CLASS: CLUSTERED LATENT ACTIONS SEPARATED FROM BODY AND SCENE FOR ACTION GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Action representations shared across bodies would let skills learned on one robot be reused on another. Existing latent-action learning, however, entangles motion meaning, what is done, with body-specific dynamics, on which body it is done, in one latent variable, so cross-body reuse of the representation remains limited. We propose Clustered Latent Actions Separated from Body and Scene (CLASS), a framework for action generation that draws latent actions into one cluster per motion category across bodies through supervised contrastive learning and conditions the action decoder on body information as a domain embedding d, or on the object position as a scene variable c in manipulation. The center of each cluster is then a drivable signal shared across bodies and scenes. On four DMC and Loco-MuJoCo bodies, per-step latent actions and centroids drive locomotion within and across bodies and benchmarks. Other bodies' centroids drive held-out body-motion combinations, and adapting about 32 parameters adds a new body. Cross-body cumulative reward drops about 80% without d, and about 88% with contrastive learning removed. With a continuity regularizer, cosine interpolation between centroids lowers the fall rate at mid-rollout switches from about 98% to 13 to 19%. In object manipulation in simulation, fixed per-action centroids drive reach, pick, place, and push across object positions, and their four-action mean success rate of 85.5% exceeds the 79.0% of per-step re-estimation. The separation thus lets a centroid computed on one body drive the same action on another body or scene, and an action can be specified by a category name, one unlabeled demonstration, or interpolation between centroids.

1 INTRODUCTION

Sharing the meaning of motions such as "running" across body structures would let skills transfer between bodies and motion switching be controlled in a structured action space. Similarly, sharing the meaning of manipulations such as "grasping" across scenes with different object placements would let the same action be invoked in any scene. A candidate representation for motion meaning is the latent action, which latent-action learning estimates as the change between two observations (Bruce et al., 2024; Gao et al., 2025; Liang et al., 2025; Chen et al., 2025), building on world models that predict future states in a compressed latent space (Ha & Schmidhuber, 2018; Hafner et al., 2019a; 2023), so that agents learn action representations from unlabeled videos.

Latent actions have been carried across bodies through a decoder trained for each body (Xu et al., 2023; Bu et al., 2025; Kim et al., 2025; Chen et al., 2024; Zheng et al., 2025), as in XSkill and UniVLA, or through an encoder or embedding of each body (Bauer et al., 2026; Yan & Lee, 2026; Zhi et al., 2026), but because a module of each body holds the mapping from a latent action to commands, only the recognition of a motion is shared across bodies. The latent space therefore contains no point that stands for "running" on every body, a skill learned on one body does not carry over to another, and the agent has no action of its own to select and execute on a body of its choice (Figure 1a). For the agent to select a motion learned on one body and execute it on another, the same motion must receive the same latent action whichever body performs it, the recognition side, and the commands of each body must be generated from the one latent action, the driving side, since without the former a latent action from another body means nothing and without the latter the body cannot be driven. The recognition side requires the latent action to carry no body information, while the driving side requires body information to produce commands that differ from body to body, so the only way to

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.

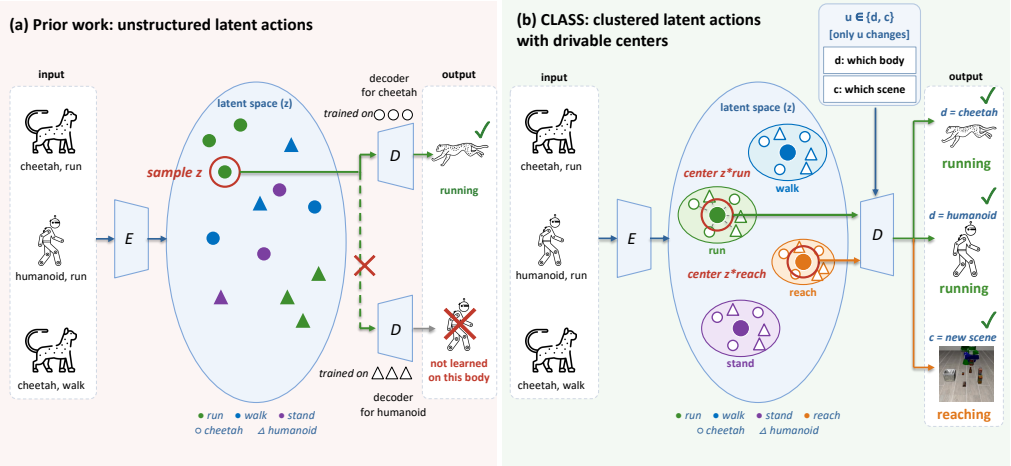


Figure 1: Framework overview. (a) Prior work with unstructured latent actions. Motion and body stay mixed in the latent space, and the same motion on different bodies forms no cluster. A sample z of the cheetah drives the decoder for the cheetah, but the decoder for the humanoid was not trained on such z, so the motion does not carry over. (b) CLASS with clustered latent actions. Contrastive learning forms one cluster per action category across bodies, and the center z^* of each cluster, not an original sample, serves as the driving signal. A single action decoder D receives z^* and the conditioning variable u, which is the domain embedding d for the body or the scene variable c for the scene. Only u changes between bodies and scenes, so the same z^*_{run} drives running on the cheetah and on the humanoid by changing d, and z^*_{reach} with c drives reaching in a new scene.

satisfy both at once is to give the body information to the decoder as a condition separate from the latent action. Because a latent action learned by prediction absorbs whatever information explains the next observation (Nikulin et al., 2025; Bu et al., 2025) and a decoder of each body keeps the body information in its weights, neither removing the body information from the latent action nor returning it as a separate condition comes about unless it is built into training explicitly. If the recognition side is built in by drawing the latent actions of each motion category into one cluster across bodies, the center of each cluster becomes a point that means "running" on every body, and the agent can hold the point to select and switch motions by itself. If a single decoder that receives the body or the object position as a separate condition serves as the driving side, the same point drives the same motion on another body or in another scene by changing only the condition.

We therefore propose CLASS (Clustered Latent Actions Separated from Body and Scene), which draws latent actions into one cluster per category across bodies by supervised contrastive learning with coarse motion-category labels and gives the body type or the object position to the action decoder as a separate condition (Figure 1b). After training, the center of each cluster is a drivable signal across bodies and scenes, and motion-category labels are needed only for training. We verify the framework on multi-body locomotion in DMC (DeepMind Control Suite) and Loco-MuJoCo (Tassa et al., 2018; Al-Hafez et al., 2023) and on Franka-arm object manipulation in simulation.

The contributions of this research are fourfold:

- **Clustered latent actions that separate motion meaning from body and scene.** With only coarse motion-category and domain labels, we learn motion meaning as the latent action and body information as the domain embedding, and the same construction holds with the object position as the condition.
- **Driving across bodies and scenes with centroids.** The centroid of each action category becomes a shared action representation across bodies and scenes. A centroid computed on one body drives motion within and across bodies on two benchmarks, and one fixed centroid drives object manipulation across object positions. Removing the domain embedding reduces cross-body reward by about 80% and removing the contrastive learning reduces it by about 88%, so both the separated condition and the cluster structure are necessary.

- **Generalization to unobserved combinations and new-body addition.** Centroids of other bodies drive body-motion combinations absent from training. A body added by adapting about 32 parameters performs motions it was not adapted on. A latent action from one unlabeled demonstration drives locomotion at 84% of centroid driving and drives object manipulation, and fixed centroids also handle object positions excluded from training.
- **Continuous motion control by switching and interpolation.** Switching the centroid mid-rollout moves behavior to another motion. With the continuity regularizer and cosine interpolation between centroids, the fall rate drops from 97–98% to 13–19%.

2 Related Work

World models and latent action learning. World models predict future states in a compressed latent space but take actions as external inputs (Ha & Schmidhuber, 2018; Hafner et al., 2019b;a; 2023; Schrittwieser et al., 2020; Sekar et al., 2020; Micheli et al., 2022; Robine et al., 2023). Approaches that extract actions from observations include world models that estimate latent actions from observation pairs and learn from video without action labels (Schmidt & Jiang, 2024; Bruce et al., 2024; Yang et al., 2023; Gao et al., 2025; Garrido et al., 2026; Liang et al., 2025; Zhou et al., 2024; Bar et al., 2025), inverse dynamics models recovering actions from observation pairs (Pathak et al., 2017; Torabi et al., 2018; Edwards et al., 2019; Ho & Ermon, 2016; Kostrikov et al., 2018) and pseudo-action labeling with them (Baker et al., 2022; Chen et al., 2022; Ye et al., 2024), and vision-language-action (VLA) foundation models (Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2024; Intelligence et al., 2025). Inter-frame motion tokenization (Chen et al., 2025; Zheng et al., 2025) and unsupervised skill discovery (Eysenbach et al., 2018; Sharma et al., 2019) represent motions or skills as latent variables without labels. These representations are re-inferred at every step, and latent plans that summarize an episode or sub-trajectory (Lynch et al., 2019; Ajay et al., 2020; Pertsch et al., 2021) are likewise inferred anew by a high-level policy. In all cases the latent structure is set only implicitly by observation-prediction or information-theoretic objectives, and the representation also captures control-irrelevant background and viewpoint (Nikulin et al., 2025).

Cross-embodiment policy learning. To adapt the policy itself to multiple bodies, some methods add body-specific modules that absorb body differences to a shared remainder (Devin et al., 2017; Chen et al., 2018), and others feed body information such as joint count and body size to a single policy that handles many bodies, even ones excluded from training (Huang et al., 2020; Gupta et al., 2022; Trabucco et al., 2022; Xiong et al., 2023; Sferrazza et al., 2024; Patel & Song, 2025; Suzuki et al., 2026; Bohlinger et al., 2024; Bohlinger & Peters, 2025; Ai et al., 2025). Large-scale cross-embodiment pretraining (O'Neill et al., 2024; Bousmalis et al., 2023; Team et al., 2024; Doshi et al., 2024; Wang et al., 2024; Reed et al., 2022) trains one foundation model on real action data of tens to hundreds of bodies, and all of these methods share module structure, observation representations, or policy weights while requiring each body's real action labels.

Conditioning on symbols and goals, and body-scene separation. A policy conditioned on a one-hot task ID or language (Rahmatizadeh et al., 2018; Jang et al., 2022; Kalashnikov et al., 2021; Hausman et al., 2018) plus a domain embedding could separate the action from the body, and a goal-conditioned policy (Lynch et al., 2019) plus a task ID could separate it from the scene. Both, however, need a label at driving time and can neither read an action from an unlabeled demonstration nor interpolate between two. Cross-body skill representations learn correspondences between bodies (Gupta et al., 2017; Kim et al., 2020; Zhang et al., 2021; Hejna et al., 2020) or extract body-independent skills or latent actions from human and robot videos for per-body decoders (Xu et al., 2023; Kim et al., 2025; Bu et al., 2025; Chen et al., 2024; 2026; Xu et al., 2026; Collins et al., 2025; Ren et al., 2025). Others map actions or poses into a common latent space and leave body specifics to per-body encoders or embeddings (Bauer et al., 2026; Yan & Lee, 2026; Zhi et al., 2026; Li et al., 2024). Physics-simulated humanoid control switches or interpolates skill embeddings and places per-skill cluster centers, but on a single body (Merel et al., 2019; Peng et al., 2022; Tessler et al., 2023; Luo et al., 2024; Dou et al., 2023; Liu et al., 2025; Rho et al., 2025). Their latent structure comes from prediction, clustering, or cross-body contrastive objectives, so assigning latents to actions needs category labels.

