COMMUTATOR-GUIDED LEARNING OF ORDER-AWARE LIE ACTIONS IN MIXED-LATENT VAES

Anonymous authors
Paper under double-blind review

ABSTRACT

A key challenge for Lie-structured variational autoencoders is to learn how actions compose and reproduce their order-dependent physical effects in decoded observations. We propose a two-phase framework separating representation learning from physical-action grounding and algebraic refinement. Phase 1 (P1) combines a categorical branch for category-related content with a continuous branch for Lie transformations. We compare unsupervised training with paired-weak supervision, which uses observation pairs sharing category or transformation state. Phase 2 (P2) first grounds latent directions using action-labeled transitions while keeping the representation model frozen (P2a), then refines these directions and Lie generators toward prescribed commutator relations while retaining finite-action targets (P2b). To connect the algebraic refinement to observable accuracy, a rollout variant (P2b+R) additionally matches decoded action trajectories through a frozen decoder. We evaluate latent actions, decoded observations and Lie structure on synthetic image benchmarks and controlled global rotations of real human-pose data, covering non-commuting $\mathrm{SE}(2)/\mathrm{SO}(3)$ actions and commuting controls. Paired-weak P1 improves mean categorical and geometric specialization, with trade-offs in reconstruction and FactorVAE (FVM) scores. In P2, algebraic refinement strengthens prescribed non-zero relations beyond grounding alone, while P2b+R achieves lower mean decoded endpoint error than Core P2b. Compared with the Commutative Lie Group VAE (CLG-VAE)-style strict control, which naively encourages commutativity by penalizing all basis-pair commutators, grounded refinement improves action-direction agreement across several image benchmarks, better recovers prescribed non-zero brackets, and reduces unwanted decoded order effects on the commuting benchmarks. These action-level gains coexist with competitive reconstruction and FVM scores relative to the strict control and external baselines. Results on the skeleton datasets likewise show improved recovery of non-commuting order effects and prescribed bracket directions and magnitudes. Together, these findings support grounding and refining the appropriate action relations rather than imposing blanket commutativity, while distinguishing algebraic agreement from accurate observable effects.

1 INTRODUCTION

Modern generative models rely on learned latent representations to capture the factors of variation underlying high-dimensional data. In visual generation, these representations organize semantic and geometric information, rather than merely compressing observations. Variational autoencoders (VAEs) provide a useful setting for studying this structure because they combine probabilistic inference with an explicit generative decoder. Geometric perspectives further examine the latent space through its curvature and local sensitivity (Arvanitidis et al., 2018; Chadebec et al., 2020). Yet accurate reconstruction and informative latent geometry do not by themselves establish that a model has learned how transformations interact. Lie-group models represent continuous changes as composable actions. Symmetry-aware approaches such as CLG-VAE (Zhu et al., 2021) and cFASL (Jung et al., 2024) encourage commuting latent generators. This is a useful bias when action order does not matter, but can be restrictive when it does: rotating and then translating an object in a global frame generally differs from performing the same operations in reverse. This motivates learning and evaluating the composition relations required by physical actions, rather than imposing

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.

commutativity by default. Simply allowing non-commuting generators is not enough. A learned commutator, which characterizes local order dependence, can have the wrong direction or magnitude for the intended physical action. Moreover, an internal operator difference may have little effect on a particular input, and its decoded consequences need not match the observed transformation. We therefore distinguish detecting an order effect from establishing that it represents the correct physical interaction. We assess order effects in learned operators, their action on individual latent states, and decoded observations, checking whether the resulting changes agree with the intended physical transformations. A second issue is the treatment of discrete generative factors. For images, Phase 1 (P1) combines categorical carriers, which provide learned starting states, with continuous Lie actions that transform them. We evaluate branch specialization under unsupervised and paired-weak training rather than assuming that separate branches ensure semantic separation. Phase 2 (P2) then grounds physical actions with the representation frozen (P2a) and refines their prescribed commuting or non-commuting relations while retaining finite-action targets (P2b). Frozen-decoder rollout additionally connects refinement to decoded effects. The main route uses known physical actions and transformation parameters; a reduced-label image extension instead combines anonymous transition discovery with few-shot grounding.

Our contributions are: (i) a staged, non-commutativity-aware method that uses separate categorical-carrier and continuous-action branches on images, grounds physical actions, refines prescribed bracket directions and magnitudes, and links actions to finite observations through frozen-decoder rollout; (ii) samplewise audits of finite Baker–Campbell–Hausdorff and endpoint audits that separates internal order sensitivity from physical-action agreement in latent and decoded spaces; (iii) anonymous transition discovery followed by few-shot grounding; and (iv) results across six datasets that shows improved agreement with prescribed non-commuting relations and reduced false order effects for commuting actions, supported by latent and decoded action audits and matched-update image controls.

1.1 Related work

Mixed-latent models represent discrete and continuous variation (Dupont, 2018), although unsupervised semantic disentanglement is not guaranteed (Locatello et al., 2019). Our image model combines these roles through a continuous Lie action applied to a categorical carrier. Transition-based approaches learn group structure from interactions and observed transformations (Caselles-Dupré et al., 2019; Painter et al., 2020; Quessard et al., 2020); we use transitions to ground physical actions before refining their prescribed algebra. The Homomorphism Autoencoder (HAE) connects latent action prediction with decoded reconstruction (Keurti et al., 2023), motivating our frozen-decoder rollout, while LieFlow informs the transition-discovery component of our reduced-label extension (Chen et al., 2026). Table 1 summarizes these connections and our stage-specific adaptations.

Table 1: Positioning relative to adjacent literature, not an exhaustive capability comparison. R denotes frozen-decoder rollout; the reduced-label route is image-only.

Focus (Stage)	Prior approaches	Our adaptation and contribution
Mixed factors (P1)	JointVAE: discrete and continuous codes; InfoGAN: code recovery (Dupont, 2018; Chen et al., 2016).	Continuous Lie actions transform categorical carriers. Branch specialization is evaluated, not assumed.
Group structure (P1/P2b)	Structured group models (Chau et al., 2020; Jung et al., 2024); CLG-VAE encourages commutativity (Zhu et al., 2021).	No commutativity penalty in P1; prescribed zero/nonzero brackets in P2b. An all-pairs commuting control provides a comparison.
Action grounding (P2a)	Transitions ground latent actions and composition (Painter et al., 2020; Quessard et al., 2020; Jin et al., 2024).	Freeze P1; learn basis combinations matching physical actions and finite order effects.
Decoded composition (P2b+R)	HAE learns group actions, including $\mathrm{SO}(3)$, through latent prediction and decoded reconstruction (Keurti et al., 2023).	Frozen-decoder rollout with bracket direction/magnitude refinement and signed order-effect audits.
Order-effect audits (P1/P2)	Latent curvature and local sensitivity (Arvanitidis et al., 2018; Chadebec et al., 2020; Khan & Storkey, 2023; Zheng et al., 2022; Zacherl et al., 2021).	Samplewise operator, acted-state, latent and decoded audits distinguish order sensitivity from physical agreement.
Reduced-label grounding	LieGAN/LaLiGAN discover symmetry generators; LieFlow models subgroup support (Yang et al., 2023; 2024; Chen et al., 2026).	Anonymous discovery and few-shot grounding precede refinement. Auxiliary coefficient flow does not supply action directions.

2 Method

Our framework separates representation learning (P1) from physical-action grounding (P2a) and algebraic refinement (P2b). P1 adapts CLG-VAE's Lie parameterization, exponential-map decoder

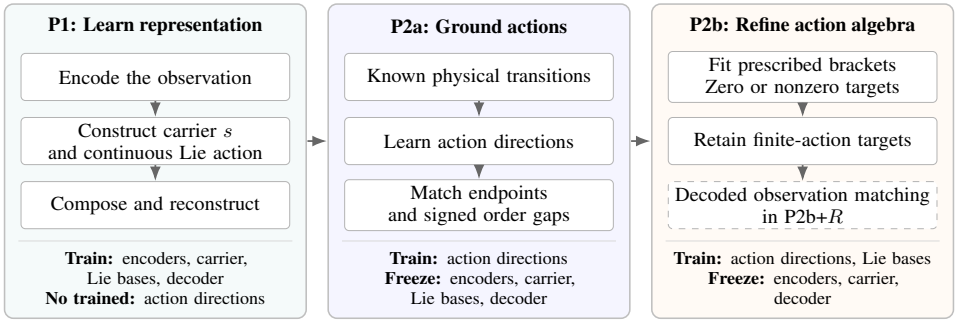

Figure 1: Two-phase framework and parameter-update scope. P1 learns the representation; P2a grounds physical actions; P2b refines their prescribed commuting or non-commuting relations. Core P2b and P2b+R update the same parameter groups; P2b+R additionally matches decoded observations through the frozen decoder (dashed box).

and feature matching (Zhu et al., 2021), but omits its decomposition and Hessian-based penalties. This leaves the bases free to commute or not, rather than imposing order independence before grounding. P2 then fits prescribed relations between physical actions, not every pair of representation bases. Figure 1 summarizes the stages and their parameter-update scopes. Detailed objectives and implementation settings appear in Appendices C and B.

2.1 P1: CARRIER REPRESENTATION AND RECONSTRUCTION

Given an input image x, P1 learns to reconstruct it using two separately parameterized encoder branches and one shared image decoder. The categorical branch constructs a learned starting vector, called a *carrier*; the continuous branch constructs a matrix that transforms this carrier. The two branches are combined before image decoding.

Categorical branch: construct the carrier. A dedicated convolutional encoder, with parameters ϕ, maps x to logits over K categorical states. Applying softmax gives the probability vector $q(x) := q_\phi(\cdot \mid x)$. A relaxed Gumbel–Softmax sample $c \in \mathbb{R}^K$ provides differentiable weights over these states (Jang et al., 2017; Dupont, 2018). These weights combine the rows of an embedding table E, and a learned projection P produces the latent carrier $s(c) = P(c^\top E) \in \mathbb{R}^m$, where m is the carrier dimension.

Continuous branch: construct the transformation. A separate convolutional encoder maps the same image x to a matrix-valued feature $\widehat{Z}(x) \in \mathbb{R}^{m\times m}$. Two prediction heads take its flattened entries as input: one outputs the Gaussian posterior mean $\mu(x) \in \mathbb{R}^d$, and the other outputs the log variance $\log\big(\sigma(x)^2\big) \in \mathbb{R}^d$, where d is the number of continuous coordinates. We sample continuous coordinates $t \in \mathbb{R}^d$ from this Gaussian posterior and construct $G(t) = \exp\big(\sum_{k=1}^d t_k A_k\big)$, where $A_k \in \mathbb{R}^{m\times m}$ are learned Lie bases. This matrix acts on the carrier to form the representation used for reconstruction; it does not itself produce an image. Its physical-action interpretation is established later in P2.

Combine the branches and reconstruct the image. The categorical branch supplies the carrier $s(c)$, while the continuous branch supplies $G(t)$. A single image decoder δ reconstructs the input from their product:

$$h = G(t)s(c) \in \mathbb{R}^m, \qquad \hat{x} = \delta(h). \tag{1}$$

Compared with CLG-VAE's continuous latent formulation, the carrier adds a separate route for category-related content. Applying the same Lie action to different carriers provides an inductive bias toward separating content from transformation. Throughout this paper, *branch specialization* means that category is more predictable from the categorical branch and geometric factors from the continuous branch, each relative to the other branch.

Following CLG-VAE (Zhu et al., 2021), image training uses $\hat{x}_{\text{rec}} = \delta(\widehat{Z}(x)s(c))$ in 20% of mini-batches, giving the encoder feature a direct reconstruction signal through the shared carrier and

decoder. Otherwise, it uses Equation (1). Inference uses only the $G(t)$ route; routing and gradient stops are detailed in Appendix B.3. For skeletons (MI-Motion and 3DPW), a shared learned carrier s_0 replaces the categorical branch: $\hat{x} = \delta(G(t)s_0)$, with the decoder producing 3D joint coordinates rather than pixels.

Training objective. P1 jointly trains the representation and reconstruction networks using four loss groups. The fit group, $\mathcal{L}_{\text{fit}} = \mathcal{L}_{\text{rec}} + \alpha\mathcal{L}_{\text{feat}}$, reconstructs the input and aligns the encoder feature with the generated Lie matrix:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}\,\ell_{\text{rec}}(\hat{x}_{\text{rec}}, x), \qquad \mathcal{L}_{\text{feat}} = \mathbb{E}\|\widehat{Z}(x) - G(t)\|_F^2. \tag{2}$$

Here $\hat{x}_{\text{rec}}$ uses the active reconstruction route, and ℓ_{rec} is BCE or SSE depending on the dataset. Feature matching follows CLG-VAE (Zhu et al., 2021). The prior group, $\mathcal{L}_{\text{prior}} = \beta\mathcal{L}_{\text{KL}}$, controls continuous posterior capacity using the Gaussian KL penalty in Equation (10) (Burgess et al., 2018; Dupont, 2018); α and β are loss weights. The code group $\mathcal{L}_{\text{code}}$ encourages categorical-code recovery (Chen et al., 2016), codebook use and embedding separation, while discouraging overconfident categorical posteriors. For code recovery, a training-only predictor Q estimates the inferred probabilities $q(x)$ from group-route reconstruction logits, without ground-truth class labels. It is not part of the reconstruction path. The leakage group $\mathcal{L}_{\text{leak}}$ penalizes squared batchwise cross-correlations between $q(x)$ and $\mu(x)$. Detaching $q(x)$ restricts its gradients to the continuous branch. Weights and dataset-specific details are given in Table 8, Appendix B.1.

The complete unsupervised image objective is

$$\mathcal{L}_{\text{P1}}^{\text{unsup}} = \mathcal{L}_{\text{fit}} + \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{code}} + \mathcal{L}_{\text{leak}}. \tag{3}$$

Skeletons use only fit and ordinary Gaussian KL terms. Paired-weak image training adds category- or transformation-matched pairs to encourage specialization (Appendix B.2, Equation (15)).

2.2 P2A: PHYSICAL-ACTION ALIGNMENT

P1 learns Lie bases but does not identify which combinations correspond to physical actions. P2a learns these combinations by matching predicted endpoints and order effects to observations linked by known transformations, while keeping P1 frozen. For example, dSprites uses 2D rotation and translations (Table 2). For each physical action a, a coefficient row $v_a \in \mathbb{R}^d$ defines its generator and finite action:

$$B_a = \sum\nolimits_{k=1}^{d} v_{a,k}A_k, \qquad H_a(\gamma) = \exp(\gamma B_a). \tag{4}$$

Here k indexes the frozen P1 Lie bases and γ is the known transformation amount, such as a rotation angle. An action therefore need not correspond to a single basis. These action matrices operate on the frozen P1 representation of each input x_n:

$$S_n^0 = \widehat{Z}^0(x_n) \text{ (images)}, \qquad S_n^0 = G^0(\mu^0(x_n))s_0^0 \text{ (skeletons)}.$$

Superscript 0 denotes frozen P1 quantities. Images retain the carrier separately, decoding a transformed state S as $\delta^0(Ss^0(c_n))$ with fixed source weights c_n. Skeleton states already include the shared carrier and are decoded as $\delta^0(S)$. For known action amounts α_n, β_n, a single action gives $S_{a,n} = H_a(\alpha_n)S_n^0$, while opposite orders give

$$S_{ab,n} = H_a(\alpha_n)H_b(\beta_n)S_n^0, \qquad S_{ba,n} = H_b(\beta_n)H_a(\alpha_n)S_n^0.$$

The rightmost matrix acts first: $S_{ab,n}$ applies b then a. Only the order changes; the source, carrier and action amounts remain fixed.

