SPHLA: SPHERICAL LATENT ACTION FLOW MATCHING FOR HUMANOID WHOLE-BODY CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Behavioral foundation models (BFMs) provide pretrained whole-body policies for humanoid robots, supporting diverse behaviors through a shared policy. The latent representation of a Forward-Backward BFM conditions whole-body control. We use this representation as a generative action space, translating high-level task specifications into latent actions executed by the BFM's actor network without synthesizing kinematic motion references. However, these latent actions lie on a hypersphere, whose geometry conventional Euclidean flow matching does not preserve. To address this limitation, we introduce SphLA, spherical latent-action flow matching, which formulates conditional generation on this hypersphere. Our approach combines geodesic probability paths, tangent velocity prediction, and exponential-map integration to preserve the latent norm throughout sampling. SphLA supports both language and vision-language conditioning through independently trained action experts that share the same spherical body latent space and actor network. Experiments show that SphLA preserves spherical latent geometry and improves text–action alignment over Euclidean flow matching. Real-robot experiments validate language-directed humanoid whole-body control.

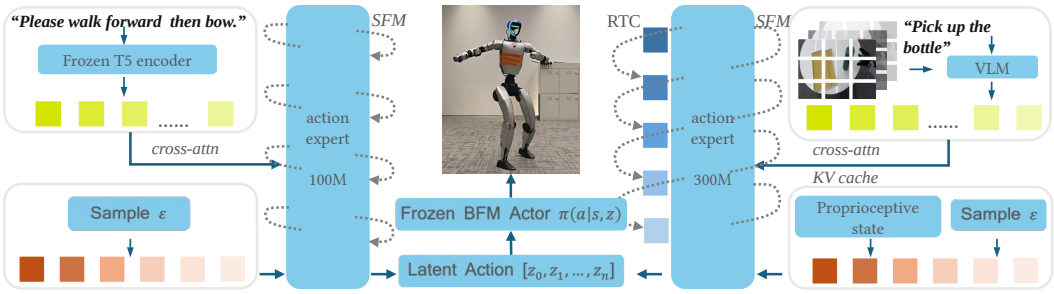


Figure 1: Overview of SphLA. Independently trained language and vision-language action experts generate trajectories in a shared spherical latent action space. A frozen BFM actor network maps these latent actions and robot observations to joint commands for whole-body control. SFM denotes spherical flow matching, and RTC denotes real-time chunking.

1 INTRODUCTION

Humanoid robots need to translate language and visual context into diverse, executable whole-body behaviors. Natural language describes the intended behavior, while execution requires prompt whole-body responses to these instructions. Bridging this gap calls for generative models that connect high-level intent with physical execution and efficiently generate diverse behaviors, enabling more natural and responsive interaction between humans and robots.

Recent behavioral foundation models (BFMs) enable a single policy to track diverse whole-body motions (Luo et al., 2026). Motion generation provides a way to connect high-level instructions to such tracking capabilities. TextOp (Xie et al., 2026) combines autoregressive latent diffusion with a universal motion tracker for streaming language control. OMG (Huang et al., 2026) couples multimodal motion generation with HoloMotion for whole-body control, while Kimodo (Rempe

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2026) generates robot-specific kinematic references that can be executed by a motion tracker. BFMs with learned motion representations can encode reference motions into latent sequences that condition whole-body control. These sequences connect motion data with physical execution and provide training targets for conditional generation. Motivated by this connection, we adopt the latent space of a Forward-Backward BFM (Touati & Ollivier, 2021; Touati et al., 2023; Li et al., 2026a) as our generation interface. The action expert maps high-level conditions directly to latent action trajectories consumed by the BFM actor network, without an intermediate kinematic motion representation.

The normalization of latent conditions derived from backward representations gives this generation space a spherical structure (Tirinzoni et al., 2025; Li et al., 2026a). Generating trajectories in this space therefore calls for a method that respects its geometry while supporting efficient inference. Diffusion models generate action sequences through iterative denoising, which can incur substantial sampling costs (Chi et al., 2023). Flow matching offers an alternative by learning velocity fields along prescribed probability paths, with the potential to generate trajectories in fewer sampling steps (Lipman et al., 2023). However, the straight-line paths used in conventional Euclidean flow matching leave the hypersphere, creating a mismatch between the generative process and the BFM latent geometry.

To address this geometric mismatch, we propose SphLA, spherical latent-action flow matching, which formulates conditional generation directly on the hypersphere. Geodesic probability paths, tangent velocity targets, and exponential-map integration preserve the latent norm throughout sampling. We instantiate this framework with independently trained text and vision-language action experts that generate latent action sequences for a shared BFM actor network. The text expert maps language instructions to latent trajectories, while the vision-language expert builds on a pretrained vision-language-action model to incorporate visual observations (NVIDIA et al., 2025). Experiments support spherical latent generation over Euclidean latent generation and direct joint-trajectory prediction for realizing language instructions through physical execution. Broader training-data coverage improves generalization to unseen text–motion pairs, while real-robot deployment demonstrates diverse and stable language-directed behaviors.

Our contributions are threefold.

- **Spherical flow matching for latent actions.** We propose SphLA for conditional generation on the spherical BFM latent space, preserving its geometry throughout sampling and improving text–action alignment over Euclidean flow matching.
- **Direct latent action generation.** We learn to generate latent trajectories that directly condition the BFM actor network, connecting high-level intent to whole-body control without synthesizing intermediate kinematic references at inference time.
- **A shared latent interface across language and vision-language conditioning.** We instantiate independently trained action experts for both conditioning modalities within the same BFM latent space. Experiments evaluate both settings and demonstrate efficient onboard generation and real-robot execution of language-directed behaviors.

2 Related Work

Behavioral Foundation Models. Recent advances in BFMs support general-purpose whole-body control, combining diverse behaviors and control objectives within a unified policy. PULSE (Luo et al., 2024) distills motion imitation skills into a latent representation for tracking and downstream control. SONIC (Luo et al., 2026) and ScaleBFM (Zeng et al., 2026) investigate scaling motion tracking through larger models, diverse reference motions, and increased training experience. HoloMotion-1 (Chen et al., 2026) and Humanoid-GPT (Qi et al., 2026) combine large motion corpora with Transformer policies to improve generalization to unseen motions. Forward-Backward representations (Touati & Ollivier, 2021; Touati et al., 2023) jointly learn task embeddings and latent-conditioned policies for zero-shot reward optimization. FB-CPR (Tirinzoni et al., 2025) incorporates unlabeled motion data to support motion tracking, goal reaching, and reward optimization, while BFM-Zero (Li et al., 2026a) extends this approach to physical humanoids. Building on these learned control representations, our work studies conditional generation in the latent action space of a pretrained FB-based BFM.

Flow Matching on Structured Geometries. Flow matching formulates generative modeling as continuous transport between distributions (Lipman et al., 2023; Tong et al., 2024). Riemannian score-based models (De Bortoli et al., 2022) and Riemannian Flow Matching (Chen & Lipman, 2024) extend this broader class of continuous generative methods to manifold-valued data. Riemannian Flow Matching Policies (Braun et al., 2024) apply geometry-aware generation to robot motions, while Geometry-Aware Image Flow Matching (Lee et al., 2026) formulates flow matching on hyperspherical image representations. These approaches account for the geometry of the target space when constructing generative dynamics. Building on this geometric principle, SphLA addresses conditional latent action generation, preserving the spherical structure of the pretrained BFM representation throughout sampling.

Language and Vision-Language Humanoid Control. Language-conditioned humanoid systems connect high-level instructions to whole-body control through different action representations. TextOp (Xie et al., 2026) combines autoregressive latent diffusion with motion tracking for streaming language control, while OMG (Huang et al., 2026) integrates modality-specific conditioning into a shared motion diffusion backbone for multimodal control. LangWBC (Shao et al., 2025) learns a language-conditioned whole-body policy, whereas RoboGhost (Li et al., 2026b) uses language-grounded motion latents to guide diffusion-based action generation. Action representation is also central to vision-language-action models, where visual observations and language jointly condition action generation. π_0 (Black et al., 2025a) predicts continuous action chunks through flow matching, while GR00T N1 (NVIDIA et al., 2025) couples a vision-language backbone with a flow-matching action expert. Humanoid systems further explore discrete motion vocabularies and learned latent representations for whole-body control (Liu et al., 2025; Jiang et al., 2026). We connect these conditioning modalities through the spherical latent space of a pretrained BFM. Independently trained language and vision-language action experts generate continuous latent action sequences for execution by the same actor network.