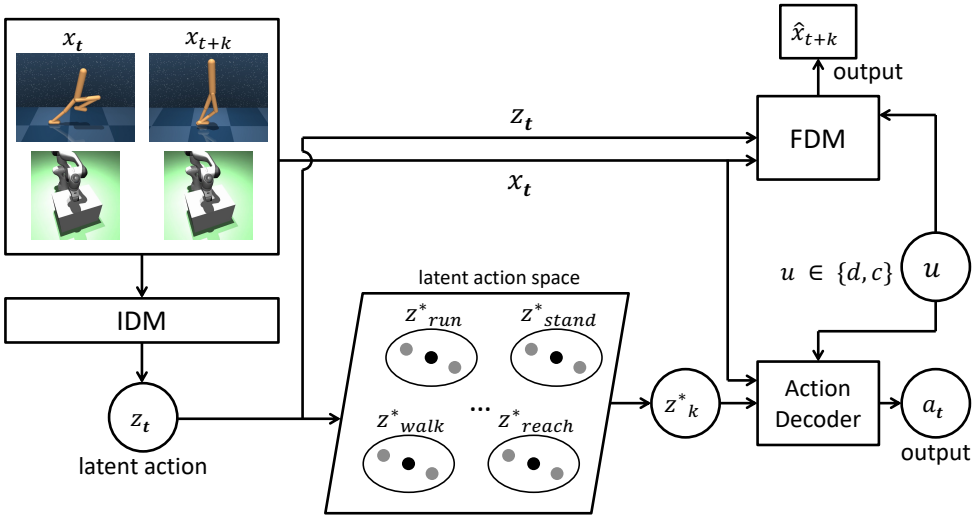


Figure 2: Overview of CLASS. The inverse dynamics model (IDM) estimates the latent action z_t from the pair of states (x_t, x_{t+k}), and the contrastive loss forms one cluster per action category in the latent action space, where the black dot of each cluster marks its centroid, such as z^*_{run}. The action decoder outputs the action a_t from x_t, the latent action, and the conditioning variable $u \in \{d, c\}$, receiving z_t in training and the centroid z^*_k of the category at driving time. The forward dynamics model (FDM) predicts $\hat{x}_{t+k}$ from x_t, z_t, and u and is used only in training.

Our positioning. Unlike methods that separate cross-body latent actions from body-specific modules (Xu et al., 2023; Kim et al., 2025; Bu et al., 2025; Bauer et al., 2026; Yan & Lee, 2026; Zhi et al., 2026), CLASS makes the center of each category a driving signal, aggregates body differences into one domain embedding for a single action decoder D, and handles whole-body motions such as walking and running on multiple bodies. Because a separated representation does not necessarily generalize to unseen combinations of factors (Higgins et al., 2017; Locatello et al., 2019; 2020; Montero et al., 2021; Mendez et al., 2022; Wang et al., 2025), the experiments test driving for body-motion combinations and object positions excluded from training.

3 Proposed Method

In this research, we build both requirements into latent-action learning, instead of leaving body information to the latent action or to a per-body decoder. Supervised contrastive learning with action-category labels, which draws the same category into one cluster regardless of the body or the object position, serves as the recognition side (§3.2), and a conditioning variable u, which supplies body or scene information to the action decoder separately from the latent action, serves as the driving side (§3.1, §3.3). We use a domain embedding $u = d$ identifying the body in locomotion and a scene variable $u = c$ computed from the object position in object manipulation, and train a model with the same encoder, losses, and action decoder in each setting (Figure 2, Appendix A).

3.1 Estimating Latent Actions

An inverse dynamics model (IDM) f_{IDM} estimates $z_t = f_{\text{IDM}}(x_t, x_{t+k})$ from a pair of the agent's own states. For x_t we use the observation o_t in locomotion and the end-effector state s_t in object manipulation. The object position is not given to the IDM and enters only through the scene variable c, so that z_t does not absorb the object position. We impose three requirements on z_t, that it explains the observed transition, that it can be converted into the real action of each body, and that the same motion receives the same z_t whichever body performs it (§3.2). A forward dynamics model (FDM), used only in training, predicts the future state $\hat{x}_{t+k}$ from (x_t, z_t, u) with the loss $\mathcal{L}_{\text{fwd}} = \|x_{t+k} - \hat{x}_{t+k}\|^2$. The action decoder D generates the real action $\hat{a}_t = D(x_t, z_t, u)$ from the same inputs with the loss $\mathcal{L}_{\text{act}} = \|a_t - \hat{a}_t\|^2$. The domain embedding d is one vector per body and stays constant during one execution, with its implementation and the treatment of direction given in Appendix A. In

object manipulation, the encoder E_c maps o_{goal}, an observation including the object position, to $c = E_c(o_{\text{goal}})$ of the same dimension as d. Because c is computed from the observation at every step, it changes during one execution when the object moves.

3.2 Structuring the Latent Space with the Meaning of the Action

With $\mathcal{L}_{\text{act}}$ and $\mathcal{L}_{\text{fwd}}$ alone, body, direction, and motion information can remain mixed in z_t, and the same motion of different bodies need not fall in the same region. We therefore introduce a supervised contrastive loss (Chen et al., 2020; Khosla et al., 2020; Laskin et al., 2020; Stooke et al., 2021) $\mathcal{L}_{\text{con}}$ that pulls together same-category samples and pushes apart different-category samples. The positive set $\mathcal{P}(i)$ contains the in-batch samples sharing the action category of anchor i, and τ is a temperature.

$$\mathcal{L}_{\text{con}} = -\sum_i \log \frac{\sum_{j \in \mathcal{P}(i)} \exp(z_i^\top z_j / \tau)}{\sum_{k \neq i} \exp(z_i^\top z_k / \tau)} \tag{1}$$

In locomotion, positives share the motion across bodies, so that the same motion of different bodies is drawn into one cluster, and in object manipulation they share a category among reach, pick, place, and push. In locomotion, an adversarial classifier that predicts the body from z_t is additionally trained through gradient reversal, which suppresses the body information that the contrastive loss leaves in z_t. We combine the three losses as follows, with $\lambda_{\text{con}} = 0.2$.

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \mathcal{L}_{\text{fwd}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}, \tag{2}$$

We also use a center loss and variance and covariance regularization of z_t (Appendix A).

3.3 Per-Category Centroids and Ways to Supply the Driving Signal

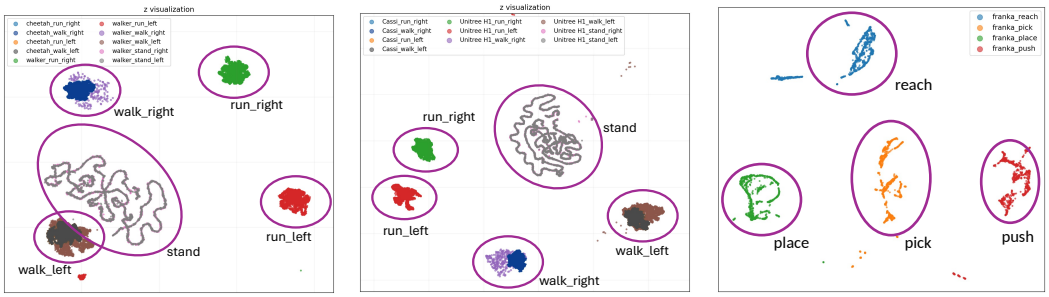
After training, we compute the centroid $z_k^* = \frac{1}{|\mathcal{Z}_k|} \sum_{z \in \mathcal{Z}_k} z$ of the latent actions $\mathcal{Z}_k$ of action category k on each body, separating directions in locomotion, and use the centroid of a category computed on one body as the driving signal for other bodies as well (Appendix A). Because $\mathcal{L}_{\text{act}}$ makes every z in a cluster convertible into a real action and $\mathcal{L}_{\text{con}}$ keeps clusters tight, their mean z_k^* also serves as a driving signal for that category. Once computed after training, a centroid can be selected by action-category name and supplied throughout an execution, with no need to re-estimate it at every step as with an individual z_t. The z_t the action decoder receives in training and the z_k^* used at driving time differ in distribution, and if the decoder reads the motion phase from small within-cluster differences, a fixed z_k^* cannot drive it. During training, we therefore replace the decoder input z_t by the within-batch mean of its category with probability p, so that $\mathcal{L}_{\text{act}}$ is learned with an input equivalent to the centroid. In locomotion, the gait phase appears in the state of joint angles and velocities, and the decoder can read it from x_t, so we use $p = 0$. Object manipulation includes phase changes in which the next movement begins while the state stays unchanged, such as lifting after the fingers have closed, so we use $p = 1$. We compare fixed driving, which feeds z_k^* throughout execution while the decoder still receives x_t, with sequential estimation, which re-estimates z_t from the latest transition each step, and evaluate closed-loop success (§4.3). The driving signal is a centroid selected by category name in the symbolic route, a z the IDM estimates from an unlabeled demonstration trajectory in the demonstration route (Appendix E), or an interpolation $z(t) = (1 - \alpha(t))z_a + \alpha(t)z_b$ between two centroids in the continuous route (Appendix J for switching and interpolation).

4 Experiments

4.1 Setup

For locomotion, we use cheetah and walker from DMC (Tassa et al., 2018) and Unitree H1 and Cassie from Loco-MuJoCo (Al-Hafez et al., 2023). Samples carry body and motion-category labels, with walk, run, and stand aligned across environments. The primary metric is cumulative rollout reward, with per-step rewards in $[0, 1]$ and episodes of at most 600 steps.

For object manipulation, a single Franka arm in MuJoCo simulation performs reach, pick, place, and push, trained on successful demonstrations from a scripted oracle controller with objects placed



(a) DMC (b) Loco-MuJoCo (c) Object manipulation (Franka)

Figure 3: UMAP projections of z colored by motion category and direction for (a) DMC and (b) Loco-MuJoCo and by action for (c) manipulation. Table 4 gives the quantitative evaluation.

in bands at the two ends of the table. For five training seeds, we evaluate 100 trials per action across the whole table and report the mean and standard deviation of success across seeds under the oracle's criterion. Appendix A gives data collection, evaluation details, and the trained models and evaluation seeds of each experiment series, Appendix I gives robustness to observation perturbations, and standard deviations in the locomotion tables are taken over evaluation rollouts.

4.2 Structure of the Latent Action Space and Contrastive Learning

Structure of the latent action space. Latent actions form five clusters by motion category and direction on DMC and Loco-MuJoCo and four clusters of reach, pick, place, and push in object manipulation (Figure 3). Walk and run gather in regions shared across bodies, so the meaning of the action is shared independently of the body. Within walk and run, direction splits into left and right regions while stand forms one cluster, so motion category and direction are represented separately.

Linear probes identify the motion category with 99.9% accuracy and the body with 80.4%, between the 75% majority baseline and the 100% from raw observations. Removing $\mathcal{L}_{\text{con}}$ raises body accuracy to 89.6% and the motion-category clusters disappear (Appendix B). The body information left at 80.4% is one reason why cross-body driving falls slightly below same-body driving in §4.3. In object manipulation, z identifies the action category with 100.0% accuracy, but a linear regression recovers the object position only with $R^2 = 0.62 \pm 0.04$ against 1.00 from raw observations, so z mainly encodes the action and c carries most of the object-position information (Appendix B).

