], as well as single-purpose state-of-the-art audio-to-motion generation baselines, such as LODGE [19] and Bailando++ [46]. Their outputs are first converted to G1 qpos with GMR [1] and then evaluated with audio-motion and physical metrics. Specifically, GENMO [17] + GMR directly generates SMPL-X motion and retargets it to G1. LODGE [19] and Bailando++ [46] are first converted to SMPL/SMPL-X-compatible representations and then retargeted. After conversion, all baselines share the same G1 qpos evaluation.

TABLE D.8: **Audio-to-motion Baselines.**

Baseline	Condition input	Original output	Conversion to G1	Final artifact
GENMO [17] + GMR	Raw audio/music embedding	SMPL-X motion	GMR retargeting	G1 `qpos_36`
LODGE [19] + GMR	Music feature `.npy`	Dance motion array	Convert to SMPL/SMPL-X, then GMR	G1 `qpos_36`
Bailando++ [46] + GMR	Raw audio/extracted feature	SMPL pickle	Convert to SMPL-X, then GMR	G1 `qpos_36`

3) Human Reference-to-Motion Baselines: We represent the human reference using the 3D coordinates of 22 human joints, i.e.,

$$\mathbf{h} \in \mathbb{R}^{T\times 22\times 3}. \tag{D.8}$$

This is essentially a *retargeting* problem, in which motion in human motion space is retargeted to G1 motion space. Here, we compare against well-established optimization-based retargeting methods, including GMR [1], PHC [31], and OmniRetarget [59], as well as the recently proposed learning-based method NMR [72].

TABLE D.9: **Human reference-to-motion Baselines.**

Baseline	Type	Intermediate representation	Final artifact
GMR [1]	Geometric/rule-based retargeting	Compact SMPL-X motion	G1 `qpos_36`
PHC [31]	Physics/fitting-based retargeting	AMASS-style SMPL motion	G1 `qpos_36`
OmniRetarget [59]	Optimization-based retargeting	Holosoma-compatible SMPL-X motion	G1 `qpos_36`
NMR [72]	Learned retargeting model	Standard SMPL-X motion	G1 `qpos_36`

D. Details on Finetuning and Scaling Experiments

a) Datasets and Evaluation Splits.: We use the AMASS CMU and WEIZMANN subsets for text-to-motion finetuning and Pico teleoperation from-scratch/finetuning comparisons. To avoid leakage, these subsets are excluded from pretraining and used only for these experiments and evaluations. For scaling experiments, we pretrain on the exact **OMG-Data** pretraining corpus. For sample-efficient finetuning with varying percentages of included data, data are controlled at the slice level. Each annotated motion segment is exhaustively sliced with a 2-second window at 30 FPS, so each slice has 60 frames and the full candidate slice pool is materialized. We shuffle this pool with a fixed random seed and keep the first specified percentage as the training set.

b) Setup.: We compare finetuning and from-scratch training under the same model size and split. Finetuning initializes from **OMG-DiT-L**, the 300M pretrained checkpoint. The from-scratch setting uses the same architecture with randomly initialized parameters. For the model-parameter scaling-law experiment, we vary only the model size and keep all other hyperparameters unchanged.

We additionally include an adaptation experiment for perceptive locomotion in Appendix E-E.

APPENDIX E
EXTENDED EXPERIMENTS AND VISUALIZATIONS

We include additional experiments and visualizations in this section. We provide runtime acceleration analysis, extended visualizations on real-time omni-modal control, finetuning experiments on perceptive locomotion tasks, and analysis of classifier-free guidance.

A. Runtime Acceleration Analysis

We report speedups from different inference optimization techniques for **OMG-DiT-B** in Table E.1. Performance is measured on a text-conditioned 60-frame future chunk at 30 FPS, corresponding to a two-second planning horizon, on a single NVIDIA A800 GPU.

Sampling time refers to the total time from receiving cached condition tensors to outputting future motion predictions, while Denoiser infer denotes the forward-pass time inside the sampler. NFE counts conditional/null CFG branch predictions; values in parentheses report backend forward calls after CFG parallelism.