To learn the coefficients, we compare these predictions with the corresponding physical-action targets. Suppressing n, let $\mathcal{W}$ contain the available action sequences, with fixed weights w_ω. The comparison map Φ returns the latent state or the decoded displacement from the source reconstruction; targets $y_\omega^\star$ are correspondingly frozen encodings or observed displacements. The carrier argument c is omitted for skeletons. The predicted and target signed order differences are

$$\Delta_{ab}^{\Phi} = \Phi(S_{ab}, c) - \Phi(S_{ba}, c), \qquad \Delta_{ab}^{\star} = y_{ab}^{\star} - y_{ba}^{\star}.$$

Endpoint and order matching then use

$$\mathcal{L}_{\text{end}} = \mathbb{E}\sum\nolimits_{\omega\in\mathcal{W}} w_\omega\,\ell_{\text{end}}(\Phi(S_\omega, c), y_\omega^\star), \quad \mathcal{L}_{\text{ord}} = \mathbb{E}\,\ell_{\text{ord}}(\Delta_{ab}^{\Phi}, \Delta_{ab}^{\star}). \tag{5}$$

Both losses compare full arrays using squared errors computed per record before averaging, not merely their magnitudes. For commuting actions, $\Delta^{\star}_{ab} = 0$, so order matching penalizes unwanted order dependence. Combining these terms gives

$$\mathcal{L}_{\mathrm{P2a}} = \mathcal{L}_{\mathrm{end}} + \lambda_{\mathrm{ord}}\mathcal{L}_{\mathrm{ord}} + \mathcal{L}_{\mathrm{guard}}. \tag{6}$$

Here λ_{ord} weights order matching. The ModelNet10-only guard discourages collapse by penalizing action-response and direction-row norms below fixed floors; it is zero elsewhere. Only v_a is optimized; all P1 parameters and persistent buffers remain frozen. Weights and dataset-specific reductions are given in Appendix C.1. P2a thus fits finite physical actions; P2b subsequently refines their prescribed commutator relations.

2.3 P2b: Prescribed algebra and decoded-action refinement

Starting from P2a, P2b updates the action directions v_a and previously frozen Lie bases A_k, with $B_a = \sum_k v_{a,k} A_k$. The encoders, carrier, decoder and source states remain fixed. For each prescribed pair $(a, b) \in \mathcal{R}$, refinement targets the following relation on a frozen source state S:

$$[B_a, B_b]S \approx \sum\nolimits_g c^g_{ab} B_g S, \qquad [B_a, B_b] = B_a B_b - B_b B_a. \tag{7}$$

The commutator captures the leading local order effect. Here g indexes physical actions and c^g_{ab} are prescribed algebra coefficients, distinct from categorical weights. Targets specify zero brackets for commuting relations and nonzero brackets otherwise. These constraints concern grounded actions, not all P1 basis pairs. The normalized closure loss is

$$\mathcal{L}_{\mathrm{alg}} = \tfrac{1}{|\mathcal{R}|} \sum\nolimits_{(a,b)\in\mathcal{R}} \mathbb{E}_S \big[\big\| [B_a, B_b]S - \sum\nolimits_g c^g_{ab} B_g S \big\|^2_F / d_{ab}(S) \big], \tag{8}$$

where $d_{ab}(S) > 0$ provides dataset-specific normalization; the Frobenius norm becomes Euclidean for vector states. Closure matches response direction and magnitude, while $\mathcal{L}_{\mathrm{scale}}$ adds nonzero-bracket amplitude constraints. We group these as $\mathcal{L}_{\mathrm{bracket}} = \lambda_{\mathrm{alg}}\mathcal{L}_{\mathrm{alg}} + \mathcal{L}_{\mathrm{scale}}$. For observable effects, $\mathcal{L}_{\mathrm{disp}}$ matches decoded displacements where configured. P2b+R adds HAE-inspired one- and two-step endpoint matching through the frozen decoder, $\mathcal{L}_{\mathrm{roll}}$ (Keurti et al., 2023).Together these form $\mathcal{L}_{\mathrm{decoded}} = \sum_{a\in\mathcal{A}} \lambda_{\mathrm{disp},a} L_{\mathrm{disp},a} + \lambda_{\mathrm{roll}}\mathcal{L}_{\mathrm{roll}}$, where $L_{\mathrm{disp},a}$ averages displacement errors for action a. The single P2b objective is

$$\mathcal{L}_{\mathrm{P2b}} = \mathcal{L}_{\mathrm{P2a}} + \lambda_{\mathrm{fin}}\mathcal{L}_{\mathrm{fin}} + \mathcal{L}_{\mathrm{bracket}} + \mathcal{L}_{\mathrm{decoded}} + \mathcal{L}_{\mathrm{anchor}}. \tag{9}$$

Core P2b sets $\lambda_{\mathrm{roll}} = 0$. Dataset-specific definitions, weights and gradient routes are given in Appendices C.2 and C.3. For images, LF+64 combines LieFlow-style anonymous transition discovery (Chen et al., 2026) with 64 labeled anchors per action, followed by reduced-label P2b+R refinement (Appendix C.4). This extension uses few-shot grounding, rather than fully unsupervised learning.

3 Experiments

3.1 Datasets

We use six datasets to test non-commuting actions, commuting controls, and the extension from image observations to real captured 3D poses. The intended P1 factors describe what we evaluate. The properties and experimental roles of each dataset are summarized in Table 2. For image observations, dSprites tests rotation–translation order effects and a commuting translation pair. ModelNet10 tests rotations about different global axes. The selected 3DShapes and Cars3D changes instead act on independent product-grid coordinates and should commute; this does not assume that arbitrary physical 3D rotations commute. For human-pose observations, MI-Motion and 3DPW use 3D joint coordinates rather than RGB images, with no categorical latent branch. MI-Motion follows the LieFlow preprocessing (Chen et al., 2026). After dataset-specific centring and scale normalization, we apply global-axis rotations with normalization scales held fixed. This provides exact single- and ordered-action targets on real poses, without treating natural articulation as global rotation. Preprocessing, dataset sizes and split construction are specified in Appendix E; evaluation splits are stated for each reported comparison.

Table 2: Datasets and experimental roles. Superscripts mark intended categorical (cat) and continuous (cont) P1 factors. Skeletons use joint coordinates and a shared carrier without a categorical code.

Dataset	Observation	Intended P1 factors	Tested P2 actions
Non-Abelian images			
dSprites (Higgins et al., 2017)	Binary render	Shape^{cat}; (position, scale, rotation)^{cont}	$\mathrm{SE}(2)$: rotation, translations
ModelNet10 (Wu et al., 2015)	Depth/normal renders	Object class^{cat}; 3D orientation^{cont}	Global-axis $\mathrm{SO}(3)$ rotations
Abelian image controls			
3DShapes (Burgess et al., 2018)	Synthetic RGB render	Shape^{cat}; (scale, orientation)^{cont}	Product-grid scale/orientation
Cars3D (Reed et al., 2015)	Rendered RGB car	Instance identity^{cat}; (elevation, azimuth)^{cont}	Product-grid elevation/azimuth
Non-Abelian real-pose tests			
MI-Motion (Peng et al., 2023)	Captured joints (20)	Shared carrier; no discrete code	Global-axis $\mathrm{SO}(3)$ rotations
3DPW (von Marcard et al., 2018)	Fitted SMPL joints (24)	Shared carrier; no discrete code	Global-axis $\mathrm{SO}(3)$ rotations

Table 3: Training comparisons, their purposes and result locations.

Comparison	Purpose and design	Results
P1 supervision	Unsupervised versus paired-weak P1 tests whether weak supervision strengthens branch specialization and decoded control under fixed-component interventions.	Table 4;Table 12 (Appendix E.1.3)
Categorical branch	On 3DShapes and Cars3D, compare categorical–continuous and continuous-only models with ordinary or Strict continuation at matched total updates. Tests the branch-and-objective design and its interaction with commutativity; removing the branch also removes its losses.	Table 13; Appendix F.2
P2 strategies	Compare P1-only (representation continuation), Strict (all-pairs commutativity without action targets), P2a-only (full-budget grounding), Core P2b (grounding and algebra refinement), P2b+R (adds rollout), and LF+64 (few-shot-grounded refinement). Tests what each training choice contributes after P1.	Table 6
Skeleton algebra	Strict versus Core P2b and P2b+R tests commutativity versus prescribed nonzero brackets, retaining physical-action supervision, anchors and matched refinement budgets.	Table 5
P2b loss components	On dSprites, add closure ($\mathcal{L}_{\mathrm{alg}}$), log magnitude ($\mathcal{L}_{\mathrm{logmag}}$) and signed projection ($\mathcal{L}_{\mathrm{proj}}$) to finite-only P2b, yielding Core P2b; separately add rollout ($\mathcal{L}_{\mathrm{roll}}$). P2a parents, finite-action supervision, anchor schedules and refinement budgets remain fixed.	Table 14; Appendix F.3
Bracket losses with rollout	On dSprites, compare finite-only+R with Core P2b+R to test whether prescribed-bracket losses contribute beyond decoded rollout. Match P2a parents, finite-action supervision, rollout settings, anchor-refresh policy, trainable parameters, decoder-buffer handling and refinement updates across five seeds.	Table 15; Appendix F.3
Grounding labels	Compare 32, 64 and 256 anchors per action on dSprites, 3DShapes and ModelNet10 to test label-budget sensitivity, holding discovery checkpoints, update budgets and other settings fixed.	Table 16; Appendix F.4

3.2 Training settings and ablations

We report means and sample standard deviations over five independent seeds. Objectives and schedules are given in Table 10 (Appendix E.1). Image P2 uses unsupervised P1 checkpoints to keep P1 supervision consistent with Strict and the external baselines; paired-weak P1 is evaluated separately. This does not match subsequent action supervision. Skeleton P1 is category-free and unsupervised. Table 3 summarizes the comparisons and their purposes. The post-P1 image comparisons share seed-specific P1 parents and evaluation records. Update budgets and exceptions are in Appendix E.1.3. Equal updates need not match compute, trainable parameter counts or supervision.

3.3 Evaluation metrics and baselines

P2 action and algebra agreement. We evaluate learned actions in latent and decoded spaces and assess agreement with prescribed Lie relations. Endpoint errors, direction cosines and order assignment measure physical-action accuracy; closure, bracket direction and magnitude measure algebraic agreement. For commuting actions, decoded false gaps measure unwanted order dependence. Together, these metrics test whether algebraic agreement is accompanied by accurate observable effects. Definitions and evaluation records appear in Appendices D.3 and D.4.

Representation quality and external baselines. Alongside the P2 action comparisons, we assess reconstruction and FVM against the internal controls and external β-VAE and cFASL baselines (Table 7). β-VAE provides a baseline without explicit group actions; cFASL provides a symmetry-aware comparison through its CLG-based commutative repository variant (Jung et al., 2024). This tests whether action-level gains coexist with competitive representation quality.

Table 4: P1 audit (mean±sample SD; five seeds). Both $q \to d$ and $\mu \to d$ measure category-prediction accuracy; lower $\mu \to d$ favours less continuous-branch leakage. Positive gaps favour intended specialization. Occupancy is normalized aggregate code entropy.Definitions are given in Appendix D.1.

	dSprites		**3DShapes**		**Cars3D**		**ModelNet10**–SO(3)	
Metric	Unsup.	Paired-weak	Unsup.	Paired-weak	Unsup.	Paired-weak	Unsup.	Paired-weak
Reconstruction ↓	28.05±7.57	67.88±11.90	30.69±5.76	33.80±7.35	59.21±1.13	77.38±3.32	69.49±2.76	68.49±1.91
FVM ↑	0.858±0.067	0.729±0.068	0.798±0.090	0.857±0.019	0.946±0.051	0.925±0.026	n/a	n/a
$q \to d$ ↑	0.461±0.120	0.519±0.051	0.326±0.117	0.638±0.235	0.337±0.096	0.375±0.039	0.134±0.006	0.230±0.012
$\mu \to d$ ↓	0.695±0.149	0.490±0.053	0.499±0.135	0.346±0.116	0.148±0.014	0.052±0.014	0.215±0.012	0.209±0.023
Category gap ↑	−0.233±0.222	0.029±0.049	−0.173±0.112	0.292±0.267	0.189±0.090	0.323±0.046	−0.082±0.014	0.021±0.027
Geometry/pose gap ↑	0.062±0.029	0.133±0.043	0.128±0.023	0.173±0.024	0.079±0.057	0.148±0.026	2.21°±1.89°	19.31°±2.20°
Occupancy ↑	0.949±0.091	0.937±0.069	0.992±0.007	0.994±0.012	1.000±0.000	0.985±0.003	0.991±0.005	0.989±0.005

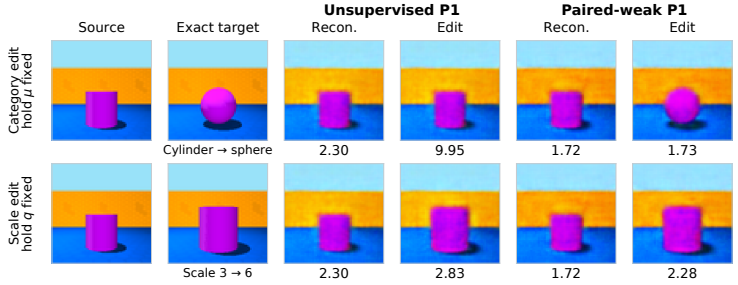

Figure 2: Categorical–continuous control on a selected 3DShapes example. Both P1 settings use the same architecture. Displayed MSE values are multiplied by 10^3.

4 Results

4.1 P1: Categorical–continuous specialization

We test whether each branch captures its intended factors. Under unsupervised P1, category prediction favours the continuous branch ($(\mu \to d) > (q \to d)$) on three of four image benchmarks. Paired-weak P1 improves mean category and geometry/pose gaps on all four, but worsens reconstruction on three and has mixed FVM effects (Table 4). Fixed-component interventions show stronger categorical control but mixed geometry effects (Table 12). Figure 2 shows a selected example, with each swap holding the other branch fixed. Categorical swaps (top) change the cylinder toward the target sphere under paired-weak P1 but largely retain it under unsupervised P1; continuous-code swaps (bottom) change scale in both settings. Edit MSEs quantify target fit.

4.2 P2: Physical actions, algebra, and decoded effects

Using the variants introduced under *P2 training strategies*, we test whether grounding and refinement improve physical-action agreement in latent and decoded spaces, alongside agreement with prescribed Lie relations. Table 5 reports the skeleton comparisons; Table 6 reports the image comparisons.

Grounded refinement versus Strict. The refined P2 variants achieve higher mean direction agreement than Strict across all four image benchmarks, improve prescribed nonzero-bracket agreement on dSprites and ModelNet10, and reduce decoded false gaps on 3DShapes and Cars3D. On both skeleton datasets, refinement also improves decoded order-gap direction and magnitude, alongside Lie-structure agreement, over Strict. Together, these results favour fitting prescribed action relations: suppressing all basis-pair commutators alone does not recover the intended physical effects.

Contribution of decoded rollout. Among the fully supervised strategies, P2b+R achieves the highest direction-agreement means on three of four image benchmarks. Compared with Core P2b, rollout lowers mean decoded endpoint error across all six datasets and improves every decoded-action metric mean on both skeleton datasets. Its benefits are not uniform: the decoded false gap decreases on 3DShapes but increases on Cars3D, where Core P2b remains best. P2a-only retains better direction agreement on ModelNet10, while Strict retains higher single-action cosine and slightly lower endpoint SSE on 3DPW, despite substantially weaker order-effect and algebraic agreement. To test whether

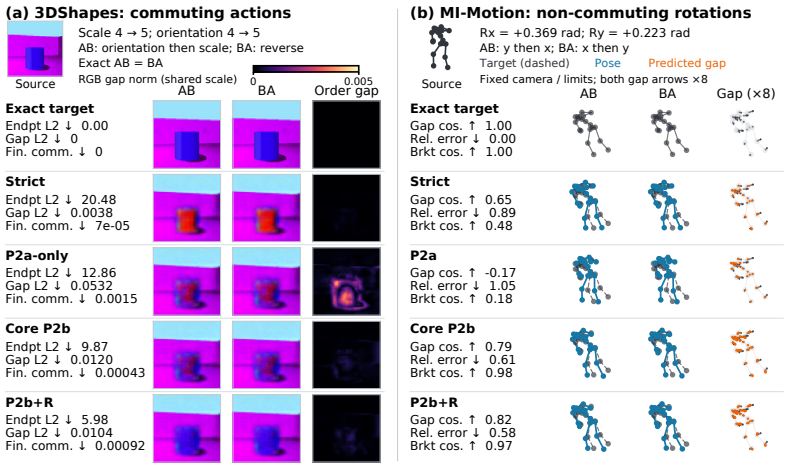

Figure 3: Selected commuting (3DShapes) and non-commuting (MI-Motion) action examples.Target and predicted skeleton gap arrows are both enlarged $8\times$.