Ablation of $\mathcal{L}_{\text{con}}$. With identical data and seed, keeping d and setting $\lambda_{\text{con}} = 0$ lowers cross-body centroid rewards from 332.55 and 312.78 to 37.39 and 39.18 for cheetah and walker, a drop of about 88% (Appendix B). Same-body driving also falls to 66–86, below the 260–275 of behavior cloning (BC) that receives no z, so the cluster structure is not a visual property but supports driving itself. In object manipulation, however, four-action mean success decreases only from $85.5 \pm 2.6\%$ to $82.7 \pm 2.1\%$, and clusters remain. The drop stays small because the replacement of §3.3 already feeds the decoder the within-batch mean of the same category, so in object manipulation the cluster structure supports driving through $\mathcal{L}_{\text{con}}$ and the replacement together.

Comparison with external baselines. For locomotion, we compare a BC conditioned on a one-hot motion and a body embedding, LAPO (Schmidt & Jiang, 2024), re-implemented to learn the latent from prediction and action reconstruction without the contrastive structure, and a BC that receives x_t and d without z (Table 1). One-hot conditioning stays about 55 below centroid driving of CLASS on the same body. The latent of LAPO drives with per-step z_t above BC but not with its centroid, and its per-step driving stays about 100 below CLASS on the same body and about 50 across bodies. A category-level driving signal therefore comes neither from specification by labels alone nor from learning from observations alone, but from latent actions learned from observations and structured by the meaning of the action.

For object manipulation, we compare goal-conditioned BC (Lynch et al., 2019), one-hot conditioned BC, and LAPO with the same data and action-decoder architecture (Table 2). CLASS exceeds

Table 1: Locomotion baselines (same data and protocol, mean ± std over 30 rollouts). CLASS (centroid) and BC rows are those of Table 3, and per-step rows use rollouts different from Table 7. Bold indicates the highest mean in each column.

method	driving signal	same-body		cross-body	
		cheetah	walker	cheetah	walker
CLASS	centroid z_k^*	419.19 ± 7.65	411.05 ± 6.41	345.58 ± 16.24	329.59 ± 15.82
CLASS	per-step z_t	**421.64** ± 10.37	**418.84** ± 8.26	**350.74** ± 14.62	**346.31** ± 16.79
one-hot conditioned BC	one-hot motion	361.85 ± 15.37	358.24 ± 12.64	-	-
LAPO (Schmidt & Jiang, 2024)	per-step z_t	326.47 ± 13.85	316.74 ± 15.85	301.72 ± 12.26	299.61 ± 14.69
LAPO (Schmidt & Jiang, 2024)	centroid	61.82 ± 25.53	55.78 ± 36.53	46.04 ± 44.65	48.35 ± 33.46
BC	none	260.79 ± 15.84	274.51 ± 14.67	-	-

Table 2: Object-manipulation baselines, ways of supplying z at driving time, and controls (success rate in %, mean ± std over five seeds, $n = 100$ per action over the whole table). Bold indicates the highest mean in each column among non-oracle rows, including ties. Oracle gives success rates only.

Configuration	reach	pick	place	push	Mean
Baselines (§4.2)					
CLASS (fixed centroid z_k^* + c)	**100.0** ± 0.00	**65.0** ± 13.81	**91.6** ± 7.11	85.2 ± 3.87	**85.5** ± 2.61
one-hot conditioned BC	98.4 ± 3.20	37.8 ± 27.58	90.8 ± 6.49	**91.6** ± 6.12	79.7 ± 5.74
LAPO (Schmidt & Jiang, 2024)	58.4 ± 3.32	0.0 ± 0.00	36.4 ± 25.43	1.4 ± 2.33	24.1 ± 6.40
goal-conditioned BC (Lynch et al., 2019)	**100.0** ± 0.00	31.4 ± 18.14	83.2 ± 9.16	44.4 ± 15.68	64.8 ± 3.86
Ways of supplying z at driving time (§4.3)					
Sequential estimation z_t	**100.0** ± 0.00	41.0 ± 33.63	90.2 ± 6.38	84.6 ± 2.58	79.0 ± 8.74
Mean z_t of one training demonstration	**100.0** ± 0.00	**65.0** ± 14.27	89.8 ± 10.11	86.4 ± 3.50	85.3 ± 3.38
Controls (§4.3)					
$z = 0$ (CLASS model)	79.4 ± 14.73	6.4 ± 8.82	40.0 ± 24.37	0.4 ± 0.49	31.6 ± 3.68
Random z (CLASS model)	66.4 ± 6.65	5.6 ± 2.50	36.4 ± 14.16	6.0 ± 5.22	28.6 ± 5.10
$c = 0$ (retrained)	44.6 ± 2.06	56.4 ± 31.80	76.0 ± 12.14	25.4 ± 6.09	50.6 ± 9.52
Oracle (ceiling)	100.0	98.0	100.0	93.0	n/a

goal-conditioned BC and LAPO, and is on par with one-hot conditioned BC in the four-action mean and in each action within seed variability.

4.3 Driving across Conditions

Driving with individual z and the necessity of the conditioning variable. We compare driving with individual z_t on the same body and across bodies, with and without d. With d, cumulative rewards on DMC and Loco-MuJoCo are 429 and 420 for the same body and 357 and 338 across bodies. The small gap between same-body and cross-body driving with d shows that d absorbs body differences and z works as a body-independent representation of the motion (confirmed in §4.4). Removing d lowers the rewards to 140 and 131 even on the same body and to 75 and 66 across bodies, a cross-body drop of about 80% (Appendix C). The decoder thus relies on d for the body-specific commands, and z alone cannot bridge the bodies. In object manipulation, setting c to zero lowers four-action mean success from $85.5 \pm 2.6\%$ to $50.6 \pm 9.5\%$ (Table 2), so conditioning on the body or scene supports driving.

Driving with centroids. Centroid driving works on the same body and across bodies, staying no more than 17 and about 30 below driving with individual z (Table 3). Zero and random z sharply reduce driving, so what supports driving is the value of z on the learned clusters. BC exceeds random z but stays 136–162 below same-body driving of CLASS, so holding z as a representation of the action itself contributes to driving. Cross-body rewards remain below same-body rewards, with gaps well above the evaluation standard error for every body.

In object manipulation, one centroid fixed per action category drives the actions across trial-varying object positions with four-action mean success of $85.5 \pm 2.6\%$. The same measure is $79.0 \pm 8.7\%$ for sequential estimation of z_t at each step from the most recent transition, $85.3 \pm 3.3\%$ for the mean z_t of one training demonstration, and $31.6 \pm 3.7\%$ and $28.6 \pm 5.1\%$ for zero and random z

Table 3: Centroid driving and controls (averaged over action categories; mean ± std).

env	body	CLASS: same-body	CLASS: cross-body	$z = 0$	z_{random}	BC
DMC	cheetah	**419.19 ± 7.65**	345.58 ± 16.24	10.14 ± 3.15	50.73 ± 10.62	260.79 ± 15.84
DMC	walker	**411.05 ± 6.41**	329.59 ± 15.82	13.64 ± 5.86	63.76 ± 12.41	274.51 ± 14.67
Loco-MuJoCo	UnitreeH1	**411.84 ± 7.25**	327.16 ± 21.36	9.62 ± 5.74	46.91 ± 17.64	251.42 ± 27.52
Loco-MuJoCo	Cassie	**413.01 ± 10.72**	363.70 ± 19.54	9.54 ± 7.32	43.71 ± 19.81	250.51 ± 31.47

(Table 2). Sequential estimation lowers pick from $65 \pm 14\%$ to $41 \pm 34\%$ because the end effector is almost stationary just before grasping and the z_t estimated from that transition departs from the centroid. Only reach remains at $79 \pm 15\%$ with $z = 0$, because the other three actions also begin with a motion toward the object. Driving in object manipulation thus also rests on the value of z on the learned clusters. Common to locomotion and object manipulation is that the centroid of each action category becomes a shared representation that stands for the same action across bodies and scenes. An individual z_t is a value re-estimated at every step from a particular transition, while a centroid, once computed after training, can be selected by action-category name, passed as it is from the body it was computed on to another body, and in object manipulation drives the same action with one vector regardless of the object position.

Driving across benchmarks and environments. We train the four bodies in a single model and perform cross-benchmark centroid driving without additional training (Appendix D). Normalized by each target body's own centroid reward, cheetah's run centroid reaches 88% on Cassie and 93% on Unitree H1, and the reverse direction from Loco-MuJoCo to DMC stays lower at 41–50%. All eight combinations still outperform zero and random z, so centroids remain valid driving signals across training environments. Connected without modification to gello sim, the trained models match their training-environment success and also drive the object in the central band excluded from training, whereas in LIBERO (Liu et al., 2023), whose implementation and control scheme differ, only reach transfers (Appendices H and L).

Measurement of the demonstration route. Driving with the mean z_t of one unlabeled trajectory's observation pairs reaches 84% of the centroid-driving reward on the same body, 81–84% across bodies within DMC, and 84–96% across benchmarks (Appendix E). In object manipulation, one demonstration unseen in training gives four-action mean success of $82.6 \pm 3.4\%$, so after training a demonstration supplies the driving signal without an action-category label.

4.4 Generalization to Combinations Unseen in Training and Adding New Bodies

Combinations excluded from training. We extend the bodies to cheetah, walker, humanoid, hopper, and ant and the motion categories to run, walk, stand, and hop, and train on 10 of the 13 tasks defined with direction variants. We drive the excluded combinations humanoid walk and walker stand with another body's centroid of the same motion category and the target body's d (Appendix G).

We compare BCs conditioned on a one-hot motion-category vector or a learned motion embedding, each with a per-body embedding, under data, decoder, and optimization settings matched to CLASS. Both evaluated combinations reach a cumulative reward of around 222, above the conditional BCs and the zero- and random-vector controls (Table 15). Walker stand reaches 55% of same-body driving with individual z when walker is included in training, so centroids drive even combinations not observed in training. In these combinations, the target body's d was not trained with the evaluated motion, so z determines the motion and d supplies only the body (Appendix A). One-hot conditioning stays at 128–144, around 60% of CLASS, and the learned embedding at 93–110. A learned embedding brings no improvement over one-hot conditioning, so the generalization comes from a structure that shares the meaning of the action across bodies, such as the motion clusters, not from a continuous motion representation. For object manipulation, we divide the table into 10 bands and exclude demonstrations from 3 of them. With fixed centroids, four-action mean success over five seeds and 100 trials per action is $78.5 \pm 7.1\%$ on the included bands and $81.6 \pm 5.8\%$ on the excluded bands (Appendix F).

Adding a body excluded from training. With all walker tasks excluded from training, adapting only about 32 domain-embedding parameters on 500 transitions with the other modules frozen

reaches 51–53% of the cumulative reward of including walker in training (Appendix F). Even without adaptation data, using the mean of the trained domain embeddings as d gives 146.6 for walk and 151.4 for run, above the 54.6 and 57.8 of a zero-initialized d (Table 12). Adapting about 3,000 parameters up to the final decoder layer reaches 56–59%. When only d is adapted, the reward with 10 transitions falls below the best value without adaptation data, but improves monotonically with the number of transitions from 50 onward. A walker adapted on 500 transitions of one motion also reaches 58–78% of direct adaptation when its adapted d is used with the other motion's centroid, with no falls observed. The adapted d thus carries body information independent of the motion, not only how to move under the adaptation motion.

