LANGUAGE-GROUNDED CONTROLLER LATENTS FOR HUMANOID WHOLE-BODY CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-conditioned humanoid motion is commonly built as a cascade in which a text-to-motion model generates a kinematic trajectory that is subsequently tracked by a separate controller. We instead generate the latent commands of a frozen whole-body controller directly from text, removing runtime retargeting and motion tracking from the primary generation path. We also incorporate differentiable kinematics into diffusion sampling to impose hand and foot target constraints directly in the controller latent space, without retraining the generator or controller. Experiments demonstrate strong language alignment, high completion rates, and effective spatial-constraint preservation after physical execution. We further demonstrate diverse behaviors in simulation and on the real Unitree G1. Beyond generation and execution, we characterize the semantic and physical structure of the frozen controller latent representation and examine whether this structure is preserved in latents generated directly from language. Our analysis shows that the 64-dimensional latent representation encodes joint configuration, locomotion direction and intensity, and gait phase in a distributed manner, with much of this structure preserved after finite scalar quantization and also present in the generated latents. We further show that text-conditioned motion characteristics can be composed in latent space to generate motions combining multiple attributes.

1 INTRODUCTION

Language-based humanoid control is commonly implemented as a cascaded pipeline. A text-to-motion model first generates a kinematic motion sequence, which is then reproduced by a physics-based motion tracker. The generated trajectory must be converted into the reference representation used by the tracker, and errors introduced during this handoff can alter the semantics of the motion or cause geometric constraints satisfied at generation time to be violated after physical execution. Therefore, high kinematic motion quality alone does not guarantee that the intended behavior will be preserved under contact dynamics and closed-loop control.

A pretrained whole-body controller already provides a compact motion interface that is directly coupled to physical execution. We ask whether this interface can be generated directly from language, rather than first generating an explicit kinematic trajectory. To this end, we train a text-conditioned diffusion model to predict its continuous pre-quantization latent representation while keeping its quantizer and proprioception-conditioned action decoder frozen. This establishes a direct path from language to executable controller commands without first generating a human or robot trajectory or performing runtime retargeting. The controller continues to operate in closed loop using the robot state, while only the high-level latent motion plan is generated from language.

This controller-native interface also supports motion control and composition within the same latent space. We incorporate differentiable kinematics into diffusion sampling so that spatial constraints, such as hand and foot targets, directly guide the latent sequence being generated. We further perform semantic motion composition by combining denoising directions from different language conditions during sampling. Both operations act directly on the latent sequence ultimately executed by the controller, without generating or modifying a separate kinematic trajectory and without retraining either the generator or the controller. In this way, language-conditioned generation, spatial guidance, and semantic composition are unified within a single controller-native latent space.

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.

At the same time, directly generating controller latents replaces explicitly interpretable joint trajectories with compact 64-dimensional commands. We therefore compare latents obtained from the frozen controller with those generated from language to investigate the motion structure represented by this interface. Our analysis shows that semantic and kinematic properties, including body laterality, locomotion characteristics, and temporal motion information, are distributed across the controller latent representation and remain substantially preserved after quantization and language-conditioned generation.

Our key contributions are summarized as follows:

- **Controller-native language generation.** We directly generate the continuous latent commands of a pretrained whole-body controller from language, removing explicit kinematic motion generation, runtime retargeting, and a separate motion-tracking stage from the primary generation path. We compare language alignment and physical execution against kinematic text-to-motion pipelines under shared prompts and controller conditions, and demonstrate diverse motions on the real Unitree G1.
- **Controllable diffusion in the execution space.** We incorporate differentiable spatial guidance to impose hand and foot targets directly during controller-latent diffusion sampling, and combine denoising directions from multiple language conditions for semantic motion composition. Both operate directly on the latent sequence executed by the frozen controller, and their effects are evaluated after closed-loop physical execution.
- **Analysis of recorded and generated controller latents.** We compare latents obtained from the frozen controller with text-generated latents to characterize the semantic and physical structure of the controller interface and examine how this structure is preserved through language-conditioned generation and quantization.

2 Related Work

Language-conditioned motion generation and control. Text-conditioned motion generation has been extensively studied in the kinematic human-motion domain. Diffusion-based methods such as MDM (Tevet et al., 2023) and MotionDiffuse (Zhang et al., 2024) generate pose sequences directly, while T2M-GPT (Zhang et al., 2023) models motion as discrete tokens. More recent systems such as Kimodo (Rempe et al., 2026) scale generation to large motion corpora and support variable-duration generation, multi-prompt sequencing, kinematic constraints, and direct G1 motion generation. Related work has also explored compositional and spatial control of generated motion. SINC (Athanasiou et al., 2023) and STMC (Petrovich et al., 2024) compose body-part or spatiotemporal motion conditions, while EnergyMoGen (Zhang et al., 2025) composes semantic and latent energies for controllable motion generation. Trajectory- and end-effector-based objectives enable spatial control during kinematic motion generation (Karunratanakul et al., 2023; Xie et al., 2024), and composable diffusion provides a general interpretation of combining conditional score directions (Liu et al., 2022). These approaches primarily operate on kinematic motion representations. In contrast, we apply both spatial guidance and semantic score composition directly while generating the latent commands executed by a frozen physics-based controller, and evaluate their effects after closed-loop physical execution.

Physics-based humanoid control and controller representations. Large-scale motion imitation has enabled physics-based humanoid controllers capable of reproducing diverse whole-body behaviors. AMP (Peng et al., 2021) introduces adversarial motion priors for naturalistic control, while PHC (Luo et al., 2023) scales robust motion imitation to large motion collections with fail-state recovery. ASE (Peng et al., 2022), CALM (Tessler et al., 2023), and PULSE (Luo et al., 2024) develop reusable latent skill or motion representations for downstream control. SONIC (Luo et al., 2026) maps motion references into a shared finite scalar quantization (FSQ) (Mentzer et al., 2024) latent space and executes the resulting commands using a proprioception-conditioned action decoder. We keep this controller interface entirely frozen and train only a language-conditioned generator to predict its existing command latents.

Recent work has also connected language to latent representations for physical humanoid control. RoboGhost (Li et al., 2026) maps language to motion latents that condition a learned policy for

executable action generation. In contrast, our method introduces no new control policy for language conditioning: the pretrained whole-body controller, its quantizer, and its action decoder remain unchanged, and only the text-conditioned generator is trained. This isolates language-to-command generation from control learning and allows language generation, spatial guidance, and semantic composition to operate directly through the controller's existing latent interface.

3 PRELIMINARIES

Frozen whole-body controller. We use the released SONIC controller for the Unitree G1 (Luo et al., 2026) without modifying the downstream control policy. Let E^r_ϕ denote SONIC's robot-motion encoder. Given a short future robot-motion reference g^r_t, the encoder produces a continuous 64-dimensional pre-quantization latent $\tilde{z}_t$, which is mapped through the finite scalar quantizer Q and consumed by the proprioception-conditioned action decoder D_ψ:

$$\tilde{z}_t = E^r_\phi(g^r_t), \qquad z_t = Q(\tilde{z}_t), \qquad a_t = D_\psi(z_t, s^p_t), \tag{1}$$

where s^p_t denotes the robot's proprioceptive state history. The action decoder operates in closed loop, using the current robot state to produce joint actions.

Our method replaces only the mapping from an externally specified motion reference to the controller latent. Specifically, we generate the continuous pre-quantization latent $\tilde{z}_t$ directly from language, while keeping the quantizer Q and action decoder D_ψ frozen. Thus, the language-conditioned generator specifies the high-level motion command, whereas physical stabilization remains handled by the pretrained controller.

SONIC additionally provides a frozen kinematic decoder K_ϕ that reconstructs robot motion from the controller latent. We use this decoder for latent analysis and differentiable spatial guidance, while leaving the controller and all SONIC components unchanged.

4 METHOD

We learn a text-conditioned generator of pre-quantization commands for the frozen controller introduced in Section 3. To construct the training data, we start from the captioned BONES-SEED motion corpus (BONES Studio, 2026), convert its motions to G1 references, and replay them through the frozen controller, retaining only successfully executed caption–latent sequences; full collection and filtering details are provided in Appendix B. Only the generator is trained, with no controller adaptation or distillation. Given a caption embedding c and requested length T, the generator produces a latent plan $\hat{\tilde{z}}_{1:T} = G_\theta(c, T)$, which is quantized and executed by the frozen controller as $\hat{a}_t = D_\psi\left(Q(\hat{\tilde{z}}_t), s^p_t\right)$ (Figure 1). This interface connects language to control without synthesizing an intermediate human or robot reference trajectory.

4.1 TEXT-CONDITIONED LATENT SEQUENCE GENERATION

The generator models a complete latent trajectory jointly, allowing attention across frames to capture action order and motion phase. We standardize the recorded pre-quantization channels as $x^0 = (\tilde{z}^* - \mu_z)/\sigma_z$ and represent variable duration with a valid-frame mask m. A frozen LLM2Vec encoder (BehnamGhader et al., 2024) supplies the caption embedding c. The denoiser is a Transformer encoder that receives the projected text condition, a diffusion-step token, and the noisy latent frames. It uses noncausal self-attention, allowing each latent frame to attend to both earlier and later frames so that the entire motion sequence can be modeled jointly rather than generated autoregressively. Each latent frame is also concatenated with a binary observation mask indicating whether that frame is given as known context or should be generated. This allows the same denoiser to generate a complete sequence from text alone or to continue from a previously generated latent prefix.

We train with the x_0 parameterization of denoising diffusion (Ho et al., 2020): a sampled diffusion index k and Gaussian noise produce a corrupted sequence x^k, from which f_θ predicts the clean normalized sequence. On a fraction of training examples, the denoiser input $x^{k,\text{in}}_b$ keeps the first few frames clean as a known prefix. These frames represent motion that is already available and serve as

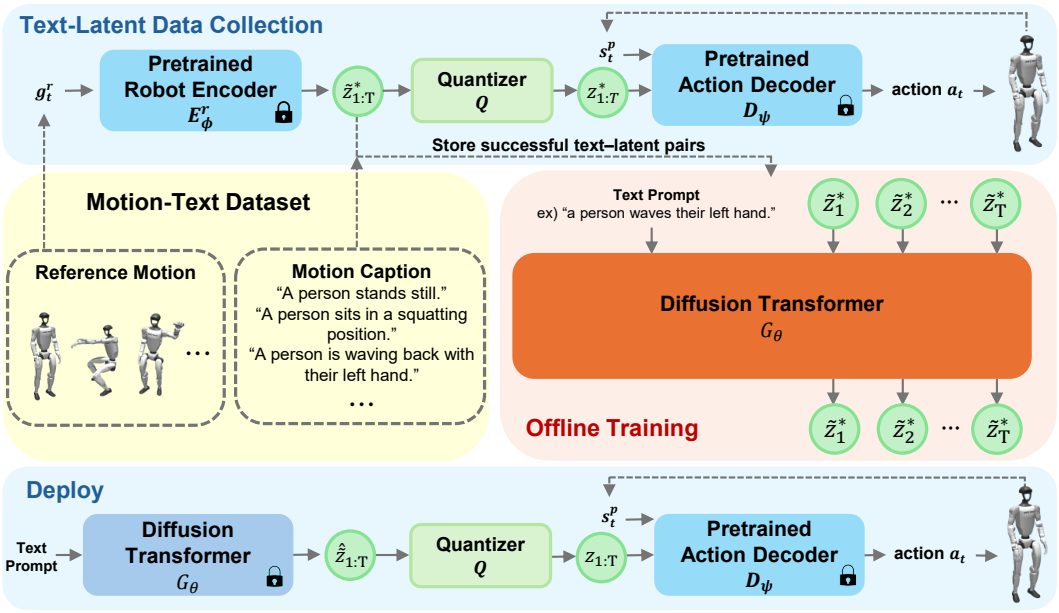


Figure 1: Overview of our text-to-controller-latent humanoid control framework. We first collect successful rollouts from the frozen pretrained robot motion tracker and store paired text–latent trajectories. A text-conditioned diffusion Transformer is then trained to generate pre-quantization latent sequences directly from language. At deployment, the generated latent sequence is projected through the fixed quantizer and executed by the pretrained state-conditioned action decoder.

temporal context for predicting the continuation. We refer to them as observed frames. Let $o_{bt} = 1$ indicate that frame t of sample b belongs to this observed prefix, and let $m_{bt} = 1$ indicate a valid, non-padded frame. We then define $u_{bt} = m_{bt}(1 - o_{bt})$, so that u selects only valid frames that are not already provided to the model and therefore must be predicted. The training objective is

$$\mathcal{L}(\theta) = \mathbb{E}_{k,\epsilon}\left[\frac{\sum_{b,t} u_{bt} \left\| \left[f_\theta(x_b^{k,\text{in}}, k_b, c_b; m_b, o_b) \right]_t - x_{bt}^0 \right\|_2^2}{64 \sum_{b,t} u_{bt}}\right]. \tag{2}$$

Accordingly, the reconstruction loss is computed only on the frames that the model must generate. Padded frames and observed-prefix frames are excluded from the loss, and padded frames are also masked from attention because they do not correspond to actual motion frames. When no observed prefix is provided, $o \equiv 0$ and the loss mask reduces to the validity mask m. This observed-prefix augmentation enables temporal continuation by conditioning each new latent window on the preceding latent prefix at inference time.