Table 5: Skeleton P2 results (mean±sample SD; five seeds). Bold: best displayed fully grounded mean, including ties; underline: LF+64 matches or improves it (not significance).

	Fully supervised grounding				Few-shot grounding
Metric	P2a	Strict	Core P2b	P2b+R	LF+64
MI-Motion — non-Abelian $\mathrm{SO}(3)$					
Latent: Single-action cosine ↑	0.562±0.017	0.780±0.015	0.854±0.018	**0.860**±0.017	0.844±0.014
Decoded: Ordered endpoint SSE ↓	0.339±0.003	0.190±0.008	0.151±0.008	**0.139**±0.006	0.140±0.007
Decoded: AB/BA gap cosine ↑	−0.016±0.083	0.587±0.025	0.785±0.007	**0.802**±0.005	0.791±0.011
Decoded: Order assignment ↑	0.455±0.185	0.984±0.004	0.996±0.002	**0.997**±0.001	0.996±0.002
Decoded: Gap-magnitude error ↓	0.123±0.002	0.120±0.002	0.029±0.001	**0.028**±0.001	0.026±0.001
Lie: Cyclic closure ↓	1.089±0.079	0.863±0.023	**0.050**±0.008	**0.050**±0.008	0.062±0.007
Lie: Mean bracket cosine ↑	0.036±0.107	0.644±0.025	**0.977**±0.004	**0.977**±0.004	0.971±0.003
Lie: Mean norm ratio → 1	0.252±0.046	0.154±0.014	0.949±0.004	**0.950**±0.004	0.952±0.011
3DPW — non-Abelian $\mathrm{SO}(3)$					
Latent: Single-action cosine ↑	0.624±0.019	**0.651**±0.018	0.598±0.015	0.605±0.013	0.573±0.046
Decoded: Ordered endpoint SSE ↓	0.511±0.046	**0.500**±0.050	0.512±0.049	0.503±0.047	0.504±0.040
Decoded: AB/BA gap cosine ↑	−0.073±0.147	0.011±0.154	0.361±0.036	**0.375**±0.028	0.369±0.024
Decoded: Order assignment ↑	0.417±0.155	0.506±0.212	0.860±0.027	**0.875**±0.022	0.855±0.022
Decoded: Gap-magnitude error ↓	0.119±0.002	0.120±0.001	0.082±0.005	**0.079**±0.005	0.074±0.005
Lie: Cyclic closure ↓	1.031±0.036	1.042±0.048	0.012±0.006	**0.011**±0.005	0.011±0.003
Lie: Mean bracket cosine ↑	−0.127±0.127	−0.020±0.094	**0.996**±0.002	**0.996**±0.002	0.997±0.001
Lie: Mean norm ratio → 1	0.154±0.046	0.139±0.029	**0.961**±0.006	**0.961**±0.005	0.953±0.008

P2a uses 1,000 post-P1 updates; all final variants use 2,000. Strict retains physical-action targets and anchors but replaces nonzero-bracket losses with all-45 basis-pair commutativity. LF+64 uses 64 labeled anchors per action for grounding, followed by fully supervised refinement.

prescribed-bracket losses provide benefits beyond decoded rollout, we compare finite-only+R with Core P2b+R on dSprites, keeping rollout and the remaining training settings matched. Adding bracket losses improves mean algebraic and decoded order-effect agreement, but slightly increases mean endpoint error (Table 15; Appendix F.3). Figure 3 illustrates improved colour and endpoint fidelity on 3DShapes and recovery of the nonzero order gap on MI-Motion in selected examples.

LF+64 achieves the lowest reported closure on both non-commuting image benchmarks and the smallest decoded false gap and finite-action commutator on 3DShapes. Its direction agreement remains lower than P2b+R on dSprites, Cars3D and ModelNet10. For skeletons, it achieves endpoint errors close to P2b+R and lower gap-magnitude errors on both datasets, while retaining strong algebraic agreement. Thus, few-shot grounding retains competitive structural results, with dataset-dependent benefits for decoded action accuracy.

Representation quality. These action-level gains accompany broadly comparable mean reconstruction and FVM scores relative to Strict (Table 7). P2b+R achieves higher mean FVM than the evaluated cFASL and VAE baselines on all three applicable benchmarks. Its mean reconstruction error is lower than cFASL on all four image benchmarks and lower than the evaluated β-VAE configurations on 3DShapes and ModelNet10, although not on dSprites and Cars3D. Thus, improved physical-action and algebraic agreement coexists with competitive reconstruction and disentanglement, rather than requiring uniform improvement in every representation-quality metric.

Table 6: Image post-P1 comparisons (mean±sample SD; five seeds). Bold: best displayed mean among P2a-only, Core P2b and P2b+R; underline: LF+64 matches or improves it.

	Controls		Fully action-supervised			Reduced-label
Metric	P1-only	Strict	P2a-only	Core P2b	P2b+R	LF+64
dSprites — non-Abelian $\mathrm{SE}(2)$						
Latent: R, T_x order-gap cosine ↑	0.142±0.202	−0.139±0.219	0.413±0.242	0.507±0.280	**0.515**±0.284	0.218±0.229
Latent: R, T_y order-gap cosine ↑	0.045±0.161	0.052±0.222	0.537±0.079	0.614±0.099	**0.622**±0.097	0.372±0.229
Lie: Equal-term closure ↓	0.797±0.082	0.670±0.007	0.723±0.103	0.529±0.063	**0.523**±0.066	0.454±0.110
Lie: R, T_y bracket cosine ↑	−0.003±0.201	0.004±0.245	0.295±0.262	0.740±0.132	**0.752**±0.126	0.545±0.327
Lie: T_x, T_y relative bracket RMS ↓	0.580±0.195	0.048±0.015	**0.256**±0.104	0.268±0.089	0.269±0.088	0.110±0.063
3DShapes — Abelian control						
Decoded: Orientation cosine ↑	−0.057±0.077	−0.058±0.086	0.016±0.131	0.445±0.096	**0.481**±0.082	0.503±0.139
Decoded: False gap ↓ ($\times 10^{-2}$)	8.517±11.909	1.607±0.747	2.173±0.977	1.457±0.109	**1.429**±0.075	1.074±0.072
Lie: Finite-action commutator ↓ ($\times 10^{-3}$)	2.483±3.583	0.308±0.307	**0.839**±0.533	0.857±0.395	1.021±0.181	0.067±0.042
Cars3D — Abelian control						
Latent: Elevation cosine ↑	0.046±0.167	0.003±0.188	0.299±0.225	0.313±0.227	**0.366**±0.189	0.311±0.113
Latent: Azimuth cosine ↑	−0.034±0.041	−0.028±0.044	−0.025±0.034	0.019±0.049	**0.066**±0.043	0.058±0.017
Decoded: Elevation cosine ↑	0.022±0.123	−0.007±0.127	0.205±0.178	0.224±0.176	**0.264**±0.149	0.263±0.092
Decoded: Azimuth cosine ↑	−0.007±0.017	−0.007±0.016	0.003±0.021	0.045±0.029	**0.077**±0.023	0.060±0.031
Decoded: False gap ↓ ($\times 10^{-2}$)	4.483±4.737	1.448±1.403	3.259±3.162	**0.568**±0.590	0.871±0.949	1.059±0.499
Lie: Finite-action commutator ↓ ($\times 10^{-3}$)	3.471±3.799	1.206±1.098	2.430±2.242	**0.551**±0.534	0.769±0.763	0.613±0.281
ModelNet10 — non-Abelian $\mathrm{SO}(3)$						
Latent: Mean single-action cosine ↑ ($\times 10^{-2}$)	−0.246±0.747	−0.208±0.774	**1.129**±0.563	0.707±0.432	0.741±0.424	0.622±1.355
Latent: Mean order-gap cosine ↑ ($\times 10^{-2}$)	0.317±0.439	0.324±0.479	**1.934**±0.249	1.524±0.388	1.520±0.385	0.770±0.821
Decoded: Order-gap cosine ↑ ($\times 10^{-2}$)	−0.230±0.913	−0.197±1.391	**3.889**±0.332	3.031±0.534	3.032±0.538	0.552±0.971
Lie: Cyclic closure ↓	1.047±0.033	1.008±0.007	1.059±0.027	**0.883**±0.081	**0.883**±0.080	0.548±0.092
Lie: Mean bracket cosine ↑	0.012±0.017	0.006±0.028	−0.010±0.024	**0.324**±0.122	**0.324**±0.122	0.396±0.065
Lie: Mean norm ratio → 1	0.207±0.067	0.079±0.034	0.233±0.032	**0.320**±0.057	**0.320**±0.057	0.461±0.069

Common seed-specific P1 parents. P1-only: representation continuation; Strict: all-45 basis-pair commutativity without physical-action targets; P2a-only: full-budget grounding. Post-P1 updates: dSprites 12,960; 3DShapes 5,064; Cars3D 1,240; ModelNet10 2,880 (1,440 for P1-only/Strict). Compute, supervision and trainable parameter counts are not matched.

Table 7: Reconstruction (↓) and FVM (↑) on common evaluation records (mean± SD; five seeds).

	External continuations		Internal variants					
Dataset / metric	cFASL	β-VAE†	P1-only	Strict	P2a-only	Core P2b	P2b+R	LF+64
dSprites / BCE ↓	30.16±2.66	**17.28**±1.05	20.81±2.95	19.72±2.80	22.06±3.67	20.22±3.36	20.22±3.36	28.04±8.57
FVM ↑	0.653±0.093	0.523±0.094	0.859±0.068	**0.862**±0.068	0.860±0.066	0.860±0.066	0.860±0.066	0.860±0.066
3DShapes / SSE ↓	97.89±8.36	28.98±2.03	**27.19**±3.43	27.85±6.05	29.07±5.54	27.90±4.44	28.27±4.59	51.44±23.86
FVM ↑	0.664±0.102	0.672±0.086	0.795±0.092	**0.797**±0.106	**0.797**±0.094	**0.797**±0.094	**0.797**±0.094	**0.797**±0.094
Cars3D / SSE ↓	64.50±1.67	**46.03**±1.07	57.26±1.47	57.72±1.27	57.45±1.34	56.74±1.46	57.11±1.68	67.62±7.53
FVM ↑	0.908±0.084	0.922±0.034	**0.946**±0.050	0.941±0.051	**0.946**±0.051	**0.946**±0.051	**0.946**±0.051	**0.946**±0.051
ModelNet10 / SSE ↓	154.77±62.97	110.56±5.63	89.02±4.13	90.50±4.09	**88.61**±4.61	90.25±2.90	90.26±2.87	100.19±6.34

5 CONCLUSION

We introduced a two-phase framework that separates representation learning from physical-action grounding and algebraic refinement. By evaluating latent actions, Lie structure and decoded observations together, the framework distinguishes algebraic agreement from accurate physical effects. P1 audits show that paired-weak supervision strengthens categorical–continuous specialization, with trade-offs in reconstruction and disentanglement. These findings emphasize that separate branches provide an inductive bias whose semantic roles must be evaluated, rather than assumed. In P2, grounded refinement improves recovery of prescribed nonzero brackets and physical order effects relative to commutativity-based controls. Frozen-decoder rollout further reduces mean decoded endpoint error relative to Core P2b across all six datasets, although improvements are not uniform across metrics. These action-level gains coexist with competitive reconstruction and FactorVAE scores relative to the internal Strict control and evaluated external baselines. Few-shot grounding also retains competitive structural agreement, with dataset-dependent decoded benefits. The central takeaway is that useful latent actions must reproduce the appropriate composition laws and their observable consequences: neither suppressing all commutators nor allowing non-commutativity alone is sufficient. Our results advance this objective in both commuting and non-commuting settings, while leaving room to strengthen semantic specialization, improve action-magnitude recovery and achieve more consistent decoded accuracy with fewer action labels.

REFERENCES

Georgios Arvanitidis, Lars Kai Hansen, and Soren Hauberg. Latent space oddity: On the curvature of deep generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJzRZ-WCZ.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Hugo Caselles-Dupré, Michael Garcia-Ortiz, and David Filliat. Symmetry-based disentangled representation learning requires interaction with environments. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/36e729ec173b94133d8fa552e4029f8b-Abstract.html.

Clément Chadebec, Clément Mantoux, and Stéphanie Allassonnière. Geometry-aware hamiltonian variational auto-encoder. *HAL Open Science*, 2020. https://hal.science/hal-02974244.

Ho Yin Chau, Frank Qiu, Yubei Chen, and Bruno A. Olshausen. Disentangling images with lie group transformations and sparse coding. *arXiv preprint arXiv:2012.12071*, 2020. URL https://arxiv.org/abs/2012.12071.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Yuxuan Chen, Jung Yeon Park, Floor Eijkelboom, Jianke Yang, Jan-Willem van de Meent, Lawson L. S. Wong, and Robin Walters. Discovering symmetry groups with flow matching. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2512.20043.

Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2008. doi: 10.1137/1.9780898717778.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Yinzhu Jin, Aman Shrivastava, and P. Thomas Fletcher. Learning group actions on latent representations. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e63309e532688c722177f81e99f94f32-Abstract-Conference.html.

Heeseung Jung, Jaehyoung Jeong, and Kangil Kim. CFASL: Composite factor-aligned symmetry learning for disentanglement in variational autoencoder. *Transactions on Machine Learning Research*, 2024. URL https://api.semanticscholar.org/CorpusID:267028511.

Hamza Keurti, Hsiao-Ru Pan, Michel Besserve, Benjamin F. Grewe, and Bernhard Schölkopf. Homomorphism autoencoder—learning group structured representations from observed transitions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 16190–16215. PMLR, 2023. URL https://proceedings.mlr.press/v202/keurti23a.html.

Asif Khan and Amos Storkey. Adversarial robustness of VAEs through the lens of local geometry. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 8954–8967. PMLR, 2023.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Scholkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4114–4124. PMLR, 2019.

Matthew Painter, Jonathon Hare, and Adam Prügel-Bennett. Linear disentangled representations and unsupervised action estimation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2020.

X. Peng, X. Zhou, Y. Luo, H. Wen, Y. Ding, and Z. Wu. The MI-Motion dataset and benchmark for 3d multi-person motion prediction. *arXiv preprint arXiv:2306.13566*, 2023. URL https://arxiv.org/abs/2306.13566.

Robin Quessard, Thomas D. Barrett, and William R. Clements. Learning disentangled representations and group structure of dynamical environments. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2020.

Scott E. Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Timo von Marcard, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using IMUs and a moving camera. In *European Conference on Computer Vision*, 2018. URL https://virtualhumans.mpi-inf.mpg.de/3DPW/.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, 2015.

Jianke Yang, Robin Walters, Nima Dehmamy, and Rose Yu. Generative adversarial symmetry discovery. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 39488–39508. PMLR, 2023. URL https://proceedings.mlr.press/v202/yang23n.html.

Jianke Yang, Nima Dehmamy, Robin Walters, and Rose Yu. Latent space symmetry discovery. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56047–56070. PMLR, 2024. URL https://proceedings.mlr.press/v235/yang24g.html.

Johannes Zacherl, Philipp Frank, and Torsten A. Ensslin. Probabilistic autoencoder using fisher information. *Entropy*, 23, 2021. URL https://api.semanticscholar.org/CorpusID:240070326.

Yijia Zheng, Tong He, Yixuan Qiu, and David P. Wipf. Learning manifold dimensions with conditional variational autoencoders. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/e04101138a3c94544760c1dbdf2c7a2d-Abstract-Conference.html.

Xinqi Zhu, Chang Xu, and Dacheng Tao. A commutative lie group vae for disentanglement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12924–12934. PMLR, 2021.

A NOTATION

Appendix B gives P1 implementation details; Appendix C specifies P2 losses and gradient routes. Metrics, evaluation records and statistical conventions are collected in Appendix D, and datasets and training settings in Appendix E. Appendix F reports supplementary results, including the supervision-matched P2b loss-component ablation.