4.5 Switching and Interpolating Latent Actions

To separate whether the value of z itself or cues in the current observation drive behavior, we switch centroids mid-rollout in the two orders walk–run–walk and run–walk–run and evaluate each 200-step phase with the reward of the destination task. CLASS keeps a cumulative reward of 160–180 in every phase on every body and switches to the motion of the new centroid, but instantaneous switching causes falls at the boundaries. An action decoder retrained on one motion category (single) earns reward only on the phases of that category, where it stays close to CLASS, so the difference comes from which categories are included in training, not from the amount of training data (Appendix J). The falls arise because z between two centroids barely appears in training, so a direct jump of z between them leaves the decoder unable to emit meaningful actions for a few steps around the switch.

For this experiment only, we add the continuity regularizer $\mathcal{L}_{\text{cont}} = \|z_t - z_{t+1}\|^2$ in training and interpolate between centroids over 30 steps at inference. We compare step (no interpolation), linear, and cosine using the boundary fall rate, the fraction of rollouts in which torso height falls below 60% of its steady value within 50 steps after a boundary. The fall rate decreases from about 98% with step to 27–31% with linear and 13–19% with cosine, which shares the same straight path and differs only in its smooth acceleration and deceleration at both ends (Table 20). The cumulative reward in the 50 steps after each boundary shows the same ordering, and cosine reaches 70–80% of its upper bound of 50 (per-boundary results in Appendix J). Falls under cosine remain at 19% on the two Loco-MuJoCo bodies, above the 13 to 14% on DMC. With the continuity regularizer, the paths between centroids thus also serve as driving signals, and the agent moves between motions with far fewer falls.

5 Conclusion

We proposed CLASS, which structures latent actions into one drivable cluster per motion category and gives the action decoder body or scene information as a separate variable, so that motion meaning is shared across bodies and scenes. The centroid of each action category served as a shared representation that stands for the same action across bodies and scenes, and a centroid computed on one body drove motion on another body, on body-motion combinations absent from training, and on a body added from little data. One unlabeled demonstration also provided a driving signal. In object manipulation in simulation, scene variables and fixed centroids drove actions across object positions. Driving required both the motion cluster structure and the separation of body information, and conditioned BCs and latent-action learning without the contrastive structure fell below CLASS. The paths between centroids, in addition to the discrete points, served as driving signals in the model trained with the continuity regularizer.

Limitations and future work. Training assumes per-sample action-category labels and true actuator commands, and the bodies and motion categories are limited to a small label set. Object manipulation uses state-only input and demonstrations from one scripted controller, and its action-decoder training needed the term replacing z_t with the category mean. Adding a held-out body reaches only 50–60% of full training, and centroid switching still leaves some falls. The pick success rate varies across training seeds (Table 2). We do not compare on the same data with cross-body skill representations that assume paired human and robot videos or per-body decoders. Future work includes more bodies and motion categories, unsupervised discovery of semantic categories, better domain-embedding initialization and fine-tuning for held-out conditions, visual input and transfer to another arm in manipulation, and deployment beyond simulation with real-world video and VLA foundation models.

AI USE STATEMENT

In this research, we used generative AI tools for editing assistance (grammar and phrasing) and for code assistance during experiment implementation, including drafting evaluation scripts that were subsequently reviewed and tested by the authors. We did not use generative AI tools to generate research ideas, experimental results, or the numerical values reported in this paper. All AI-assisted text and code were reviewed and verified by the authors, and we take responsibility for the final content of this paper, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This research uses only publicly available simulation benchmarks (DeepMind Control Suite, Loco-MuJoCo, and LIBERO) and a MuJoCo simulation of a Franka arm, and does not involve human subjects, personal data, or real-robot deployment. The Loco-MuJoCo motion-capture data are used under their published licenses (Appendix K). We do not foresee direct negative societal impacts specific to this work. As with other robot-learning methods, downstream deployment on physical systems would require safety validation beyond the scope of this paper.

REPRODUCIBILITY STATEMENT

Architectural and training details, including all loss weights and optimizer settings, are given in Appendix A, together with the correspondence between each experiment and the trained model it uses. Data generation for the two locomotion benchmarks and the object-manipulation simulation and the evaluation protocols (episode length, number of rollouts, reward normalization, success criteria) are described in §4.1 and Appendix A; the standard deviations of the main conditions are reported as $\pm$ in the main tables, and the model variance across training seeds in Appendix D and Appendix G. Code, trained checkpoints, and the data-generation scripts will be released upon publication.

REFERENCES

Bo Ai, Liu Dai, Nico Bohlinger, Dichen Li, Tongzhou Mu, Zhanxin Wu, K. Fay, Henrik I. Christensen, Jan Peters, and Hao Su. Towards embodiment scaling laws in robot locomotion. In *Conference on Robot Learning (CoRL)*, 2025.

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. Opal: Offline primitive discovery for accelerating offline reinforcement learning. *arXiv preprint arXiv:2010.13611*, 2020.

Firas Al-Hafez, Guoping Zhao, Jan Peters, and Davide Tateo. Locomujoco: A comprehensive imitation learning benchmark for locomotion. *arXiv preprint arXiv:2311.02496*, 2023.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15791–15801, 2025.

Erik Bauer, Elvis Nava, and Robert K. Katzschmann. Latent action diffusion for cross-embodiment manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Nico Bohlinger and Jan Peters. Multi-embodiment locomotion at scale with extreme embodiment randomization. *arXiv preprint arXiv:2509.02815*, 2025.

Nico Bohlinger, Grzegorz Czechmanowski, Maciej Krupka, Piotr Kicki, Krzysztof Walas, Jan Peters, and Davide Tateo. One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion. In *Conference on Robot Learning (CoRL)*, 2024.

Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, Alex X Lee, Maria Bauzá, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, et al. Robocat: A self-improving generalist agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

Tao Chen, Adithyavairavan Murali, and Abhinav Gupta. Hardware conditioned policies for multi-robot transfer learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Xi Chen, Ali Ghadirzadeh, Tianhe Yu, Jianhao Wang, Alex Yuan Gao, Wenzhe Li, Liang Bin, Chelsea Finn, and Chongjie Zhang. Lapo: Latent-variable advantage-weighted policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 36902–36913, 2022.

Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. IGOR: Image-GOal representations are the atomic control units for foundation models in embodied AI. *arXiv preprint arXiv:2411.00785*, 2024.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, Jianyu Chen, and Jiang Bian. villa-X: Enhancing latent action modeling in vision-language-action models. In *International Conference on Learning Representations (ICLR)*, 2026.

Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19752–19763, 2025.

Jeremy A. Collins, Loránd Cheng, Kunal Aneja, Albert Wilcox, Benjamin Joffe, and Animesh Garg. AMPLIFY: Actionless motion priors for robot learning from videos. *arXiv preprint arXiv:2506.14198*, 2025.

Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2169–2176. IEEE, 2017.

Ria Doshi, Homer Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *Conference on Robot Learning (CoRL)*, 2024.

Zhiyang Dou, Xuelin Chen, Qingnan Fan, Taku Komura, and Wenping Wang. C·ase: Learning conditional adversarial skill embeddings for physics-based characters. In *SIGGRAPH Asia 2023 Conference Papers*, 2023. doi: 10.1145/3610548.3618205.

Ashley Edwards, Himanshu Sahni, Yannick Schroecker, and Charles Isbell. Imitating latent policies from observation. In *International conference on machine learning*, pp. 1755–1763. PMLR, 2019.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Shenyuan Gao, Siyuan Zhou, Yilun Du, Jun Zhang, and Chuang Gan. Adaworld: Learning adaptable world models with latent actions. *arXiv preprint arXiv:2503.18938*, 2025.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild, 2026. URL https://arxiv.org/abs/2601.05230.

Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1703.02949.

Agrim Gupta, Linxi Fan, Surya Ganguli, and Li Fei-Fei. Metamorph: Learning universal controllers with transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations (ICLR)*, 2018.

Donald J. Hejna, Pieter Abbeel, and Lerrel Pinto. Hierarchically decoupled imitation for morphological transfer. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4159–4171, 2020. URL https://arxiv.org/abs/2003.01709.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Wenlong Huang, Igor Mordatch, and Deepak Pathak. One policy to control them all: Shared modular policies for agent-agnostic control. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4455–4464. PMLR, 2020.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning (CoRL)*, volume 164 of *Proceedings of Machine Learning Research*, pp. 991–1002. PMLR, 2022.

Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. MT-Opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. UniSkill: Imitating human videos via cross-embodiment skill representations. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, 2025.

Kuno Kim, Yihong Gu, Jiaming Song, Shengjia Zhao, and Stefano Ermon. Domain adaptive imitation learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5286–5295, 2020. URL https://arxiv.org/abs/1910.00105.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *arXiv preprint arXiv:1809.02925*, 2018.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on Robot Learning (CoRL)*, 2017.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pp. 5639–5650. PMLR, 2020.

Tianyu Li, Hyunyoung Jung, Matthew Gombolay, Yong Kwon Cho, and Sehoon Ha. CrossLoco: Human motion driven control of legged robots via guided unsupervised reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2024.

Anthony Liang, Pavel Czempin, Matthew Hong, Yutai Zhou, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Nian Liu, Libin Liu, Zilong Zhang, Zi Wang, Hongzhao Xie, Tengyu Liu, Xinyi Tong, Yaodong Yang, and Zhaofeng He. Learning uniformly distributed embedding clusters of stylistic skills for physically simulated characters. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM)*, 2025. doi: 10.1145/3746027.3755577.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4114–4124, 2019. URL https://arxiv.org/abs/1811.12359.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6348–6359, 2020. URL https://arxiv.org/abs/2002.02886.

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

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.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on Robot Learning (CoRL)*, 2019.

Jorge A. Mendez, Marcel Hussing, Meghna Gummadi, and Eric Eaton. CompoSuite: A compositional reinforcement learning benchmark. In *Proceedings of the 1st Conference on Lifelong Learning Agents (CoLLAs)*, volume 199 of *Proceedings of Machine Learning Research*, pp. 982–1003, 2022. URL https://arxiv.org/abs/2207.04136.

Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. Neural probabilistic motor primitives for humanoid control. In *International Conference on Learning Representations (ICLR)*, 2019.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Milton Llera Montero, Casimir J. H. Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey S. Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=qbH974jKUVy.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors. *arXiv preprint arXiv:2502.00379*, 2025.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Austin Patel and Shuran Song. GET-Zero: Graph embodiment transformer for zero-shot embodiment generalization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

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), 2022. doi: 10.1145/3528223.3530110.

Karl Pertsch, Youngwoon Lee, and Joseph Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on robot learning*, pp. 188–204. PMLR, 2021.

Rouhollah Rahmatizadeh, Pooya Abolghasemi, Ladislau Bölöni, and Sergey Levine. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3758–3765, 2018.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022.

Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Seungeun Rho, Aaron Trinh, Danfei Xu, and Sehoon Ha. Reference grounded skill discovery. *arXiv preprint arXiv:2510.06203*, 2025.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. *arXiv preprint arXiv:2303.07109*, 2023.

Dominik Schmidt and Minqi Jiang. Learning to act without actions. In *International Conference on Learning Representations (ICLR)*, 2024.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International conference on machine learning*, pp. 8583–8592. PMLR, 2020.

Carmelo Sferrazza, Dun-Ming Huang, Fangchen Liu, Jongmin Lee, and Pieter Abbeel. Body transformer: Leveraging robot embodiment for policy learning. In *Conference on Robot Learning (CoRL)*, 2024.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International conference on machine learning*, pp. 9870–9879. PMLR, 2021.