At inference, the trained denoiser generates a latent sequence of the requested duration, specified by the valid-frame mask. We use DDIM (Song et al., 2021) for sampling and classifier-free guidance (Ho & Salimans, 2021), enabled by text dropout during training, to strengthen conditioning on the input caption. The resulting clean normalized latent sequence is mapped back to the controller's pre-quantization latent space by undoing the channel-wise normalization and is then passed through the frozen FSQ before execution. Further architecture and optimization details are given in Appendix B.

4.2 Controllable Generation in the Controller Latent

Differentiable spatial guidance. We impose hand or foot targets without retraining the generator or controller. For a clean estimate $\hat{x}_t^0$, let $\hat{\tilde{z}}_t = \mu_z + \sigma_z \odot \hat{x}_t^0$. We project this latent through FSQ before the frozen kinematic decoder K_ϕ, matching the representation consumed by the controller, and use a straight-through gradient through the hard quantizer:

$$z_t^{\text{ste}} = \hat{\tilde{z}}_t + \text{sg}\Big(Q(\hat{\tilde{z}}_t) - \hat{\tilde{z}}_t\Big), \tag{3}$$

where sg denotes stop-gradient. Differentiable forward kinematics applied after K_ϕ gives the pelvis-frame position $p_{j,t}$ of constrained link j. For target $p^*_{j,t}$, the sampling objective is

$$\mathcal{L}_{\text{spatial}} = \sum_{(j,t)\in\mathcal{C}} w_{j,t} \left\| p_{j,t} - p^*_{j,t} \right\|_2^2, \tag{4}$$

where $\mathcal{C}$ denotes the constrained links and frames and $w_{j,t}$ their temporal weights. During an initial fraction of DDIM sampling, we move the clean estimate along the negative normalized gradient, scale the step by the latent norm, and taper it near the target. Score composition, when used, is applied before this geometric update. The denoiser, FSQ, kinematic decoder, and action decoder remain frozen, and constraint accuracy is evaluated after closed-loop execution.

Semantic composition during diffusion sampling. Because the denoiser predicts controller latents, semantic conditions can also be combined inside the same sampling process. For captions $c_1, \ldots, c_M$, all conditions are evaluated from the same noisy sample at DDIM step k:

$$d_i^k = f_\theta(x^k, k, c_i; m, o) - f_\theta(x^k, k, \varnothing; m, o), \tag{5}$$

$$\hat{x}^0_{\text{score}} = \hat{x}^0_\varnothing + s \sum_{i=1}^{M} w_i d_i^k, \qquad \hat{x}^0_\varnothing = f_\theta(x^k, k, \varnothing; m, o). \tag{6}$$

This is the x_0-prediction form of composable diffusion (Liu et al., 2022). Unlike learned energy composition (Zhang et al., 2025), it combines the caption-conditional directions already provided by our generator and produces one controller-latent sequence throughout denoising. The directions depend on the current noisy trajectory and diffusion step; Equation 6 therefore does not assume fixed semantic vectors, a disentangled latent space, or independent control of each attribute. It provides conditional semantic composition during generation, with the strength and interaction of attributes determined by the joint denoising trajectory.

Optional executable constraint projection. For higher-precision spatial constraints, we optionally refine a completed sample by decoding it to joint targets $q^0_{1:T} = K_\phi(Q(\tilde{z}_{1:T}))$, solving inverse kinematics only over the pelvis-to-end-effector chain, and mapping the edited trajectory back to the controller interface:

$$q^*_{1:T} = \arg\min_{q\in\mathcal{Q}} \mathcal{L}_{\text{spatial}}(q) + \lambda_p \|q - q^0\|_2^2 + \lambda_s \|\Delta^2(q - q^0)\|_2^2, \qquad \tilde{z}^*_t = E^r_\phi(R(q^*_{1:T})_t)\,, \tag{7}$$

where $\mathcal{Q}$ denotes the robot joint limits and R reconstructs the future-reference window, including velocities, required by the frozen robot encoder. The re-encoded command is quantized and executed by the same frozen action decoder. This decode–constrain–re-encode path is a separate optional final projection; the sampling-time spatial guidance and semantic composition above both generate controller latents directly. No generator or controller component is updated.

5 Experiments

5.1 Evaluation setup

We evaluate 1,000 held-out clip-length prompts from our captioned G1 corpus, excluding their clips from TMR (Petrovich et al., 2023) evaluator training. All methods receive identical prompts and durations and use the same frozen SONIC controller for physical execution. MDM (Tevet et al., 2023), MLD (Chen et al., 2023), T2M-GPT (Zhang et al., 2023), and MoMask (Guo et al., 2024) use our captioned G1 corpus and text embeddings; Kimodo-G1 (Rempe et al., 2026) uses its released checkpoint trained on the same source data. Each baseline first generates G1 kinematics, which SONIC encodes and tracks, whereas our model directly generates SONIC commands. Full configurations are in Appendix D.

We measure text alignment with TMR R@3, FID, and Diversity. Physical metrics are FID_{exec} against ground-truth executions, completion without a fall during the command, and fall rate including the 2 s standing hand-over. For kinematic baselines, ΔR@3 is the alignment change after tracking; Diversity is interpreted relative to the reference value. Additional metric definitions and rollout details are in Appendix D.3.

Table 1: Text–motion alignment and physical executability.

	Generated motion			Executed motion					
Method	R@3 ↑	FID ↓	Diversity	R@3 ↑	ΔR@3	FID_{exec} ↓	Diversity	Compl. % ↑	Fall % ↓
Ground truth	0.952	0.00	21.06	0.881	–	0.0	19.37	97.7	2.5
MDM (Tevet et al., 2023)	0.953	22.39	20.34	0.874	−0.079	23.9	19.25	91.3	10.1
MLD (Chen et al., 2023)	0.965	18.17	21.34	0.887	−0.078	13.1	19.70	93.9	7.0
T2M-GPT (Zhang et al., 2023)	0.921	6.80	20.48	0.827	−0.094	8.7	19.09	95.9	5.3
MoMask (Guo et al., 2024)	0.959	8.23	20.87	**0.892**	−0.067	7.3	19.50	96.8	4.5
Kimodo-G1 (Rempe et al., 2026)	**0.968**	**4.73**	21.15	0.880	−0.088	8.7	19.74	95.5	6.1
Ours (text → latent)		–		0.879	–	**6.9**	19.58	**97.0**	**3.4**

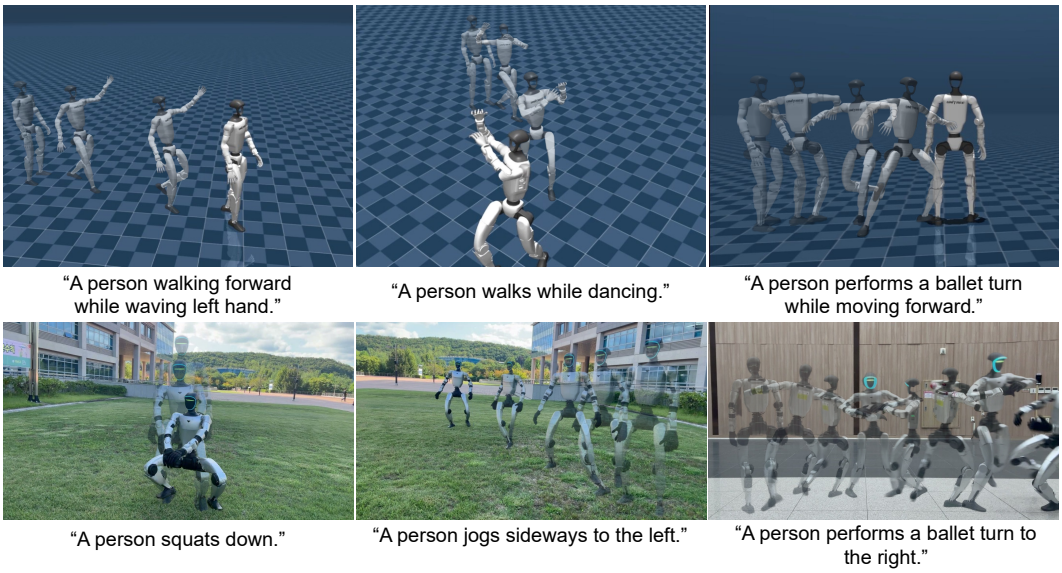
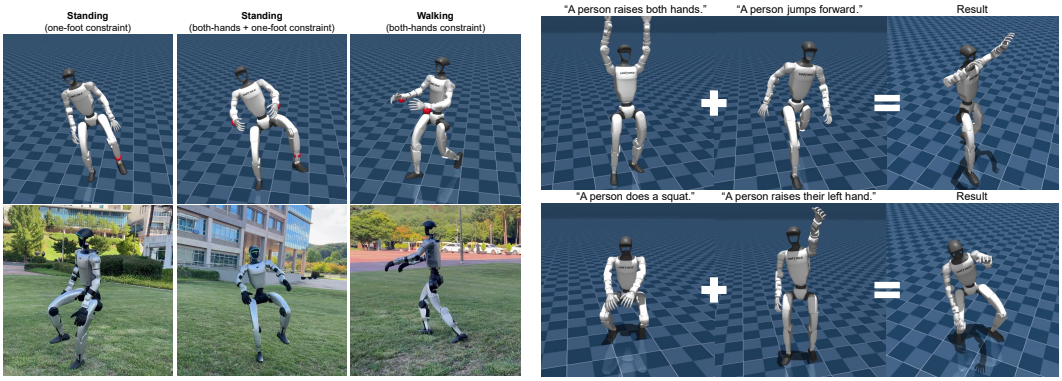


Figure 2: Diverse language-conditioned motions generated directly as controller latents and executed in MuJoCo and on the real Unitree G1.

5.2 Language-conditioned motion execution

Text–motion alignment. We first evaluate how well the generated motions preserve the semantics of the input text before considering physical executability. Among the kinematic generators, Kimodo-G1 achieves the strongest overall text–motion alignment, with an R@3 of 0.968 and an FID of 4.73. MLD achieves similarly strong text alignment with an R@3 of 0.965, but its FID is substantially higher at 18.17. Unlike these baselines, our method does not generate an intermediate kinematic motion trajectory that can be evaluated before execution. It instead generates controller latents directly, so its text–motion alignment is measured only on the motion executed by the frozen controller. We therefore leave the generated-motion columns undefined for our method and compare it with the baselines after controller execution.