3 METHOD

SphLA formulates conditional motion generation in the latent action space of a pretrained BFM. As illustrated in Figure 1, separate text and vision-language action experts generate latent trajectories for a shared BFM actor network. The spherical structure of this representation determines the training paths, velocity supervision, and sampling updates.

3.1 PROBLEM FORMULATION

Given a conditioning input c, SphLA generates a sequence of latent actions $Z = (z^{(1)}, \ldots, z^{(L)})$, where L denotes the number of motion frames. Each latent conditions a pretrained BFM actor network $\pi(a \mid s, z)$ that maps the latent and robot state to an execution action. The actor remains frozen during both training and deployment.

The generation targets are derived from reference motions through the backward encoder B of a Forward-Backward BFM (Tirinzoni et al., 2025; Li et al., 2026a). For each reference state s^+, the normalized representation is

$$z = R\,\frac{B(s^+)}{\|B(s^+)\|}, \qquad z \in \mathbb{S}_R^{d-1}, \qquad R = \sqrt{d}, \tag{1}$$

where d is the latent dimension and R is the sphere radius. Framewise encoding yields a fixed-norm latent trajectory that, paired with its conditioning input, forms a training example (Z, c). Appendix A describes the underlying Forward-Backward representation.

Flow matching models this conditional distribution through a velocity field along probability paths (Lipman et al., 2023; Tong et al., 2024). For each frame, the conventional straight-line path $z_t = (1-t)z_0 + tz_1$ generally leaves the sphere even when both endpoints lie on it. SphLA therefore uses spherical paths for training and geometry-preserving updates for sampling. Here $t \in [0, 1]$ denotes flow time, distinct from the motion frame index.

3.2 SPHERICAL FLOW MATCHING

Geodesic Paths. Following Riemannian flow matching (Chen & Lipman, 2024), we connect a source latent z_0 and a target latent z_1 through a spherical geodesic. Let $\omega = \arccos(\langle z_0, z_1 \rangle / R^2)$ denote the angle between the endpoints. The interpolation path is

$$z_t = \frac{\sin((1-t)\omega)}{\sin \omega} z_0 + \frac{\sin(t\omega)}{\sin \omega} z_1, \qquad \|z_t\| = R. \tag{2}$$

Differentiating this path yields the target velocity $\dot{z}_t$, which lies in the tangent space at z_t. Paths are constructed framewise, while the network jointly predicts velocities across the latent sequence to model temporal dependencies. Appendix B provides the velocity expression and its geometric properties.

Training Objective. The network output $v_\theta(z_t, t, c)$ is supervised directly by the target velocity $\dot{z}_t$. Since the target is tangent to the sphere, the squared regression error decomposes into orthogonal tangential and radial terms,

$$\begin{aligned} \mathcal{L}_{\mathrm{CFM}} &= \mathbb{E}\big[\|v_\theta(z_t, t, c) - \dot{z}_t\|^2\big] \\ &= \underbrace{\mathbb{E}\big[\|v_{\tan} - \dot{z}_t\|^2\big]}_{\mathcal{L}_{\tan}} + \underbrace{\mathbb{E}\big[\|v_{\mathrm{rad}}\|^2\big]}_{\mathcal{L}_{\mathrm{rad}}}, \end{aligned} \tag{3}$$

where $v_{\mathrm{rad}} = \langle z_t, v_\theta \rangle z_t / R^2$ and $v_{\tan} = v_\theta - v_{\mathrm{rad}}$. Both $v_{\tan}$ and $\dot{z}_t$ lie in the tangent space, whereas v_{rad} is normal to it. Their orthogonality eliminates the cross term when expanding the squared regression error. Direct regression therefore fits the tangential velocity while penalizing radial components, with unit weight on both terms. The expectation is taken over paired training examples, source latents, and flow times, and the loss is averaged over valid motion frames. For text-conditioned training, logit-normal time sampling places greater emphasis on earlier flow times.

The text expert additionally predicts the standardized motion length using an auxiliary Smooth L1 loss $\mathcal{L}_{\mathrm{len}}$. The complete objective for the text expert is

$$\mathcal{L}_{\mathrm{text}} = \mathcal{L}_{\mathrm{CFM}} + \lambda_{\mathrm{len}} \mathcal{L}_{\mathrm{len}}. \tag{4}$$

Appendix C.1 details flow-time sampling and length supervision.

Spherical Sampling. At each sampling step, the predicted velocity is projected onto the tangent space,

$$\mathrm{proj}_{T_z}(v) = v - \frac{\langle z, v \rangle}{R^2} z. \tag{5}$$

For a tangent increment $u = \Delta t \, \mathrm{proj}_{T_z}(v_\theta(z, t, c))$, the spherical exponential map updates the latent as

$$z \leftarrow \exp_z(u) = \cos\left(\frac{\|u\|}{R}\right) z + R \sin\left(\frac{\|u\|}{R}\right) \frac{u}{\|u\|}, \tag{6}$$

with the continuous limit $\exp_z(0) = z$. Each update preserves the latent norm to numerical precision.

3.3 INSTANTIATION FOR HUMANOID WHOLE-BODY CONTROL

Figure 1 instantiates SphLA with independently trained text and vision-language action experts sharing a BFM latent action space and actor network. The frozen backward encoder provides latent training targets from reference motions, while the frozen actor executes the generated latent sequences.

The text expert uses frozen FLAN-T5-base (Chung et al., 2024) to encode text tokens. It samples complete latent sequences from spherical noise. A DiT-style Transformer (Peebles & Xie, 2023) predicts the velocity field through text cross-attention and self-attention over the latent sequence, with flow time incorporated through adaptive layer normalization. A separate length head pools text features to predict the motion duration. Users can either specify the duration or let this prediction determine the generation horizon.

The vision-language expert builds on a GR00T-based backbone (NVIDIA et al., 2025) and conditions on images, language, and proprioception. Its action outputs comprise continuous BFM body latents and hand joint targets. Body latents use spherical flow matching, while hand targets use Euclidean flow matching, with a weighted objective combining the two. Body latents are rescaled to the actor's expected radius before execution.

For text-conditioned execution, the generated sequence is supplied frame by frame to the BFM actor. The VLA expert generates successive action chunks, using real-time chunking (Black et al., 2025b) to connect overlapping predictions. In both cases, the actor converts body latents and robot-state observations into motor actions. Generation runs separately from the low-level execution loop. Appendices C and C.4 detail the architecture, implementation, and deployment protocol.

4 Experiments

The experiments evaluate four aspects of SphLA: *(i)* spherical versus Euclidean flow matching for text-conditioned latent action generation; *(ii)* action representation choices and comparisons with existing text-driven humanoid systems; *(iii)* whole-body control under vision-language conditioning; and *(iv)* real-robot execution and onboard generation efficiency.

4.1 Spherical versus Euclidean Latent-Action Generation

We investigate how the geometry of the generative path affects text-conditioned latent action generation. SphLA and the Euclidean flow-matching baseline (FM) use the same BONES-SEED training data (Bones Studio, 2026), text conditioning, DiT architecture, and optimization budget. Both models generate actions in the same BFM latent space, while differing in their interpolation paths, velocity supervision, and integration rules.

We measure constraint preservation using the relative radial error $e_r(z_t) = |\|z_t\|_2 - R|/R$. Figure 2 reports the mean error across 24 validation trajectories using midpoint integration, measured before terminal normalization. SphLA preserves the latent radius to numerical precision throughout sampling. Across the evaluated budgets for the number of function evaluations (NFE), FM's peak mean radial error ranges from 25.1% to 29.3%, and its endpoint mean radial error ranges from 0.52% to 1.16%. FM outputs require terminal normalization to satisfy the spherical latent constraint. This operation removes radial deviations at the endpoint while leaving the intermediate generation path unchanged.