Symbol	Meaning
K, d, m	categorical states, continuous coordinates/Lie bases, and carrier dimension (matrix side length)
$x, \hat{x}, x^\star$	source observation, reconstruction, and exact transformed target
$q_\phi, q(x), c$	categorical posterior, its probability vector for observation x, and relaxed Gumbel–Softmax sample
μ, σ, t	continuous posterior mean, standard deviation, and sampled coordinates
$E, P, s(c), s_0$	image codebook, projection, categorical carrier, and shared skeleton carrier
$A_k, G(t)$	P1 Lie basis and $\exp(\sum_k t_k A_k)$
$\widehat{Z}, S_n^0, S_{\omega,n}$	encoded group feature, frozen P2 source state, and state after action word ω; S is a matrix for images and a vector for skeletons
$h(Z, c)$	acted carrier $Zs(c)$
δ, D_{img}	observation-valued decoder; $D_{\text{img}}(Z, c) = \delta(Zs(c))$ on images (including sigmoid)
$Q, o_{\text{rec}}, o^{\text{MI}}$	training-only code predictor and raw image-decoder logits for reconstruction/code recovery
$\hat{x}_{\text{rec}}$	observation reconstructed through the active P1 training route
$0, \text{ref}$	superscripts for the frozen P1 parent and the recorded continuation reference
c_n in P2	protocol-fixed carrier weights; not necessarily a sampled Gumbel vector
sg	stop-gradient (detach) operator
$v_a, B_a, H_a(\gamma)$	named direction, generator $\sum_k v_{a,k} A_k$, and $\exp(\gamma B_a)$
c^g_{ab}	prescribed action-algebra coefficients, distinct from categorical weights c_n
$\mathcal{A}, \mathcal{R}, \mathcal{W}_n$	named actions, prescribed relations, and available action words
$\Phi_{\text{lat}}, \Phi_{\text{obs}}$	latent-state and decoded-observation-displacement endpoint maps
$\mathcal{D}^0, w_\omega, \nu_\omega$	frozen observation map and fixed endpoint/rollout word weights
$y^\star_{\omega,n}$	exact target for word ω in the selected endpoint space
$r_{ab}, u_{ab}, u^{\text{ref}}_{ab}$	live bracket, live algebra target, and detached fixed-parent target response
$d_{ab}(S), \eta_{ab}$	positive closure denominator and response scale for commuting RMS
$\mathcal{L}_{\text{roll}}$	frozen-decoder observation loss on one-step and ordered two-step targets
$D^{\text{BCH},0}, d^0, \widetilde{\Delta}^\theta$	P1 operator, acted-carrier, and decoded order diagnostics

B P1: OBJECTIVES AND IMPLEMENTATION DETAILS

The encoder–carrier–decoder path is defined in Section 2.1. This appendix expands its grouped image objective, then specifies the category-free and paired-weak variants and the training schedules. Evaluation probes and interventions are described separately in Appendix D.

B.1 IMAGE LOSS GROUPS AND COMPONENT DEFINITIONS

The following definitions implement the four groups in Equation (3), expanded in Table 8. Expectations average over the minibatch; sg denotes stop-gradient.

Reconstruction and feature matching ($\mathcal{L}_{\text{fit}}$). The fit losses are defined in Equation (2). Let o_{rec} denote raw image-decoder logits, so the reconstructed observation is $\hat{x}_{\text{rec}} = \text{sigmoid}(o_{\text{rec}})$. The implementation evaluates BCE using o_{rec} on dSprites; the other image datasets use SSE between $\hat{x}_{\text{rec}}$ and the input. Both sum over image entries before averaging over observations; feature matching sums over matrix entries. In contrast, the skeleton feature term in Equation (14) additionally divides by m^2. The selected feature coefficient $\alpha = \texttt{hy_rec}$ is 1.0 for dSprites, 3DShapes and ModelNet10, and 0.7 for Cars3D.

Table 8: Expanded P1 loss groups. Coefficients follow the selected dataset-specific schedules; a zero coefficient omits a term.

Group / definition	Weighted terms
Fit (2)	$\mathcal{L}_{\text{fit}} = \mathcal{L}_{\text{rec}} + \alpha\mathcal{L}_{\text{feat}}$
Prior (10)	$\mathcal{L}_{\text{prior}} = \beta\mathcal{L}_{\text{KL}}$
Code use (11, 12)	$\mathcal{L}_{\text{code}} = \lambda_{\text{MI}}\mathcal{L}_{\text{MI}} + \lambda_{\text{usage}}\mathcal{L}_{\text{usage}} + \lambda_{\text{conf}}\mathcal{L}_{\text{conf}} + \lambda_{\text{sep}}\mathcal{L}_{\text{sep}}$
Leakage (13)	$\mathcal{L}_{\text{leak}} = \lambda_{q\mu}\mathcal{L}_{q\mu}$

Posterior capacity ($\mathcal{L}_{\text{prior}}$). For $\mu_b := \mu(x_b)$ and componentwise log variance $\ell_b := \log\big(\sigma(x_b)^2\big)$, the image implementation uses

$$\begin{aligned} \text{KL}_b &= \tfrac{1}{2}\sum_j \big(\mu_{b,j}^2 + \exp(\ell_{b,j}) - 1 - \ell_{b,j}\big), \\ \mathcal{L}_{\text{KL}} &= a_{\text{ann}}^{j_{\text{opt}}} \left|\mathbb{E}_b \text{KL}_b - C_{j_{\text{opt}}}\right|, \\ C_{j_{\text{opt}}} &= \min(C_{\max}, j_{\text{opt}} C_{\max}/J_{\text{lead}}). \end{aligned} \tag{10}$$

Here j_{opt} is the optimization step, a_{ann} the configured annealing factor, and J_{lead} the capacity-ramp duration. These symbols are distinct from the carrier s and P2 source state S. Table 8 multiplies this penalty by β. Skeletons instead use the ordinary mean Gaussian KL, without the capacity target or annealing factor (Equation (14)).

Categorical-code use ($\mathcal{L}_{\text{code}}$). For a minibatch of B observations, write $q_b := q(x_b)$, $q_b(k) = q_\phi(k \mid x_b)$, and $\bar{q} = B^{-1}\sum_b q_b$. Let u be the uniform categorical distribution, with $u_k = 1/K$. The recovery, usage and confidence terms are

$$\mathcal{L}_{\text{MI}} = -\mathbb{E}_b \sum_k q_b(k) \log Q(k \mid o_b^{\text{MI}}), \quad \mathcal{L}_{\text{usage}} = \text{KL}(u\|\bar{q}), \quad \mathcal{L}_{\text{conf}} = \log K - \mathbb{E}_b H(q_b). \tag{11}$$

Here $H(q) = -\sum_k q(k)\log q(k)$, and Q is an auxiliary predictor applied to the exponential-map decoder logits o^{MI}. Its target is the soft encoder posterior, not a semantic label or a hard class target. This is an InfoGAN-inspired recovery loss (Chen et al., 2016), not a claim that the full objective is the original InfoGAN bound. Usage discourages unused codes; minimizing the positive confidence penalty discourages overconfident individual posteriors. For the embedding-separation term, let $\widetilde{E}$ contain the row-normalized categorical embeddings:

$$\mathcal{L}_{\text{sep}} = K^{-2}\|\widetilde{E}\widetilde{E}^\top - I_K\|_F^2. \tag{12}$$

This separates embedding rows, not semantic classes. These competing regularizers do not guarantee informative categories or semantic separation; aggregate occupancy and per-sample entropy are audited separately.

Leakage control ($\mathcal{L}_{\text{leak}}$). The batch matrices $q \in \mathbb{R}^{B\times K}$ and $\mu \in \mathbb{R}^{B\times d}$ stack the rows $q(x_b)^\top$ and $\mu(x_b)^\top$, respectively. Each column is centered and divided by its batch root-mean-square deviation (with 10^{-6} added to the variance). Their empirical cross-correlation $C_{q\mu}$ gives

$$\mathcal{L}_{q\mu} = (Kd)^{-1}\|C_{\text{sg}(q),\mu}\|_F^2. \tag{13}$$

The squared entries are averaged over all Kd combinations. Detaching q directs this penalty to the continuous encoder, so the categorical branch cannot reduce it by collapsing its posterior.

B.2 Category-free and paired-weak objective variants

The image components above define $\mathcal{L}_{\text{P1}}^{\text{unsup}}$ in Equation (3). The following variants retain that notation and are reported separately.

Skeletons: reconstruction and prior only. MI-Motion and 3DPW provide 3D joint coordinates, without an intended categorical factor in this experiment. We omit the discrete encoder and use one

learned carrier $s_0 \in \mathbb{R}^{16}$: $h(Z) = Zs_0$, $\hat{x} = \delta(G(t)s_0)$. Their P1 objective is

$$\mathcal{L}_{\mathrm{P1}}^{\mathrm{skel}} = \underbrace{\mathcal{L}_{\mathrm{rec}} + 0.1\mathcal{L}_{\mathrm{feat}}^{\mathrm{skel}}}_{\text{reconstruction}} + \underbrace{10^{-3}\mathcal{L}_{\mathrm{KL}}^{\mathrm{std}}}_{\text{posterior regularity}} . \tag{14}$$

Here $\mathcal{L}_{\mathrm{rec}}$ is coordinate SSE per skeleton, $\mathcal{L}_{\mathrm{feat}}^{\mathrm{skel}} = m^{-2}\mathbb{E}\|\widehat{Z} - G(t)\|_F^2$, and $\mathcal{L}_{\mathrm{KL}}^{\mathrm{std}}$ is the ordinary Gaussian KL to $\mathcal{N}(0, I)$, without a capacity schedule. There is no categorical penalty or auxiliary $\widehat{Z}$-decoding route for skeletons.

Paired weak-supervision extension. When relational metadata are available, a separately reported variant uses

$$\mathcal{L}_{\mathrm{P1}}^{\mathrm{pair}} = \mathcal{L}_{\mathrm{P1}}^{\mathrm{unsup}} + \underbrace{\lambda_{q\text{-cons}}\mathcal{L}_{q\text{-cons}} + \lambda_{\mathrm{equiv}}\mathcal{L}_{\mathrm{equiv}}}_{\text{same category; different action state}} + \underbrace{\lambda_{\mu\text{-inv}}\mathcal{L}_{\mu\text{-inv}} + \lambda_{\mathrm{MI}}^{\mathrm{sched}}\mathcal{L}_{\mathrm{swap}}}_{\text{matched action state; different category}} . \tag{15}$$

Action state means the dataset-specific continuous configuration. For a same-category/different-action-state pair (x, x'), $\mathcal{L}_{q\text{-cons}}$ is the symmetric Jensen–Shannon divergence between $q_\phi(\cdot \mid x)$ and $q_\phi(\cdot \mid x')$, while $\mathcal{L}_{\mathrm{equiv}}$ predicts the known relative action from $\mu(x') - \mu(x)$ with a small head. For an action-matched/different-category pair, $\mathcal{L}_{\mu\text{-inv}}$ matches the two continuous means and $\mathcal{L}_{\mathrm{swap}}$ exchanges their categorical carriers while reconstructing the paired targets. The swap term follows the active MI schedule, and unavailable relations are omitted. Action state denotes rendered geometry for dSprites, scale–orientation for 3DShapes, elevation–azimuth for Cars3D, and global $SO(3)$ orientation for ModelNet10. Matching is exact on the factorial benchmarks and uses the recorded rotation tolerance on ModelNet10; relative-action targets can be periodic or matrix-valued. These relations supply semantic side information even though they are not classifier targets, so this variant is weakly supervised and is never merged with the unsupervised result.

B.3 Sampling, reconstruction routes, and schedules

The auxiliary route described in Section 2.1 uses $\widehat{Z}$ with probability `forward_eg_prob` $= 0.2$, and $G(t)$ otherwise. Both act on the categorical carrier before decoding. The auxiliary $\widehat{Z}$-route detaches both the categorical sample and the resulting carrier $s(c)$, blocking gradients to the categorical encoder, embedding table and carrier projection on that route. The main route follows the selected snapshot's categorical-gradient schedule. Code recovery and evaluation reconstruction always use the exponential-map route. This is the bottleneck feature-sharing implementation inherited from Zhu et al. (2021), with our carrier retained on both routes.

The selected image runs use soft Gumbel–Softmax weights (`hard=False`); dataset-specific temperatures are recorded in Table 9. On the main image reconstruction route, the complete projected carrier is detached during the first 3,000 optimization steps and released thereafter. Code recovery uses its own undetached exponential-map route.

Active schedules (image benchmarks). Categorical MI is off for 3,000 steps, ramps linearly to its configured weight by step 8,000, and is then fixed. Confidence and embedding separation are inactive for the first 3,000 steps. Aggregate usage uses multipliers $10, 5, 2$ before step 3,000, from 3,000 to 8,000, and thereafter. The cross-branch penalty begins at its configured separation-start step (8,000 in the selected image runs). Zero coefficients omit terms, as in Table 8; dataset coefficients and training settings are listed in Appendix E.1 and the saved code snapshots.

C P2 objectives and supervision

This appendix specifies the losses underlying the concise P2 method: P2a endpoint alignment, P2b algebra and decoded-action refinement, the category-free skeleton specialization, and the reduced-label image extension. P1 objectives are in Appendix B; diagnostics and evaluation metrics are in Appendix D.

C.1 P2a: action states, targets, and objective

Shared action notation; dataset-specific states. P2 uses a frozen source state S_n^0. It is the encoded matrix $\mathrm{sg}[\widehat{Z}^0(x_n)]$ on the image benchmarks, but the acted vector $\mathrm{sg}[G^0(\mu^0(x_n))s_0^0]$ on

skeletons. For images, c_n denotes the fixed carrier weights used by the selected protocol; it is omitted for skeletons. Define the frozen observation map $\mathcal{D}^0(S, c) = D^0_{\mathrm{img}}(S, c)$ for matrix states and $\mathcal{D}^0(S) = \delta^0(S)$ for skeleton vectors. This distinction avoids treating a skeleton carrier vector as an encoded image matrix.

Actions and the rightmost-first convention follow Section 2.2. P2a uses the frozen P1 bases; P2b updates action bases while keeping the source-state bases and carrier fixed. Predictions and targets use the same transformation parameters. Fixed increments may equivalently be absorbed into B_a.

P2a: match actions and their order. The endpoint map is either the latent state or the decoded displacement:

$$\Phi_{\mathrm{lat}}(S, c) = S, \qquad \Phi_{\mathrm{obs}}(S, c) = \mathcal{D}^0(S, c) - \mathcal{D}^0(S^0_n, c). \tag{16}$$

Targets $y^\star_{\omega,n}$ are frozen encodings of transformed observations for latent matching, or exact observation displacements $x^\star_{\omega,n} - x_n$ for decoded matching. The skeleton latent target uses the frozen P1 bases and the same shared carrier as its source. The available single/ordered sequences $\mathcal{W}_n$ enter the endpoint and signed-gap losses defined in Equation (5). The fixed weights w_ω and full-vector squared-error normalization are dataset-specific. Image P2a uses latent targets except on ModelNet10, which uses decoded displacements; skeleton P2a uses latent carrier-vector targets. The target gap is zero for a *prescribed commuting action pair*, not because the current learned bracket happens to vanish. ℓ^Φ_{ord} uses squared full-gap error, with relative target normalization on ModelNet10 and the fixed pair weights in its configuration. Only the action-direction rows are updated (20 or 30 scalars for the selected action sets). The ModelNet10-only $\mathcal{L}_{\mathrm{guard}}$ in Equation (6) is defined below.

Dataset-specific endpoint reductions. For latent image matching (dSprites, 3DShapes and Cars3D), $\ell_\Phi(p, y) = m^{-2}\|p - y\|^2_F$: the implementation averages over matrix entries. Category-free skeleton endpoints instead use $\|p - y\|^2_2$, summed over vector entries. The selected ModelNet10 route uses decoded displacements and the relative full-vector error $\rho(p, y) = \|p - y\|^2_2/(\|y\|^2_2 + \epsilon)$. Its target is the observed displacement, not a latent encoding.

dSprites directly fits both ordered endpoints and their signed matrix-valued gap on valid interior sources. Cars3D fits both orders to the same encoded target and penalizes a nonzero gap. 3DShapes fits exact single-factor scale and orientation endpoints, with separate means over active groups; its commuting loss uses mixed coefficients constructed from the minibatch. Thus $\mathcal{W}_n$ means the available endpoints of that protocol, not a universally active sum of every single and ordered target. Fixed weights and calibration choices are recorded in Table 11. All endpoint and gap losses compare complete vectors or matrices rather than scalar norms.