Motion executability. Passing kinematic generations through the frozen motion tracker generally reduces their text–motion alignment. R@3 decreases by 0.079 for MDM, 0.078 for MLD, 0.094 for T2M-GPT, 0.067 for MoMask, and 0.088 for Kimodo-G1 relative to their generated kinematic motions. Among the kinematic generation baselines, MoMask retains the highest executed R@3 of 0.892, followed by MLD at 0.887 and Kimodo-G1 at 0.880. Our method does not introduce a separate tracking stage because its output is already the latent command consumed by the controller. It achieves an executed R@3 of 0.879, close to the ground-truth execution score of 0.881 and comparable to the strongest tracked kinematic pipelines. Among all compared text-to-motion generators, our method achieves the lowest execution FID of 6.9, the highest completion rate of 97.0%, and the lowest fall rate of 3.4%. Thus, direct generation at the controller's latent interface preserves competitive text alignment after physical execution while providing the strongest executability across the compared generation pipelines.



(a) **Spatially Constrained Motion Generation.** (b) **Compound Motion Generation in MuJoCo.**

Figure 3: **Controllable generation in the executable controller latent. (a)** Sampling-time spatial guidance, with red spheres marking the requested end-effector targets in MuJoCo and corresponding constrained motions shown on the real Unitree G1. **(b)** Score composition generates compound motions in MuJoCo by combining raising both hands with jumping forward (top) and squatting with raising the left hand (bottom). Appendix E.5 quantifies attribute retention and interference.

5.3 Controllable generation in the controller latent

Spatial constraints guidance. We evaluate 500 prompt–constraint pairs: 250 general motion prompts, each paired with one sampled hand constraint (left, right, or both), and 250 separately sampled stationary prompts, each paired with one foot constraint (left or right). Targets are selected from six reachable pelvis-frame locations and held fixed throughout the motion. Kimodo-G1 uses its released kinematic constraint interface, whereas our guidance differentiates the spatial objective through FSQ and the frozen kinematic decoder during latent sampling. Both methods receive identical prompts and targets.

We additionally evaluate the optional projection in Equation 7. For our outputs, the frozen kinematic head decodes either a free or guided latent sequence, chain IK moves the requested wrist or ankle to its target while penalizing large and nonsmooth joint corrections, and the frozen robot encoder converts the edited motion back to controller latents before execution. The same IK and re-encoding procedure is applied to unconstrained Kimodo-G1 motion, so the comparison separates its effect from the generator that supplies the initial motion. Appendix C.2 gives the full protocol and optimization settings.

Table 2: Spatial-constraint accuracy after physics execution. Error is mean/median target distance, and success denotes reaching within 10 cm without falling.

Method	All			Hand			Foot		
	Error ↓ (cm)	Succ. 10 ↑ (%)	Fall ↓ (%)	Error ↓ (cm)	Succ. 10 ↑ (%)	Fall ↓ (%)	Error ↓ (cm)	Succ. 10 ↑ (%)	Fall ↓ (%)
Kimodo-G1 (constraint) (Rempe et al., 2026)	13.1 / 12.0	31	2.8	10.4 / 9.8	46	4.4	15.9 / 15.7	16	1.2
Kimodo-G1 (free + IK) (Rempe et al., 2026)	8.4 / 8.0	65	3.8	7.1 / 5.8	73	4.4	9.6 / 9.4	56	3.2
Kimodo-G1 (constraint + IK) (Rempe et al., 2026)	8.6 / 8.2	61	2.6	6.6 / 5.3	81	4.0	10.6 / 10.7	41	1.2
Ours (guidance)	10.9 / 9.4	53	3.2	12.0 / 10.7	42	3.6	9.8 / **8.6**	63	2.8
Ours (free + IK)	7.9 / 7.7	69	1.6	6.6 / 5.5	78	2.4	9.3 / 9.2	61	**0.8**
Ours (guidance + IK)	**7.2 / 7.1**	**75**	**1.2**	**5.6 / 4.8**	**86**	**1.6**	**8.9** / 8.7	**65**	**0.8**

Table 2 shows that, compared with Kimodo-G1's native constraint interface, our sampling-time guidance reduces the overall mean/median executed error from 13.1/12.0 cm to 10.9/9.4 cm and increases success from 31% to 53%. Applying the optional projection further improves geometric accuracy for both Kimodo-G1 and our method. With the same projection, Kimodo-G1 reaches 8.4/8.0 cm error and 65% success, while our guidance followed by IK reaches 7.2/7.1 cm and 75% success. Within our method, compared with free generation followed by IK, guidance reduces

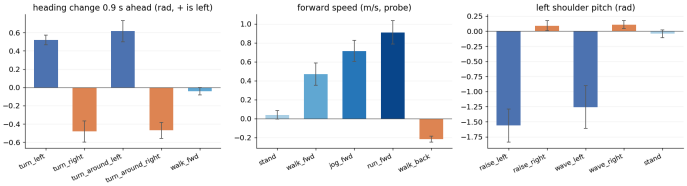
the error from 7.9/7.7 cm to 7.2/7.1 cm and increases success from 69% to 75%, indicating that sampling-time guidance provides a useful initialization for the subsequent projection.

To quantify where geometric accuracy is lost across the controller interface, we additionally track the constrained target error through each stage of our guidance+IK pipeline. The error is only 0.3 cm immediately after IK, increases to 3.3 cm after re-encoding the edited motion into controller latents, and reaches 7.2 cm after closed-loop physical execution. This stage-wise analysis directly shows the generation-to-execution gap introduced by re-encoding and closed-loop control.

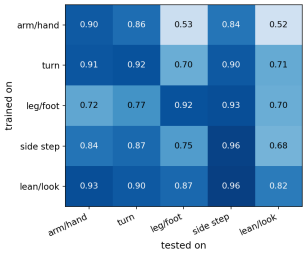
Semantic motion composition. The same controller-native sampling interface also supports semantic composition. At every DDIM step, Equation 6 combines multiple text-conditioned denoising directions into one latent trajectory, which is then consumed directly by the frozen controller. Figure 3(b) shows two examples after physics execution: jumping forward while raising both hands and squatting while raising the left hand. Both attributes are visibly expressed in each generated motion without first composing decoded joint trajectories.

Composition is conditional and graded rather than disentangled. Appendix E.5 quantifies the boundary: compatible actions can be expressed together but may be weaker than their single-caption generations, and changing the weights trades their relative strength. Opposing requests such as left versus right or forward versus backward generally do not cancel into an average pose; the diffusion trajectory instead converges to one motion basin for a given seed. We therefore claim semantic composition during controller-latent generation, without claiming arbitrary latent arithmetic or independent control of semantic attributes. Generated latents also reflect contrasting text-conditioned motion attributes, including direction, locomotion intensity, and laterality (Figure 4a).

5.4 Analysis of Generated Controller Latents



(a) Same-noise generations under contrasting text conditions for turning direction, locomotion intensity, backward motion, and arm laterality.



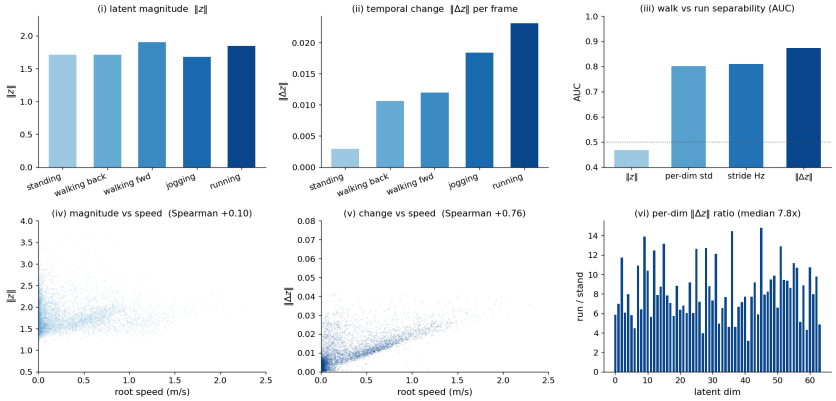
(b) Cross-action transfer accuracy of left/right classifiers trained and evaluated across action families.

Figure 4: **Language-conditioned contrasts in generated controller latents.**

Distributed structure of generated controller latents. The composition results show that multiple semantic conditions can be reflected within a single executable latent sequence. We therefore analyze how the generated controller latents relate to motion semantics. We find that all 64 latent dimensions are active, and that semantic changes are distributed across multiple channels rather than localized to a small set of coordinates (Appendix Figure 8a). Under spatial constraints, the magnitude of the latent change strongly correlates with the achieved end-effector correction (Spearman $\rho = 0.89$). Moreover, a linear readout from the latent edit predicts the requested 3-D target position with $R^2 = 0.54$–0.66 (Appendix Figure 13), indicating that substantial geometric information is encoded in the latent change despite being distributed across multiple dimensions.

Temporal dynamics versus latent magnitude. Motion characteristics are expressed more clearly through temporal variation than through the absolute magnitude of the latent. Walking and running occupy similar latent-magnitude ranges, making them difficult to distinguish from latent magnitude alone. In contrast, the frame-to-frame latent change $|\Delta z|$ and oscillation frequency increase with motion speed (Figures 5a and 5b). This suggests that locomotion intensity and gait rhythm are represented more strongly through latent temporal dynamics than through static latent magnitude.

(a) Walking and running have similar latent magnitudes, while temporal variation separates them more strongly.



(b) Same value range, different rhythm: three latent channels of one walking clip and one running clip.

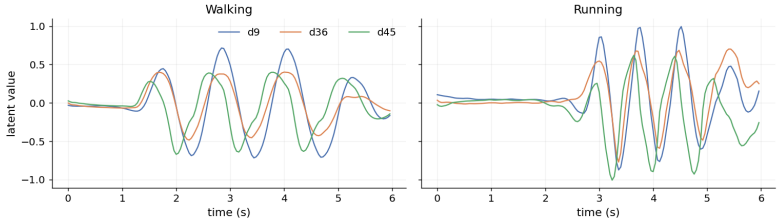


Figure 5: **Locomotion intensity is carried mainly by temporal dynamics.** Top: walking and running have similar latent magnitudes, whereas per-frame change, class AUC, and speed correlation distinguish them. Bottom: three example latent channels occupy similar ranges in walking and running but oscillate at different rates.

Action-dependent directional structure. We further find that directional information such as left–right laterality is partially shared across different action families, as shown by the shared laterality direction (Appendix Figure 12a) and cross-action transfer of left/right classification (Figure 4b). However, laterality remains more strongly expressed within the same action family. Together with the distributed and temporal analyses above, these results suggest that motion semantics in the controller latent are represented through a combination of distributed features, temporal dynamics, and action-dependent components rather than through simple fixed directions or latent magnitude alone.

6 Conclusions and Limitations

We propose a method that directly generates the latent commands of a frozen whole-body controller from language, avoiding explicit kinematic motion generation, runtime retargeting, and a separate motion-tracking stage. Our method maintains strong text–motion alignment after physical execution, with R@3 comparable to kinematic pipelines while achieving low execution FID, high completion, and a low fall rate. The same controller-native latent interface also supports end-effector spatial constraints and semantic motion composition.

The method is limited by the motion space that the frozen controller can reliably execute. The latent representation is structured but not disentangled, and no individual coordinate or global direction has a fixed semantic meaning. Semantic composition does not guarantee independent attribute control: compatible attributes can attenuate one another, while conflicting conditions can converge to a single motion mode. Our quantitative evaluation is in-distribution with respect to the generator training corpus, and the real-robot experiments are qualitative. Finally, the latent plan is generated before execution: the action decoder operates in closed loop, but the generator does not perform online replanning.

AI USE STATEMENT

In this work, we used generative AI tools to assist with software implementation, code editing and debugging, and drafting and editing portions of the manuscript. We reviewed all AI-assisted work. AI-assisted code was verified through direct execution and experimental testing, and AI-assisted text was reviewed and revised by the authors for technical accuracy and consistency with the reported results. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The main paper describes the model formulation, controller interface, training objective, spatial-guidance method, and semantic-composition rule. Additional implementation details, including rollout collection, preprocessing, model architecture, optimization, and sampling, are provided in Appendix B; spatial-guidance details and constraint evaluation are provided in Appendix C; baseline configurations and evaluation protocols are provided in Appendix D; and composition sweeps and latent-analysis procedures are provided in Appendix E. These details are intended to support reproduction of the reported experiments and results.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol. SINC: Spatial composition of 3D human motions for simultaneous action generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9984–9995, 2023.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2Vec: Large language models are secretly powerful text encoders. In *Conference on Language Modeling (COLM)*, 2024.