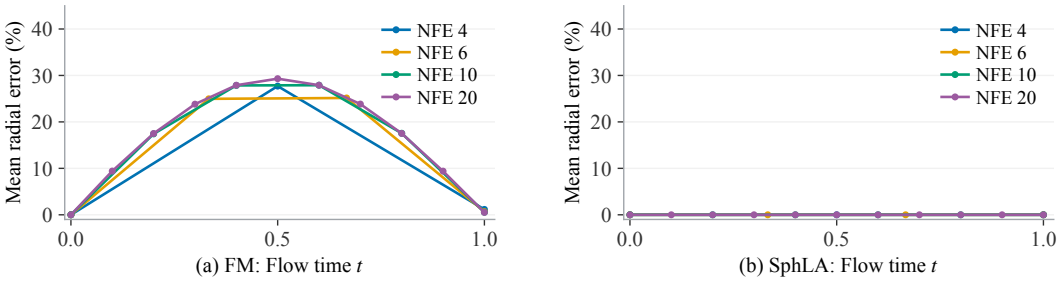


Figure 2: Mean relative radial error for (a) FM and (b) SphLA across NFE budgets using midpoint sampling, averaged over 24 validation trajectories before terminal normalization.

Generation quality is evaluated using a frozen text–latent dual encoder trained on paired text and reference latent sequences, following text-to-motion evaluation practice (Guo et al., 2022). R@1/2/3 measures text–action alignment through retrieval among 32 candidates, while latent FID and Diversity characterize distributional similarity and variation in the learned embedding space. Evaluation uses 4,189 held-out trajectories with reference durations and three generation seeds. FM outputs are normalized before evaluation. The evaluator and metric definitions are provided in Appendix D.2.

At NFE 10, SphLA improves R@1 over FM by 1.92 percentage points, with consistent gains in R@2 and R@3, and reduces latent FID from 3.320 to 2.790 (Table 1). Diversity remains close

Table 1: Text-conditioned latent action generation on the BONES-SEED validation split at NFE 10 using midpoint integration. Generated results are averaged over three generation seeds; bold indicates the best retrieval and latent FID scores among the compared methods.

Method	R@1 ↑	R@2 ↑	R@3 ↑	Latent FID ↓	Diversity
FM	0.9058	0.9585	0.9689	3.320	20.152
SphLA	**0.9250**	**0.9763**	**0.9852**	**2.790**	20.177
Real reference	0.9498	0.9906	0.9951	—	20.199

to the real-reference value, indicating improved alignment and distributional similarity without a marked loss of motion variation.

These alignment gains persist across midpoint sampling budgets of NFE 4–20 (Figure 3), with R@1 varying by less than 0.6 percentage points for each method. Increasing the budget reduces latent FID for both methods. FM achieves lower latent FID at NFE 4, whereas SphLA achieves lower values at NFE 10 and 20.

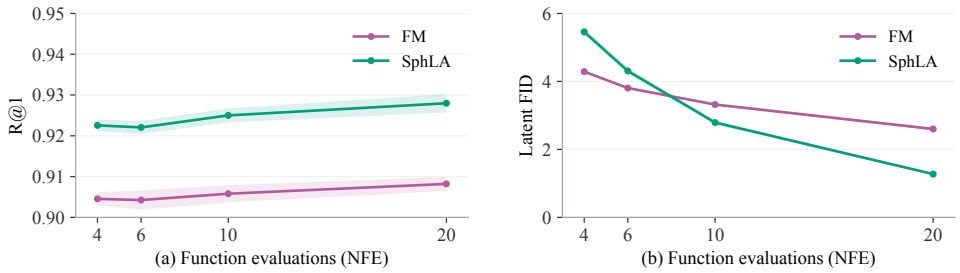


Figure 3: Text-conditioned generation quality versus function evaluations (NFE): (a) R@1 and (b) latent FID. Midpoint integration uses two evaluations per step. Shaded bands indicate one standard deviation across generation seeds.

To examine how generated latent actions reflect motion semantics, Figure 4 visualizes their t-SNE embedding, colored by action identity and instructed body involvement. Walking and running form distinguishable local concentrations, while upper- and lower-body instructions occupy partially overlapping regions. These patterns suggest local organization related to motion semantics in the generated BFM latents.

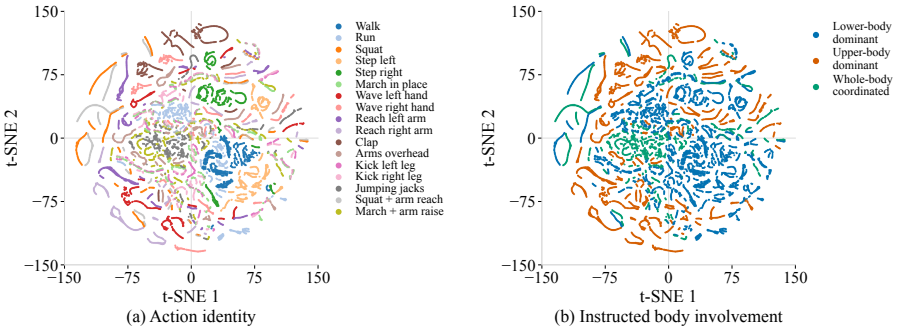


Figure 4: Semantic organization of generated latent actions. A shared t-SNE embedding is colored by (a) action identity and (b) instructed body involvement.

4.2 Comparison with Alternative Action-Generation Pipelines

We evaluate text–motion alignment on physically executed trajectories using a dual encoder trained with bidirectional contrastive learning on BONES-SEED text–motion pairs, following TMR (Petrovich et al., 2023). The evaluator scores MuJoCo execution trajectories against their corresponding

instructions. We report R@1/2/3 for text–motion retrieval, FID and Diversity for motion distribution and variation, and fall rate for physical stability.

Joint-space FM, Latent-space FM, and SphLA share the BONES-SEED training data, frozen BFM actor, evaluation cases, and retrieval candidates. Joint-space FM predicts kinematic trajectories that are encoded into BFM latents for execution, whereas Latent-space FM and SphLA directly generate latent trajectories using Euclidean and spherical flow matching, respectively. The evaluation set contains 100 distinct captions randomly sampled outside the generators' training set, spanning nine action categories, including walking, jogging, jumping, dancing, and manipulation. Each caption is evaluated at 2, 4, and 6 seconds with a shared generation seed, yielding 300 executions per method.

Table 2: Comparison of action-generation approaches through physical execution in MuJoCo using a shared text–motion evaluator. Bold indicates the best result among generation methods.

	Robustness	Text–motion alignment			Motion distribution	
Method	Fall rate ↓	R@1 ↑	R@2 ↑	R@3 ↑	FID ↓	Diversity
Reference motions	—	0.9800	0.9800	0.9800	—	1.3785
Joint-space FM	0.0733	0.7693	0.8500	0.8833	0.2137	1.3735
Latent-space FM	0.0733	0.7560	0.8420	0.8687	0.2266	1.3728
SphLA	**0.0600**	**0.8107**	**0.8793**	**0.8993**	**0.1991**	1.3757

SphLA achieves the strongest text–motion alignment among the three approaches on unseen instructions (Table 2). It improves R@1 over Latent-space FM by 5.47 percentage points, with consistent gains in R@2 and R@3 and a lower FID, supporting the role of spherical geometry in preserving motion semantics beyond training captions. Higher retrieval scores and lower FID than Joint-space FM further support latent generation as an effective alternative to predicting joint trajectories. SphLA also records a lower observed fall rate, with 18 falls (6.00%) compared with 22 (7.33%) for each baseline across 300 executions. Together, these results connect the choice of generation space and its geometry to instruction fidelity and physical stability under unseen language conditions.

Evaluation of OMG highlights the influence of annotations and evaluators on cross-system comparisons. We compare SphLA with OMG 300M using 63 caption pairs linked by BONES-SEED motion names. Both caption versions drive generation of sequences lasting 2 seconds, which are executed in MuJoCo and scored against their input captions by both evaluators. SphLA achieves higher R@1/2/3 across all tested combinations, reaching R@1 of 46.98% versus 25.08% for OMG 300M with OMG captions and its official evaluator. Caption changes affect the evaluators differently, showing that retrieval results reflect both generated behavior and its correspondence with the learned evaluation space. Appendix E.4 details this comparison, while Appendix E.5 summarizes results reported by TextOp, OMG, and ADAPT under their respective evaluation protocols.