ModelNet10 P2a anti-contraction guards. Let v^{ref} be the fixed initialization used for P2a calibration and $B^{\mathrm{ref}}_a = \sum_k v^{\mathrm{ref}}_{a,k} A^0_k$. Define $\psi(p, q) = [\max(0, \log((q + \epsilon)/(p + \epsilon)))]^2$. Then

$$\mathcal{L}_{\mathrm{guard}} = \lambda_{\mathrm{rf}} \mathbb{E}_{n,a} \psi(\|B_a S^0_n\|_F, \|B^{\mathrm{ref}}_a S^0_n\|_F) + \lambda_{\mathrm{vf}} \mathbb{E}_a \psi(\|v_a\|_2, \|v^{\mathrm{ref}}_a\|_2).$$

This guard is used only on ModelNet10 and is zero for all other datasets. It penalizes response and direction norms below their reference values; it does not supply an additional target action.

C.2 P2b: grouped objective and image-specific terms

C.2.1 Closure and nonzero-bracket scale

P2b retains the P2a targets and trains current action generators $B_a = \sum_k v_{a,k} A_k$. For direction-only gradient routes, write

$$B^{\mathrm{dir}}_a = \sum_k v_{a,k}\, \mathrm{sg}(A_k).$$

This has the same forward value as B_a, but its losses update only the direction rows. The frozen bases used to construct source states are separate from these current action bases.

For each prescribed relation $(a, b) \in \mathcal{R}$, with known structure constants c^g_{ab}, define

$$r_{ab}(S) = [B_a, B_b]S, \qquad u_{ab}(S) = \sum_g c^g_{ab} B_g S. \tag{17}$$

For a commuting relation, $u_{ab} = 0$. Equation (8) averages normalized full-response errors. The denominator is specified by dataset, not by a universal detachment or stabilization convention:

- On dSprites, the three denominators are

$$\|B^{\mathrm{ref}}_{T_y} S\|_F^2 + \epsilon, \qquad \|B^{\mathrm{ref}}_{T_x} S\|_F^2 + \epsilon, \qquad \tfrac{1}{2}(\|B^{\mathrm{ref}}_{T_x} S\|_F^2 + \|B^{\mathrm{ref}}_{T_y} S\|_F^2) + \epsilon.$$

  The numerator's nonzero targets remain live; only these reference normalizations are frozen. References refresh at the recorded continuation boundary.
- On ModelNet10, $d_{ab}(S) = \|B_g S\|_F^2 + \epsilon$ for the cyclic target g, with gradients through the target and denominator.
- On the Abelian image controls, $d_{ab}(S) = \frac{1}{2}(\|B^{\mathrm{ref}}_a S\|_F^2 + \|B^{\mathrm{ref}}_b S\|_F^2) + \epsilon$, and $u_{ab} = 0$.
- On skeletons, $d_{ab}(S) = \max(\|B_g S\|_2, \epsilon)^2$, again with a live cyclic target.

The closure stabilizer is 10^{-8} in these implementations. The skeleton clamp-then-square convention is not the same formula as adding ϵ to a squared norm.

On dSprites and ModelNet10, additional nonzero-bracket terms compare $r = r_{ab}(S)$ with the detached semantic response $u^{\mathrm{ref}} = \mathrm{sg}[\sum_g c^g_{ab} B^{\mathrm{ref}}_g S]$ of the recorded reference parent (P2a, or the preceding refinement link where configured):

$$\begin{aligned}
\mathcal{L}_{\mathrm{logmag}} &= \mathbb{E}\left[\left(\log \frac{\|r\|_F + \epsilon}{\|u^{\mathrm{ref}}\|_F + \epsilon}\right)^2\right], \\
\mathcal{L}_{\mathrm{proj}} &= \mathbb{E}\left[\left(\frac{\langle r, u^{\mathrm{ref}}\rangle_F}{\|u^{\mathrm{ref}}\|_F^2 + \epsilon} - 1\right)^2\right], \\
\mathcal{L}_{\mathrm{scale}} &= \lambda_{\mathrm{logmag}} \mathcal{L}_{\mathrm{logmag}} + \lambda_{\mathrm{proj}} \mathcal{L}_{\mathrm{proj}}.
\end{aligned} \tag{18}$$

For the dSprites component runs, the implemented log-magnitude stabilizer replaces the squared log ratio above by

$$\left[\log \max\left(\frac{\|r\|_F}{\|u^{\mathrm{ref}}\|_F + 10^{-8}}, 10^{-6}\right)\right]^2.$$

The ratio, not the individual norms, is clamped; no epsilon is added to the numerator. References come from the immediate continuation parent. Log magnitude targets response size; signed projection targets amplitude along the required direction. Closure also penalizes components in the wrong direction. For skeletons, the selected scale term instead penalizes the log of the mean bracket norm ratio for each cyclic relation; finite-response cosine and log-norm terms anchor the P2a action (Appendix C.3). Thus the main endpoint and closure losses are samplewise, but not every auxiliary term uses the same reduction across datasets.

C.2.2 Decoded displacement matching

For the Abelian image controls, zero closure alone cannot identify what the actions mean. We therefore match decoded displacements separately for each action. Let $\mathcal{N}_a$ contain minibatch records with a nonzero amount of action a. Define

$$L_{\mathrm{disp},a} = \frac{1}{|\mathcal{N}_a|} \sum_{n \in \mathcal{N}_a} \frac{\left\| \mathcal{D}^0(e^{\gamma_n B^{\mathrm{dir}}_a} S^0_n, c_n) - \mathcal{D}^0(S^0_n, c_n) - (x^\star_{a,n} - x_n) \right\|_2^2}{\|x^\star_{a,n} - x_n\|_2^2 + 10^{-8}}. \tag{19}$$

Here γ_n is the supplied single-action parameter, $c_n = q^0(x_n)$, and $B^{\mathrm{dir}}_a = \sum_k v_{a,k}\, \mathrm{sg}(A_k)$. Detaching the bases restricts this loss to updating direction rows through the frozen decoder. An absent action contributes zero. The objective sums the separately weighted action means, $\sum_a \lambda_{\mathrm{disp},a} L_{\mathrm{disp},a}$, without dividing by $|\mathcal{A}|$.

In reduced-label ModelNet10, the corresponding decoded-semantic term is also active; its normalization and scope are specified in Appendix C.4.

C.2.3 FROZEN-DECODER ROLLOUT

The HAE-inspired rollout (Keurti et al., 2023) compares complete decoded observations after one or two actions with their targets:

$$\mathcal{L}_{\mathrm{roll}} = \mathbb{E}_n \sum_{\omega \in \mathcal{W}_n^{\mathrm{roll}}} \nu_\omega \, \ell_{\mathrm{obs}}\big(\mathcal{D}^0(S_{\omega,n}, c_n), x^\star_{\omega,n}\big) \, . \tag{20}$$

The endpoint set and weights are protocol-specific. Ordered states use current action generators on the frozen source state; the observation map is the frozen decoder defined in Appendix C.1. The observation loss is per-image summed BCE on dSprites, pixel SSE on the other image benchmarks, and coordinate SSE on skeletons. In these observation-space equations, image decoding includes the final sigmoid. Numerically, dSprites evaluates BCE using the corresponding pre-sigmoid logits; pixel SSE on the other image datasets compares sigmoid outputs with observations. These targets come from the pipeline's existing action records; rollout does not introduce a separately trained HAE or an additional decoder. Gradients pass through the frozen observation map to direction rows and action bases, without updating decoder parameters or buffers.

For the fully action-supervised image pipeline, dSprites averages twelve endpoints (four for each of three action pairs) on valid interior sources; 3DShapes and Cars3D average four endpoints; ModelNet10 averages nine (three single and six ordered). Skeletons separately average single and ordered coordinate errors as defined in Appendix C.3. Reduced-label endpoint sets and weights follow Appendix C.4, rather than being assumed identical to the fully action-supervised pipeline.

C.2.4 FINITE-ACTION RETENTION AND GRADIENT ROUTES

In Equation (9), $\mathcal{L}_{\mathrm{P2a}}$ denotes retained grounding terms evaluated with the current actions and the continuation's specified reference and gradient conventions. $\mathcal{L}_{\mathrm{fin}}$ groups additional or reweighted finite-action penalties, as specified for each dataset below.

In all four fully action-supervised image pipelines, retained P2a losses detach the current action bases and update direction rows only. The Abelian-image displacement loss $\mathcal{L}_{\mathrm{disp}}$ also updates only directions. Closure, nonzero-bracket scale losses, and rollout can update both directions and bases. The skeleton retained endpoint losses likewise use live action bases. These gradient stops do not replace current action bases by the frozen P1 bases used to construct source states.

dSprites. In addition to retained, direction-only P2a, P2b uses live-basis relative latent endpoint terms. Write $Z^\star_\omega = \widehat{Z}^0(x^\star_\omega)$ for the frozen encoding of the transformed target, with the record index suppressed:

$$L_{\mathrm{hom1}} = \mathbb{E}_{\mathrm{single}} \frac{\|S_\omega - Z^\star_\omega\|_F^2}{\|Z^\star_\omega - S^0\|_F^2 + \epsilon}, \qquad L_{\mathrm{hom2}} = \mathbb{E}_{\mathrm{ordered}} \frac{\|S_\omega - Z^\star_\omega\|_F^2}{\|Z^\star_\omega - S^0\|_F^2 + \epsilon}. \tag{21}$$

The averages use valid interior sources and cover six single and six ordered endpoints across the three action pairs. Their selected weights are 4 and 3. Single endpoints enter here even though dSprites P2a directly fitted ordered endpoints. Some ordered targets are intentionally used again under a different, displacement-relative normalization and a live-basis gradient path. They should not be described as wholly new targets or as the same error counted under an unexplained alias.

ModelNet10. Let $\mathcal{L}^{\mathrm{MN}}_{\mathrm{P2a}}$ denote the ModelNet10 specialization of $\mathcal{L}_{\mathrm{P2a}}$, and let $L^{\mathrm{obs}}_{\mathrm{ordered}}$ be the mean of its six relative ordered decoded-endpoint errors. The selected final retention gives these ordered errors weight 8 rather than 1:

$$\mathcal{L}^{\mathrm{MN}}_{\mathrm{retain}} = \mathcal{L}^{\mathrm{MN}}_{\mathrm{P2a}} + 7L^{\mathrm{obs}}_{\mathrm{ordered}}.$$

The calibrated signed-gap weight is retained without an additional rescaling. There is no separate dSprites-style relative homomorphism term. This retained route detaches the action bases. When its anti-contraction guard is reused during P2b, the response-floor reference is

$$B^{\mathrm{floor}}_a = \mathrm{sg}\left(\sum_k v^{\mathrm{init}}_{a,k} A_k\right),$$

using current detached bases and the fixed initial direction rows $v^{\mathrm{init}} = v^{\mathrm{ref}}$ from the P2a guard calibration. It is not a permanently frozen P1 generator once other losses have changed the action bases.

Weighted-group conventions. The groups $\mathcal{L}_{\text{scale}}$ and $\mathcal{L}_{\text{anchor}}$ include their internal weights. Image scale terms follow Equation (18); skeleton scale and finite-response reductions follow Appendix C.3. Inapplicable terms have zero weight, and Core P2b omits rollout by setting $\lambda_{\text{roll}} = 0$.

C.2.5 Reference anchors and Abelian specialization

For each image P2b continuation, superscript ref denotes its recorded immutable reference. ModelNet10 response and magnitude targets use P2a, while parameter anchors use the continuation's source checkpoint. With $\rho_F(p, y) = \|p - y\|_F^2/(\|y\|_F^2 + \epsilon)$, the shared response and parameter anchors are

$$\begin{aligned}
\mathcal{L}_{\text{resp}} &= |\mathcal{A}|^{-1} \sum_{a \in \mathcal{A}} \mathbb{E}_S \, \rho_F(B_a S, B_a^{\text{ref}} S), \\
\mathcal{L}_{\text{basis}} &= \frac{\sum_k \|A_k - A_k^{\text{ref}}\|_F^2}{\sum_k \|A_k^{\text{ref}}\|_F^2 + \epsilon}, \qquad \mathcal{L}_{\text{dir}} = \frac{\sum_a \|v_a - v_a^{\text{ref}}\|_2^2}{\sum_a \|v_a^{\text{ref}}\|_2^2 + \epsilon}, \\
\mathcal{L}_{\text{anchor}} &= \lambda_{\text{resp}} \mathcal{L}_{\text{resp}} + \lambda_{\text{basis}} \mathcal{L}_{\text{basis}} + \lambda_{\text{dir}} \mathcal{L}_{\text{dir}}.
\end{aligned} \tag{22}$$

Dataset-specific fixed scales and relation weights are frozen after the stated calibration screen and recorded in the released manifests.

Abelian image specializations. The zero-target loss is Equation (8) with $u_{ab} = 0$ and the Abelian denominator specified above. The 3DShapes response anchor retains the scale action only; Cars3D averages the elevation and azimuth response anchors. Both use separate direction-row anchors $\|v_a - v_a^{\text{ref}}\|_2^2/(\|v_a^{\text{ref}}\|_2^2 + \epsilon)$, with their own configured weights, rather than one pooled row denominator. The basis anchor remains the pooled relative error above.

C.3 Category-free skeleton objective

The code applies each action to a frozen source vector $h^0 = G^0(\mu^0(x))s_0^0$, where superscript 0 denotes the P1 parent. Exact rotated observations are encoded with the same frozen bases and carrier to give latent targets. Let L^h_{single} be the mean latent endpoint SSE over the three axes, L^h_{ordered} the mean over the three cyclic pairs and their two orders, and L^h_{gap} the mean squared predicted-minus-target ordered-gap error. Their decoded-coordinate counterparts are L^x_{single} and L^x_{ordered}. Then

$$\mathcal{L}^{\text{skel}}_{\text{P2a}} = L^h_{\text{single}} + L^h_{\text{ordered}} + \tfrac{1}{2} L^h_{\text{gap}}.$$

For $\mathcal{C} = \{(x, y, z), (y, z, x), (z, x, y)\}$, write $r_{ab} = [B_a, B_b]h^0$, $u_c = B_c h^0$, and $\rho_{ab} = \|r_{ab}\|_2 / \max(\|u_c\|_2, \epsilon)$, with $\epsilon = 10^{-8}$. The implemented structural terms are

$$C_{\text{skel}} = \tfrac{1}{3} \sum_{(a,b,c) \in \mathcal{C}} \mathbb{E} \, \frac{\|r_{ab} - u_c\|_2^2}{\max(\|u_c\|_2, \epsilon)^2}, \qquad M_{\text{br}} = \tfrac{1}{3} \sum_{(a,b,c) \in \mathcal{C}} \left[\log \max(\mathbb{E}\, \rho_{ab}, \epsilon)\right]^2. \tag{23}$$

Here u_c is live, not detached; the expectation is the sampled source minibatch. In particular, M_{br} is the log of the mean norm ratio, not a per-sample log-ratio penalty.

Finite-response anchors use $p_a = (\exp(\gamma B_a) - I)h^0$ and detached P2a-parent responses p_a^{ref}. Their losses are

$$D_{\text{resp}} = \mathbb{E}_{x,a,\gamma}[1 - \cos(p_a, p_a^{\text{ref}})], \qquad M_{\text{resp}} = \mathbb{E}_{x,a,\gamma} \left[\left(\log \frac{\max(\|p_a\|_2, \epsilon)}{\max(\|p_a^{\text{ref}}\|_2, \epsilon)} \right)^2 \right].$$

These preserve finite-action direction and magnitude, respectively; they are not the image pipeline's signed bracket-projection loss. With P2a-parent parameters $A^{\text{ref}}, v^{\text{ref}}$, define mean-entry anchors $A_{\text{anc}} = (dm^2)^{-1} \sum_k \|A_k - A_k^{\text{ref}}\|_F^2$ and $V_{\text{anc}} = (3d)^{-1} \sum_a \|v_a - v_a^{\text{ref}}\|_2^2$. The exact continuation is

$$\begin{aligned}
\mathcal{L}^{\text{skel}}_{\text{P2b+R}} = \mathcal{L}^{\text{skel}}_{\text{P2a}} &+ 0.25(L^x_{\text{single}} + L^x_{\text{ordered}}) + C_{\text{skel}} + 0.1 M_{\text{br}} \\
&+ 0.5 D_{\text{resp}} + 0.1 M_{\text{resp}} + 10 A_{\text{anc}} + V_{\text{anc}}.
\end{aligned} \tag{24}$$

Only the direction rows and Lie bases receive updates. The source-state bases, continuous encoder, shared carrier, and observation decoder remain fixed. Thus source reconstruction is unchanged even though the action bases are refined. Endpoint and closure errors are evaluated per source before averaging. The bracket-scale term instead uses the batch-average ratio defined above; the generator matrices themselves are shared across sources.