BONES Studio. BONES-SEED: Skeletal everyday embodiment dataset. `https://bones.studio/datasets/seed`, 2026. Human motion dataset with natural-language annotations.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Katie E. Everett, Lechao Xiao, Mitchell Wortsman, Alexander A. Alemi, Roman Novak, Peter J. Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, and Jeffrey Pennington. Scaling exponents across parameterizations and optimizers. In *International Conference on Machine Learning (ICML)*, pp. 12666–12700, 2024.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. MoMask: Generative masked modeling of 3D human motions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2151–2162, 2023.

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

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision (ECCV)*, pp. 423–439, 2022.

Zhengyi Luo, Jinkun Cao, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10895–10904, 2023.

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 (ICLR)*, 2024.

Zhengyi Luo et al. SONIC: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117):eaed4592, 2026.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations (ICLR)*, 2024.

Mayank Mittal et al. Isaac Lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40 (4):1–20, 2021.

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions on Graphics*, 41(4):1–17, 2022.

Mathis Petrovich, Michael J. Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Mathis Petrovich, Or Litany, Umar Iqbal, Michael J. Black, Gül Varol, Xue Bin Peng, and Davis Rempe. Multi-track timeline control for text-driven 3D human motion generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1911–1921, 2024.

Davis Rempe, Mathis Petrovich, et al. Kimodo: Scaling controllable human motion generation. Technical report, NVIDIA, 2026. arXiv:2603.15546.

Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H. Bermano. Human motion diffusion as a generative prior. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. CALM: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH Conference Proceedings*, 2023. doi: 10.1145/3588432.3591541.

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

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. OmniControl: Control any joint at any time for human motion generation. In *International Conference on Learning Representations (ICLR)*, 2024.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2M-GPT: Generating human motion from textual descriptions with discrete representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14730–14740, 2023.

Jianrong Zhang, Hehe Fan, and Yi Yang. EnergyMoGen: Compositional human motion generation with energy-based diffusion model in latent space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17592–17602, 2025.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. MotionDiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4115–4128, 2024. doi: 10.1109/TPAMI.2024.3355414.

APPENDICES

A Frozen Controller Interface **13**

B Training Details **14**

B.1 Rollout collection and preprocessing 14

B.2 Architecture and training 14

B.3 Diffusion sampling and temporal continuation 16

C Spatial Guidance **17**

C.1 Differentiable guidance through FSQ 17

C.2 Constraint evaluation protocol 17

D Experimental Details **18**

D.1 Baseline setup 18

D.2 Kinematic-to-SONIC execution 19

D.3 Evaluation protocol and metrics 19

D.4 Additional qualitative results 20

E Language–Latent Analysis and Composition **21**

E.1 Analysis data and protocol 21

E.2 Distributed and temporal latent structure 21

E.3 Spatial guidance as a controlled latent intervention 25

E.4 Text contrasts and denoising dynamics 26

E.5 Composition capability and limitations 27

A FROZEN CONTROLLER INTERFACE

We use the released G1 robot encoder, FSQ quantizer, action decoder, and auxiliary kinematic decoder from SONIC (Luo et al., 2026). The Unitree G1 has 29 actuated joints. At each 50 Hz control step, the robot encoder receives ten future reference samples spaced by 0.1 s. Each sample contains 29 joint positions, 29 joint velocities, and a six-dimensional root-orientation representation, producing a 640-dimensional reference input. The encoder outputs a 64-dimensional continuous pre-quantization latent $\tilde{z}_t$, which is mapped by the frozen FSQ to the quantized command z_t. Root translation and root linear velocity are not explicit inputs to the robot encoder.

The action decoder receives the 64-dimensional quantized command together with ten steps of proprioceptive history. Each state contains projected gravity, base angular velocity, joint positions, joint velocities, and the previous action, giving 930 state features. The decoder therefore receives 994 inputs and outputs 29 joint-position control actions. Isaac Lab (Mittal et al., 2025) uses a 0.005 s physics step with decimation four, corresponding to 200 Hz physics and 50 Hz control. The auxiliary kinematic decoder maps the quantized latent back to the 640-dimensional reference representation and is used by the proposed spatial guidance.

The diffusion model is trained on the continuous encoder output before FSQ. We denote this latent by $\tilde{z}_t$, the quantized command by $z_t = Q(\tilde{z}_t)$, and the normalized diffusion target by $x_t^0 = (\tilde{z}_t - \mu_z)/\sigma_z$. Normalization is reversed before the frozen quantizer is applied. All controller components remain frozen throughout generator training.

Table 3: Frozen G1 controller interface used throughout the experiments.

Component	Interface
Robot reference encoder	640 reference features $\rightarrow$ 64 continuous latents
FSQ	Two groups of 32 channels; 32 levels per channel
Action decoder	64 quantized latents + 930 state features $\rightarrow$ 29 actions
Kinematic decoder	64 quantized latents $\rightarrow$ 640 reference features
Controller / physics rate	50 Hz / 200 Hz
Reference / state spacing	0.1 s / 0.02 s

B Training Details

B.1 Rollout collection and preprocessing

We use approximately 288 hours of motion-caption data from BONES-SEED as the source dataset. Using the temporal annotations provided with the original motions, we construct training data at multiple temporal scales, including full-take motions, atomic segments corresponding to individual actions, and multi-event sequences containing consecutive actions. As a result, overlapping intervals from the same source motion can appear as separate training samples, yielding approximately 388 hours of motion-caption segments in total. All source motions are resampled to 50 Hz, and invalid, physically implausible, or discontinuous motions are filtered out before use.

The resulting motion segments are executed with the frozen SONIC controller in Isaac Lab, where closed-loop rollouts are collected using 4096 parallel environments. We retain only episodes that successfully complete the entire reference motion, while tracking failures and incomplete episodes interrupted by motion-library resampling are discarded. Consequently, the final training set contains only motions that SONIC can successfully execute to completion under closed-loop control. Each retained rollout stores the continuous encoder latent before FSQ, its motion identifier, and state information used only for replay and diagnostics; the sequence generator itself does not consume robot state. Generator training uses motion segments of 16–512 frames (0.32–10.24 s).

Captions are encoded with the frozen LLM2Vec model (BehnamGhader et al., 2024) based on Llama-3-8B-Instruct. The pooled 4096-dimensional text feature is cached, and when multiple captions are available for a motion, one caption is selected when the corresponding episode is sampled. Latent normalization statistics are estimated channel-wise from the recorded controller latents, with the standard deviation floored at 10^{-3}.

B.2 Architecture and training

The denoiser is a 16-layer Transformer encoder with width 1024, eight attention heads, feed-forward width 2048, GELU activations, and noncausal self-attention. It jointly processes the projected text condition, diffusion timestep, and noisy latent sequence. Each 64-dimensional noisy latent frame is concatenated with a 64-dimensional binary observation mask, producing 128 input channels per frame. The model predicts the complete clean 64-dimensional latent sequence.

Sequences are padded to at most 512 frames, and a validity mask excludes padding from attention and the reconstruction loss. Independently for each training example, prefix augmentation is activated with probability 0.1. When active, a prefix length ℓ is sampled uniformly from $\{1, \ldots, 25\}$ and the first $\min(\ell, T)$ valid frames are supplied as observed context for a sequence of length T. These observed frames are excluded from the loss, teaching the model to continue from previously generated latent context. Text conditioning is independently dropped with probability 0.1 for classifier-free guidance.

For batch element b, the normalized clean target is corrupted according to

$$x_b^k = \sqrt{\bar{\alpha}_{k_b}} x_b^0 + \sqrt{1 - \bar{\alpha}_{k_b}} \epsilon_b, \tag{8}$$

Algorithm 1 Training text-conditioned latent sequence diffusion

Require: Rollout latents $\tilde{z}^*$, captions c, latent statistics (μ_z, σ_z); $\theta_{\mathrm{EMA}} \leftarrow \theta$

for each optimization step **do**
  Sample a batch $\{(\tilde{z}^*_b, c_b, T_b)\}_{b=1}^{B}$ and form validity mask m
  $x^0 \leftarrow (\tilde{z}^* - \mu_z)/\sigma_z$
  Independently sample $k_b \sim \mathcal{U}\{1, \ldots, K\}$ and $\epsilon_b \sim \mathcal{N}(0, I)$
  $x^k_b \leftarrow \sqrt{\bar{\alpha}_{k_b}} x^0_b + \sqrt{1 - \bar{\alpha}_{k_b}} \epsilon_b$
  $o \leftarrow \mathbf{0}$
  Independently for each b, activate prefix observation with probability 0.1
  If activated, sample $\ell_b \sim \mathcal{U}\{1, \ldots, 25\}$ and mark the first $\min(\ell_b, T_b)$ valid frames in o_b
  $x^{k,\mathrm{in}} \leftarrow (1 - o) \odot x^k + o \odot x^0$
  Independently set $c_b \leftarrow \varnothing$ with probability 0.1
  $\hat{x}^0 \leftarrow f_\theta(x^{k,\mathrm{in}}, k, c;\ m, o)$
  $u \leftarrow m \odot (1 - o)$
  $\mathcal{L} \leftarrow \dfrac{\sum_{b,t,d} u_{bt} \left(\hat{x}^0_{btd} - x^0_{btd}\right)^2}{64 \sum_{b,t} u_{bt}}$
  Update θ using $\mathcal{L}$
  if the optimization step is a multiple of 10 **then**
    $\theta_{\mathrm{EMA}} \leftarrow 0.995\theta_{\mathrm{EMA}} + 0.005\theta$
  end if
end for

and the model predicts x^0 directly. With u_{bt} denoting valid, unobserved frames, the training loss is

$$\mathcal{L} = \frac{\sum_{b,t,d} u_{bt} \left(\hat{x}^0_{btd} - x^0_{btd}\right)^2}{64 \sum_{b,t} u_{bt}}. \tag{9}$$

This is the same masking convention as in Equation 2. No additional velocity, contact, kinematic, or control-reward losses are used.

The learning rate warms up linearly over 2,000 updates to 10^{-4} and is effectively constant thereafter: the cosine decay of the configured schedule has lowered it only to 9.9×10^{-5} at the reported checkpoint. EMA parameters are updated every ten optimizer steps with decay 0.995. All reported results use the EMA weights of the checkpoint at 30,000 updates. Training was performed on an NVIDIA RTX PRO 6000 Blackwell GPU.

Table 4: Sequence-generator training and sampling configuration used for the reported results.

Setting	Value
Transformer layers / width	16 / 1024
Attention heads / FF width	8 / 2048
Trainable parameters	140,890,176
Maximum sequence length	512 frames
Training diffusion steps	1000
Prediction target	x_0
Text dropout	0.1
Observed-prefix augmentation	$p = 0.1$; $\ell \sim \mathcal{U}\{1, \ldots, 25\}$
Optimizer	Adam-atan2 (Everett et al., 2024)
Learning rate	10^{-4}
Batch size	128
Gradient clipping	Global norm 1.0
DDIM sampling steps / η	100 / 0
Classifier-free guidance scale	2.0
Training hardware	1× NVIDIA RTX PRO 6000 Blackwell (96 GB)

B.3 DIFFUSION SAMPLING AND TEMPORAL CONTINUATION

At inference, classifier-free guidance combines conditional and unconditional clean-sequence predictions as $\hat{x}^0 = \hat{x}^0_\varnothing + s(\hat{x}^0_c - \hat{x}^0_\varnothing)$. Unless otherwise specified, we use 100 DDIM steps, $\eta = 0$, and guidance scale $s = 2.0$. After the final clean estimate, channel-wise normalization is inverted and the frozen FSQ produces the command sequence consumed by the controller.