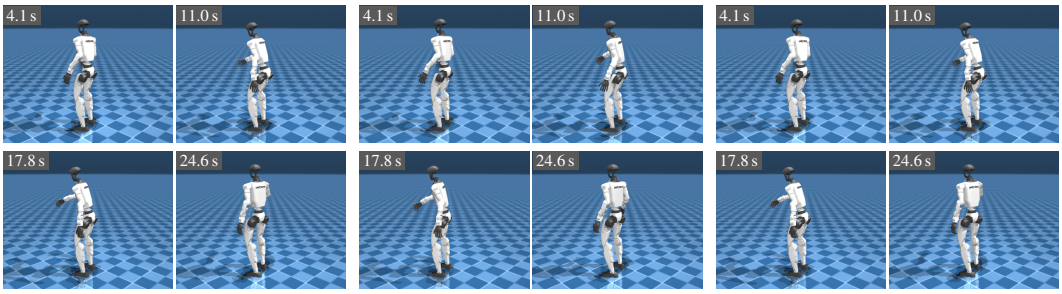
4.3 Vision-Language-Conditioned Whole-Body Control

SphLA extends to vision-language conditioning through a separately trained GR00T action expert. The expert retains the multimodal conditioning and DiT backbone, while applying spherical flow matching to body latent generation. Visual observations, language instructions, and proprioception condition the expert, and the same frozen BFM actor used in the text route executes its predicted latents through whole-body joint commands.

Training uses 138 teleoperated demonstrations of a bottle-manipulation task. Evaluation covers ten training episodes under recorded conditioning, followed by physics replay. The expert receives recorded observations, while the actor receives the evolving simulated robot state and controls all 29 body joints. Reference latents and predictions from FM and SphLA are executed through the same actor with actuator settings aligned to its training configuration. Joint-angle MAE and root-relative MPJPE measure agreement between the executed trajectories and the original demonstrations.

At NFE 1, SphLA achieves a joint-angle MAE of 7.62° and a root-relative MPJPE of 49.46 mm, compared with 8.30° and 52.29 mm for FM (Table 3). Reference-latent replay yields 7.59° and 48.97 mm, placing SphLA close to the execution fidelity obtained with reference latents. All ten SphLA replays complete without falls. These results support spherical latent generation within

a VLA expert under recorded conditioning. Representative replays are shown in Figure 5; the sampling-budget comparison and evaluation details are provided in Appendix F.



(a) Reference latents (b) FM (c) SphLA

Figure 5: Body-motion replay under recorded vision-language conditioning. Each panel shows four matched timestamps; FM and SphLA use NFE 1 and share the reference replay's BFM actor.

Table 3: Physics replay on ten training episodes. Errors against the original demonstrations are averaged equally across episodes; both generative methods use NFE 1.

Latent source	29-joint MAE (°) ↓	Root-relative MPJPE (mm) ↓
Reference latents	7.59	48.97
FM	8.30	52.29
SphLA	**7.62**	**49.46**

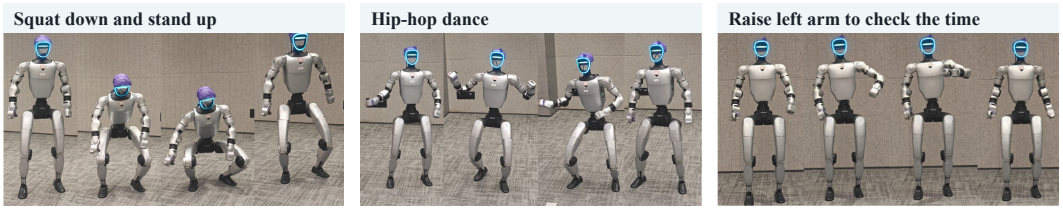
4.4 Onboard Generation and Real-Robot Execution

SphLA translates language instructions into expressive whole-body behaviors on the Unitree G1, with both generation and execution running onboard. The text encoder and DiT generator operate on the Orin NX, while the BFM actor executes generated latent sequences at 50 Hz. Beyond specifying an action, language can convey the attitude and expressive style with which it is performed. The real-robot examples span diverse actions and greeting gestures, including enthusiastic arm raising, a welcoming gesture with both arms, and a right-hand wave (Figure 6(a,b)). These examples suggest that SphLA captures aspects of expressive intent in language and translates them into variations in whole-body behavior, supporting more expressive human–robot interaction.

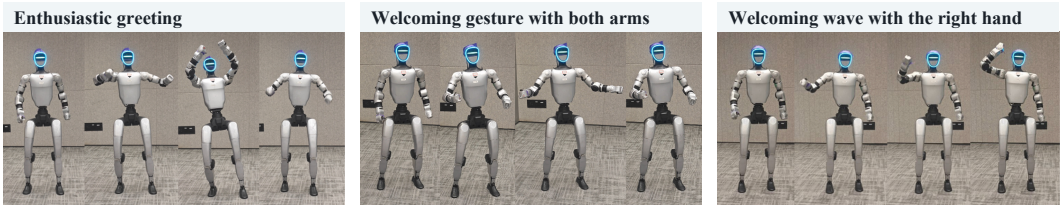
Duration conditioning allows the same language instruction to be realized within different requested time horizons. Given an identical prompt, SphLA generates motion sequences lasting 2, 4, and 6 seconds in simulation, with variations in pose and pace that preserve the intended behavior across durations (Figure 6(c)). Together with the greeting examples, these results illustrate control over both behavioral expression and motion duration.

We observe during training that increasing the amount of paired data improves generalization to unseen text–motion pairs. With only 64 or 500 training pairs, the model fits the training data well, but validation alignment peaks early and then declines, indicating overfitting. With 20,000 pairs and the full training set, validation alignment remains near 0.95 through 200,000 updates. This sustained validation performance suggests that broader data coverage helps the velocity field capture structure shared across text-conditioned latent trajectories and generalize beyond individual training pairs. Learning curves and the metric definition are provided in Appendix G.2.

Low generation latency supports rapid responses to language instructions in interactive humanoid control. With FP32 inference on the Orin NX, SphLA prepares complete motion sequences lasting 5 and 10 seconds in an average of 0.866 and 1.683 seconds, respectively. The corresponding 95th-percentile latencies remain close to the means, indicating consistent preparation times. On the H20, average preparation latency ranges from 0.143 to 0.200 seconds for requested durations ranging from 1 to 10 seconds (Table 4). These measurements demonstrate efficient generation across different temporal horizons, supporting onboard deployment and low-latency server-based interaction.



(a) Expressive behaviors (real robot)



(b) Greeting styles (real robot)



(c) Same prompt at different durations (simulation)

Figure 6: Language-directed whole-body behaviors with SphLA. Real-robot sequences show (a) diverse expressive actions and (b) variations in greeting gestures. Simulation sequences in (c) illustrate the same text instruction realized over different requested durations. Frames progress from left to right, with timestamps indicating elapsed execution time.

Table 4: Motion preparation latency (s) on onboard and server hardware across requested durations. Mean and 95th-percentile (P95) latencies are measured over 100 requests per duration on each platform with cached text conditioning.

		Requested motion duration (s)									
Platform	Statistic	1	2	3	4	5	6	7	8	9	10
Orin (onboard)	Mean	0.414	0.534	0.679	0.842	0.866	1.067	1.250	1.466	1.662	1.683
	P95	0.434	0.539	0.687	0.851	0.880	1.081	1.264	1.476	1.676	1.699
H20 (offline)	Mean	0.143	0.147	0.148	0.150	0.150	0.160	0.162	0.192	0.192	0.200
	P95	0.178	0.193	0.178	0.177	0.193	0.193	0.195	0.216	0.232	0.236

5 Conclusion

SphLA connects high-level intent with humanoid whole-body control by generating latent action trajectories for a pretrained BFM actor. By respecting the spherical geometry of this action space, it preserves the latent norm throughout sampling and improves text–action alignment over Euclidean flow matching. Independently trained experts support language and vision-language conditioning within the same framework. Experiments demonstrate expressive real-robot behaviors, controllable motion duration in simulation, and efficient onboard generation. These results highlight how geometry-aware generation can connect learned control representations with diverse and responsive humanoid behaviors.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript polishing, literature retrieval, data processing and visualization, and code refinement. The authors reviewed the AI-assisted outputs and take full responsibility for the accuracy and integrity of the final manuscript.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems*, 2025a.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems*, volume 38, pp. 37596–37620, 2025b.