C.4 LABELS-HIDDEN DISCOVERY AND FEW-SHOT GROUNDING

This subsection specifies the reduced-label *image* route. It differs from the fully action-supervised objective in its finite losses, inferred coordinates and anchor use. The skeleton LF+64 comparison in Table 5 instead uses few-shot grounding followed by refinement with full action targets; the image-specific objectives below must not be interpreted as its training recipe.

C.4.1 INFER COEFFICIENTS OF ANONYMOUS OBSERVED TRANSITIONS

Public discovery records contain x_n, x'_n and an opaque identifier used for record alignment, never as a network input. Freeze all P1 parameters and buffers. Form standardized transition features

$$f_n = [\,\mu^0(x'_n) - \mu^0(x_n),\ \mathrm{vec}(\widehat{Z}^0(x'_n) - \widehat{Z}^0(x_n))\,]. \tag{25}$$

Standardization uses discovery-training statistics. For a coefficient vector $w_n \in \mathbb{R}^{10}$, define

$$B(w_n) = \sum_k w_{n,k} A^0_k, \qquad S'_n = \exp(B(w_n)) S^0_n, \qquad S^0_n = \widehat{Z}^0(x_n).$$

Decoding retains the source categorical probabilities as carrier weights.

First fit a free vector w_n, initialized to zero, for each training pair:

$$\mathcal{L}_{\mathrm{free}} = L_Z + 0.02 L_x + 10^{-5}\mathbb{E}_n d^{-1}\|w_n\|_2^2, \quad L_Z = \mathbb{E}_n \frac{\|S'_n - \widehat{Z}^0(x'_n)\|_F^2}{\max(\|\widehat{Z}^0(x'_n)\|_F^2, 10^{-6})}. \tag{26}$$

The decoded term L_x is mean-pixel BCE on dSprites. On the other images it is

$$L_x = \mathbb{E}_n \frac{\|\mathcal{D}^0(S'_n, c_n) - x'_n\|_2^2}{\max(\|x'_n - x_n\|_2^2, 10^{-6})}.$$

Thus discovery uses latent *target energy*, not the latent displacement normalization used in continuation.

C.4.2 CLUSTER DIRECTIONS AND LEARN A TRANSITION PREDICTOR

The slot count J is fixed at three for dSprites/ModelNet10 and two for 3DShapes/Cars3D; the method does not discover the number of actions. Let $M_n = B(w_n)$, $r_n = \|M_n\|_F$, and $U_n = M_n / \max(r_n, 10^{-8})$. Projective k-means groups opposite directions together:

$$j^\star_n = \arg\max_j |\langle U_n, C_j\rangle_F|, \qquad \zeta_n = \mathrm{sign}\langle U_n, C_{j^\star_n}\rangle_F.$$

The centers C_j have unit Frobenius norm and are updated using the normalized means of sign-aligned cluster members $\zeta_n U_n$. Zero signs are taken as $+1$. Initial anonymous coefficient rows and fixed pseudo-amplitudes are

$$\bar{w}_j = \mathrm{mean}_{n:j^\star_n = j}(\zeta_n w_n), \quad \tilde{v}^0_j = \frac{\bar{w}_j}{\max(\|\sum_k \bar{w}_{j,k} A^0_k\|_F, 10^{-8})}, \quad \xi^\star_n = \zeta_n \max(r_n, 10^{-8}). \tag{27}$$

The amplitude target is the signed individual matrix norm, not a projection onto the final center.

An inference MLP maps f_n to slot logits $\ell_{\mathrm{slot},n}$ and scalar amplitude ξ_n. With $p_n = \mathrm{softmax}(\ell_{\mathrm{slot},n}/0.35)$, the selected hard forward assignment uses straight-through gradients:

$$p^{\mathrm{ST}}_n = \mathrm{sg}(\mathrm{onehot}(\arg\max p_n) - p_n) + p_n, \qquad w_n = \xi_n \sum_j p^{\mathrm{ST}}_{n,j} \tilde{v}_j.$$

The inference network and anonymous rows are trained jointly:

$$\begin{aligned}\mathcal{L}_{\mathrm{joint}} = {} & L_Z + 0.02 L_x + 0.20[\mathrm{CE}(\ell_{\mathrm{slot}}, j^\star) + 0.05\,\mathrm{MSE}(\xi, \xi^\star)] \\ & + 0.05 D_{\mathrm{KL}}(\bar{p}\|U_J) + 0.01\mathbb{E}_n H(p_n).\end{aligned} \tag{28}$$

CE and MSE average over records; $\bar{p}$ is the mean soft slot distribution, and U_J is uniform over slots. The pseudo-targets remain fixed. The positive entropy penalty encourages confident *action-slot* assignments. This is different from P1's penalty against overconfident *categorical-carrier* posteriors: p_n and $q(x_n)$ are different variables with different roles.

C.4.3 Fit a separate coefficient-distribution flow

Detach the inferred coefficients after transition-predictor training. A LieFlow-style network (Chen et al., 2026) models their distribution:

$$z \sim \mathcal{N}(0, I_d), \quad w_{\tau_\mathrm{f}} = (1-\tau_\mathrm{f})z + \tau_\mathrm{f}\ \mathrm{sg}(w), \quad \mathcal{L}_\mathrm{flow} = \mathbb{E}\, d^{-1} \| u_\theta(w_{\tau_\mathrm{f}}, \tau_\mathrm{f}) - (\mathrm{sg}(w) - z) \|_2^2. \tag{29}$$

Here $\tau_\mathrm{f} \in [0,1]$ is flow time, distinct from P1 coordinates t and the Gumbel–Softmax temperature τ. The implementation uses LieFlow's sinusoidal-time-embedding MLP with input/output dimension ten and hidden width 64, and its power time sampler with skewness three. Only the flow parameters θ receive gradients. Coefficient generation is audited using 40 Euler steps and mean/covariance comparisons.

This flow does not refine the anonymous generators or P1, and its samples do not supply downstream action directions. Downstream uses the slot/inference model. The implementation therefore combines anonymous transition fitting with an auxiliary coefficient-distribution flow; it is not an unchanged LieFlow reproduction.

C.4.4 Ground the slots with 64 action-labeled anchors per action

For 64 training transitions per action, reveal the physical names and amounts γ_n. The permutation π maximizes anchor slot-to-action agreement. For dSprites/ModelNet10, correctly assigned anchors $\mathcal{C}_j$ fit a signed slope

$$s_j = \frac{\sum_{n \in \mathcal{C}_j} \gamma_n \xi_n}{\sum_{n \in \mathcal{C}_j} \gamma_n^2}, \qquad v^\mathrm{init}_{\pi(j)} = s_j \tilde{v}_j, \qquad (\widehat{a}, \widehat{\gamma}) = (\pi(j), \xi / s_j). \tag{30}$$

The stored calibration supplies numerically valid slopes. For 3DShapes/Cars3D, anchor amounts are signed unit grid steps: a majority sign $s_j \in \{-1, 1\}$ sets $v^\mathrm{init}_{\pi(j)} = s_j \tilde{v}_j$ and $\widehat{\gamma} = s_j \operatorname{sign}(\xi)$, taking a zero amplitude as positive.

dSprites uses this closed-form initialization directly. The other datasets also optimize copied direction rows on the same anchors with bases fixed:

$$\mathcal{L}_\mathrm{align} = E_\mathrm{anchor} + 0.05\, \frac{\|V - V^\mathrm{init}\|_F^2}{\max(\|V^\mathrm{init}\|_F^2, 10^{-8})}, \tag{31}$$

Here V stacks direction rows and $\mathcal{C}_a$ contains all 64 selected action-labeled anchors for action a. It is distinct from the correctly assigned subset $\mathcal{C}_j$ used for slot-slope fitting. For action-a anchors, let $T_n = e^{\gamma_n B_a} S_n^0$ and $T_n^\mathrm{dir} = e^{\gamma_n B_a^\mathrm{dir}} S_n^0$. Define

$$E_\mathrm{anchor} = \frac{1}{|\mathcal{A}|} \sum_a \frac{1}{|\mathcal{C}_a|} \sum_{n \in \mathcal{C}_a} \left[\frac{\|T_n - \widehat{Z}^0(x'_n)\|_F^2}{\max(\|\widehat{Z}^0(x'_n) - S_n^0\|_F^2, 10^{-8})} + \frac{\|\mathcal{D}^0(T_n^\mathrm{dir}, c_n) - \mathrm{sg}(\mathcal{D}^0(S_n^0, c_n)) - (x'_n - x_n)\|_2^2}{\max(\|x'_n - x_n\|_2^2, 10^{-8})} \right]. \tag{32}$$

The anchor-fitting budgets are 500, 252, and 500 updates on 3DShapes, Cars3D, and ModelNet10. No additional action-labeled examples are introduced. When this term is retained during P2b, its latent component updates directions and bases; the decoded component updates directions only.

C.4.5 Refine using predicted coordinates and observed endpoints

Apply the frozen predictor independently to (x, x_a) and (x, x_b). Freeze their grounded action indices and amounts. Continuation training keeps ordered records only when both slot confidences are at least 0.80 and the predicted slots differ. These confidences use $\mathrm{softmax}(\ell_\mathrm{slot})$, without the training temperature 0.35. Single-transition records are not filtered by this rule. The ordered states are

$$S_{ab} = e^{\widehat{\gamma}_a B_{\widehat{a}}} e^{\widehat{\gamma}_b B_{\widehat{b}}} S^0, \qquad S_{ba} = e^{\widehat{\gamma}_b B_{\widehat{b}}} e^{\widehat{\gamma}_a B_{\widehat{a}}} S^0. \tag{33}$$

Public images x_a, x_b, x_{ab}, x_{ba} provide endpoint targets. Ordinary finite-action losses use predicted action types and transformation amounts rather than their hidden ground-truth values.

Let E_1, E_2 denote mean single/ordered latent endpoint errors: their numerators are $\|S_\omega - \widehat{Z}^0(x_\omega)\|_F^2$ and their denominators use target displacement energy $\|\widehat{Z}^0(x_\omega) - S^0\|_F^2$. dSprites adds 10^{-8} to this energy; the other images clamp it at 10^{-8}. More precisely,

$$E_1 = \tfrac{1}{2}\left[E_{\text{single pool}} + \tfrac{1}{2}(E_a + E_b)\right], \qquad E_2 = \tfrac{1}{2}(E_{ab} + E_{ba}),$$

where $E_{\text{single pool}}$ averages the separate single-transition pool, and E_a, E_b, E_{ab}, E_{ba} average the indicated endpoints from retained ordered records. These reduced-label continuations have no separate signed order-gap loss.

Write $L_{\text{disp},a}$ for the per-action version of Equation (19), using inferred action coordinates on ordinary records and a target-energy denominator clamped at 10^{-8}. Their complete groupings are

$$\mathcal{L}_{\text{LF,DS}} = 4E_1 + 3E_2 + 1.5\mathcal{L}_{\text{alg}} + \mathcal{L}_{\text{scale}} + \lambda_{\text{resp}}\mathcal{L}_{\text{resp}} + \mathcal{L}_{\text{basis}} + \mathcal{L}_{\text{dir}} + \lambda_{\text{roll}}\mathcal{L}_{\text{roll}}, \tag{34}$$

$$\mathcal{L}_{\text{LF,Ab}} = 4E_1 + 3E_2 + 4E_{\text{anchor}} + \sum_a \lambda_{\text{disp},a} L_{\text{disp},a} + \lambda_{\text{alg}}\mathcal{L}_{\text{alg}} + 20\mathcal{L}_{\text{resp}} + \mathcal{L}_{\text{basis}} + \mathcal{L}_{\text{dir}} + \lambda_{\text{roll}}\mathcal{L}_{\text{roll}}, \tag{35}$$

$$\mathcal{L}_{\text{LF,MN}} = 4E_1 + 3E_2 + 4E_{\text{anchor}} + \lambda_{\text{sem}}L_{\text{sem}} + \lambda_{\text{alg}}\mathcal{L}_{\text{alg}} + \mathcal{L}_{\text{scale}} + 20\mathcal{L}_{\text{resp}} + \mathcal{L}_{\text{basis}} + \mathcal{L}_{\text{dir}} + \lambda_{\text{roll}}\mathcal{L}_{\text{roll}}. \tag{36}$$

Component roles are as defined in Section C.2, with the reduced-label route's own recorded references and calibrations. Both action responses are anchored on the Abelian reduced-label route. ModelNet10's L_{sem} is the mean of relative decoded single-action displacement errors over the three axes, with bases detached, as in Equation (19), but LF decoded-displacement terms clamp target displacement energy at 10^{-8}, rather than adding epsilon. Thus this decoded semantic route is active on reduced-label ModelNet10, although the fully action-supervised image displacement term was Abelian-only. The LF ModelNet10 route does not use the anti-contraction guard of fully action-supervised P2a.

The three dSprites links use $(\lambda_{\text{logmag}}, \lambda_{\text{proj}}, \lambda_{\text{resp}})$ of $(0, 0, 10), (0.05, 0, 20), (0.03, 0.20, 20)$, respectively. References refresh between links, and rollout enters only on the final link. There is no continuing anchor endpoint loss on dSprites. The other three datasets retain E_{anchor} throughout P2b. Their rollout averages four observed endpoints per ordered record, including reduced-label ModelNet10, rather than its fully action-supervised nine-endpoint rollout. Remaining 3DShapes, Cars3D, and ModelNet10 coefficients are calibrated on training records and then fixed across the dataset's seeds. dSprites calibrates its final rollout coefficient separately per seed as $0.1\|\nabla L_{\text{base}}\|/\|\nabla L_{\text{roll}}\|$, using the recorded training-only gradient screen; L_{base} is that link's objective without rollout.

Supervision and update accounting. Only the selected 64 training anchors per action reveal names and transformation parameters to grounding losses; training metadata assemble these balanced anchor sets. Other continuation records use frozen inferred coordinates and observed endpoints, with prescribed rather than discovered algebra. Continuation updates copied direction rows and action bases; the encoders, carrier, decoder, buffers and inference head remain fixed. This is not end-to-end unsupervised learning. Budgets, including coefficient-flow fitting and anchor optimization, are given in Appendix E.1.

D Diagnostics, metrics, and evaluation protocol

These are frozen-model measurements, not additional training losses. Representation probes and interventions test P1 branch roles; finite-order diagnostics separate operator, state and decoded effects; physical-action metrics evaluate the target-grounded P2 models. Fixed evaluation records are specified in Appendix D.4; supplementary results are in Appendix F.

D.1 P1 representation probes and interventions

Frozen probes. For each dataset and seed, compared checkpoints use identical deterministic probe indices. With the model frozen, separate ridge probes predict categorical and geometric factors from $q(x)$ and $\mu(x)$. Probe labels never update the representation. Features are standardized using probe-training statistics. Ridge strength is selected from $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$ by validation balanced

accuracy, or minimum mean geodesic error for rotation, before refitting on training and validation data.

Let $a_f(z)$ denote held-out balanced accuracy for predicting factor f from branch z. The category gap is $a_{\mathrm{cat}}(q) - a_{\mathrm{cat}}(\mu)$. For the factorial image datasets, the geometry gap is $|\mathcal{G}|^{-1} \sum_{g \in \mathcal{G}} [a_g(\mu) - a_g(q)]$, where $\mathcal{G}$ contains scale, orientation and horizontal/vertical position for dSprites; scale and orientation for 3DShapes; and elevation and azimuth for Cars3D. For ModelNet10, ridge regression predicts rotation-matrix entries, followed by projection onto $\mathrm{SO}(3)$. The pose gap is the mean geodesic error from q minus that from μ, in degrees. Positive gaps therefore favour the intended branch roles.