Multi-condition semantic composition. For M caption conditions, the implementation embeds every caption independently with the same frozen LLM2Vec encoder. At each DDIM step it evaluates the unconditional estimate $\hat{x}^0_\varnothing$ and the conditional estimates $\hat{x}^0_i$ from the same noisy sequence, then forms

$$\hat{x}^0_{\text{comp}} = \hat{x}^0_\varnothing + s \sum_{i=1}^{M} w_i \left(\hat{x}^0_i - \hat{x}^0_\varnothing\right). \tag{10}$$

The result is used for a single DDIM update. Under the usual classifier-free-guidance score interpretation, this corresponds to the weighted product of experts

$$p_{\text{comp}}(x^k) \propto p_\varnothing(x^k)^{\,1-s\sum_i w_i} \prod_{i=1}^{M} p(x^k \mid c_i)^{\,sw_i}. \tag{11}$$

This is the multi-condition extension of composable diffusion (Liu et al., 2022), implemented here in the clean-latent prediction parameterization. Related motion methods compose pretrained diffusion priors or learned latent energies (Shafir et al., 2024; Zhang et al., 2025); our semantic-composition path uses the existing conditional directions of the trained controller-latent generator and requires no additional training. We do not assume that the resulting product of conditional densities produces a logical conjunction of motion attributes. Appendix E.5 separately measures every requested attribute and physical stability. If differentiable spatial guidance is active, its clean-estimate hook is applied after Equation 10, so the semantic and geometric conditions act on the same latent trajectory.

The interactive prompt parser uses

```
1.0 * a person jogs forward
+ 1.0 * a person raises their left arm @4
```

to set $w_1 = w_2 = 1$ and request four seconds for the entire intervention. At the 50 Hz control rate, this produces one 200-frame sample. Explicit weights are used exactly as written: the example stacks two full-strength guidance directions and is not a normalized 50–50 interpolation. When all weight prefixes are omitted, the parser assigns $w_i = 1/M$, preserving the aggregate guidance scale of a single caption. If one component has an explicit weight, every component must specify one. Components are separated by a plus sign surrounded by spaces; a trailing `@N` sets the duration of the complete sample.

Temporal continuation. For motions longer than a single generation window, we use overlapping-window generation. The first window is generated from text alone, while each later window receives the final frames of the preceding window as an observed clean prefix. The duplicated overlap is retained only once in the assembled trajectory. The default overlap used by the long-horizon evaluation path is 25 frames (0.5 s). The full latent sequence is assembled before physical execution, so this procedure does not use robot-state feedback or online replanning.

For sentence-wise commands, each sentence provides a separate text condition and may include an explicit duration, such as `wave the left hand @3. squat @4`. Later segments receive the final five frames of the preceding segment as observed context, generate the requested continuation, and discard the duplicated prefix during concatenation. Thus, both long single-caption motions and multi-action instructions use previously generated latent context to preserve temporal continuity.

C SPATIAL GUIDANCE

C.1 DIFFERENTIABLE GUIDANCE THROUGH FSQ

Spatial guidance is applied to the current denoiser estimate of the clean normalized latent. We first undo latent normalization,

$$\hat{\tilde{z}}_t = \mu_z + \sigma_z \odot \hat{x}_t^0. \tag{12}$$

Because the frozen kinematic decoder consumes quantized controller latents, guidance evaluates the spatial objective after FSQ. We retain gradients through the hard quantizer with the straight-through form

$$z_t^{\mathrm{ste}} = \hat{\tilde{z}}_t + \mathrm{sg}\Big(Q(\hat{\tilde{z}}_t) - \hat{\tilde{z}}_t\Big), \tag{13}$$

whose forward value equals the quantized latent while its backward Jacobian with respect to $\hat{\tilde{z}}_t$ is treated as identity.

The frozen kinematic decoder reconstructs the corresponding reference pose, and differentiable forward kinematics gives the pelvis-frame position $p_{j,t}$ of constrained link j. For target $p_{j,t}^*$, we use

$$\mathcal{L}_{\mathrm{spatial}} = \sum_{(j,t)\in\mathcal{C}} w_{j,t} \left\| p_{j,t} - p_{j,t}^* \right\|_2^2. \tag{14}$$

During the configured guidance portion of DDIM sampling, the current clean latent estimate is moved along the negative normalized gradient of this loss. The update magnitude is scaled by the latent magnitude and tapered as the predicted end-effector approaches the requested target. The denoiser, FSQ, kinematic decoder, and action decoder remain frozen.

C.2 CONSTRAINT EVALUATION PROTOCOL

We evaluate spatial guidance on 500 prompt–constraint pairs drawn from the evaluator-held-out pool: 250 hand prompts covering everyday actions (walking, standing, turning, greeting, reaching, gesturing, and sitting; 1–7 s, locomotion allowed) and 250 separately drawn stationary foot prompts (root travel at most 0.15 m). Each hand prompt carries one constraint on the left hand, right hand, or both hands, and each foot prompt carries one constraint on the left or right foot. For each constrained link, the target is sampled from six reachable pelvis-frame positions and held fixed throughout the generated motion. Both methods receive the same prompts and target locations.

Constraint error is the distance between the requested pelvis-frame target and the corresponding executed wrist or ankle position. We report mean and median error, the fraction of episodes satisfying all active constraints within 5 cm and 10 cm without falling, and the fall rate during execution. Hand and foot results are reported separately in addition to the pooled evaluation. The no-guidance variants use the same prompts and targets but omit the sampling-time spatial objective.

On the shared prompt–target pairs, the fall-rate difference between our guidance and constrained Kimodo-G1 is $+0.4$ percentage points with a paired 95% bootstrap confidence interval of $[-1.6, 2.4]$. Relative to our free samples, guidance changes the fall rate by -1.4 points, with an interval of $[-3.2, 0.4]$. The constraint objective therefore does not measurably destabilize execution in this evaluation.

Post-generation executable projection. The optional IK rows in Table 2 use the following fixed procedure. For our method, we first quantize the generated latent and decode the first future joint target at every control step with SONIC's frozen kinematic head. For Kimodo-G1, we start from its unconstrained generated joint trajectory. We then optimize only the joints on the pelvis-to-constrained-link chain over the full constraint interval, with all other joints fixed. The objective is the squared end-effector target error in Equation 7, a joint-correction penalty with $\lambda_p = 10^{-3}$, and a second-difference penalty with $\lambda_s = 10^{-2}$. We use 300 Adam steps with learning rate 0.1 and enforce the joint limits from the G1 model.

After IK, joint velocities are recomputed from the edited sequence and the ten-sample future-reference windows expected by the frozen SONIC robot encoder are reconstructed. The encoder

produces a new continuous 64-D pre-FSQ command at every control step; these commands are quantized and passed to the unchanged state-conditioned action decoder. For our latent input, reconstruction uses a fixed level root because the generated command does not explicitly contain horizontal root position; for Kimodo-G1, its generated root trajectory is retained. The target is pelvis-relative in both cases. All IK, re-encoding, and execution settings are identical across the free and guided variants, and neither the generator nor any SONIC component is updated.

D Experimental Details

D.1 Baseline setup

The kinematic baselines generate robot motion trajectories before execution by the frozen SONIC controller, whereas our method directly generates the controller latent sequence. The locally trained MDM (Tevet et al., 2023), MLD (Chen et al., 2023), T2M-GPT (Zhang et al., 2023), and MoMask (Guo et al., 2024) baselines share a 66-dimensional G1 motion representation containing 29 joint positions, 29 joint velocities, three-dimensional heading-frame root linear velocity, root yaw velocity, root height, and three-dimensional projected gravity. Absolute horizontal root position and absolute yaw are excluded. All four baselines use the same caption library and frozen pooled LLM2Vec text condition as our method.

MDM. Our G1 adaptation uses an eight-layer Transformer encoder with width 512 and predicts the clean normalized 66-dimensional motion sequence with 1000 diffusion steps. The reported checkpoint is trained for 60,000 updates with batch size 256.

MLD. The first stage trains a Transformer VAE that compresses a complete motion sequence to a 256-dimensional latent. A second-stage text-conditioned diffusion model is then trained in this learned motion-latent space while the VAE remains frozen. The VAE and diffusion stages are each trained for 60,000 updates.

T2M-GPT. The G1 adaptation uses a temporal VQ-VAE (van den Oord et al., 2017) with a 512-entry codebook followed by an 18-layer autoregressive Transformer over motion codes. The VQ-VAE downsamples time by a factor of eight, and both training stages use 300,000 updates with batch size 256.

MoMask. The G1 adaptation follows the released MoMask recipe: a residual VQ-VAE with six quantizers, each a 512-entry codebook, over the same convolutional encoder–decoder with eightfold temporal downsampling, followed by a masked Transformer over the base-layer codes and a residual Transformer over the remaining layers (eight layers and width 384 each). The RVQ-VAE is trained for 300,000 updates with batch size 256, and each Transformer for 180,000 updates with batch size 64. Generation uses ten masked-prediction steps with classifier-free guidance 4.0 for the base layer and guidance 5.0 for the residual layers.

Kimodo-G1. We use the released `kimodo-g1-seed` checkpoint (Rempe et al., 2026). It generates G1 kinematics at 30 Hz using its native motion representation and released pretraining. No additional human-to-G1 retargeting or inverse-kinematics stage is inserted before SONIC execution.

Table 5: Training settings for the locally trained kinematic baselines.

Model / stage	Learning rate	Updates
MDM	10^{-4}	60,000
MLD VAE	10^{-4}	60,000
MLD diffusion	10^{-4}	60,000
T2M-GPT VQ-VAE	2×10^{-4}	300,000
T2M-GPT Transformer	10^{-4}	300,000
MoMask RVQ-VAE	2×10^{-4}	300,000
MoMask masked Transformer	2×10^{-4}	180,000
MoMask residual Transformer	2×10^{-4}	180,000

D.2 KINEMATIC-TO-SONIC EXECUTION

For the 66-dimensional baselines, horizontal root position and yaw are reconstructed by integrating the predicted velocities, while the predicted root height and joint angles define the remaining kinematic state. Joint velocities are recomputed from the reconstructed trajectory. The resulting motion is converted to SONIC's reference representation, resampled to 50 Hz, and packed into the ten-frame future-reference format expected by the frozen robot encoder. Kimodo's 30 Hz output is similarly resampled before SONIC encoding.

The frozen SONIC encoder converts each reference window to the same 64-dimensional pre-FSQ latent interface used by our method. All baseline command streams then pass through the identical frozen FSQ and state-conditioned action decoder. Root translation does not enter the SONIC robot-reference input. This bridge is used only for kinematic baselines; our method directly produces the latent commands consumed by FSQ.

D.3 EVALUATION PROTOCOL AND METRICS

The main language-conditioned evaluation uses a fixed list of 1,000 prompts shared across methods. The TMR-based motion evaluator (Petrovich et al., 2023) is trained with the corresponding evaluation clips removed from its training set.

Text–motion scoring uses the same evaluator for generated kinematic trajectories and physically executed trajectories. R@3 measures whether the paired caption retrieves the motion within the top three candidates inside seeded groups of 32. FID compares Gaussian fits of motion embeddings to those of the corresponding reference clips, and Diversity measures pairwise spread in the same embedding space.

For physical execution, each environment first follows a standing command for 2 s before the generated sequence begins. A fall is detected after an initial grace period when root tilt exceeds 60° or root height drops below 0.3 m. Completion is the fraction of episodes that reach the end of the generated sequence without falling. Fall rate additionally includes falls during the subsequent 2 s standing hand-over. The same initialization, controller, and fall detector are used for all compared methods.

D.4 ADDITIONAL QUALITATIVE RESULTS

Figure 6 provides additional qualitative examples of language-conditioned motions in simulation and on the physical Unitree G1.