Kei Suzuki, Jing Liu, Ye Wang, Chiori Hori, Matthew Brand, Diego Romeres, and Toshiaki Koike-Akino. Embedding morphology into transformers for cross-robot policy learning. *arXiv preprint arXiv:2603.00182*, 2026.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

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 2023 Conference Proceedings*, 2023. doi: 10.1145/3588432.3591541.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Brandon Trabucco, Mariano Phielipp, and Glen Berseth. AnyMorph: Learning transferable policies by inferring agent morphology. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21677–21691. PMLR, 2022.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

Ren Wang, Xin Wang, Tongtong Feng, Xinyue Gong, Guangyao Li, Yu-Wei Zhan, Qing Li, and Wenwu Zhu. Improving compositional generalization in cross-embodiment learning via mixture of disentangled prototypes. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25)*, Dublin, Ireland, 2025. ACM. doi: 10.1145/3746027.3754499. URL `https://doi.org/10.1145/3746027.3754499`.

Zheng Xiong, Jacob Beck, and Shimon Whiteson. Universal morphology control via contextual modulation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38286–38300. PMLR, 2023.

Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. XSkill: Cross embodiment skill discovery. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, 2023.

Runze Xu, Yiluo Zhang, Jian Wang, Yu Wang, and Jincheng Yu. Motion-focused latent action enables cross-embodiment VLA training from human egovideos. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026.

Yashuai Yan and Dongheui Lee. Learning a unified latent space for cross-embodiment robot control. *arXiv preprint arXiv:2601.15419*, 2026.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

Qiang Zhang, Tete Xiao, Alexei A. Efros, Lerrel Pinto, and Xiaolong Wang. Learning cross-domain correspondence for control with dynamics cycle-consistency. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2012.09811.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22508–22519, 2025.

Heng Zhi, Wentao Tan, Lei Zhu, Fengling Li, Jingjing Li, Guoli Yang, and Heng Tao Shen. MOTIF: Learning action motifs for few-shot cross-embodiment transfer. *arXiv preprint arXiv:2602.13764*, 2026. URL https://arxiv.org/abs/2602.13764.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A Architecture and Training Details

Architectures. The inverse dynamics model f_{IDM}, the forward dynamics model f_{FDM}, and the action decoder D are 4-layer MLPs (width 256, ReLU), and the latent action has $d_z = 64$. The IDM takes the pair (x_t, x_{t+k}) with $k = 4$, and the FDM predicts x_{t+k}. In locomotion x_t is the observation o_t, and in object manipulation it is the 7-dimensional state s_t of end-effector position, velocity, and gripper. In manipulation, d is replaced by c, the 16-dimensional output of a 3-layer MLP E_c (width 256) applied to the 4-dimensional o_{goal} that contains the object position, and E_c is shared by the FDM and D. Layer counts, widths, and d_z are the same in both settings.

Domain embedding and direction. The embedding d is a lookup table (`nn.Embedding`) with one 16-dimensional entry per body, which the FDM and D each hold and index identically, so one entry amounts to about 32 parameters (§4.4). Motion category and direction are carried by z, and the centroids z_k^* are computed from the training z_t per body, category, and direction. At driving time D receives the d of the target body, and same-body driving uses the target body's own centroid while cross-body driving uses the centroid of another body with the same category and direction. For a combination excluded from training (§4.4), D receives the d of the target body as it is, and for an added body (Appendix F) only the d of the new body is adapted.

Losses. We set $\lambda_{\mathrm{act}} = \lambda_{\mathrm{fwd}} = 1.0$ and $\lambda_{\mathrm{con}} = 0.2$ with temperature $\tau = 0.2$ in Eq. (1), and the positive set $\mathcal{P}(i)$ consists of the in-batch samples with the same category label. In addition to Eq. (2), we use a center loss that pulls z_t toward the same-category batch centroid (weight 0.05) and variance and covariance regularization of z_t (weights 1.0 and 0.1). Locomotion additionally trains the body classifier of §3.2 through gradient reversal (weight 0.2). The probability of replacing the decoder input z_t by the same-category batch mean is $p = 0$ in locomotion and $p = 1$ in manipulation (§3.3).

Cross-embodiment dimension handling. In locomotion, observations and actions are zero-padded to their maximum dimensions across the training bodies so that a single f_{IDM}, f_{FDM}, and D handle all bodies. Manipulation has one body and uses no padding.

Table 4: Linear probes and cluster metrics for z. Manipulation values are means over five training seeds ($\pm$ denotes standard deviation).

Setting / metric	$\lambda_{\text{con}} = 0.2$	$\lambda_{\text{con}} = 0$	Reference
Locomotion: category accuracy	0.999	0.763	chance 0.333; majority 0.500
Locomotion: body accuracy	0.804	0.896	raw 1.000; chance 0.500; majority 0.750
Locomotion: silhouette	0.445	-0.153	n/a
Manipulation: category accuracy	1.000	n/a	chance 0.250; raw 1.000
Manipulation: position at t, R^2	0.62 ± 0.04	n/a	raw 1.00
Manipulation: initial position, R^2	0.22 ± 0.04	n/a	raw 0.58
Manipulation: silhouette	0.972 ± 0.001	n/a	n/a

Data. For DMC, we collect run and walk from cheetah and run, walk, and stand from walker, storing 200 episodes per body-motion combination covering both directions. Cheetah walk is generated by capping the running speed. For Loco-MuJoCo, we use Unitree H1 and Cassie and segment the mocap trajectories (Lafan1, AMASS) into the walk, run, and stand categories used in DMC. Object manipulation uses a MuJoCo simulation with one Franka arm and a 5 cm cube on a table, and the state has 7 dimensions and the action 4 (end-effector displacement and gripper command). Training uses 400 successful demonstrations per action from a scripted oracle controller with objects placed in bands at the two ends of the table, and states, observations, and actions are standardized with the training-data means and standard deviations. On the same trial sequence, the oracle succeeds at 100% for reach, 98% for pick, 100% for place, and 93% for push.

Optimization. We use AdamW (Loshchilov & Hutter, 2017) (learning rate 2×10^{-4}, weight decay 10^{-4}, batch size 256) for 300,000 steps in locomotion and 100,000 steps in manipulation, sampling each batch uniformly from the pooled trajectories of all training domains.

Trained models and evaluation seeds. Unless stated otherwise, locomotion trains one model per configuration, and the $\pm$ in the tables is the standard deviation over evaluation rollouts. Driving within a benchmark (Tables 3 and 7) and switching (Table 19) use one model per benchmark, while cross-benchmark driving (Table 9), training-seed variance (Table 10), and the demonstration route (Table 11) are evaluated with one model over the four bodies. The body addition of Appendix F starts from a model trained with all walker tasks excluded, and the held-out combinations (Tables 15 and 16) are driven with a model trained on 10 of the 13 tasks over five bodies. The conditional BCs are trained separately on the same data, and the BC for included combinations is trained on all 13 tasks. Interpolation (Tables 20 and 21) uses a model retrained with the continuity regularizer $\mathcal{L}_{\text{cont}}$. Driving and switching use 30 rollouts per body-motion condition, interpolation 20 seeds per environment, body, schedule, and switching pattern, and the tables of Appendices D–G 20 episodes. Manipulation trains five models per configuration (training seeds 0 to 4), evaluates each on 100 trials per action, and reports mean $\pm$ std across the models.

Hardware and training time. Each locomotion model takes about 3 hours on one A100 GPU, and the total compute including preliminary runs and unreported ablations is about 180 GPU-hours (about 60 runs). Manipulation takes about 25 minutes per seed for 100,000 steps on a 14-core CPU.

B Information in Latent Actions and the Contrastive-Loss Ablation

Table 4 gives the quantitative evaluation of Figure 3 in §4.2. Linear probes use an 80/20 split of 4,000 samples, with logistic regression for category and body and linear regression for object position (Table 4). For manipulation, we sample 1,000 pairs (x_t, x_{t+4}) per category for each of five training seeds. Body accuracy from z is 80.4% (majority 75%, raw observations 100%), and object-position R^2 is 0.62 at time t and 0.22 at episode start, showing that some body and scene information remains.

Removing the contrastive loss substantially lowers locomotion centroid-driving rewards (Table 5). In manipulation, nearest-centroid accuracy remains 1.000, while pick and place decrease, push improves,

Table 5: Locomotion contrastive-loss ablation (same data and training seed; d retained). Cumulative rewards are cheetah / walker. Re-measurement at $\lambda_{\text{con}} = 0.2$ uses rollouts different from Table 3.

λ_{con}	Same body	Cross body
0.2 (re-measured)	416.00 / 415.21	332.55 / 312.78
0	66.47 / 85.67	37.39 / 39.18

Table 6: Manipulation contrastive-loss ablation (same data; mean $\pm$ std over five training seeds; 100 trials per model and action). Success rates are percentages.

Metric	$\lambda_{\text{con}} = 0.2$	$\lambda_{\text{con}} = 0$
Reach success (%)	100.0 ± 0.00	100.0 ± 0.00
Pick success (%)	65.0 ± 13.81	58.0 ± 8.46
Place success (%)	91.6 ± 7.11	82.4 ± 13.67
Push success (%)	85.2 ± 3.87	90.2 ± 5.42
Mean success (%)	85.5 ± 2.61	82.7 ± 2.12
Nearest-centroid accuracy	1.000	1.000
Separation ratio	52 ± 3	3.6 ± 0.2

and mean success changes from 85.5% to 82.7% (Table 6). The separation ratio is the minimum centroid distance divided by the mean within-cluster radius.

C Conditioning Ablations and Manipulation Design

We drive the same and different bodies with individual z_t extracted from observation pairs, comparing the presence and absence of d (Table 7). Removing d lowers mean cross-body rewards from 357 to 75 on DMC and from 338 to 66 on Loco-MuJoCo, and also lowers same-body rewards. The scene-variable ablation for manipulation ($c = 0$) is given in Table 2. Figure 4 shows rollouts of driving with individual z, and Figure 5 shows rollouts of centroid driving.

Using the locomotion configuration unchanged yields 0% pick success even after 200,000 training steps. To address the mismatch between training with z_t and driving with a fixed centroid, we replace decoder inputs with the same-category batch mean at $p = 1$. Push demonstrations include 10 stationary steps after reaching the goal, and the IDM input at each time is reduced from the 11-dimensional (s_t, o_t) to the 7-dimensional s_t (Table 8). Increasing the center loss alone leaves pick success low, and reducing the decoder to 3 layers and d_z to 6 lowers place and push success.

Table 7: Driving with individual z (cumulative reward; mean $\pm$ std over 30 rollouts per body-motion condition).