Bones Studio. BONES-SEED: Skeletal everyday embodiment dataset. `https://huggingface.co/datasets/bones-studio/seed`, 2026.

Max Braun, Noémie Jaquier, Leonel Rozo, and Tamim Asfour. Riemannian flow matching policy for robot motion learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5144–5151, 2024.

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. *arXiv preprint arXiv:2605.15336*, 2026.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, 2024.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. In *Advances in Neural Information Processing Systems*, 2022.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3D human motions from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5152–5161, 2022.

Siqiao Huang, Kun-Ying Lee, Dongming Qiao, Guanqi He, Zhenyu Wang, Yitang Li, Shaoting Zhu, and Hang Zhao. OMG: Omni-modal motion generation for generalist humanoid control. *arXiv preprint arXiv:2606.10340*, 2026.

Haoran Jiang, Jin Chen, Qingwen Bu, Li Chen, Modi Shi, Yanjie Zhang, Delong Li, Chuanzhe Suo, Chuang Wang, Zhihui Peng, and Hongyang Li. WholeBodyVLA: Towards unified latent VLA for whole-body loco-manipulation control. In *International Conference on Learning Representations*, 2026.

Junho Lee, Kwanseok Kim, and Joonseok Lee. Geometry-aware image flow matching. In *International Conference on Machine Learning*, 2026.

Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, Alessandro Lazaric, Matteo Pirotta, and Guanya Shi. BFM-Zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning. In *International Conference on Learning Representations*, 2026a.

Zhe Li, Yangyang Wei, Boan Zhu, Yibo Peng, Tao Huang, Pengwei Wang, Zhongyuan Wang, Cheng Chi, Chang Xu, and Shanghang Zhang. From language to locomotion: Retargeting-free humanoid control via motion latent guidance. In *International Conference on Learning Representations*, 2026b.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Zhirui Liu, Kaiyang Ji, Ke Yang, Yahao Fan, Jingyi Yu, Ye Shi, and Jingya Wang. Commanding humanoid by free-form language: A large language action model with unified motion vocabulary. *arXiv preprint arXiv:2511.22963*, 2025.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. In *International Conference on Learning Representations*, 2024.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Jinhyung Park, David Sami, Zi Wang, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi Fan, and Yuke Zhu. SONIC: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117):eaed4592, 2026.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Mathis Petrovich, Michael J. Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9488–9497, 2023.

Zekun Qi, Xuchuan Chen, Jilong Wang, Chenghuai Lin, Yunrui Lian, Wenyao Zhang, Xinqiang Yu, He Wang, and Li Yi. Humanoid generative pre-training for zero-shot motion tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20834–20844, 2026.

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 preprint arXiv:2603.15546*, 2026.

RoboParty Lab Team. UFO: An unsupervised reinforcement learning framework for humanoid control. https://github.com/Roboparty/UFO, 2026.

Yiyang Shao, Bike Zhang, Qiayuan Liao, Xiaoyu Huang, Yuman Gao, Yufeng Chi, Zhongyu Li, Sophia Shao, and Koushil Sreenath. LangWBC: Language-directed humanoid whole-body control via end-to-end learning. In *Robotics: Science and Systems*, 2025.

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. In *International Conference on Learning Representations*, 2025.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In *Advances in Neural Information Processing Systems*, 2021.

Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *International Conference on Learning Representations*, 2023.

Yan Wu, Chenhao Li, Kaifeng Zhao, Gen Li, Marco Hutter, and Siyu Tang. ADAPT: Agile diffusion action priors for robust and steerable online text-driven humanoid control. *arXiv preprint arXiv:2609.00677*, 2026.

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.

Weishuai Zeng, Kangning Yin, Xiaojie Niu, Shunlin Lu, Weixiang Zhong, Jiahe Chen, Feiyu Jia, Xiao Chen, Zirui Wang, Furui Xu, Ming Zhou, Kailin Li, Weinan Zhang, He Wang, Li Yi, Dahua Lin, Jiangmiao Pang, and Jingbo Wang. Scaling behavior foundation model for humanoid robots. *arXiv preprint arXiv:2607.15163*, 2026.

A Forward-Backward Representations and the Actor Interface

Forward-Backward (FB) representations factorize future state occupancy to connect task embeddings with latent-conditioned policies. The backward encoder and BFM actor in SphLA use pretrained weights released by UFO (RoboParty Lab Team, 2026) and remain frozen during training and deployment.

Successor measure and task embeddings. A policy π induces the discounted successor measure

$$M^{\pi}(X \mid s, a) = \sum_{k=0}^{\infty} \gamma^k \Pr\big(s_{k+1} \in X \mid s, a, \pi\big), \tag{7}$$

where X is a set of states and γ is the discount factor. FB representations (Touati & Ollivier, 2021; Touati et al., 2023) approximate it as

$$M^{\pi_z}(\mathrm{d}s' \mid s, a) \approx F(s, a, z)^{\top} B(s')\, \rho(\mathrm{d}s'), \tag{8}$$

where ρ is a base measure over states. The forward map F, backward map B, and policies $\{\pi_z\}$ indexed by $z \in \mathbb{R}^d$ are learned jointly. Integration against a reward r gives $Q_r^{\pi_z}(s, a) \approx F(s, a, z)^{\top} z_r$, with $z_r = \mathbb{E}_{s' \sim \rho}[B(s')r(s')]$ serving as a task descriptor. FB-CPR (Tirinzoni et al., 2025) incorporates unlabeled behaviors through conditional policy regularization; BFM-Zero (Li et al., 2026a) extends this approach to physical humanoids.

Spherical actor interface. The normalization in Eq. (1) fixes the norm of each reference-frame embedding to $R = \sqrt{d}$. Framewise encoding thus yields spherical latent trajectories that condition the frozen actor and serve as SphLA's generation targets.

B Spherical Geometry and Numerical Verification

Conditional paths and velocity targets. The Euclidean baseline uses

$$z_t = (1 - t)z_0 + tz_1, \qquad \dot{z}_t = z_1 - z_0. \tag{9}$$

Differentiating the spherical path in Eq. (2) gives

$$\dot{z}_t = \frac{\omega}{\sin \omega} \left[-\cos((1 - t)\omega)z_0 + \cos(t\omega)z_1 \right]. \tag{10}$$

For $0 < \omega < \pi$, the orthonormal directions $e_1 = z_0/R$ and $e_2 = (z_1 - \cos\omega\, z_0)/(R \sin\omega)$ give $z_t = R[\cos(t\omega)e_1 + \sin(t\omega)e_2]$ and $\dot{z}_t = R\omega[-\sin(t\omega)e_1 + \cos(t\omega)e_2]$. Hence,

$$\langle z_t, \dot{z}_t \rangle = 0, \qquad \|\dot{z}_t\| = R\omega. \tag{11}$$

The target therefore remains tangent and follows the geodesic at constant speed.

Norm preservation during integration. For $z \in \mathbb{S}_R^{d-1}$, sampling uses the tangent increment $u = \Delta t\, [v_\theta - \langle v_\theta, z \rangle z / R^2]$. For $u \neq 0$, orthogonality gives

$$\left\| \exp_z(u) \right\|^2 = R^2 \cos^2\left(\frac{\|u\|}{R} \right) + R^2 \sin^2\left(\frac{\|u\|}{R} \right) = R^2. \tag{12}$$

The zero-increment case leaves z unchanged. Norm preservation holds even for an imperfect velocity prediction, although prediction and integration errors still affect the trajectory on the sphere.

Numerical verification. Random-batch checks verify path norms, tangency, speed, and integration (Table 5). Four exponential-map steps recover a constant-speed geodesic oracle's endpoint within $3.3 \times 10^{-7} R$, versus $1.0 \times 10^{-1} R$ for Euler steps with radial retraction. This oracle result does not imply exact integration of a learned, state-dependent velocity field.