Occupancy is $H(\bar{q})/\log K$, where $\bar{q} = N^{-1} \sum_n q(x_n)$ over probe-test observations and H is Shannon entropy. It measures aggregate code use, not per-observation confidence.

Interventions. For a category intervention, we hold the continuous code fixed and replace or interpolate the categorical posterior toward that of a factor-matched target image. For a geometry intervention, we hold the categorical code fixed and replace the continuous code using a same-category image with different geometry.

For edited output $\hat{x}_{\mathrm{edit}}$, source x and target $x^\star$, edit MSE measures target fit. Target advantage is $\mathrm{MSE}(\hat{x}_{\mathrm{edit}}, x) - \mathrm{MSE}(\hat{x}_{\mathrm{edit}}, x^\star)$, using mean pixel error. A positive advantage means that the edit is closer to the target than to the source; it does not measure improvement relative to the unedited reconstruction.

D.2 Finite ordered-action diagnostics

We diagnose the selected P1 checkpoint, denoted by superscript 0. For sample x_b, these deterministic diagnostics use posterior means $t_b^0 := \mu^0(x_b)$, rather than posterior samples. For generators A_i^0, A_j^0 and coordinates $t_{i,b}^0, t_{j,b}^0$, the finite BCH deviation is

$$D_{ij,b}^{\mathrm{BCH},0} = \left\| \exp(t_{i,b}^0 A_i^0 + t_{j,b}^0 A_j^0) - \exp(t_{i,b}^0 A_i^0) \exp(t_{j,b}^0 A_j^0) \right\|_F . \tag{37}$$

This operator-level statistic screens finite non-additivity; it does not by itself show that the discrepancy moves the state on which the generators act.

To avoid applying the probed coordinates twice, first construct a common base state containing only the non-probed coordinates:

$$h_{-ij,b}^0 := \exp\left(\sum_{k \notin \{i,j\}} t_{k,b}^0 A_k^0 \right) s^0(c_b). \tag{38}$$

Writing $G_{k,b}^0 = \exp(t_{k,b}^0 A_k^0)$, the two ordered states are

$$h_{ij,b}^0 = G_{i,b}^0 G_{j,b}^0 h_{-ij,b}^0, \qquad h_{ji,b}^0 = G_{j,b}^0 G_{i,b}^0 h_{-ij,b}^0. \tag{39}$$

Thus both orders act on exactly the same detached categorical sample c_b, carrier $s^0(c_b)$, and all non-probed continuous coordinates; only the action order changes. The normalized state displacement is

$$d_{ij,b}^0 = \frac{\|h_{ij,b}^0 - h_{ji,b}^0\|_2}{\|h_{-ij,b}^0\|_2 + \varepsilon_h}. \tag{40}$$

It tests whether swapping the actions actually moves this sample. In a stabilizer case, $(G_i^0 G_j^0 - G_j^0 G_i^0) h_{-ij,b}^0 = 0$, even though the two operators need not agree on other states. The BCH difference in Equation 37 is a distinct operator diagnostic.

For decoder parameters θ, decode the fixed ordered states as

$$x_{ij,b}^\theta = \delta_\theta\big(\mathrm{sg}(h_{ij,b}^0)\big), \qquad x_{ji,b}^\theta = \delta_\theta\big(\mathrm{sg}(h_{ji,b}^0)\big), \tag{41}$$

and measure

$$\widetilde{\Delta}_{ij,b}^\theta = \frac{\sqrt{\|x_{ij,b}^\theta - x_{ji,b}^\theta\|_2^2 + \varepsilon_\Delta^2}}{\|x_b\|_2 + \varepsilon_x}. \tag{42}$$

Here θ is fixed to the P1 decoder for the audit, and positive $\varepsilon_h, \varepsilon_x, \varepsilon_\Delta$ are numerical stabilizers. We use $\Delta^\theta_{ij,b}$ for the unsmoothed numerator $\|x^\theta_{ij,b} - x^\theta_{ji,b}\|_2$. These are diagnostics, not an additional hinge-training objective.

The three quantities answer different questions: $D^{\mathrm{BCH},0}$ measures operator-level non-additivity, d^0 measures the order effect on the current carrier, and $\widetilde{\Delta}^\theta$ measures the decoder-visible order gap. A small gap on one carrier does not establish a commuting algebra. Nonzero d^0 with a small decoded gap reveals attenuation by the decoder. Conversely, a small but nonzero state gap can be amplified by decoder sensitivity; it is not by itself a false physical order effect. Identical states must decode identically. A *false commuting gap* is defined only against a known commuting action pair. These audits screen P1; exact targets and prescribed brackets evaluate P2a and P2b. For skeletons the same definitions replace $s^0(c_b)$ by the shared s^0_0.

When a renderer provides exact ordered targets x_{AB} and x_{BA}, we additionally measure target accuracy, assignment correctness, magnitude and direction recovery, and commuting controls. These tests are distinct from reconstruction of the original input and are required for a claim that an ordered output is correct rather than merely different.

Local commutator interpretation. For $X = t_i A_i$ and $Y = t_j A_j$, the BCH expansion gives

$$\begin{aligned} \exp(X)\exp(Y) &= \exp\big(X + Y + \tfrac{1}{2}[X,Y] + \mathcal{O}_3\big)\,, \\ \exp(Y)\exp(X) &= \exp\big(X + Y - \tfrac{1}{2}[X,Y] + \mathcal{O}_3\big)\,, \end{aligned} \tag{43}$$

where $\mathcal{O}_3$ contains terms of total degree at least three (Higham, 2008). The bracket is therefore the leading local source of order dependence, but it does not by itself identify the physical action, its sign, or its finite endpoint. The fully action-supervised P2 route uses supplied transformation parameters and exact transformed observations for those purposes. Reduced-label P2 uses few-shot grounding and frozen inferred coordinates as specified in Appendix C.4; neither route is end-to-end unsupervised symmetry discovery.

D.3 Physical-action and Lie metrics

Let p_ω and y_ω denote predicted and target endpoints in the evaluated latent or decoded space. Single-action direction is the cosine between predicted and target displacements from their common source. Order-gap direction is the cosine between $\Delta^\Phi = p_{ab} - p_{ba}$ and $\Delta^\star = y_{ab} - y_{ba}$. Order assignment is

$$\mathbb{E}\,\mathbf{1}[d(p_{ab}, y_{ab}) + d(p_{ba}, y_{ba}) < d(p_{ab}, y_{ba}) + d(p_{ba}, y_{ab})]\,, \tag{44}$$

where d is Euclidean distance for image endpoint-L2 audits and squared Euclidean distance for skeletons. This is the fraction of records whose true AB/BA pairing beats the swapped pairing, not category accuracy. Joint endpoint L2 averages the two true-pairing distances; skeleton endpoint tables instead report mean per-observation SSE over normalized joint coordinates. Decoded gap-magnitude error is $\mathbb{E}\big|\|\Delta^\Phi\|_2 - \|\Delta^\star\|_2\big|$. For a commuting pair, decoded false gap is $\mathbb{E}\|\hat{x}_{ab} - \hat{x}_{ba}\|_2$.

For a nonzero prescribed relation, define the bracket response $r_{ab}(S) = [B_a, B_b]S$ and target response $u_{ab}(S) = \sum_g c^g_{ab} B_g S$. Bracket cosine and norm ratio are

$$\mathbb{E}_S \frac{\langle r_{ab}, u_{ab}\rangle_F}{\max(\|r_{ab}\|_F, \epsilon)\max(\|u_{ab}\|_F, \epsilon)}, \qquad \mathbb{E}_S \frac{\|r_{ab}\|_F}{\|u_{ab}\|_F + \epsilon}. \tag{45}$$

For these bracket metrics, $\epsilon = 10^{-8}$ on dSprites and 10^{-12} on ModelNet10. Skeletons use $\epsilon = 10^{-8}$ and replace the norm-ratio denominator with $\max(\|u_{ab}\|_2, \epsilon)$. The Frobenius norm reduces to the Euclidean norm for vector states. Reported mean cyclic metrics average equally over the three cyclic $\mathrm{SO}(3)$ relations.

For dSprites evaluation, all responses and normalizers use the generators of the checkpoint being evaluated. For an evaluated source state S, define

$$\eta_{xy}(S)^2 = \tfrac{1}{2}\big(\|B_{T_x} S\|_F^2 + \|B_{T_y} S\|_F^2\big).$$

The translation–translation relative bracket RMS is

$$\left[\mathbb{E}_S \frac{\|[B_{T_x}, B_{T_y}]S\|_F^2}{\eta_{xy}(S)^2 + 10^{-8}}\right]^{1/2}. \tag{46}$$

The reported equal-term closure is

$$C_{\mathrm{SE}(2)} = \tfrac{1}{3}\mathbb{E}_S \left[\frac{\|[B_R, B_{T_x}]S - B_{T_y}S\|_F^2}{\|B_{T_y}S\|_F^2 + 10^{-8}} + \frac{\|[B_R, B_{T_y}]S + B_{T_x}S\|_F^2}{\|B_{T_x}S\|_F^2 + 10^{-8}} + \frac{\|[B_{T_x}, B_{T_y}]S\|_F^2}{\eta_{xy}(S)^2 + 10^{-8}} \right]. \tag{47}$$

Unlike the dSprites training loss, these evaluation normalizers do not use continuation-reference generators. ModelNet10 cyclic closure averages $\|r_{ab} - u_{ab}\|_F^2/(\|u_{ab}\|_F^2 + 10^{-8})$ over sources and the three cyclic relations, using current-checkpoint target responses. Skeleton closure uses the live target and clamping in Equation (23).

The finite-action commutator reported for 3DShapes and Cars3D is

$$C_{\mathrm{finite}} = \mathbb{E}_n \left\| H_a(\alpha_n)H_b(\beta_n) - H_b(\beta_n)H_a(\alpha_n) \right\|_F ,$$

using the same action amounts as the corresponding endpoint audit. This is an unnormalized operator norm: it does not multiply the commutator by a source state.

Ideal nonzero bracket cosine and norm ratio are one; ideal closure, commuting residuals, endpoint error, gap-magnitude error and false gap are zero.

D.4 Evaluation records and statistical reporting

Reported aggregates use five independent training seeds; means and sample SDs are computed over seeds, not individual evaluation records. Within each seed, compared checkpoints use the same fixed records. Cumulative stage references and continuation-budget comparisons are distinguished in Appendix E.1.3.

Image action records. dSprites uses 1,536 fixed development records: 512 for each of R, T_x, R, T_y, and T_x, T_y, evaluated at unit action scale. 3DShapes uses 4,096 records per seed, with factor-set seeds $18007 + s$ and FVM seeds $92000 + s$, $s \in \{0, \ldots, 4\}$. Its scale and periodic orientation steps are one grid unit. The rollout recipe is selected with the disjoint development factor seed 348007 before the five-seed evaluation. Cars3D uses 4,096 fixed records per seed; its elevation–azimuth gap is a commuting negative control. ModelNet10 uses 640 development records per seed; its selected recipe is fixed after seed-0 development.

Skeleton action records. MI-Motion uses 512 sources from development sequences disjoint from training; 3DPW uses 512 sources from official validation sequences. Each source receives three single-axis rotations and both orders of the cyclic xy, yz, zx pairs. The MI-Motion evaluator seed is 36001. These sources are not used for P1 or P2 optimization. The saved 1,536-source confirmation/test pools are not evaluated in the reported category-free comparison.

Preservation measurements. Image reconstruction follows the posterior-mean, soft-code, current-basis protocol in Table 7. FVM evaluated directly from a frozen encoder cannot measure changes confined to the Lie bases. Parameter and buffer scope are distinguished for the component ablation in Appendix F.3.

E Datasets and training configurations

This appendix specifies preprocessing, dataset sizes and training settings. Loss definitions are in Appendices B and C; fixed evaluation records are in Appendix D.4.

Controlled image benchmarks. dSprites contains 64×64 binary images generated from shape, scale, orientation, and horizontal and vertical position. Their cardinalities are 3, 6, 40, 32, and 32, respectively. We assign shape to the categorical branch. 3DShapes contains 64×64 RGB images

generated from object shape, scale, orientation, object colour, wall colour, and floor colour; object shape has four values and is assigned to the categorical branch. Cars3D contains 17,568 RGB images generated from car identity, azimuth, and elevation, with 183, 24, and 4 values, respectively. Car identity is assigned to the categorical branch and viewpoint to the continuous branch.

ModelNet10–$SO(3)$ identity–pose benchmark. The benchmark uses the official orientation-aligned ModelNet10 collection, containing 4,899 meshes in ten classes with its official train/test split. Each mesh is centered and normalized while preserving its official orientation, then rendered deterministically with orthographic depth and surface-normal channels. Class identity defines the intended categorical factor, while the frozen geometric probe predicts the supplied rotation matrix. Category labels and rotations are used only for splitting and evaluation in unsupervised P1. The separately labeled paired-weak variant uses them only to construct identity- and rotation-matched relations, as described in Equation 15; they are not category-classifier targets. Because the sampled views do not form a balanced factorial grid, FVM is not reported.

MI-Motion real-skeleton stress test. MI-Motion contains 3D multi-person motion sequences (Peng et al., 2023). Following LieFlow's preprocessing (Chen et al., 2026), we subtract pelvis joint 12 and divide each 20-joint frame by its vertical head-to-foot height: the height of head joint 0 minus the mean height of foot joints 16 and 19, producing 60-coordinate observations. The inherited preprocessing discards frames below 0.3 times the median height computed over all source sequences before splitting. The retained 538,625 frames come from 215 source sequences. Complete sequences are split into 166 train, 19 development, and 30 confirmation sequences, containing 430,018, 53,571, and 55,036 frames, respectively, with zero sequence overlap. P1 receives no scene, sequence, person, frame, rotation, or action label. For P2, exact single and ordered targets are produced by applying controlled global-axis rotations with magnitudes sampled in $[0.15, 0.50]$ radians after normalization, without recomputing the height scale after rotation. These targets evaluate the cyclic $\mathfrak{so}(3)$ law in real skeleton coordinates; they do not assert that natural human actions are global rotations or that the model discovers their symmetries.

3DPW real captured-pose replication. 3DPW provides in-the-wild video sequences with SMPL body fits (von Marcard et al., 2018). We use only the fitted 3D joints, not RGB: all 24 joints are retained, pelvis-centred, and divided by each actor's median torso-plus-leg-chain scale, yielding 72-coordinate observations. The official split is unchanged: 24 train sequences (23,543 actor-frames), 12 validation sequences (10,858), and 24 test sequences (40,219), with no sequence overlap. The frozen action records contain 64 train anchors, 512 validation-development sources, and 1,536 official-test sources. The category-free results in this revision use only the 512 development sources; no category-free official-test result is reported. P2 applies the same controlled global-axis rotations as MI-Motion; this audits viewpoint equivariance of real captured poses, not natural articulation or video appearance. The data are used under the official non-commercial scientific-research terms; we do not redistribute raw or prepared 3DPW data.

Representation sizes. Every reported model uses a ten-dimensional continuous posterior. The categorical codebooks for dSprites, 3DShapes, Cars3D, and ModelNet10 contain 3, 4, 183, and 10 entries, with carrier-embedding dimensions 64, 64, 32, and 64, respectively. MI-Motion and 3DPW have no categorical codebook: each uses one shared learned 16-dimensional carrier.

E.1 Selected training configurations

E.1.1 P1 configurations

All selected image-benchmark P1 models use Adam with learning rate 2×10^{-4}, a ten-dimensional continuous posterior, capacity 10, auxiliary encoded-feature reconstruction probability 0.2, $\lambda_{\text{shape_usage}} = 1$, $\lambda_{\text{shape_sep}} = 10^{-3}$. Reconstruction is binary cross-entropy for dSprites and squared error for the RGB/depth benchmarks. Table 9 gives the dataset-specific launcher arguments. "Endpoint" is the number of completed epochs; the corresponding checkpoint index is one smaller. ModelNet paired training deliberately changes only λ_{MI} from 0.4 to 0.6 during its final 20 epochs.

For dSprites, the selected checkpoints are epochs 52 (U) and 48 (PW); for 3DShapes and Cars3D they are epochs 40 and 249, respectively; for ModelNet10 they are epochs 149 (U) and 169 (PW).