Environment	z source	With d		Without d	
		same body	cross body	same body	cross body
DMC	cheetah_run	451.62 ± 5.72	357.35 ± 18.54	153.13 ± 20.64	76.57 ± 42.35
DMC	cheetah_walk	411.75 ± 8.34	361.34 ± 23.51	142.76 ± 26.41	78.53 ± 53.12
DMC	walker_run	436.13 ± 6.48	382.92 ± 20.58	146.38 ± 23.74	73.65 ± 36.21
DMC	walker_walk	442.74 ± 8.16	379.15 ± 23.74	153.63 ± 28.64	86.82 ± 25.37
DMC	walker_stand	405.33 ± 10.42	302.57 ± 19.64	103.62 ± 36.43	61.75 ± 35.86
Loco-MuJoCo	UnitreeH1_run	441.26 ± 8.85	349.17 ± 19.53	137.84 ± 31.64	68.52 ± 42.21
Loco-MuJoCo	UnitreeH1_walk	416.93 ± 10.53	378.42 ± 27.62	128.41 ± 41.32	69.69 ± 53.74
Loco-MuJoCo	UnitreeH1_stand	427.85 ± 21.74	344.28 ± 35.74	139.73 ± 56.21	72.95 ± 51.87
Loco-MuJoCo	Cassie_run	407.31 ± 9.26	318.52 ± 26.74	128.94 ± 57.31	67.53 ± 61.42
Loco-MuJoCo	Cassie_walk	408.14 ± 14.63	301.94 ± 36.35	119.75 ± 65.85	52.87 ± 73.45

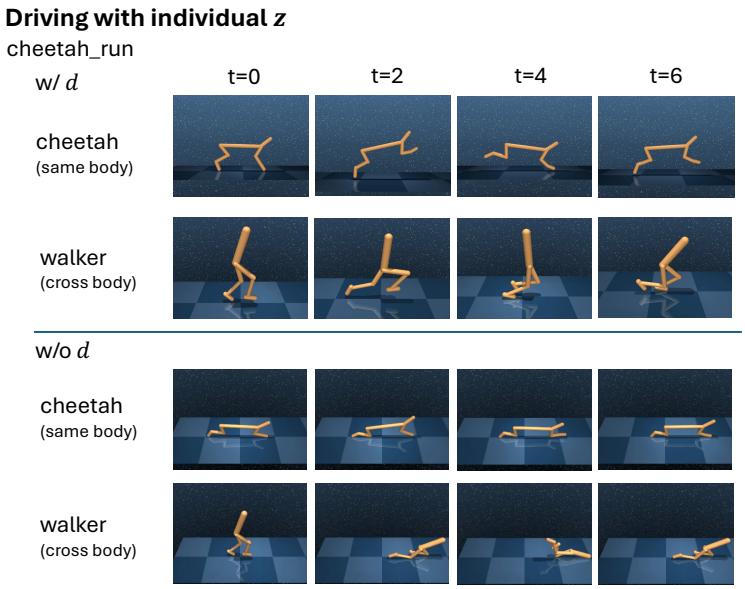


Figure 4: Rollouts of driving with individual z (Table 7). The z estimated from cheetah run drives running on the same body (cheetah) and on a different body (walker) with d, whereas without d posture is lost even on the same body, consistent with the reward drops in Table 7.

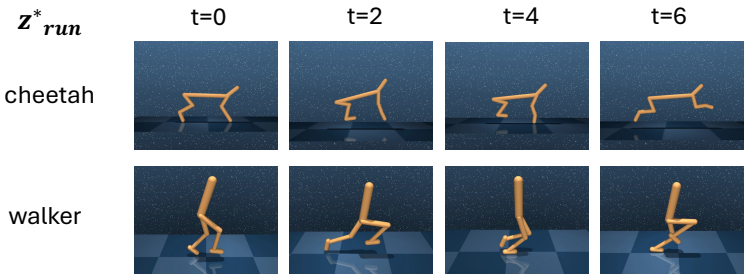


Figure 5: Rollouts of centroid driving (Table 3). The single centroid z^*_{run} drives running on both cheetah and walker at a quality comparable to driving with individual z.

D ALL CROSS-BENCHMARK PAIRS AND TRAINING-SEED VARIANCE

Table 9 reports all eight cross-benchmark pairs using one model over four bodies. Because reward scales differ, compare each row with the target body driven by its own centroid.

Table 8: Manipulation design comparisons (success rates in %; objects anywhere on the table; 100 trials per action). The first row uses 200,000 steps and seed 0; other rows use 100,000 steps and mean $\pm$ std over five training seeds. Unless changed in the row label, the IDM input at each time is 11-dimensional, the decoder has 4 layers, and $d_z = 64$.

Configuration	Reach	Pick	Place	Push	Mean
Locomotion setting ($p = 0$, no stationary steps)	100.0	0.0	28.6	43.0	42.9
$\lambda_{\text{center}} = 10$ ($p = 0$, no stationary steps)	100.0 $\pm$ 0.00	0.8 $\pm$ 0.75	46.7 $\pm$ 24.23	62.6 $\pm$ 23.02	52.5 $\pm$ 9.81
Replacement $p = 1$, no stationary steps	100.0 $\pm$ 0.00	57.2 $\pm$ 22.59	83.1 $\pm$ 10.44	29.0 $\pm$ 10.26	67.3 $\pm$ 7.81
$p = 1$, 10 stationary steps	100.0 $\pm$ 0.00	55.0 $\pm$ 27.88	82.0 $\pm$ 10.18	84.4 $\pm$ 11.94	80.4 $\pm$ 5.38
$p = 1$, 10 stationary steps, 3-layer decoder, $d_z = 6$	99.8 $\pm$ 0.40	64.0 $\pm$ 16.38	57.3 $\pm$ 38.77	71.2 $\pm$ 15.68	73.1 $\pm$ 15.57
$p = 1$, 10 stationary steps, 7-dimensional s_t input (main)	100.0 $\pm$ 0.00	65.0 $\pm$ 13.81	91.6 $\pm$ 7.11	85.2 $\pm$ 3.87	85.5 $\pm$ 2.61

Table 9: All eight cross-benchmark centroid-driving pairs (single joint model; mean ± std over 20 episodes).

Source centroid	Target body	CLASS	$z = 0$	z_{random}	Same-body reference
cheetah_run	UnitreeH1	259.8 ± 16.0	57.3 ± 19.3	38.5 ± 10.7	280.1 ± 40.8
cheetah_run	Cassie	225.5 ± 58.7	55.0 ± 36.1	48.4 ± 26.3	255.4 ± 83.7
walker_walk	UnitreeH1	151.8 ± 17.1	60.3 ± 17.9	42.4 ± 16.3	188.0 ± 19.5
walker_walk	Cassie	164.2 ± 18.5	64.4 ± 19.1	43.9 ± 19.6	194.6 ± 11.0
UnitreeH1_walk	cheetah	177.0 ± 34.8	68.0 ± 33.3	74.9 ± 30.5	394.4 ± 27.5
UnitreeH1_walk	walker	190.0 ± 59.5	72.5 ± 21.4	56.8 ± 40.7	390.2 ± 26.6
Cassie_run	cheetah	158.4 ± 17.8	63.3 ± 22.7	40.3 ± 12.7	388.0 ± 19.5
Cassie_run	walker	198.6 ± 134.4	51.4 ± 48.2	46.4 ± 12.4	395.4 ± 83.7

Table 10: DMC centroid driving, mean ± std over five training seeds (20 episodes per model).

Condition	Mean ± std across training seeds
Same body: cheetah run	398.1 ± 11.6
Same body: walker run	392.7 ± 10.6
Same body: walker walk	404.7 ± 11.1
Same body: walker stand	391.8 ± 9.7
Cross body: cheetah run → walker	341.3 ± 11.4
Cross body: walker run → cheetah	352.9 ± 6.1

To separate training-seed variation from rollout variation, we train five four-body models with seeds 0–4 using the same data and settings, then evaluate four same-body and two cross-body DMC conditions (20 episodes, 600 steps). Table 10 reports mean ± std across training seeds. Same-body rewards exceed cross-body rewards for all five seeds. Manipulation also reports mean ± std over five training seeds.

E Demonstration Route and One-Shot Driving

In the demonstration route, we average the z estimated by the IDM from observation pairs of one unlabeled trajectory and combine it with the d of the target body. This reaches 84% of centroid driving on the same body, 81 to 84% across bodies within DMC, and 84 to 96% from Unitree H1 to walker and cheetah across benchmarks (Table 11).

For object manipulation, we measure nearest-centroid classification of the mean z_t using 50 unseen demonstrations per action, collected with a different random seed from training. One-shot evaluation uses 10 unseen demonstrations per action, where recognition uses the same nearest-centroid classification and execution fixes each demonstration's mean z as the driving signal. Mapping and recognition reach 100% for all actions in all five seeds. Execution success is $100.0 \pm 0.00\%$ for reach, $54.0 \pm 12.00\%$ for pick, $91.6 \pm 9.18\%$ for place, and $85.0 \pm 7.80\%$ for push (mean ± std over five seeds), with variation remaining in pick.

Table 11: Driving from one demonstration (cumulative reward; 20 episodes). The demonstration column reports mean ± std over five trajectories.

Trajectory → body	One demonstration	Centroid	$z = 0$	z_{random}
cheetah run → walker run	292.5 ± 1.9	346.4	14.7	57.7
walker run → walker run (same body)	346.7 ± 3.8	411.6	11.7	47.7
walker run → cheetah run	279.0 ± 7.2	346.0	68.2	41.5
Unitree H1 run → walker run (cross-benchmark)	163.7 ± 1.9	194.1	14.7	47.7
Unitree H1 walk → cheetah run (cross-benchmark)	167.7 ± 6.3	175.1	68.2	61.5

Table 12: Walker without adaptation and with 10–500 adaptation transitions (cumulative reward; 20 episodes). Values without adaptation are mean ± std. Walk and run are evaluated rightward.

Setting	Transitions N	walk	run
nearest_agent	0	116.6 ± 6.0	115.4 ± 4.8
global_mean	0	146.6 ± 6.0	151.4 ± 4.8
Zero d	0	54.6 ± 4.8	57.8 ± 6.8
Adapt d only	10	90.3	85.9
Adapt d only	50	145.8	114.3
Adapt d only	100	183.3	154.5
Adapt d only	500	200.8	211.4
Adapt d + final layer	100	188.4	193.7
Adapt d + final layer	500	**216.7**	**234.8**
Full training	n/a	390.2	395.4

Table 13: Evaluation on another motion after adapting walker's d only on 500 transitions of one motion (20 episodes; mean ± std).

Adapted on	Evaluated motion (centroid)	Reward	$z = 0$	z_{random}
walk	run (cheetah_run_right)	123.1 ± 5.0	44.0 ± 6.5	22.9 ± 5.9
run	walk (h1_walk_right; cross-benchmark)	157.6 ± 7.3	55.3 ± 8.1	26.0 ± 8.4

F Held-Out Body Adaptation and Unseen Object Positions

We train with all walker tasks excluded. To adapt d only, we freeze the IDM, forward dynamics model, and action decoder, extract z from walker observation pairs without action labels, and minimize $\mathcal{L}_{\mathrm{act}}$ and $\mathcal{L}_{\mathrm{fwd}}$. We update one entry of d (about 32 parameters), and a separate condition also updates the decoder final layer (about 3k parameters in total). Without adaptation data, the mean of the learned embeddings (global_mean) outperforms reuse of the closest body's embedding (nearest_agent) and zero initialization (Table 12). Adaptation with $N = 10$ transitions falls below this best result, but performance improves with more transitions. At $N = 500$, adapting d alone reaches 51 to 53% of full training, and also updating the final layer reaches 56 to 59%.

Driving walker with another motion's centroid and the d adapted on a single motion reaches 58 to 78% of direct adaptation on the evaluated motion (Table 13). No falls were observed in the rollouts of these two conditions.

For object manipulation, we divide the table's x range of 0.42 to 0.65 m into 10 equal-width bands and exclude demonstrations from bands 3, 6, and 9, counted from the ends. We evaluate the included and excluded bands separately (Table 14). The four-action means are similar, while pick and place vary across seeds in both sets of bands.