Numerical precision. Geometric operations, including inner products, use float32 regardless of the network's compute dtype. This avoids the bfloat16 rounding of $1 - 10^{-6}$ to 1, which weakens guards around $\arccos$ and $\sin\omega$.

Table 5: Numerical checks of spherical paths and integration in FP32. Oracle tests use the analytic velocity field.

Property	Expected	Measured residual
Endpoint norm	$\|z_0\| = R$	2.9×10^{-6}
Interpolant on sphere	$\|z_t\| = R\,\forall t$	3.8×10^{-6}
Velocity target tangent	$\langle \dot{z}_t, z_t \rangle = 0$	2.6×10^{-7}
Constant geodesic speed	$\|\dot{z}_t\| = R\omega$	5.7×10^{-6}
Finite-difference check	$\Delta z_t / \Delta t \to \dot{z}_t$	3.2×10^{-4}
Boundary conditions	$z_t\|_{t=0} = z_0,\ z_t\|_{t=1} = z_1$	0
Exponential map endpoint (oracle, N=4)	recovers z_1	$3.3 \times 10^{-7}\,R$
Exponential map preserves norm (no retraction)	$\|z\| = R$	1.9×10^{-6}
Euler + retraction (oracle, N=4)	recovers z_1	$1.0 \times 10^{-1}\,R$
Mixed path: body factor on sphere	$\|z_{\text{body}}\| = R$	2.9×10^{-6}
Mixed path: hand factor	Euclidean interpolant	0

C Network Architecture and Implementation

C.1 Text Action Expert

The text expert combines a frozen T5 encoder with a DiT of 8 blocks, width 768, and 12 attention heads. Each block applies text cross-attention, bidirectional motion self-attention, and a feed-forward layer. Flow time modulates these branches through adaptive layer normalization; rotary embeddings encode motion positions. Linear projections connect the latent space to the Transformer width. The body latent dimension is $d = 256$ and the actor-interface radius is $R = 16$.

The expert samples complete sequences from spherical noise without motion-history inputs. An attention-pooling head predicts the standardized frame count from text features when no duration is supplied, within the supported sequence range. A supplied duration instead sets the generation horizon. A learned null text embedding supports classifier-free guidance. For a clip of L frames, the length objective is

$$\mathcal{L}_{\text{len}} = \mathbb{E}_{(L,c)}\left[\text{SmoothL1}\left(h_\phi(c) - \frac{L - \mu_L}{\sigma_L}\right)\right], \tag{13}$$

where μ_L and σ_L are training length statistics. This term augments velocity regression as specified in Section 3.2.

Training configuration. Training uses batch size 64, length bucketing, learning rate 4.8×10^{-4}, 2,000 warmup steps, and 200,000 updates. The length-loss weight is 0.2 and the exponential moving average decay is 0.9999. Flow time is sampled as $t = \text{sigmoid}(\xi)$ with $\xi \sim \mathcal{N}(-0.61, 1.49^2)$ and clipped to $[10^{-4}, 1-10^{-4}]$, emphasizing earlier flow times. Sampling budgets and guidance weights accompany the corresponding evaluations.

C.2 Vision-Language Action Expert

Shared latent representation. The VLA expert jointly predicts the same BFM body latents and 14 hand-action channels. Body-channel normalization bounds of $[-R, R]$ reduce min-max normalization to $\bar{z} = z/R$, placing the body representation on a unit sphere during training and sampling. Multiplication by R restores the actor's scale at execution. Since $|z_i| \le \|z\| = R$, these bounds contain all valid latents. Hand channels use per-channel normalization.

Training objective. The SphLA VLA expert uses masked MSE with body and hand weights of 32 and 1 to balance normalized target scales. Body targets follow spherical paths; hand targets follow linear interpolation. The FM comparison configuration is specified in Appendix F.2.

C.3 VLA Integration and Real-Time Chunking

Sampling projects body velocities onto the current tangent space before exponential-map updates and uses Euler updates for hand channels. Real-time chunking (RTC) is enabled for all reported

VLA comparisons, using a 20-frame overlap, a 6-frame frozen prefix, and velocity ramp rate 3; additional spherical interpolation (SLERP) is disabled. Appendix F provides the evaluation protocol.

C.4 Real-Robot Deployment

The text generator and frozen BFM actor run in separate processes on the Unitree G1. Generated sequences enter a ready queue and play after an explicit start command. The actor consumes body latents directly at 50 Hz with robot-state feedback. A standing latent is held during idle periods; spherical interpolation connects it to the sequence at entry and exit. These playback transitions are separate from VLA chunk overlap.

The onboard Orin NX service uses PyTorch FP32 inference with resident T5 and DiT models and scaled dot-product attention. Text features are cached by instruction; conditioning features and cross-attention keys and values are reused across sampling steps. The service transfers latents to CPU memory and prepares playback transitions before marking a plan ready. Appendix G reports execution examples and latency.

D Text-Conditioned Generation Evaluation

D.1 Dataset and Preprocessing

The comparisons use BONES-SEED (Bones Studio, 2026), excluding mirrored motions and clips lasting 10 seconds or longer. Joint-space FM, Latent-space FM, and SphLA share this filtered training set. The latent models target framewise backward encodings; Joint-space FM targets kinematic trajectories containing root poses and joint positions.

D.2 Evaluation Protocol

Text–latent evaluator. A dual encoder maps real text–latent sequence pairs to shared embeddings using a text projection and a temporal Transformer with masked mean pooling. Training uses symmetric contrastive loss, treating matching descriptions and annotations of the same segment as positives. The frozen evaluator is shared across methods. Its text and sequence embeddings, u_i and v_i, describe whole sequences, distinct from the backward encoder's per-frame control latents.

Evaluation set and repeated sampling. Evaluation uses 4,189 trajectories from the generator's original validation holdout at their reference lengths. Retrieval uses 32-candidate groups and five fixed random regroupings. Each regrouping evaluates 4,160 queries in complete groups, omitting 29 samples after permutation; latent FID uses all 4,189 trajectories. Methods share reference data, candidates, and generation seeds 42, 43, and 44. Standard deviations use the population convention across these seeds and measure generation variability for fixed trained models. Table 1 reports means with midpoint sampling at NFE 10 and batch size one. R@1 standard deviations are 0.0021 for FM and 0.0017 for SphLA; latent FID standard deviations are 0.040 and 0.067. Figure 3 uses the same conditions with batch size 32. Each midpoint update requires two velocity evaluations. At the main-table setting, omitting FM's terminal normalization changes R@1 by less than 0.02 percentage points and latent FID by less than 0.03, preserving the ranking.

Metric definitions. Retrieval ranks sequences by cosine similarity to u_i. R@K counts queries whose paired sequence is among the top K candidates, for $K \in \{1, 2, 3\}$, using paired-sequence hits rather than multi-positive retrieval. Latent FID compares the means and sample covariances of real and generated embeddings:

$$\mathrm{FID}_{\mathrm{latent}} = \|\mu_r - \mu_g\|_2^2 + \mathrm{tr}\Big(\Sigma_r + \Sigma_g - 2(\Sigma_r^{1/2}\Sigma_g\Sigma_r^{1/2})^{1/2}\Big). \tag{14}$$

Diversity is the mean distance between $P = 300$ randomly sampled pairs of distinct generated sequences, $\frac{1}{P}\sum_{p=1}^{P}\|v_{a_p} - v_{b_p}\|_2$, with pair-sampling seed 42. Reference diversity uses the same procedure. FID and Diversity use unnormalized sequence embeddings; retrieval uses unit-normalized embeddings. All three metrics operate in the learned text–latent feature space.

Radial traces. Figure 2 uses 24 stratified validation trajectories, midpoint sampling, and generation seed 42. Relative radial error $|\|z\| - R|/R$ is averaged over valid frames per trajectory and then across trajectories. Traces include the initial state and completed updates, excluding internal midpoint stages and terminal normalization. SphLA's maximum recorded relative error is 3.58×10^{-7}. For sphere-valued endpoints with angle ω, the Euclidean conditional path satisfies

$$\|(1-t)z_0 + tz_1\|_2^2 = R^2\left[1 - 2t(1-t)(1-\cos\omega)\right]. \tag{15}$$

Orthogonal endpoints yield midpoint norm $R/\sqrt{2}$ and approximately 29.3% radial deviation. This describes the conditional training path; recorded traces measure the learned sampling trajectories.