TABLE E.1: **Runtime Acceleration Analysis.** Inference time is measured while generating a 60-frame future chunk with text conditioning, running **OMG-DiT-B** on a single NVIDIA A800 GPU. We report the mean and standard deviation over five steady-state runs.

Optimization	Sampling (ms)	Denoiser infer (ms)	NFE (calls)	Speedup
PyTorch eager	1297.9 ± 12.5	1265.0 ± 12.4	100 (100)	1.0×
+ `torch.compile`	467.9 ± 2.9	435.6 ± 2.8	100 (100)	2.8×
+ ONNX Runtime CUDA	358.2 ± 24.2	337.0 ± 22.7	100 (50)	3.6×
+ TensorRT engine	127.3 ± 7.3	100.9 ± 4.3	100 (50)	10.2×
+ TensorRT FP16	99.8 ± 5.2	73.3 ± 5.2	100 (50)	13.0×
+ DiT cache	51.7 ± 1.5	31.2 ± 1.3	38 (19)	25.1×

B. Extended Visualizations on Real-Time Omni-Modal Control

We provide extended qualitative visualizations on real-time omni-modal control. Motions are generated by the same pretrained model in real time, with conditions unseen during training. As shown in Figure E.2, **OMG** generates diverse and robot-executable motions conditioned on various signals, showcasing strong capabilities as a foundation model for generalist humanoid control.