G Conditioned BC and Object-Manipulation Controls

We use 13 tasks, including direction variants, across four motion categories (run, walk, stand, hop) and five bodies (cheetah, walker, humanoid, hopper, ant). The held-out comparison trains on 10 tasks after excluding humanoid walk, walker stand, and cheetah leftward run. Cheetah leftward run is a reference result because no environment rewards leftward running. One-hot BC conditions on the motion category and body embedding, matching the data, four-layer MLP decoder with hidden

Table 14: Generalization to unseen object positions (success rate in %; mean ± std over five seeds; $n = 100$ per action).

Object position	reach	pick	place	push	Mean
Training bands	100 ± 0	47 ± 24	79 ± 9	88 ± 6	78.5 ± 7.1
Excluded bands	100 ± 0	61 ± 35	67 ± 16	98 ± 3	81.6 ± 5.8

Table 15: Combinations excluded from training (20 episodes; mean ± std; matched data, decoder architecture, and optimization). Full results are in Table 16.

combination (source centroid)	CLASS	one-hot BC	learned-embedding BC	controls ($z = 0$ / z_{random})
humanoid walk (walker_walk)	**222.2 ± 25.73**	128.3 ± 24.62	110.1 ± 0.85	66.8 ± 5.74 / 63.7 ± 6.23
walker stand (humanoid_stand)	**222.6 ± 16.75**	144.3 ± 50.03	92.9 ± 47.56	42.4 ± 25.35 / 34.8 ± 16.12

Table 16: Conditioned BC comparison (cumulative reward; 20 episodes). One-hot and learned-embedding columns report mean ± std; controls report mean reward only. Controls replace the motion input of one-hot BC with zero/random vectors.

Setting	Task	One-hot	Learned embedding	Controls (zero / random)
All-task training	cheetah run (right)	341.8 ± 34.0	385.3 ± 70.5	45.9 / 65.2
All-task training	walker run (right)	342.7 ± 96.2	365.3 ± 82.9	30.0 / 50.5
All-task training	walker walk (right)	367.8 ± 74.4	396.1 ± 60.6	33.8 / 66.2
All-task training	walker stand (right)	337.9 ± 68.6	353.8 ± 45.7	41.3 / 78.9
Held-out	walker stand (right)	144.3 ± 50.03	92.9 ± 47.56	39.0 / 37.8
Held-out	humanoid walk (right)	128.3 ± 24.62	110.1 ± 0.85	69.4 / 69.2
Held-out	cheetah run (left; reference)	96.2 ± 31.2	89.0 ± 36.1	78.3 / 52.0
Same-model control	walker walk (right)	375.6 ± 48.3	386.4 ± 57.9	44.1 / 51.9
Same-model control	humanoid stand (right)	210.4 ± 39.9	184.7 ± 83.6	35.0 / 48.9

size 256, and optimization settings to CLASS. We also compare a 64-dimensional learned motion embedding. The one-hot conditioned BC of Table 1 is trained in the same way on the DMC data of Table 7 (cheetah and walker). In Table 16, "all-task training" uses a BC trained on all 13 tasks. "Held-out" and "same-model control" are the excluded and trained combinations, respectively, of a BC trained on 10 tasks.

Object-manipulation comparisons match the data, action decoder, optimization settings, and evaluation trial sequence. Goal-conditioned BC predicts a_t from (s_t, c) without z, while one-hot conditioned BC replaces z with a 64-dimensional learned embedding per action category. LAPO is our re-implementation that removes the contrastive and center losses, variance and covariance regularization, and input replacement, and learns from forward-dynamics prediction and action reconstruction. The locomotion LAPO of Table 1 is trained in the same way on the data of Table 7 and driven with per-step z_t or with its centroid. The $c = 0$ condition is retrained with the scene variable fixed to zero. Table 2 in the main text gives per-action comparisons of baselines and driving signals. Sequential estimation infers z_t at each step from the latest transition, and driving from one training demonstration fixes its mean z_t. Sequential estimation lowers pick success relative to the fixed centroid. Zero and random z reduce success while retaining some reach performance. Retraining with $c = 0$ mainly lowers reach and push success.

H Transfer to gello sim

The gello sim environment uses the same robot model and control scheme as the training environment, with the communication path used for the physical robot. We connect it by translating the coordinate frame and moving to the start pose, leaving the model and centroids unchanged. Appendix L reports transfer to LIBERO, which uses a different implementation and control scheme. We compare base with dart05, trained on demonstrations collected with DART (Laskey et al., 2017), which adds disturbances of standard deviation 0.05 to the position component of the actions. The four-action means for base and dart05 are $85.5 \pm 2.6\%$ and $83.6 \pm 6.0\%$ in the training environment, and $84.8 \pm 3.5\%$ and $81.1 \pm 6.6\%$ in gello sim. In the central band excluded from training, they are $85.0 \pm 8.0\%$ and $80.7 \pm 10.9\%$, respectively (Table 17).

I Robustness to Observation Perturbation

On DMC with d, we measure cumulative reward for single-category driving under additive Gaussian observation noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ or high-frequency vibration whose sign reverses at every step. We

Table 17: Transfer to gello sim: success rate (%) with a fixed centroid per action; mean ± std over five seeds. n is the number of trials per seed. Central denotes the band excluded from training.

Model	Environment	Position	n	Reach	Pick	Place	Push
base	training	whole	100	100.0 ± 0.00	65.0 ± 13.81	91.6 ± 7.11	85.2 ± 3.87
base	gello sim	whole	50	100.0 ± 0.00	59.6 ± 15.30	93.9 ± 5.63	85.6 ± 3.88
dart05	training	whole	100	100.0 ± 0.00	52.8 ± 14.08	90.4 ± 11.28	91.2 ± 2.32
dart05	gello sim	whole	50	100.0 ± 0.00	47.6 ± 16.27	88.6 ± 11.37	88.4 ± 1.96
base	gello sim	central	20	100.0 ± 0.00	63.0 ± 27.31	100.0 ± 0.00	77.0 ± 6.78
dart05	gello sim	central	20	100.0 ± 0.00	37.0 ± 27.31	86.0 ± 18.55	100.0 ± 0.00

Table 18: Cumulative reward under observation perturbations (DMC, with d) at the largest evaluated strength, $\sigma = 0.20$. The reference uses the with-d, same-body condition of Table 7.

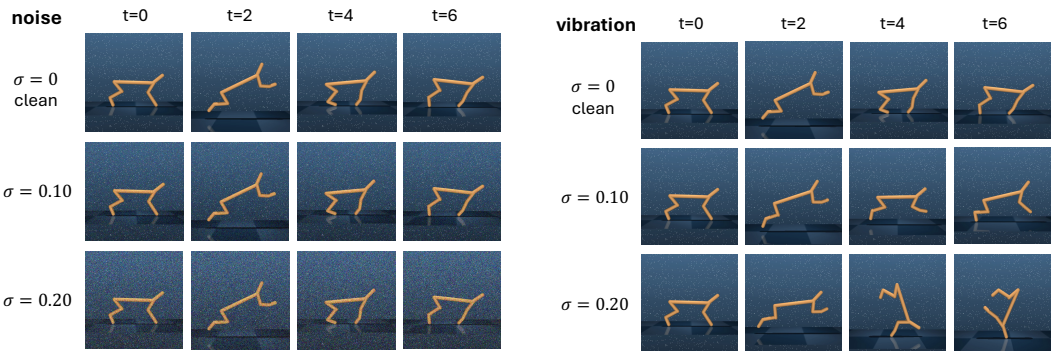
Condition	Reference	Additive noise ($\sigma = 0.20$)	High-frequency vibration ($\sigma = 0.20$)
cheetah_run	451.62	428.63	249.61
cheetah_walk	411.75	378.92	248.74
walker_run	436.13	410.85	274.97
walker_walk	442.74	409.46	261.83
walker_stand	405.33	353.18	263.49

also randomly permute the observation dimensions. The evaluated model is the main model of Table 7 without additional regularizers.

At the largest strength, additive noise causes a smaller reward drop than vibration (Table 18). For cheetah_run, reward decreases from the reference of 451.62 to 428.63 under additive noise and 249.61 under vibration. Under vibration, driving that initially succeeds cannot be sustained. Figure 6 shows the perturbed input observations and the rollouts of the driven agent at $\sigma = 0.10$ and 0.20.

J Details of Switching and Interpolation

The switching experiment in §4.5 uses walk–run–walk and run–walk–run, with 200 steps per phase. Each phase is evaluated using the reward of the task associated with its target centroid z_{target}. We feed the same z sequence to the CLASS decoder and a decoder retrained on a single category (single). The single-category decoder drives its trained category but earns low reward on the other category (Table 19). Figure 7 shows representative frames of CLASS around the switch, where the motion changes but the agent falls at the boundary.



(a) additive noise (b) high-frequency vibration

Figure 6: Input observations under each perturbation and the corresponding rollouts of the driven agent (cheetah_run, with d) at $\sigma = 0, 0.10$, and 0.20. (a) Additive Gaussian noise. (b) High-frequency vibration, under which driving is not sustained at $\sigma = 0.20$. Table 18 reports the largest strength only.

Table 19: Phase-wise cumulative reward under z switching (200 steps per phase; w: walk, r: run). CLASS rows and cheetah single-category rows report mean ± std.

Environment	Method	Body	w → r → w: ϕ_1 / ϕ_2 / ϕ_3	r → w → r: ϕ_1 / ϕ_2 / ϕ_3
DMC	CLASS	cheetah	**165.46 ± 23.64 / 178.35 ± 26.75 / 162.74 ± 13.81**	**178.54 ± 15.63 / 165.89 ± 20.84 / 181.24 ± 12.63**
DMC	CLASS	walker	**168.35 ± 16.36 / 172.84 ± 31.64 / 172.73 ± 26.31**	**173.16 ± 16.42 / 175.84 ± 27.52 / 174.95 ± 21.97**
DMC	single (walk)	cheetah	164.72 ± 12.23 / 12.47 ± 3.29 / 146.43 ± 23.18	12.53 ± 4.15 / 145.89 ± 31.74 / 14.37 ± 4.26
DMC	single (walk)	walker	157.53 / 24.65 / 148.15	18.79 / 146.85 / 12.51
DMC	single (run)	cheetah	15.43 ± 5.67 / 168.52 ± 20.74 / 12.67 ± 21.85	174.75 ± 17.42 / 16.61 ± 20.15 / 169.35 ± 19.75
DMC	single (run)	walker	18.12 / 171.83 / 16.43	171.25 / 19.43 / 169.43
Loco-MuJoCo	CLASS	UnitreeH1	**168.58 ± 14.63 / 172.61 ± 26.89 / 167.03 ± 20.16**	**175.86 ± 34.32 / 167.46 ± 25.68 / 173.59 ± 23.21**
Loco-MuJoCo	CLASS	Cassie	**165.79 ± 17.44 / 168.45 ± 23.15 / 161.57 ± 13.68**	**168.75 ± 25.34 / 167.51 ± 14.74 / 168.36 ± 23.47**
Loco-MuJoCo	single (walk)	UnitreeH1	152.85 / 13.53 / 153.76	24.64 / 137.58 / 18.64
Loco-MuJoCo	single (walk)	Cassie	149.76 / 16.37 / 146.84	21.51 / 137.58 / 17.84
Loco-MuJoCo	single (run)	UnitreeH1	17.35 / 169.52 / 14.61	164.74 / 14.26 / 162.86
Loco-MuJoCo	single (run)	Cassie	13.27 / 157.35 / 17.93	165.26 / 12.21 / 162.39

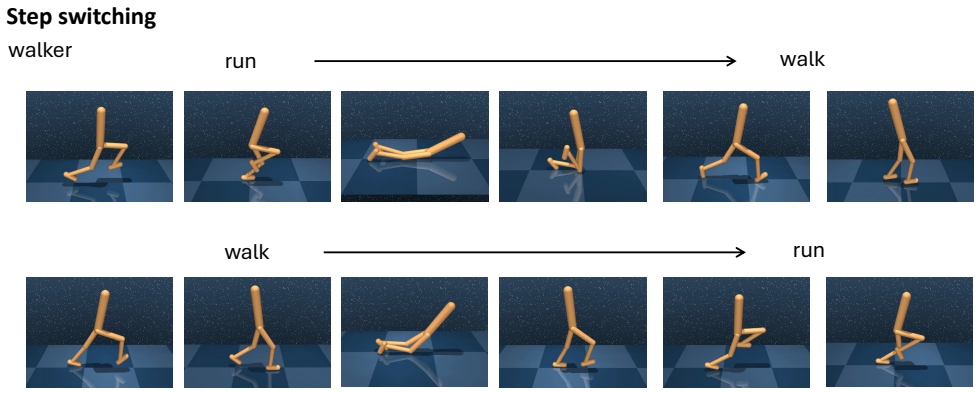


Figure 7: Rollouts of CLASS in the switching experiment (Table 19) on walker with step switching, run→walk (top) and walk→run (bottom). The agent moves into the new motion category after the switch but falls at the switch boundary, because the z traversed between two centroids is not usable by the action decoder.