Semantic visualization. The t-SNE analysis uses 17 action categories with three subject variants each ("A person," "Someone," and "The person"). Seeds 42, 43, and 44 produce 153 trajectories, each containing 250 latent actions or 5 seconds at 50 Hz. Generation uses text conditioning without motion history, 20 sampling steps, and guidance weight 1.0. All 38,250 frames enter one two-dimensional t-SNE embedding with perplexity 30 and seed 42, without category labels. Both panels of Figure 4 show these same points. Instructions determine the body-involvement labels, comprising eight lower-body, six upper-body, and three whole-body categories, rather than measured joint activity.

D.3 First-Order and Midpoint Sampling

FM uses the standard explicit midpoint update (Lipman et al., 2024). SphLA constructs the intermediate state with a half-step exponential map, evaluates velocity there, and projects it onto the original tangent space for the full exponential-map update. This preserves the spherical norm.

Table 6 compares samplers at NFE 10 using 4,189 validation conditions, batch size 32, terminal normalization, and three seeds. First-order sampling uses ten steps; midpoint uses five. Midpoint lowers latent FID by 20.0% for FM and 25.5% for SphLA, with unchanged FM R@1 and slightly lower SphLA R@1. Its benefit is therefore distributional similarity; SphLA outperforms FM in R@1 and latent FID with either sampler.

Table 6: First-order versus midpoint sampling at NFE 10. Results are averaged over three generation seeds; bold indicates the better method for each sampler and metric.

	First-order (10 steps)		Midpoint (5 steps)		FID reduction
Method	R@1 ↑	Latent FID ↓	R@1 ↑	Latent FID ↓	with midpoint
FM	0.9058	4.151	0.9058	3.320	20.0%
SphLA	**0.9292**	**3.743**	**0.9250**	**2.790**	25.5%

E Alternative Action-Generation Pipelines

E.1 Compared Systems and Execution

Joint-space FM, Latent-space FM, and SphLA share the BONES-SEED training data described in Appendix D.1 and the same frozen BFM actor. Joint-space FM predicts root positions, root orientations, and joint positions. A centered three-frame moving average with replicated boundaries smooths predictions before denormalization. Root quaternions are normalized with consistent signs across frames, and the frozen backward encoder maps the reference to BFM latents. One extra generated frame supplies the initial finite difference and is discarded after encoding. Latent-space FM and SphLA generate latent sequences directly.

Each rollout begins with 0.2 seconds of PD settling, 2 seconds of actor-controlled standing, and 0.6 seconds tracking the first task frame, retaining the actor's observation history throughout. Scoring uses only the requested task interval at 50 Hz, recorded in full even after a fall.

TextOp and OMG use their own execution pipelines with BeyondMimic-based and HoloMotion trackers, respectively. Their published results appear in Appendix E.5; the caption diagnostic in Appendix E.4 examines cross-system scoring sensitivity.

E.2 Evaluation Set

The BONES-SEED evaluation set contains 100 distinct captions randomly sampled outside the generators' training set. It covers walking (12), jogging (12), jumping (12), dancing (12), turning (10), manipulation (12), idle behavior (12), crouching (10), and stretching (8). Each caption is evaluated at all three target durations of 2, 4, and 6 seconds, yielding 300 caption–duration conditions, with 100 per duration. All methods use the same conditions, generation seed 0, and classifier-free guidance weight 2.0, producing 300 executions per method.

E.3 Execution Evaluator and Metrics

The evaluator trains on BONES-SEED text–reference-motion pairs and scores recorded executions. Forward kinematics produces positions of 30 G1 body points relative to the initial root position and heading, giving 90 features per frame at 50 Hz. Text features come from frozen FLAN-T5-base with a 96-token limit. The motion encoder has a convolutional frontend and six Transformer layers of width 512, four attention heads, rotary embeddings, and dropout 0.1. Masked mean pooling and projection produce unit-normalized 512-dimensional text and motion embeddings.

Bidirectional contrastive training uses temperature 0.07 and treats matching positive identifiers or normalized captions as positives. Source groups and positive identifiers are disjoint across the supplied splits. The epoch-30 checkpoint uses length-based batches of up to 32 sequences, AdamW with learning rate 10^{-4} and weight decay 0.01, and seed 42; singleton remainders are skipped. This evaluator is shared across methods and differs from the text–latent evaluator in Appendix D.2.

Retrieval. Each execution supplies one text query and its paired motion. The 31 negative candidates are sampled uniformly without replacement from cases in other action categories, excluding identical captions. Five fixed assignments with seed 42 yield 1,500 queries per method. R@K measures paired-motion hits among the top K cosine similarities, for $K \in \{1, 2, 3\}$. Cross-category negatives make this a test of cross-category retrieval rather than fine-grained distinctions within a category.

FID and Diversity. FID compares the means and sample covariances of 300 executed-motion embeddings with those of 100 reference motions deduplicated by source identity. Each reference is encoded once at its full original length, without duration-based cropping or replication. Diversity averages Euclidean distance over 300 random pairs of distinct execution indices with seed 0. Both metrics use unit-normalized 512-dimensional embeddings, unlike the unnormalized latent features in Appendix D.2. All executions, including falls, contribute to retrieval, FID, and Diversity. Reference retrieval uses the same evaluator and 1,500 fixed queries. Reference motions are not physically executed, so their fall rate and self-comparison FID are omitted.

Fall rate. A fall occurs if root height drops below 0.35 meters during the recorded task interval. Fall rate divides the count by 300, excluding initialization. This criterion measures stability, not semantic task completion.

E.4 Generation across Caption Versions and Evaluators

The comparison in Table 7 uses 63 pairs of BONES-SEED and OMG captions associated through motion base names. Pairing identifies the motion name shared across the datasets; actor takes, temporal crops, and semantic details can differ. Each caption independently conditions SphLA and OMG 300M, yielding 126 executions per model. All sequences last 2 seconds and use generation seed 0, with guidance weights of 1 for SphLA and 2.5 for OMG 300M.

Each MuJoCo execution is scored against its generation caption using our trained text–motion evaluator and OMG's released T5-3B evaluator (`step_004000.pt`). Recordings at 50 Hz are resampled to 60 frames at 30 Hz for the OMG evaluator, retaining each evaluator's input normalization. For each model, retrieval candidates are drawn from its 126 executions, including both caption versions. Each query has one paired execution and 31 distinct negatives, sampled in five assignments with seed 42. Candidate identities and positive positions are shared across models and evaluators.

Executions generated from the other caption of the same motion can serve as negatives. Scores average 315 queries per model and text version.

Table 7: Retrieval across caption versions and evaluators on 63 paired motion annotations. Each caption drives generation and scores the resulting execution. Values are percentages.

		Our evaluator			OMG evaluator		
Model	Caption	R@1	R@2	R@3	R@1	R@2	R@3
SphLA	BONES-SEED	65.40	79.37	84.76	41.27	64.76	77.14
OMG 300M	BONES-SEED	6.35	13.97	18.41	9.52	17.46	24.76
SphLA	OMG	37.46	55.87	61.59	46.98	66.03	74.92
OMG 300M	OMG	9.21	14.60	20.95	25.08	34.60	41.59

SphLA outperforms OMG 300M in all retrieval metrics across the four settings (Table 7). BONES-SEED captions favor SphLA under our evaluator, while OMG captions improve its R@1/2 under the OMG evaluator. Because caption changes affect both generation and scoring, this comparison measures their combined influence, rather than evaluator sensitivity alone.

E.5 Published Results under Original Evaluation Protocols

Table 8 summarizes means from the arXiv v1 papers. TextOp (Xie et al., 2026) evaluates generated segments and MuJoCo executions on BABEL validation text streams (Tables III and IV). M+G and G denote trackers trained on motion-capture plus generated motions and generated motions alone. OMG (Huang et al., 2026) reports generation quality and tracking fall rate separately (Table 1); FID is converted from 10^{-2} display units. ADAPT (Wu et al., 2026) reports R@1/3/5 on successful rollouts with changing instructions, without FID or Diversity (Table 1); RC denotes residual correction. TextOp and ADAPT success rates are not converted to fall rates. Differences in data, evaluators, and evaluated motion stages preclude direct ranking across these protocols or against Table 2.