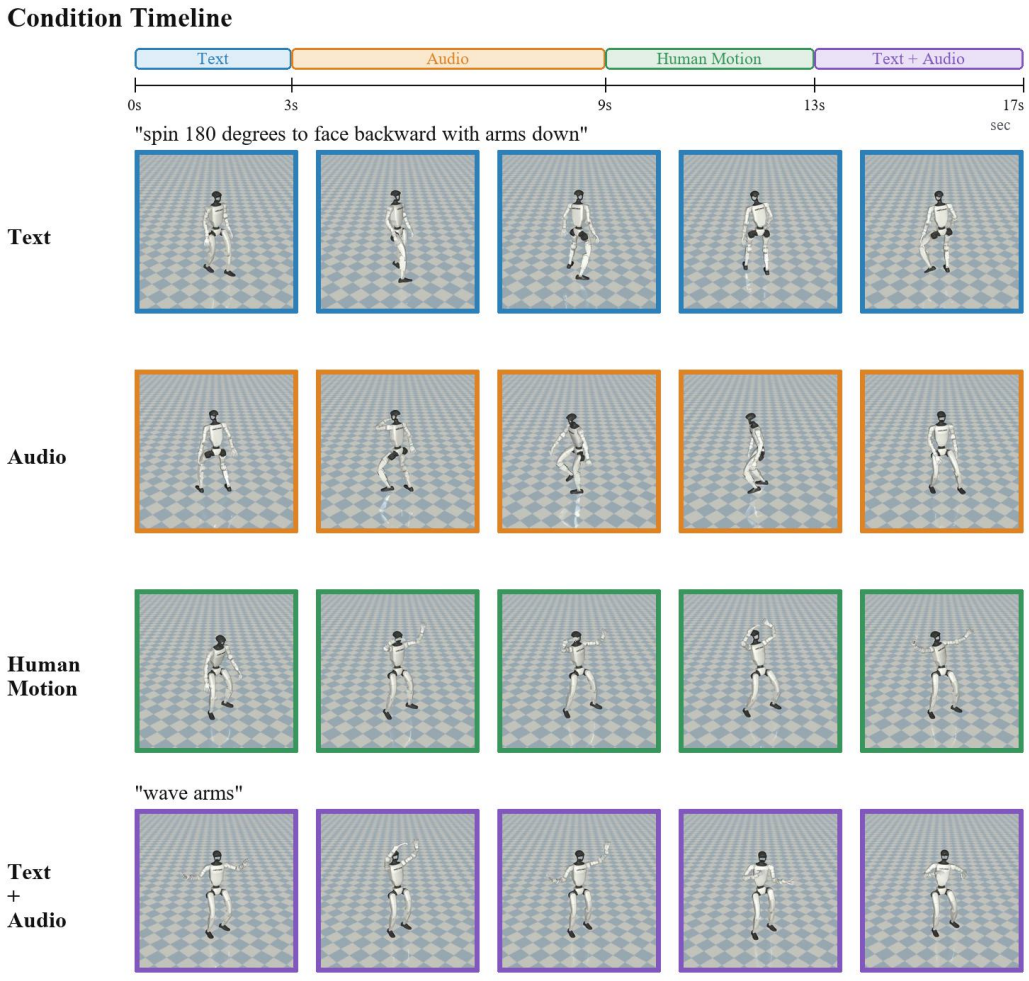


Fig. E.1: **Interactive Control: Composition in the Temporal Horizon.**

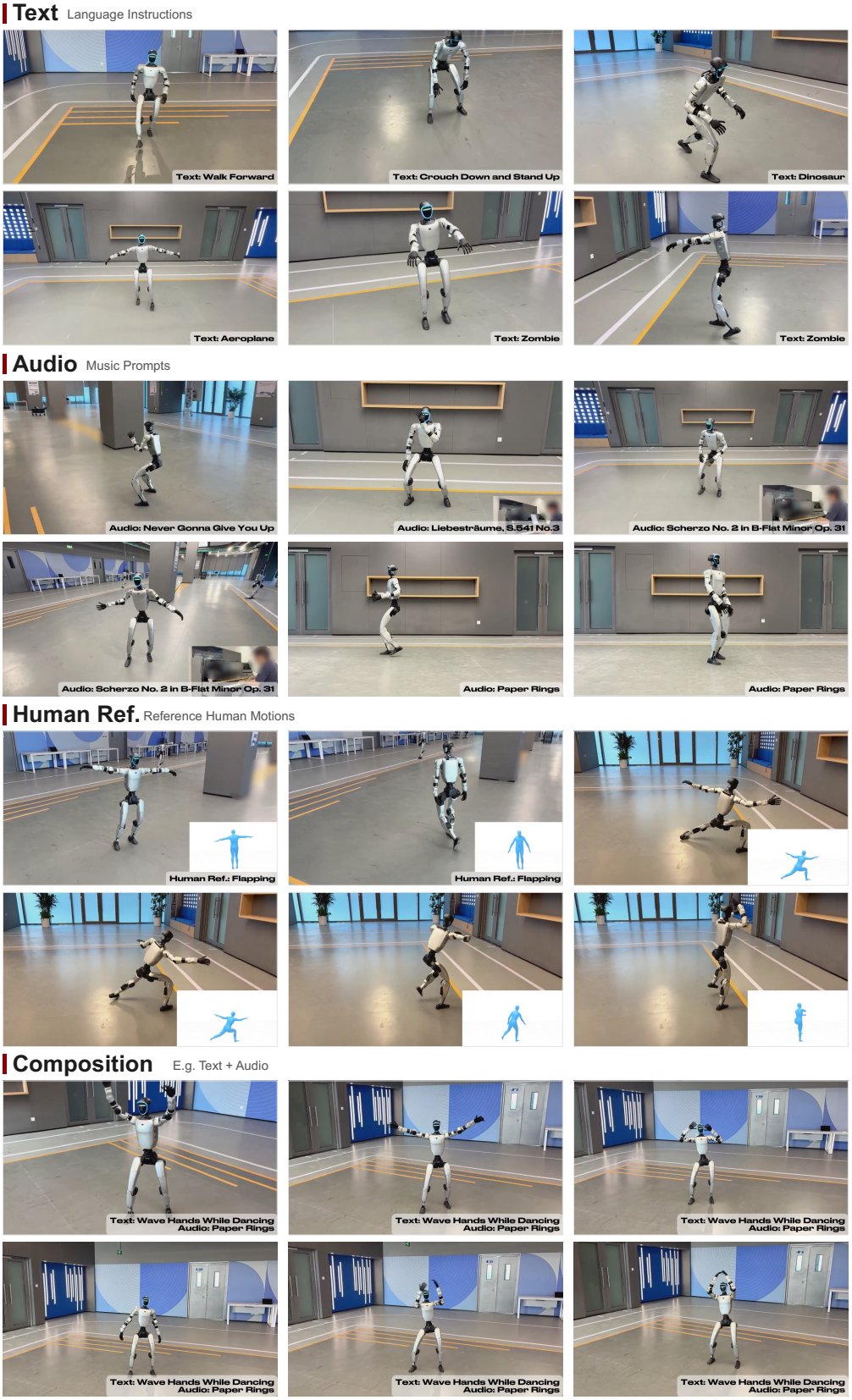


Fig. E.2: **Qualitative visualization.** Unitree G1 execution sequences produced by **OMG** under text, audio, human-reference, and composed