For the interpolation experiment, we separately train a model with $\mathcal{L}_{\text{cont}} = \|z_t - z_{t+1}\|^2$ added to the loss in §3.2. At inference, we compare step (no interpolation), linear, and cosine over a 30-step switching window. Linear and cosine follow the same straight path, and cosine accelerates and decelerates smoothly at its ends. Boundary fall rate is the fraction of rollouts whose torso height falls below 60% of its steady value within 50 steps after a boundary. The steady value is the mean torso height under single-centroid driving of the same body. We average 20 seeds per condition over four boundaries across the two patterns. Table 20 reports fall rates, and Table 21 reports the secondary metric, the boundary-window reward, namely the cumulative reward in the 50 steps after each switch boundary, with an upper bound of 50. Tables 22 and 23 give the breakdown at each of the four boundaries.

At all four boundaries, fall rates are lowest for cosine, then linear, then step, with the reverse ordering for boundary-window reward (Tables 22 and 23). Falls remain under cosine, with rates of 13.45 to

Table 20: Boundary fall rate (%; mean ± std; 20 rollout seeds per condition, averaged over four boundaries). All schedules use the model trained with the continuity regularizer. Per-boundary values are in Table 22.

body	step	linear	cosine
cheetah	97.47 ± 6.22	28.61 ± 4.74	**13.45 ± 4.87**
walker	97.76 ± 8.54	26.63 ± 5.32	**14.04 ± 7.03**
UnitreeH1	98.13 ± 13.63	31.28 ± 14.86	**19.34 ± 12.40**
Cassie	98.07 ± 12.24	30.77 ± 10.73	**19.45 ± 6.12**

Table 21: Boundary-window reward under interpolation (cumulative reward in the 50 steps after each switch boundary; upper bound 50; four-boundary average; mean ± std). Fall rates are in Table 20.

Environment	Body	Boundary-window reward: step / linear / cosine
DMC	cheetah	3.12 ± 1.37 / 26.04 ± 8.53 / **38.36** ± **10.67**
DMC	walker	2.64 ± 1.20 / 27.38 ± 6.43 / **38.71** ± **8.26**
Loco-MuJoCo	UnitreeH1	2.25 ± 1.18 / 33.45 ± 9.53 / **36.60** ± **15.63**
Loco-MuJoCo	Cassie	2.57 ± 1.31 / 32.57 ± 12.64 / **36.59** ± **10.95**

Table 22: Boundary fall rate (%) at each of the two boundaries (b_1, b_2) of each switching pattern, by schedule and body. Each cell aggregates 20 rollout seeds.

Environment	Body	Schedule	walk→run→walk		run→walk→run	
			b_1 (walk→run)	b_2 (run→walk)	b_1 (run→walk)	b_2 (walk→run)
DMC	cheetah	step	97.25	96.82	98.56	97.25
DMC	cheetah	linear	27.25	27.24	31.37	28.57
DMC	cheetah	cosine	**14.64**	**12.75**	**12.85**	**13.56**
DMC	walker	step	97.84	98.57	97.38	97.25
DMC	walker	linear	25.96	24.47	27.54	28.56
DMC	walker	cosine	**15.25**	**13.75**	**12.34**	**14.82**
Loco-MuJoCo	UnitreeH1	step	98.25	98.61	98.42	97.25
Loco-MuJoCo	UnitreeH1	linear	31.43	30.64	31.68	31.38
Loco-MuJoCo	UnitreeH1	cosine	**20.64**	**19.74**	**18.26**	**18.74**
Loco-MuJoCo	Cassie	step	98.24	97.59	97.82	98.64
Loco-MuJoCo	Cassie	linear	30.75	30.15	31.33	30.85
Loco-MuJoCo	Cassie	cosine	**20.75**	**20.64**	**17.85**	**18.57**

19.45% (Table 20). Figure 8 shows a run→walk switch under the continuity regularizer and cosine interpolation.

K Licenses for Existing Assets

Table 24 lists asset versions, licenses, and URLs. Assets are used under their licenses with credit to the original works. No modified asset is redistributed. Locomotion was trained with MuJoCo 3.3.7, PyTorch 2.3.1, and NumPy 1.24.4. Object manipulation, which uses the MuJoCo Menagerie Panda, was trained with MuJoCo 3.4.0, PyTorch 2.9.1, and NumPy 2.2.6. LIBERO evaluation uses robosuite 1.4.1 and MuJoCo 2.3.7.

L Details of the Transfer to LIBERO

LIBERO (Liu et al., 2023) uses the same robot, a Franka Panda, with robosuite torque-based operational space control. Its meshes, inertial parameters, and control differ from the MuJoCo Menagerie Panda controlled by joint position servos during training, and it includes unseen everyday

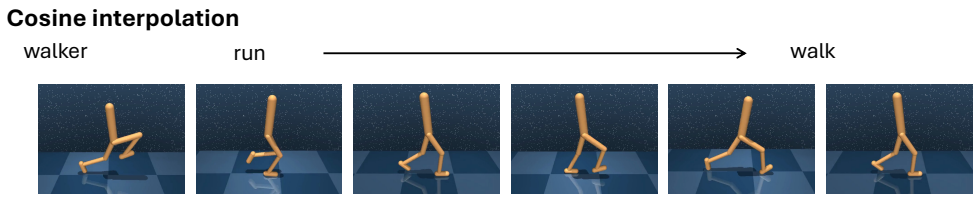


Figure 8: A run→walk switch on walker under the continuity regularizer and cosine interpolation (Table 20). In this rollout the boundary fall of Figure 7 does not occur, and the agent moves to the new motion while keeping its balance.

Table 23: Boundary-window reward (cumulative reward in the 50 steps after each switch boundary; upper bound 50) at each of the two boundaries (b_1, b_2) of each switching pattern, by schedule and body.

Environment	Body	Schedule	walk→run→walk		run→walk→run	
			b_1 (walk→run)	b_2 (run→walk)	b_1 (run→walk)	b_2 (walk→run)
DMC	cheetah	step	3.67	3.31	2.85	2.64
DMC	cheetah	linear	26.75	25.13	24.75	27.53
DMC	cheetah	cosine	**38.63**	**37.84**	**39.12**	**37.86**
DMC	walker	step	2.75	2.46	2.51	2.85
DMC	walker	linear	27.52	28.43	27.15	26.43
DMC	walker	cosine	**38.53**	**39.34**	**37.12**	**39.85**
Loco-MuJoCo	UnitreeH1	step	2.53	2.35	1.75	2.35
Loco-MuJoCo	UnitreeH1	linear	33.31	32.75	33.58	34.16
Loco-MuJoCo	UnitreeH1	cosine	**36.53**	**35.87**	**37.84**	**36.15**
Loco-MuJoCo	Cassie	step	2.75	2.47	2.58	2.46
Loco-MuJoCo	Cassie	linear	31.64	32.75	34.16	31.74
Loco-MuJoCo	Cassie	cosine	**35.74**	**36.31**	**36.72**	**37.58**

Table 24: Licenses, versions, and URLs of the existing assets used in this research.

Asset	Type	Version	License	URL
DeepMind Control Suite (`dm_control`)	Simulator / benchmark	1.0.34	Apache 2.0	`https://github.com/google-deepmind/dm_control`
Loco-MuJoCo	Simulator / benchmark	1.0.0	MIT	`https://github.com/robfiras/loco-mujoco`
PyTorch	Deep-learning framework	2.3.1 (locomotion), 2.9.1 (manipulation)	BSD-3-Clause	`https://github.com/pytorch/pytorch`
MuJoCo	Physics simulator	3.3.7 (locomotion), 3.4.0 (manipulation), 2.3.7 (LIBERO)	Apache 2.0	`https://github.com/google-deepmind/mujoco`
NumPy	Numerical library	1.24.4 (locomotion), 2.2.6 (manipulation)	BSD-3-Clause	`https://github.com/numpy/numpy`
MuJoCo Menagerie (`franka_emika_panda`)	Robot model	-	Apache 2.0	`https://github.com/google-deepmind/mujoco_menagerie`
LIBERO	Benchmark	-	MIT	`https://github.com/Lifelong-Robot-Learning/LIBERO`
robosuite	Simulation environment	1.4.1	MIT	`https://github.com/ARISE-Initiative/robosuite`

objects. A conversion layer translates the coordinate frame, moves the end effector to its initial pose, and applies per-axis gain compensation $(2, 2, 3.5)$, while the model and centroids remain unchanged. We use the base and dart05 models from Appendix H, trained with objects in the end bands of the table (0.42 to 0.45 m and 0.57 to 0.65 m). On the can, with five seeds and 30 trials per seed, dart05 reaches $100.0 \pm 0.00\%$ on reach but only $4.7 \pm 3.40\%$ on pick, showing that grasping does not transfer (Table 25). Even reach drops to $46.0 \pm 7.72\%$ for base. Evaluation in the central band excluded from training (0.45 to 0.57 m) uses only ten trials with training seed 1, so its sample size differs from the whole-table evaluation. With base, the end effector stalls 6–7 cm before the hover point, and the tested gain changes did not improve reach or pick. The gripper opens to 8 cm and the model observes each object only as a point, limiting grasping of different shapes to its training experience.

Table 25: Success rate (%) in LIBERO with a fixed centroid per action. Whole: mean ± std over five seeds with 30 trials each; central: ten trials with training seed 1, in the band excluded from training.

Model	Object position	Reach	Pick	Place	Push
base	whole	46.0 ± 7.72	3.3 ± 5.16	8.0 ± 10.02	2.7 ± 5.33
dart05	whole	100.0 ± 0.00	4.7 ± 3.40	25.3 ± 22.37	13.3 ± 9.19
base	central	10.0	0.0	10.0	30.0
dart05	central	100.0	0.0	100.0	10.0