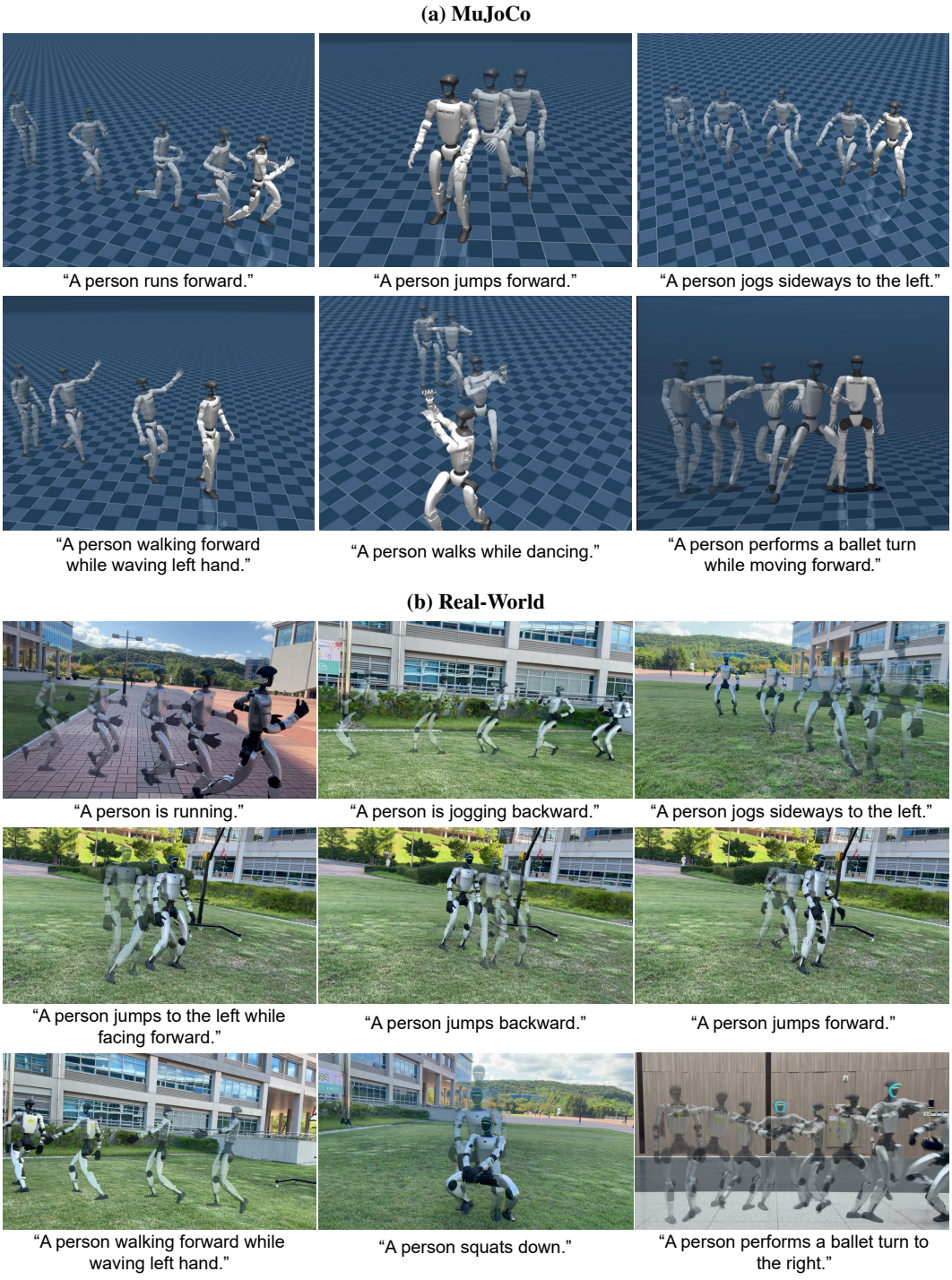


Figure 6: **Additional Qualitative Results.** **(a)** MuJoCo simulation. **(b)** Real-world execution on the Unitree G1.

E LANGUAGE–LATENT ANALYSIS AND COMPOSITION

E.1 ANALYSIS DATA AND PROTOCOL

We construct a post-hoc analysis set of 5,668 captioned clips and 513,089 frames sampled at 25 Hz, drawn from a subset of the training library. Each clip appears twice: once as the frozen SONIC robot encoder's latent for the recorded motion, and once as the latent our model generates from the same caption. The two versions cover the same clips with the same number of frames each, so the encoder side acts as a matched reference rather than a differently sized corpus. Every frame carries the 64-D continuous pre-FSQ latent, its FSQ-quantized value, and the 66-D heading-invariant kinematic representation used in the baseline study. The latter contains joint positions and velocities, heading-frame root velocity, yaw rate, root height, and projected gravity. The analysis does not update the controller, generator, text encoder, or quantizer.

Caption-derived groups are formed by deterministic lexical rules applied to all available captions for a clip. For paired contrasts, a clip matching both sides is excluded; gait groups similarly exclude conflicting gait words. These groups are used only for class summaries and contrast directions. The regression probes use measured frame-level kinematics from all clips and do not use the caption groups. Because lexical groups can include captions that describe additional actions, we treat them as reproducible weak labels rather than manually verified action classes. The source package includes the original caption and clip-level measurements for all analyzed clips, the plotted values behind every analysis figure, and an index of the corresponding figure captions.

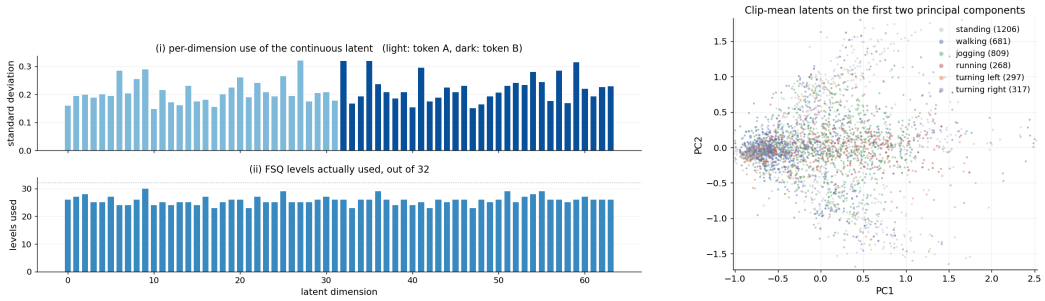
For linear regression probes (Alain & Bengio, 2016), clips are assigned to an 80/20 train/test split so that frames from one clip cannot appear in both sets. Inputs and targets are standardized using training statistics, and ridge regularization is $\lambda = 10^{-2}$. The nonlinear probe is a 64–256–256–1 MLP with GELU activations, trained for three epochs with Adam at 10^{-3} and batch size 8192. Logistic left/right probes use cross-validation over clip-mean latents. Probe results quantify information recoverable by the specified model class; they do not establish a coordinate-wise factorization or a causal control direction.

Generated-motion analysis uses the 30,000-update EMA checkpoint, 100 deterministic DDIM steps, and classifier-free guidance scale 2.0. The prompt-contrast suite contains 16 prompts with eight shared seeds and 200 generated frames per prompt. Score composition uses 32 shared seeds per caption pair; the conflicting-caption study uses 16 seeds. We report decoded quantities before execution and the corresponding quantities measured after closed-loop execution. This paired reporting is essential because a latent or decoded edit may be semantically clear while being dynamically unstable.

E.2 DISTRIBUTED AND TEMPORAL LATENT STRUCTURE

No continuous dimension is inactive in either representation: no dimension has a standard deviation below half the median. For the frozen encoder (Figure 7), the per-dimension standard deviation ranges from 0.148 to 0.320 (median 0.205), and the number of observed FSQ levels ranges from 23 to 30 of 32 (median 26); in a PCA over frame latents, the first principal component explains 24.2% of the variance, the first five 65.8%, and the first ten 85.0%. For latents generated from text (Figure 8), the corresponding statistics are 0.141–0.351 (median 0.206), 23–29 occupied FSQ levels (median 25), and 29.6%, 67.1%, and 84.7% explained variance, respectively. Both representations allocate 47% and 53% of their variance to the first and second 32-D tokens and require 13 principal components to explain 90% of the total variance.

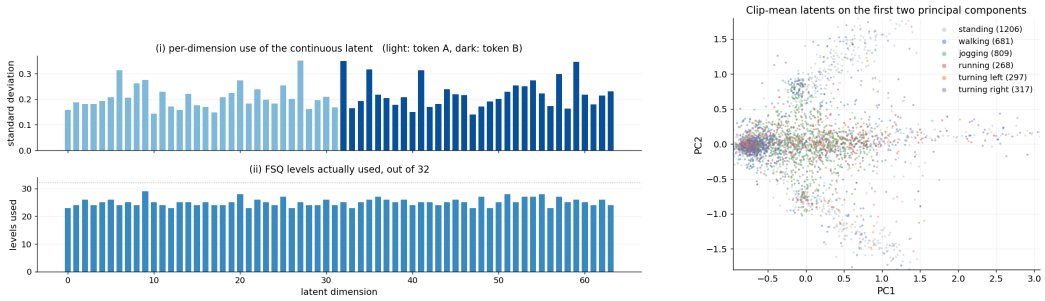
The agreement in per-dimension variability, FSQ utilization, token-wise variance, and principal-component structure indicates that the text-conditioned generator reproduces the broad statistical organization of the frozen controller latent space rather than collapsing onto a restricted subset of latent dimensions. Both tokens carry every tested quantity, so the two-token organization should not be interpreted as a semantic partition. The following analyses examine whether this reproduced latent structure also preserves motion- and language-dependent semantics.



(a) Standard deviation and used FSQ levels per dimension. Every channel carries signal and the two 32-D tokens are comparably loaded.

(b) Clip means on the first two principal components. Caption groups overlap, so leading variance is not a semantic map.

Figure 7: **The frozen controller latent space is broadly used and not semantically ordered.** Latents produced by SONIC's robot encoder, on the same clips the generated analysis covers. Every channel is occupied and the leading principal components do not separate caption groups, so the code is spread over many dimensions rather than organized along simple interpretable axes.



(a) The same per-dimension summary for generated latents. The occupancy pattern of Figure 7a is retained, with no dead channel.

(b) Clip means of generated latents on the first two principal components. Caption groups overlap here as well.

Figure 8: **Latents generated from text occupy the controller's space in the same way.** Our generator, evaluated on the same clips and the same number of frames per clip as Figure 7, so the two summaries are directly comparable. The per-dimension occupancy and the overlap of caption groups both reproduce the reference; the leading component is somewhat more dominant, indicating mild concentration rather than a different occupancy of the space.

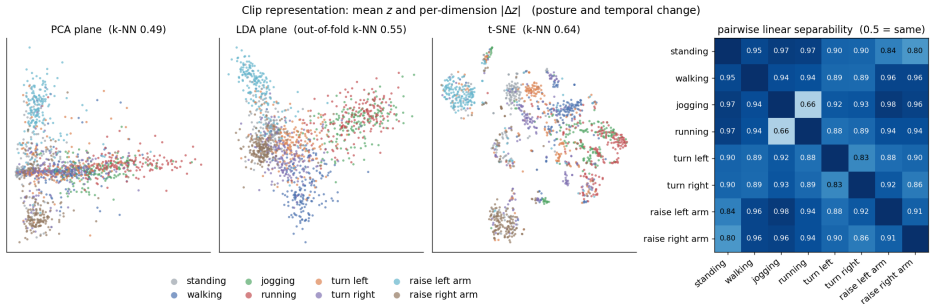


Figure 9: **Generated motion classes are distributed across posture and temporal change.** Each generated clip is represented by its mean latent and per-dimension mean absolute temporal change. The same eight motion classes are shown under PCA, LDA, and t-SNE projections, followed by pairwise linear separability in the full representation.