text-audio conditions. Frames are uniformly sampled within each sequence, and embedded prompts are preserved.

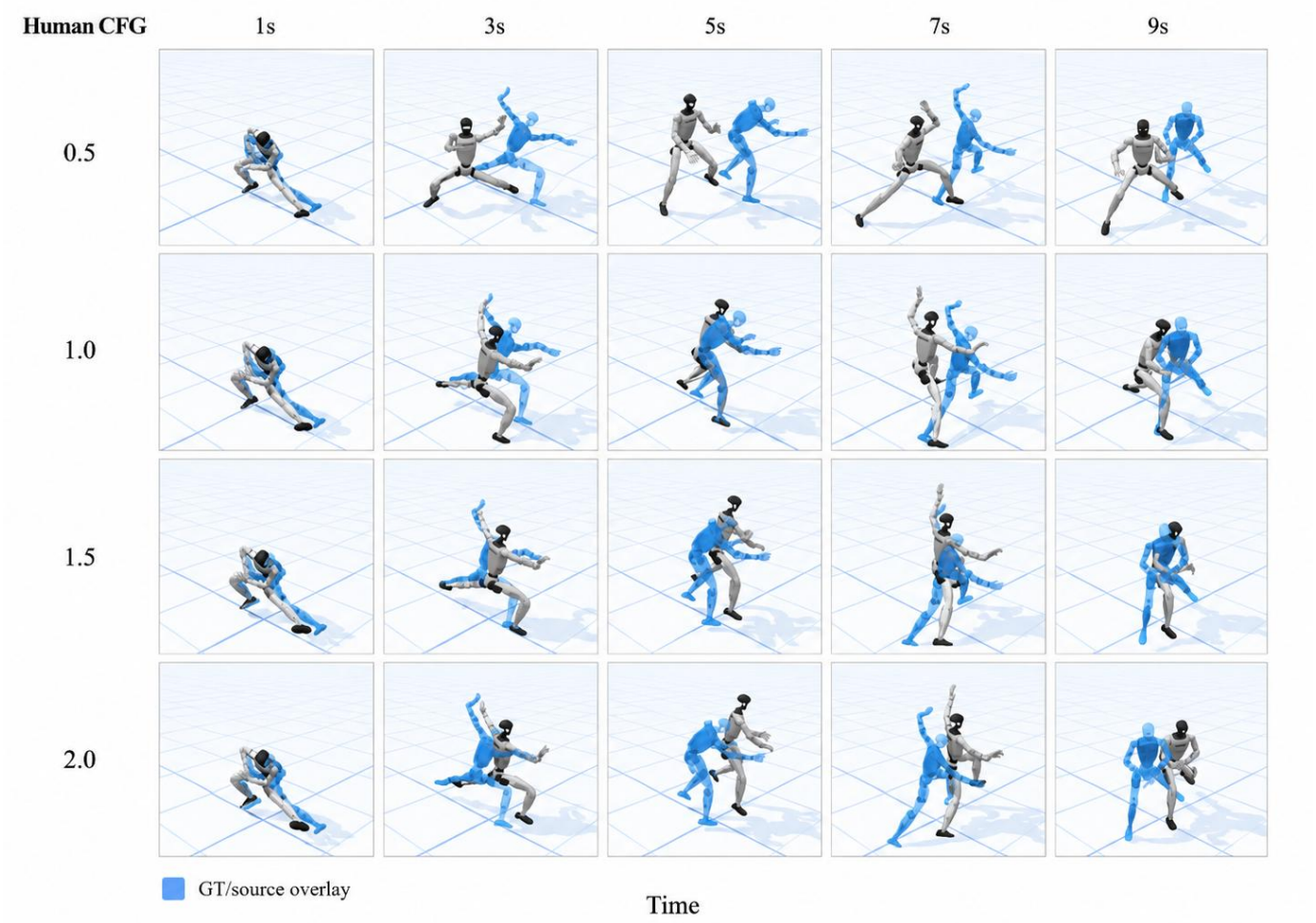


Fig. E.3: **Human-reference CFG sweep.** The translucent reference overlay shows the target motion, and the opaque robot shows the generated motion.