Table 8: Results reported by TextOp, OMG, and ADAPT under their original protocols. Retrieval and fall rates use a 0–1 scale; dashes denote unreported metrics.

Method	Fall rate ↓	R@1 ↑	R@2 ↑	R@3 ↑	FID ↓	Diversity
TextOp (generation)	—	0.300	0.481	0.607	3.072	9.220
TextOp (M+G)	—	0.273	0.453	0.574	3.454	8.386
TextOp (G)	—	0.281	0.464	0.577	3.371	8.409
OMG-L	0.0029	0.5850	0.7490	0.8145	0.0955	1.39
OMG-XL	0.0078	0.6543	0.8037	0.8691	0.0603	1.39
ADAPT (w/o RC)	—	0.5950	—	0.7800	—	—
ADAPT	—	0.4460	—	0.6495	—	—

F Vision-Language-Action Evaluation

F.1 Evaluation Protocol

Evaluation uses recorded egocentric observations, language, and 46-dimensional proprioception from training episodes $0, 1, 2, 4, 5, 6, 7, 8, 9, 10$, totaling 10,633 frames. Prediction uses a 40-frame horizon and a 20-frame execution window. Physics replay supplies simulated-state feedback to the BFM actor, while the expert retains recorded observations rather than receiving new rendered views.

Latent angular error measures angles between unit-normalized predicted and demonstrated body latents, excluding hand outputs. Errors are averaged within episodes and then equally across episodes, without smoothing or temporal realignment.

F.2 Physical Execution

All 29 body joints, including six wrist joints, remain active during replay; separate hand outputs are not executed. The shared frozen actor uses its training actuator configuration, including wrist gains, effort and velocity limits, armature, and friction.

Joint-angle MAE averages absolute errors over the 29 joints and replay frames. Root-relative MPJPE averages Euclidean errors over 30 body positions (the pelvis and 29 links). Each trajectory is translated by its own pelvis position, without rotational alignment, and distances are reported in millimeters. Each metric is computed per episode and then averaged equally across episodes. Reference-latent replay also passes through the actor and physics simulation, so its error against the original demonstration can be nonzero.

The FM and SphLA experts use the same multimodal backbone, demonstration set, training budget of 30,000 updates, batch size of 32, and prediction horizon of 40 frames. Their body loss weights differ (0.307 for FM and 32 for SphLA); SphLA uses hand weight 1. Thus, the comparison covers two training configurations with a shared execution pipeline. All ten replays complete without falls for both methods at every evaluated budget and for reference latents.

F.3 Generation Budget

Both experts use one network evaluation per integration step. We compare NFE 1, 2, and 4 with RTC enabled, using the same ten episodes and replay configuration (Table 9). Each setting uses one saved prediction per episode; generation seeds were not recorded, so the comparison does not use matched random draws across settings.

Table 9: VLA generation across sampling budgets with RTC enabled. Latent angular error, 29-joint MAE, and root-relative MPJPE are averaged equally over ten training episodes.

Method	NFE	Latent error (°) ↓	29-joint MAE (°) ↓	MPJPE (mm) ↓
Reference latents	—	—	7.59	48.97
FM	1	26.06	8.30	52.29
	2	14.15	7.68	50.47
	4	12.20	7.72	50.57
SphLA	1	6.51	7.62	49.46
	2	13.40	7.74	50.08
	4	19.54	8.01	51.06

SphLA attains its lowest latent and execution errors at NFE 1. More evaluations improve FM's latent error, with its execution errors lowest at NFE 2; FM outperforms SphLA on all three metrics at NFE 4. SphLA at NFE 1 approaches FM's execution accuracy at NFE 2 or 4, supporting single-evaluation generation in this setting. NFE measures network evaluations, not wall-clock latency.

G Real-Robot Results and Generalization

G.1 Behavior and Duration Examples

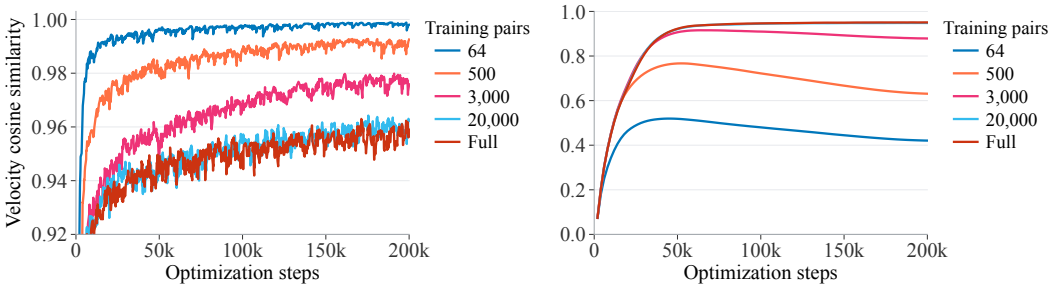
Figure 6(a,b) shows real-robot execution; panel (c) shows simulation. Panel (a) includes squatting and standing, hip-hop dancing, and checking the time with the left arm. The recorded squat and watch prompts are "A person squats down and then stands back up" and "A person swiftly raises their left arm in front of their chest and tilts their head downward to check the time."

Panel (b) uses "A person raises both arms overhead in celebration, pauses, then lowers them," "A person stands still, then raises their arms and spreads them wide in a warm, welcoming gesture before lowering them again," and "A person stands and waves their right hand." Figure labels abbreviate these prompts, which differ in both action content and wording.

Panel (c) uses "The dancer is grooving by clapping their hands while swaying side to side" with durations of 2, 4, and 6 seconds. Frames occur at $(0.5, 1, 1.5)$, $(1, 2, 3)$, and $(2, 3.5, 5)$ seconds, respectively.

G.2 Data Scale and Velocity-Field Generalization

The data-scale study uses 64, 500, 3,000, 20,000, and the full set of training pairs, with identical architecture, hyperparameters, and 200,000 updates. Velocity alignment averages cosine similarity between predicted velocities and flow-matching targets over valid latent frames at sampled flow times. It measures local direction prediction along the training probability path; Section 4.1 evaluates fully sampled trajectories.



(a) Training

(b) Validation

Figure 7: Effect of training-data scale on velocity alignment. Panels show (a) training and (b) validation curves over 200,000 updates; vertical scales differ.

Figure 7 complements the main-text comparison with the 3,000-pair setting, which shows a smaller training–validation gap and milder validation decline than the 64- and 500-pair runs. The 20,000-pair and full-data runs have nearly overlapping validation curves around 0.95, outperforming the smaller-data settings despite lower training alignment. Curves use training-log exports without additional smoothing or uncertainty estimates. This study evaluates velocity prediction, not duration-conditioned execution across data scales.

G.3 Inference Latency

Timing protocol. Table 4 uses "A person waves the right hand" with requested durations of 1 to 10 seconds, or 50 to 500 frames. One warmup request per duration is excluded, followed by 100 requests per duration in a balanced random interleaving. Requests are processed individually with cached text features after warmup (Appendix C.4).

Wall-clock timing runs from the local client's request to detection of the complete plan in the ready queue, polled every 20 ms. It includes service handling, generation, CPU transfer, playback preparation, and readiness detection; model loading, initial text encoding, the manual playback trigger, and physical execution are excluded. Duration counts generated frames at 50 Hz, excluding playback transitions. Onboard standard deviations range from 0.004 to 0.010 seconds; the main table reports means and 95th percentiles.

Server reference. The H20 run uses the same instruction, duration order, warmup, and timing boundary for 1,000 requests. It uses PyTorch 2.11.0, CUDA 12.8, FP32, 20 geodesic integration steps, guidance weight 1, and EMA weights at update 190,000, with matrix-multiplication TF32 disabled. The service runs without the robot policy or playback process. The timings characterize the complete deployment configurations, rather than hardware speed in isolation.