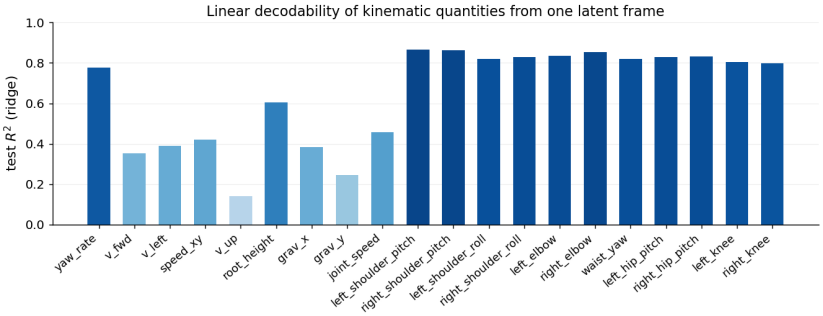
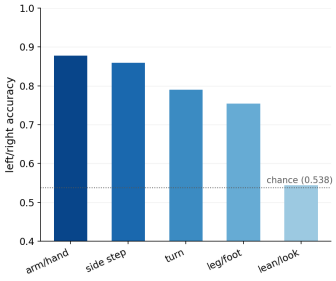
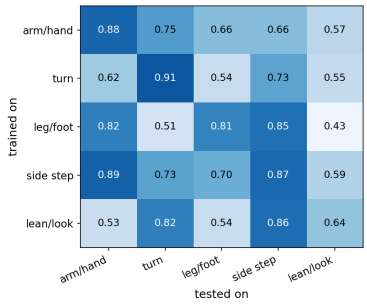


Figure 10: Linear (ridge) test-set R^2 for kinematic quantities decoded from one continuous controller-latent frame. Pose variables are strongly linearly recoverable, while locomotion quantities are less so and gain most from the nonlinear readout in Table 6.

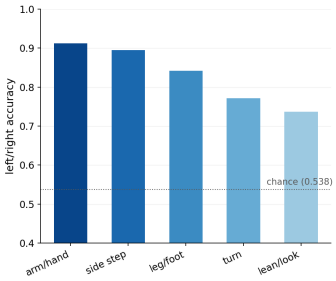


(a) One balanced left−right direction, applied to every family with a per-family median threshold. Only lean/look falls back to chance.

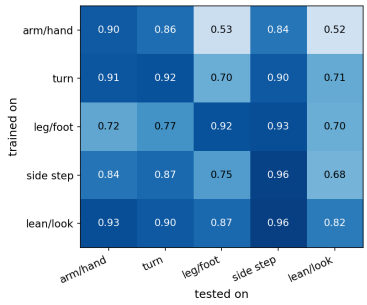


(b) Left/right classifiers trained on one action family and tested on another. The diagonal is within-family cross-validation.

Figure 11: **The same laterality analysis on the frozen encoder's latents**, restricted to the 5,668 clips the generated analysis covers, with the same captions, the same balancing and the same 0.538 chance level as Figure 12.



(a) One balanced left−right direction, applied to every family with a per-family median threshold. All five families stay well above chance, so a shared laterality direction exists.



(b) Left/right classifiers trained on one action family and tested on another. The diagonal is within-family cross-validation.

Figure 12: **Laterality is partly shared across action families.** Action families are defined by captions that mention exactly one side and exactly one family, balanced to 57 clips per side per family. Chance is 0.538, because the shared-axis estimator selects the better of the two sign conventions. Within-family accuracy averages 0.906 and cross-family transfer averages 0.796, so a common direction carries most of the signal while each family keeps a component of its own. The lean/look family is the smallest, with 78 and 57 clips.

Table 6: Clip-disjoint probe performance. "FSQ" fits the same linear readout after quantization.

Target	Linear R^2	FSQ linear R^2	MLP R^2
Yaw rate	0.778	0.766	0.811
Forward velocity	0.352	0.337	0.746
Planar speed	0.419	0.382	0.733
Root height	0.604	0.590	0.740
Left shoulder pitch	0.868	0.865	0.945
Right shoulder pitch	0.862	0.856	0.940
Left knee	0.804	0.780	0.862

Linear probes recover substantial joint-configuration information from the controller latent: across all 29 joint angles the linear R^2 averages 0.662 (0.282 to 0.868) (Figure 10). Quantization causes only a small drop for the displayed quantities (Table 6). These results show that joint-level kinematic information remains recoverable after FSQ, while not implying that individual latent dimensions correspond to independent motion variables.

Left/right structure is shared across action families but is not completely action-independent. Left/right identity is recoverable from clip-mean latents at 0.933 over all left-only versus right-only captions, yet the mean-difference directions of individual contrasts are only partly aligned (cosine 0.47 between turning and arm raising). With the action families balanced to 57 clips per side per family, a single left-minus-right direction classifies arm/hand, side-step, leg/foot, turn, and lean/look clips at 0.912, 0.895, 0.842, 0.772, and 0.737, respectively (Figure 12a). A left/right classifier trained on one family transfers to the others with mean accuracy 0.796, compared with 0.906 within-family (Figure 12b). These results indicate that laterality contains a shared component while retaining action-specific structure.

Frozen-encoder control on the same clips. Generated latents could separate left from right more easily simply by being more prototypical than recorded motion. We therefore repeat the analysis on the frozen encoder's latents for the same 5,668 clips, with the same captions and the same balancing to 57 clips per side per family (Table 7 and Figure 11). The gap persists under this matched comparison: within-family accuracy is 0.821 for encoder latents and 0.906 for generated latents, while cross-family transfer is 0.667 and 0.796, respectively. The gap is not a collapse: for four of the five families the distance between the left and right class means grows by 30 to 74% from encoder to generated latents, while the spread of each class along that direction changes by at most 5%. For lean/look, the corresponding shared-axis accuracies are 0.544 and 0.737; this is also the one family the robot cannot express, since the G1 has no neck and a recorded "look to the left" survives retargeting only as a slight waist yaw. The larger separation in generated latents may partly reflect classifier-free guidance, which uses $s = 2.0$ at sampling time and can amplify caption-conditioned differences. We therefore interpret this comparison as evidence that laterality remains strongly represented after generation, rather than that generation improves the underlying semantic structure.

Table 7: Laterality read from the frozen encoder's latents and from the generated latents on the same clips. Both use the identical 5,668 clips and captions, balanced to 57 clips per side per family, so the sign-selecting chance level is 0.538 for both columns.

	Encoder latents	Generated latents
Shared-axis accuracy by family		
arm/hand	0.877	0.912
side step	0.860	0.895
leg/foot	0.754	0.842
turn	0.789	0.772
lean/look	0.544	0.737
Within-family mean	0.821	0.906
Cross-family transfer	0.667	0.796

Together, these results suggest that the generated controller latents preserve a distributed semantic structure rather than decomposing motion into independent axes. Related behaviors share reusable directions, such as laterality, while action-specific components remain necessary to represent the full motion.

The temporal code reflects movement intensity more strongly through variation than absolute norm (Figure 5a). Over clips, mean $\|\Delta\tilde{z}\|$ has Spearman correlation 0.76 with root speed and 0.62 with joint-angle variability, while per-channel latent standard deviation correlates 0.85 with joint-angle variability. In clips of at least four seconds, the dominant standardized-latent frequency falls within 0.13 Hz of the knee stride frequency for 52% of standing, 73% of walking, 62% of jogging, and 61% of running clips; Figure 5b shows one walking and one running clip as an example. These results, together with Table 8, characterize the latent as a temporally structured future command rather than a static style label.

Table 8: Linear R^2 when the latent at time t predicts a kinematic target at $t + \tau$. Positive shifts test future-command information.

Target	-1.00s	-0.50s	-0.25s	0.00s	$+0.25$s	$+0.50$s	$+1.00$s
Yaw rate	.143	.235	.391	**.778**	.505	.319	.162
Forward velocity	.216	.270	.314	.352	.385	**.404**	.353
Planar speed	.249	.329	.406	.419	.478	**.489**	.415
Left shoulder pitch	.431	.516	.628	**.868**	.838	.615	.487
Left knee	.320	.431	.507	**.804**	.781	.619	.505
Root height	.385	.438	.495	.604	**.645**	.548	.498

E.3 Spatial guidance as a controlled latent intervention

The distributed structure above is read off unguided samples. The guided runs of Appendix C.2 let us probe the same structure by intervening on it: a free and a guided sample share their caption and their noise draw, so their difference Δz isolates what the spatial constraint did and nothing else. Figure 13 summarizes the 500 prompt–constraint pairs (84 left-hand, 83 right-hand, 83 both-hands, 125 left-foot and 125 right-foot constraints).

The size of the edit tracks the geometry it has to achieve. Across pairs, $\|\Delta z\|$ correlates with the distance the link actually moved at Spearman $\rho = 0.89$ (against 0.85 with the distance it was asked to move), at a slope of 4.8 latent units per meter. The edit is not a small perturbation: it averages 0.66 ± 0.28 of the free sample's own latent norm, and it is spread over the whole clip and over all 64 channels rather than concentrated at the constrained frames or in a few coordinates.

The direction of the edit identifies which link was constrained. Left- and right-hand edits are close to orthogonal (cosine 0.14) while each aligns with the both-hands edit (0.74 and 0.74); the two feet share more with each other (0.39); and hand and foot edits are unrelated ($|\cos| \leq 0.14$). These directions also partly reuse the language-derived laterality structure of Appendix E.2: the hand left$-$right edit has cosine 0.54 with the caption-derived arm axis and the foot left$-$right edit 0.49 with the caption-derived leg axis, while the foot edit is orthogonal to the arm axis (0.06). The correspondence is partial rather than exact, as the hand edit also retains cosine 0.41 with the leg axis.

Finally, the edit carries the target itself: a linear readout of Δz recovers the requested pelvis-frame target position with $R^2 = 0.66$, 0.61 and 0.54 for the forward, lateral and vertical coordinates. Geometric information is therefore present in the controller latent in a recoverable form, and, like the semantic structure of the previous subsection, it is distributed across the representation rather than localized to particular dimensions.

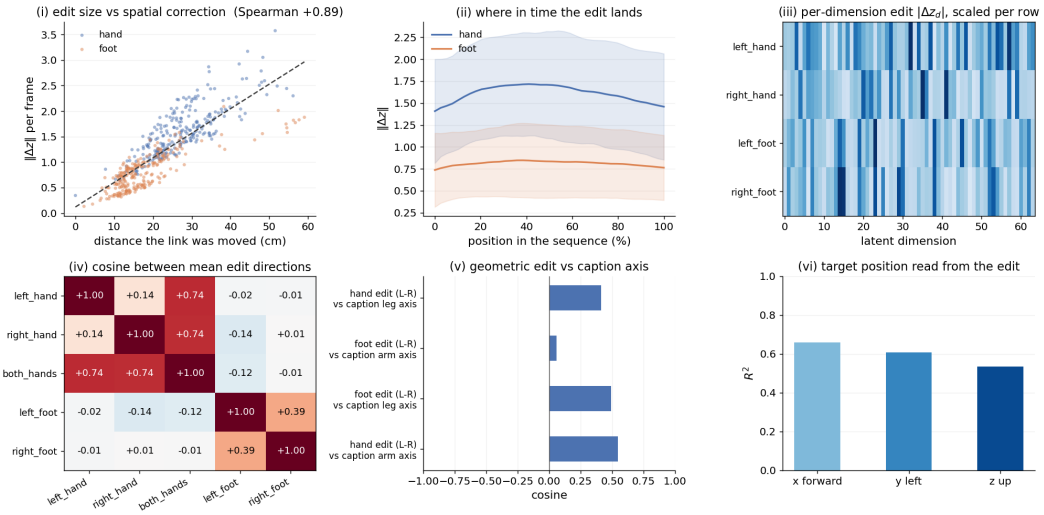


Figure 13: **Spatial constraints as controlled interventions in generated controller latents.** Free and constrained samples use identical text and noise. The panels report how latent-edit magnitude scales with the achieved link displacement, where the edit occurs in time and across dimensions, similarities among link-specific edit directions, alignment with caption-derived body-side axes, and linear readout of the requested 3-D target from the edit.