C. Interactive Control with Temporal Composition

We showcase our model's capability for real-time *interactive* control. We feed the model time-varying commands from different modalities over the temporal horizon. As shown in Figure E.1, **OMG** follows local temporal conditions while maintaining smooth transitions between conditions.

D. Analysis on Classifier-Free Guidance

In this section, we focus on answering the following questions:

1) For single-modality conditioning, does classifier-free guidance improve instruction following? How does the scale of classifier-free guidance affect performance?
2) For compositional modality conditioning, does increasing the guidance scale for one modality steer the behavior toward better alignment with instructions in that modality?

a) Single-Modality Classifier-Free Guidance.: We aim to understand whether classifier-free guidance improves the instruction-following capability of **OMG** through the lens of human-reference conditioning. As shown in Figure E.3, increasing the guidance scale for human-reference motion makes the generated motion more aligned with the reference, with the effect becoming more pronounced over longer horizons. We further measure MPJPE and g-MPJPE between the generated motion and ground truth under varying guidance scales. As shown in Table E.2, moderate guidance improves alignment, whereas overly strong guidance hurts performance.

TABLE E.2: **Human-reference CFG sweep.** We report MPJPE and g-MPJPE under varying classifier-free guidance scales for human-reference motion across different rollout lengths. MPJPE and g-MPJPE are reported in millimeters; lower is better.

Human CFG	MPJPE	g-MPJPE	1s	3s	5s	7s	9s
0.5	190.3	557.0	62.7	407.2	236.1	213.2	91.5
1.0	96.6	261.8	42.9	59.3	117.3	144.0	216.6
1.5	74.0	218.9	38.5	51.6	108.9	93.2	139.2
2.0	71.0	343.9	38.3	60.4	109.5	82.6	73.4

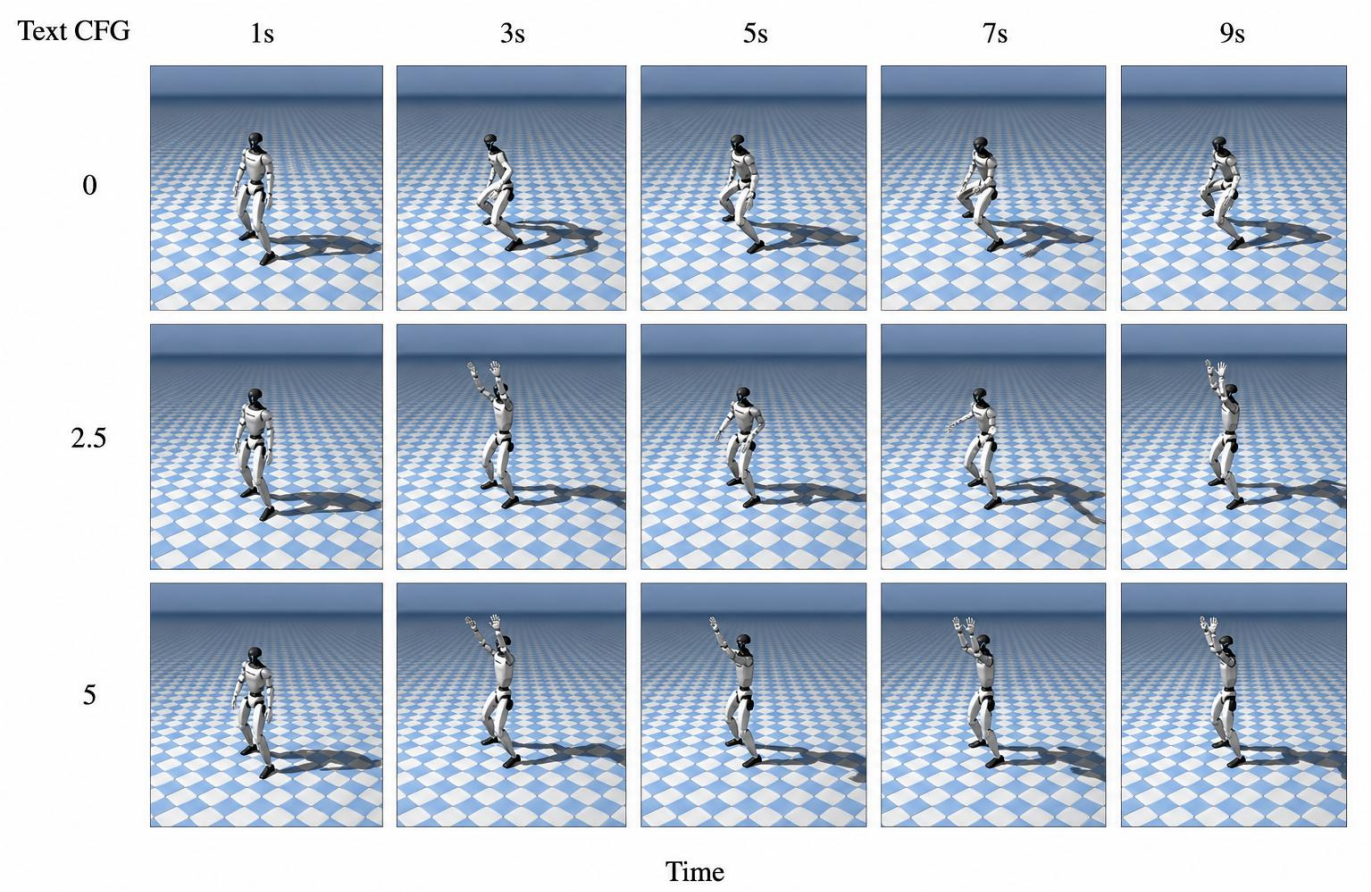


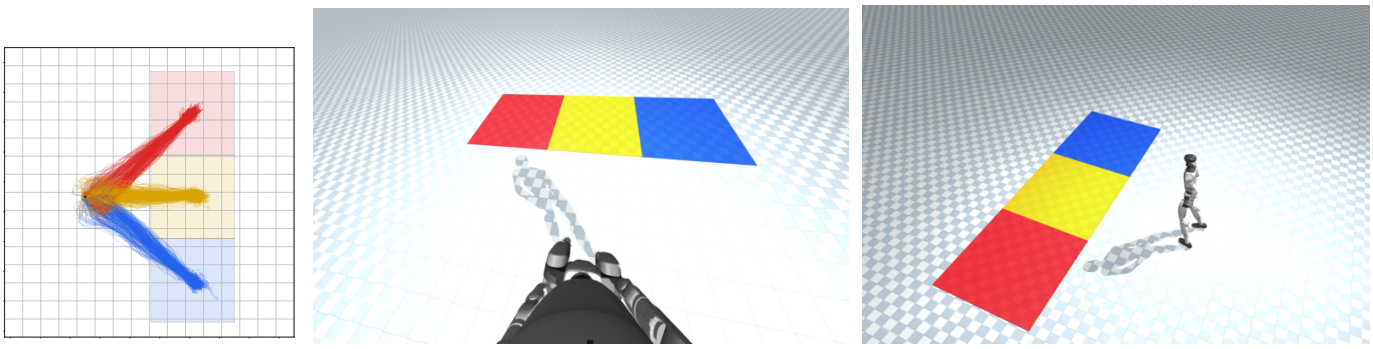
Fig. E.4: **Text-audio CFG sweep.** Columns show snapshots over time and rows vary the text guidance scale while keeping the audio condition fixed. Larger text guidance improves adherence to the language instruction in the composed audio-language setting.

b) Multi-Modality Classifier-Free Guidance.: Next, we study the effects of guidance under multi-modal conditioning. As shown in Figure E.4, we compose text and audio conditions: the audio condition specifies the dance rhythm, while the language condition asks the robot to raise its hands. Increasing the text guidance scale strengthens semantic alignment to the language instruction while preserving the audio-conditioned temporal structure.