E.4 Text contrasts and denoising dynamics

The text generator reproduces the directions measured in the library. With eight shared seeds, "turn left" and "turn right" decode to $+0.521 \pm 0.054$ and -0.482 ± 0.115 rad of heading change over the 0.9-s reference horizon. Left- and right-arm raise prompts decode to -1.563 ± 0.274 and -1.603 ± 0.161 rad at the named shoulder, respectively, while the opposite shoulder remains near its standing value. The linear forward-velocity probe returns 0.040 ± 0.045, 0.470 ± 0.116, 0.714 ± 0.112, and 0.911 ± 0.125 m/s for standing, walking, jogging, and running, and -0.215 ± 0.031 m/s for walking backward. Although the linear probe compresses the absolute velocity scale, it preserves the expected ordering across locomotion intensities and the sign of backward motion; we therefore interpret these values as relative readouts rather than calibrated estimates of physical speed. Figure 4a visualizes these contrasts. These readouts also carry over to execution: per control frame, the nonlinear readout of the generated latent correlates with the executed kinematics at $r = 0.87$ for forward velocity, 0.90 for yaw rate, 0.97 for left-shoulder pitch, and 0.86 for root height (Figure 14).

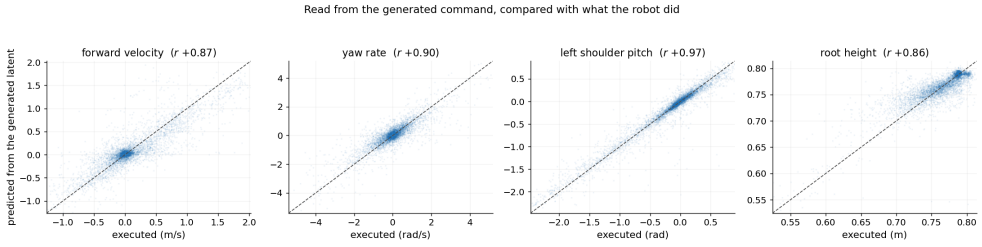


Figure 14: **Execution consistency of generated controller latents.** Readouts from generated latents predict quantities measured from executed robot motion.

Figure 15 explains why equal score weights need not cancel and need not form a conjunction. Conditional directions for opposite turns, opposite arms, and forward/backward gait have early cosine -0.36, -0.14, and -0.74, but late cosine $+0.81$, $+0.84$, and $+0.93$. Jogging plus arm raising changes from -0.54 to $+0.29$. Early disagreement breaks symmetry; later, both condition-specific predictions are evaluated near the same selected motion and can point toward that basin. Thus these

are trajectory-dependent denoising directions rather than two fixed latent vectors whose sum has a globally defined semantic meaning. Early denoising therefore enters a motion basin and the later conditional directions adapt to that shared trajectory, so Equation 6 defines a conditional composition rule rather than arithmetic over fixed opposite semantic axes.

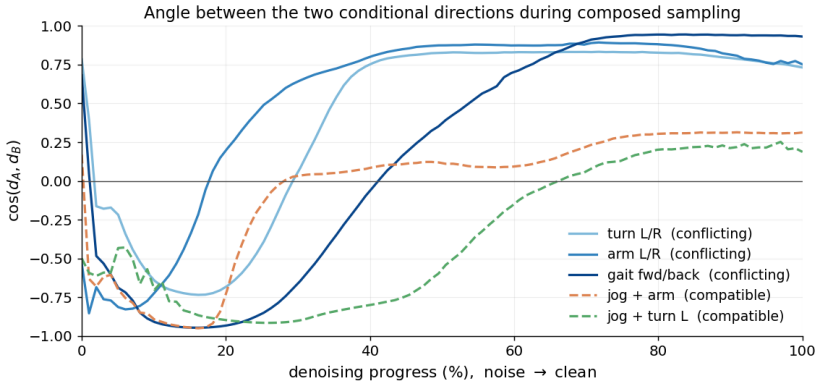


Figure 15: Cosine similarity between two caption-conditional denoising directions over 100 DDIM steps. Conflicting and gait–upper-body pairs initially oppose one another. As the sample becomes clean, the directions align after the trajectory has entered a particular motion basin. Curves average four shared seeds.

E.5 Composition capability and limitations

Compatible multi-attribute generation. The main semantic-composition path combines caption-conditional denoising directions at every DDIM step and generates one controller-latent sequence. The qualitative examples in Figure 3(b) show that this process can express two recognizable attributes after physics execution: a jump forward with both hands raised and a squat with the left hand raised. Table 9 broadens the test to ten compatible caption pairs, three weight settings, 32 shared seeds, and both decoded and executed measurements. It reports attribute retention relative to the corresponding single-caption generation and the share of seeds that fall. Figure 16 shows where an added clause acts at fixed noise: the latent change is spread over many channels (the eight most affected channels carry 25–28% of it), while the executed joint-space change concentrates in the arm the clause names.

Attenuation and weight trade-offs. The sweep separates the existence of compound behavior from full-strength independent control. Equal weights are where the two attributes coexist at all: after execution, squatting plus a left-hand wave retains 70% of the squat and 53% of the wave, running plus a left-hand wave 55% and 49%, and jogging plus a left-arm raise 58% and 41%. In the other seven pairs the arm attribute is largely absent at equal weight even though the locomotion attribute survives, as in walking plus a right-hand wave (67% and 0%) or turning left plus a right-arm raise (72% and 0%). Changing the weights trades one attribute for the other rather than isolating them: for jogging plus a left-arm raise the executed retentions move from 80/8% at $1.5/0.5$ to 2/79% at $0.5/1.5$. Simultaneous expression also costs stability, since the pairs that express both attributes are the pairs that fall: squatting plus a wave falls in 22% of seeds at equal weight and 38% at $1.5/0.5$, against at most 3% in any pair whose arm response is absent. A single natural-language "A while B" caption and per-dimension precision weighting are likewise inconsistent in the supplementary sweeps. These results support graded multi-attribute generation, not a guarantee that both single-caption amplitudes are preserved.

Table 9: Score-space composition over ten caption pairs and three weight settings (32 shared seeds, 200 frames). Each attribute is reported as retention, the composed value divided by the value the caption reaches alone (100 % = full single-caption strength), from the decoded kinematics (Gen.) and after execution on the G1 (Exec.). The A attribute is a locomotion or posture quantity (forward, backward or lateral speed, heading change, or height drop); the B attribute is arm elevation, the mean height of the named wrist above its shoulder in the pelvis frame. Measuring the arm this way keeps the two attributes independent: each A caption on its own leaves the B attribute at zero. Fall is the share of the 32 seeds that fall during execution. Weights w_A/w_B multiply the classifier-free guidance scale $s = 2$.

Pair (A + B)	w_A/w_B	Gen. A (%)	Gen. B (%)	Exec. A (%)	Exec. B (%)	Fall
jogs forward + raises their left arm	1.0/1.0	34	40	58	41	0.19
	1.5/0.5	65	8	80	8	0.12
	0.5/1.5	2	80	2	79	0.00
walks forward + waves their right hand	1.0/1.0	65	0	67	0	0.00
	1.5/0.5	99	0	97	0	0.00
	0.5/1.5	8	14	6	12	0.00
walks forward + raises both arms	1.0/1.0	30	2	36	2	0.03
	1.5/0.5	101	0	101	0	0.00
	0.5/1.5	4	81	0	82	0.00
walks backward + waves their left hand	1.0/1.0	42	0	60	0	0.00
	1.5/0.5	92	0	95	0	0.00
	0.5/1.5	12	23	19	23	0.00
jogs forward + raises their right arm	1.0/1.0	29	13	37	12	0.00
	1.5/0.5	69	0	71	0	0.00
	0.5/1.5	3	68	1	67	0.00
runs forward + waves their left hand	1.0/1.0	41	51	55	49	0.09
	1.5/0.5	82	23	93	23	0.06
	0.5/1.5	3	91	4	91	0.00
turns to the left + raises their right arm	1.0/1.0	62	0	72	0	0.00
	1.5/0.5	108	0	106	0	0.00
	0.5/1.5	-2	38	12	38	0.00
turns to the right + waves their left hand	1.0/1.0	60	0	71	0	0.00
	1.5/0.5	95	0	96	0	0.00
	0.5/1.5	-3	30	2	30	0.00
walks sideways to the left + raises their left arm	1.0/1.0	53	3	58	2	0.00
	1.5/0.5	98	0	94	0	0.00
	0.5/1.5	5	76	1	74	0.00
squats down + waves their left hand	1.0/1.0	25	64	70	53	0.22
	1.5/0.5	80	25	103	8	0.38
	0.5/1.5	-18	94	8	90	0.00

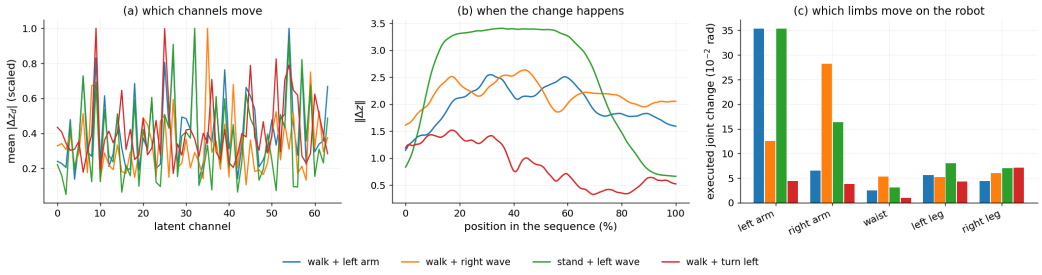


Figure 16: **Localization of a text-induced change at fixed noise.** Adding one clause changes a distributed set of latent channels and a characteristic portion of the sequence. The executed joint-space change is concentrated in the body part named by an arm clause, while a turning clause primarily changes the legs.

Conflicting semantic conditions. Compatible actions should be distinguished from mutually exclusive requests. As quantified in Table 10, left/right, forward/backward, stand/run, and squat/upright pairs generally do not produce an arithmetic midpoint: they select one mode or, for some seeds, a weak neutral outcome. Opposite arm requests converge to the left-arm mode in 10 of 16 seeds and are neutral in six, and opposite walking directions converge to backward motion in all 16 seeds, while turn pairs are divided among one side, the other side and neutral outcomes rather than averaging reliably to no turn. The selected mode is not predicted by whichever single-caption response has the larger magnitude, and caption-order agreement is 1.00 for every pair, which rules out an artefact of evaluating the first caption before the second. The early-negative and late-positive direction cosines in Figure 15 provide a diffusion-level explanation: conditions first compete while the sample is noisy, then become aligned after the trajectory has entered one basin. Consequently, our claim is that multiple text conditions can jointly shape a controller-latent generation trajectory. We do not claim a disentangled representation, arbitrary latent arithmetic, independent semantic controls, or a guaranteed logical conjunction.

Table 10: Equal-weight ($w_A = w_B = 1$) composition of ten conflicting caption pairs over 16 shared seeds. The outcome is the pair's metric centred between the two single-caption medians and signed so that A is positive; a seed is assigned to A or B when it exceeds 35 % of the single-caption strength, otherwise it is neutral. Swapping the caption order reproduces the per-seed assignment in the stated fraction of seeds.

Pair (A / B)	Metric	Single A	Single B	Composed	A/B/neutral	Order agr.
turns to the left / turns to the right	yaw (rad)	$+0.44$	-0.52	$+0.06$	8/2/6	1.00
turns around to the left / turns around to the right	yaw (rad)	$+0.55$	-0.61	$+0.20$	9/2/5	1.00
raises their left arm / raises their right arm	armdiff (rad)	$+1.64$	-1.56	$+0.78$	10/0/6	1.00
waves their left hand / waves their right hand	armdiff (rad)	$+1.40$	-1.23	$+0.60$	9/0/7	1.00
walks forward / walks backward	vfwd (m/s)	$+0.92$	-0.62	-0.45	0/16/0	1.00
jogs forward / jogs backward	vfwd (m/s)	$+1.41$	-1.17	-0.47	0/14/2	1.00
walks sideways to the left / walks sideways to the right	vleft (m/s)	$+0.56$	-0.52	$+0.14$	7/2/7	1.00
stands still / runs forward	speed (m/s)	$+0.06$	$+1.57$	$+0.17$	16/0/0	1.00
squats down / stands upright	height (m)	$+0.49$	$+0.79$	$+0.75$	0/16/0	1.00
raises both arms / lowers both arms to the sides	bsp (rad)	-1.93	-0.10	-1.26	5/0/11	1.00