E. Adaptation to New Modalities: Perceptive Locomotion

a) Task and Environment.: We evaluate whether the pretrained motion prior can be adapted to a new egocentric visual control task. We place three adjacent, differently colored square targets in front of the robot. The robot receives an egocentric RGB observations from a mounted camera and categorical target-color commands, and asked to locomote to the commanded square.

b) Training Data.: We leverage Kimodo [42] to generate demonstrations in advance, using its capability to condition on 2D paths. We collect 300 episodes per target color. Each episode contains 210 frames at 30 Hz, corresponding to roughly 7 s of source motion, and includes an appended target-hold segment so that the reference motion stops after reaching the target. To



(a) **Dataset.** (b) **Egocentric RGB input.** (c) **Third-person layout.**

Fig. E.5: **Perceptive Locomotion setup.** The dataset contains 300 Kimodo-generated demonstrations per target color. The policy observes only low-resolution egocentric RGB and a discrete color command. The goal is to follow the command and locomote to the corresponding color.

TABLE E.3: **Success Rate on Perceptive Locomotion.** We report success rate over 90 rollouts.

Initialization	Step	Success Rate
Scratch	500	44/90 (48.9%)
Pretrained	500	**53/90 (58.9%)**

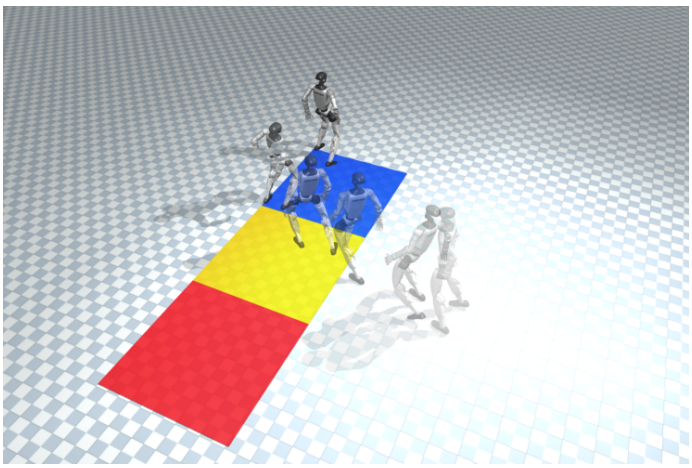


Fig. E.6: **Third-person timelapse of a successful rollout by the pretrained checkpoint.** The robot enters the commanded blue target.

train the diffusion model, we sample 2 s windows using the same G1 state representation as the motion-generation model. The visual input is the current mounted-camera RGB frame resized to 64×64 and encoded as RGB patches. For fair comparison, we disable the language prompt and instead inject the target as a learned categorical color embedding for {yellow, blue, red}.

c) Model and Training Details.: We compare two initializations under the same architecture and optimization hyperparameters. The *scratch* model is randomly initialized, while the *pretrained* model initializes the motion backbone from the 300M mixed-modality **OMG-DiT-L**. The newly introduced RGB patch encoder, visual cross-attention, and target-color embedding are trained in both cases. The visual patch gate is initialized to 0.05 so that visual conditioning is active from the start of finetuning. Both models are trained on one GPU with a local batch size of 16, and a learning rate of 6×10^{-5}.

d) Evaluation and Results.: At test time, we use online replanning from a canonical standing G1 state. Each replan samples a 2 s motion window, executes the first 0.5 s, and then replans from the updated state; the rollout budget is 210 source frames. We evaluate 30 held-out validation rollouts per target color, for 90 rollouts per checkpoint. Because this experiment is intended to measure whether the model can visually ground the target, we measure success as follows: a rollout succeeds if, at some time in the trajectory, the root/pelvis horizontal position lies within the commanded target square during execution. As shown in Table E.3, pretraining yields a higher success rate than training from scratch, showing that the pretrained motion prior transfers positively to new scenarios.