Table 9: Selected image-benchmark P1 configurations (reproducibility settings, not result or ablation metrics). U denotes unsupervised training and PW denotes paired weak supervision. The weak objectives use relational metadata only: λ_q is categorical invariance, λ_{pose} is the launcher name for relative-action equivariance, and $\lambda_{\mu\text{-sp}}$ is continuous invariance for action-matched pairs. Carrier-swap reconstruction reuses the MI schedule.

Dataset	Setting	Endpoint	Batch	K/e	τ	β	λ_{MI}	λ_{shape}	$\lambda_{q\text{-}\mu}$	λ_q	λ_{pose}	$\lambda_{\mu\text{-sp}}$	hy_rec
dSprites	U	53	256	3/64	.67	1.2	.6	.1	.5	0	0	0	1.0
	PW	49	256	3/64	.67	1.2	.4	.1	.5	1	1	0	1.0
3DShapes	U	41	256	4/64	.67	.5	.6	.1	0	0	0	0	1.0
	PW	41	256	4/64	.67	.5	.6	.1	2.5	1	0	0	1.0
Cars3D	U	250	64	183/32	1.5	.5	.5	1.0	2.5	0	0	0	.7
	PW	250	64	183/32	1.5	.5	.4	1.0	2.5	1	1	0	.7
ModelNet10–$SO(3)$	U	150	64	10/64	.67	.5	.6	.1	2.5	0	0	0	1.0
	PW	$150 + 20$	64	10/64	.67	.5	$.4 \to .6$	.1	2.5	1	1	1	1.0

Table 10: Dataset-specific training configurations. Known actions provide transformation metadata, not category labels. Schedules give cumulative P2a and P2b-stage training, including rollout where enabled: epochs for images and optimizer updates for skeletons. They are not common compute budgets across datasets.

Dataset / actions	Target algebra	P2a endpoint map	P2b+rollout signal	Schedule
dSprites: R, T_x, T_y	$[B_R, B_x] = B_y$, $[B_R, B_y] = -B_x$, $[B_x, B_y] = 0$	Latent group state	Retained one-/two-step endpoints, closure, log-norm, signed projection, anchors and final-link frozen-decoder rollout	$2 + 3$ epochs
3DShapes: scale, orientation	$[B_s, B_o] = 0$	Latent group state	Retained endpoints, decoded displacements, zero closure, anchors and rollout	$2 + 1$ epochs
Cars3D: elevation, azimuth	$[B_e, B_a] = 0$	Latent group state	Retained endpoints, decoded displacements, zero closure, anchors and rollout	$3 + 2$ epochs
ModelNet10: global x, y, z	Three cyclic $\mathfrak{so}(3)$ relations	Decoded displacement	Retained endpoints, cyclic closure, log-norm, signed projection, anchors and rollout	$9 + 7$ epochs
MI-Motion: global x, y, z	$[B_x, B_y] = B_z$, $[B_y, B_z] = B_x$, $[B_z, B_x] = B_y$	Latent carrier vector	Retained single/ordered endpoints, cyclic closure, bracket log-norm, finite-response direction/magnitude anchors and frozen-decoder rollout	$1000 + 1000$ updates
3DPW: global x, y, z	Same cyclic $\mathfrak{so}(3)$ relations	Latent carrier vector	Same P2b+rollout objective in normalized 24-joint coordinates	$1000 + 1000$ updates

dSprites and ModelNet10 use periodic or matrix-valued relative-action targets in the paired arm. Cars3D uses same-identity/different-pose and same-pose/different-identity pairs. 3DShapes uses same-shape/different-geometry and same-geometry/different-shape pairs. No factor label is used as a classifier target.

MI-Motion and 3DPW use the shared-carrier construction $h = G(t)s_0$, with MLP observation networks for 60 and 72 skeleton coordinates, respectively. Five unsupervised P1 seeds per dataset use Adam at 2×10^{-4}, batch size 512, and 20 training epochs. The fixed P2 parent is checkpoint index 18 (19 completed epochs), preserving the established selection rule rather than selecting a new best epoch. P1 retains only coordinate reconstruction SSE, 0.1 mean-entry group-feature MSE, and 10^{-3} ordinary Gaussian KL; there is no capacity schedule or categorical loss. Both models have ten continuous coordinates, ten 16×16 Lie bases, and one shared 16-dimensional carrier. The continuous encoder uses two 256-unit hidden layers; the decoder uses two 256-unit hidden layers. MI-Motion has 840 P1 updates per epoch and a 10,000-source development cap; 3DPW has 46 updates per epoch and uses all 10,858 validation observations. P1 development averages use the reconstruction evaluation set; the P2 action results use the separate 512-source records specified in Appendix D.4.

E.1.2 P2 CONFIGURATIONS

Selected continuation settings. Table 11 gives trainable scopes and coefficients. dSprites uses three one-epoch refinement links with link-local references. ModelNet10 uses nine P2a epochs and seven refinement epochs; rollout replaces the final refinement link.

Table 11: Selected staged-continuation configurations (reproducibility settings, not result or ablation metrics). GB denotes a one-minibatch gradient-balance screen fixed before continuation.

Dataset	Stage	Schedule	Updated parameters	Active coefficients
dSprites	P2a	2	30 action-direction scalars	$\lambda_{\text{lat}} = 1$, $\lambda^{z}_{\text{sgn}} = 114.781$, $\lambda^{z}_{\text{comm}} = 12.451$
dSprites	P2b+rollout	3 after P2a	30 directions + ten 10×10 Lie bases	final link: $\lambda_{\text{close}} = 1.5$, $\lambda_{\text{hom1}} = 4$, $\lambda_{\text{hom2}} = 3$, $\lambda_{\text{resp}} = 20$, anchors $= 1/1$, $\lambda_{\text{logmag}} = .03$, $\lambda_{\text{proj}} = .20$, $\lambda_{\text{roll}} = 4.29537 \times 10^{-4}$; projected $R, T_x : R, T_y = 1 : 3$
3DShapes	P2a	2	20 action-direction scalars	$\lambda_{\text{lat}} = 1$, scale/orientation weights $= .943735/1.056265$, $\lambda^{z}_{\text{comm}} = 1.007127$
3DShapes	P2b+rollout	1 after P2a	20 directions + ten 20×20 Lie bases	$\lambda^{s}_{\text{disp}} = 275.194171$, $\lambda^{o}_{\text{disp}} = 255.362127$, $\lambda_{\text{ab}} = 144.847561$, $\lambda^{s}_{\text{resp}} = 20$, anchors $= 1/1/1$, $\lambda_{\text{roll}} = 0.03285715$
Cars3D	P2a	3	20 action-direction scalars	$\lambda_{\text{lat}} = 1$; commuting term GB to 20% of target gradient
Cars3D	P2b+rollout	2 after P2a	20 directions + ten 10×10 Lie bases	$\lambda^{e}_{\text{disp}} = 252.857726$, $\lambda^{a}_{\text{disp}} = 5619.830768$, $\lambda_{\text{ab}} = 283.037151$, $\lambda_{\text{resp}} = 20$, anchors $= 1/1/1$, $\lambda_{\text{roll}} = 0.10152570$
ModelNet10–$SO(3)$	P2a	$6 + 3$	30 action-direction scalars	initial signed GB ratio .25; continuation ratio 2, pair priorities $.8 : 1 : 1.2$, response/direction floors $= 20/1$
ModelNet10–$SO(3)$	P2b+rollout	7 after P2a	30 directions + ten 20×20 Lie bases	$\lambda_{\text{close}} = 3.468112$, $\lambda_{\text{resp}} = 20$, $\lambda_{\text{logmag}} = 0.111233$, $\lambda_{\text{proj}} = 1.268473$, anchors $= 1/1$, $\lambda_{\text{roll}} = 0.01799974$
MI-Motion–$SO(3)$	P2a	1,000 updates	30 action-direction scalars	single/ordered latent endpoints plus 0.5 weighted latent gap error
MI-Motion–$SO(3)$	P2b+rollout	1,000 updates after P2a	30 directions + ten 16×16 Lie bases	$\lambda_{\text{roll}} = .25$, $\lambda_{\text{close}} = 1$, $\lambda_{\text{logmag}} = .1$, response cosine/log-norm $= .5/.1$, basis/direction anchors $= 10/1$
3DPW–$SO(3)$	P2a	1,000 updates	30 action-direction scalars	same latent endpoint and gap objective as MI-Motion
3DPW–$SO(3)$	P2b+rollout	1,000 updates after P2a	30 directions + ten 16×16 Lie bases	$\lambda_{\text{roll}} = .25$, $\lambda_{\text{close}} = 1$, $\lambda_{\text{logmag}} = .1$, response cosine/log-norm $= .5/.1$, basis/direction anchors $= 10/1$

E.1.3 Continuation controls and update accounting

The internal Strict control continues the P1 objective without physical-action targets and adds the unnormalized all-pairs basis penalty

$$\mathcal{L}_{\text{Strict}} = \mathcal{L}_{\text{P1}} + \frac{\lambda_{\text{comm}}}{45} \sum_{i<j} \|A_i A_j - A_j A_i\|_F^2. \tag{48}$$

Matrix entries are summed and the 45 basis pairs are averaged. The coefficient is calibrated once on seed-0 training minibatches using Lie-basis gradient norms and then held fixed across seeds. This is a commutativity control, not an action-supervision-matched test of the algebraic loss; the latter is provided by Appendix F.3.

The dSprites, 3DShapes and Cars3D continuation comparisons use 12,960, 5,064 and 1,240 post-P1 optimizer updates, respectively. For ModelNet10, the recorded P1-only and Strict checkpoints have 1,440 post-P1 updates, whereas the action-trained comparison uses 2,880. These ModelNet10 entries therefore do not form a fully update-matched comparison. Matching nominal epochs alone is insufficient when the update counts per epoch differ.

Rollout and reduced-label update allocation. The fully action-supervised rollout and reduced-label pipelines use 12,960 (dSprites), 5,064 (3DShapes), 1,240 (Cars3D) and 2,880 (ModelNet10) post-P1 updates. LF+64 allocates these to discovery, grounding and refinement as $5184 + 0 + 7776$, $2876 + 500 + 1688$, $740 + 252 + 248$ and $1120 + 500 + 1260$, respectively. dSprites uses a closed-form anchor map with no grounding optimizer updates. Calibration updates are recorded separately.

F Additional analyses

This appendix reports the branch interventions and structural-loss controls that complement the main representation and P2 results.

F.1 P1 branch interventions

Table 12 tests decoded control by changing one branch while holding the other fixed. For paired-weak dSprites, categorical interventions are target-directed in five of five seeds, whereas geometry interventions are target-directed in two of five.

Table 12: P1 fixed-component intervention target advantages (mean±sample SD over five seeds). Positive values mean that changing only the named branch moves the decoded output toward its factor-matched target. The factorial image benchmarks support this audit; ModelNet10's nonfactorial sampled views do not provide the required matched intervention targets.

Dataset	Setting	Category ↑	Geometry ↑
dSprites	Unsupervised	-0.00693 ± 0.00585	0.00136 ± 0.00092
	Paired weak	0.00105 ± 0.00064	0.00013 ± 0.00047
3DShapes	Unsupervised	-0.0017 ± 0.0007	0.0005 ± 0.0002
	Paired weak	0.0015 ± 0.0009	0.0006 ± 0.0001
Cars3D	Unsupervised	-0.01113 ± 0.00233	0.02436 ± 0.00156
	Paired weak	0.00606 ± 0.00240	0.01607 ± 0.00148

Table 13: Categorical-branch and commutativity ablations on 3DShapes and Cars3D (mean±sample SD; five seeds). Total optimizer-update budgets are matched within each dataset. Ordinary excludes the added all-pairs commutativity penalty; +Strict includes it.

	Categorical–continuous		Continuous-only	
Metric	Ordinary	+Strict	Ordinary	+Strict
3DShapes				
Reconstruction SSE ↓	27.186±3.427	27.851±6.054	31.262±7.048	32.992±7.821
FVM ↑	0.795±0.092	0.797±0.106	0.772±0.093	0.772±0.091
$\mathcal{C}$ ($\times 10^{-3}$)	353.606±87.053	2.758±1.383	1167.795±371.057	8.817±4.966
Cars3D				
Reconstruction SSE ↓	57.258±1.466	57.722±1.265	64.606±0.510	65.114±0.413
FVM ↑	0.946±0.050	0.941±0.051	0.931±0.045	0.904±0.055
$\mathcal{C}$ ($\times 10^{-3}$)	207.172±55.051	5.419±2.681	399.651±74.825	16.048±1.982

Removing the categorical branch also removes its associated losses; compute and parameter counts are not matched. $\mathcal{C}$ is the mean squared basis commutator; the scale applies to means and SDs. It is scale-dependent, not a physical-action error.

F.2 Categorical-branch and commutativity ablations

We compare categorical–continuous and continuous-only models, each with ordinary or Strict continuation, under matched total optimizer-update budgets (Table 13). This tests the representation design and the effect of adding commutativity penalties. On both benchmarks, categorical–continuous models achieve lower mean reconstruction error and higher mean FVM than continuous-only models, with and without Strict continuation. Strict reduces basis commutators in both architectures, but increases mean reconstruction error and has mixed effects on FVM. These results support the categorical-branch design and its associated objectives on these two benchmarks.

F.3 P2b loss components and rollout extension

Table 14 connects the ablation to the groups in Equation (9). Starting from finite-only P2b, the first four columns cumulatively add closure $\mathcal{L}_{\mathrm{alg}}$, log magnitude, and signed projection (the latter two form $\mathcal{L}_{\mathrm{scale}}$), yielding the complete Core P2b objective. The fifth column reports the existing P2b+R extension, which adds decoded rollout $\mathcal{L}_{\mathrm{roll}}$.

All five variants share seed-specific P2a parents and 7,776 refinement updates (three links of 2,592), or 12,960 total post-P1 updates. Only the 1,030 direction/basis parameters are optimized. Finite-action terms and anchor schedules are retained; *finite-only* omits the added bracket losses, not the shared native objective or anchors. Closure weights are $1.5/1.5/1.5$, log-magnitude weights $0/0.05/0.03$, and projection weights $0/0/0.20$ when the respective term is enabled. Shared single/ordered finite-action weights are $4/3$, response-anchor weights $10/20/20$, and basis/direction anchor weights $1/1$. References refresh from each link's own parent.

Rollout is enabled only in the final refinement link, with $\lambda_{\mathrm{roll}} = 4.29537 \times 10^{-4}$; it does not add another training epoch. All columns use the same development records and metric definitions. The rollout implementation also locks decoder spectral-normalization buffers, whereas the four original controls permit small buffer changes. It is therefore shown as an update-matched extension, not a strictly isolated loss-only comparison.

Table 14: dSprites P2b loss components and rollout extension (mean±sample SD; five seeds). All variants share P2a parents, evaluation records and refinement update budgets. The first four columns add structural terms cumulatively; the rollout extension also differs in normalization-buffer handling, as described above.

	Structural loss components				Rollout extension
Metric	Finite-only P2b	$+\mathcal{L}_{\text{alg}}$	$+\mathcal{L}_{\text{logmag}}$	Core P2b ($+\mathcal{L}_{\text{proj}}$)	P2b+R ($+\mathcal{L}_{\text{roll}}$)
Decoded joint endpoint L2 ↓	8.498 ± 0.815	8.553 ± 0.854	8.550 ± 0.874	8.549 ± 0.882	8.509 ± 0.867
Closure ↓	0.665 ± 0.044	0.540 ± 0.059	0.533 ± 0.061	0.529 ± 0.063	0.523 ± 0.066
R, T_x bracket cosine ↑	0.443 ± 0.253	0.657 ± 0.331	0.616 ± 0.322	0.619 ± 0.324	0.631 ± 0.327
R, T_x norm ratio → 1	0.164 ± 0.063	0.186 ± 0.076	0.214 ± 0.071	0.216 ± 0.072	0.219 ± 0.074
R, T_y bracket cosine ↑	0.454 ± 0.226	0.742 ± 0.127	0.720 ± 0.137	0.740 ± 0.132	0.752 ± 0.126
R, T_y norm ratio → 1	0.159 ± 0.072	0.174 ± 0.074	0.199 ± 0.073	0.205 ± 0.076	0.208 ± 0.077
T_x, T_y relative bracket-response RMS ↓	0.451 ± 0.173	0.245 ± 0.074	0.262 ± 0.085	0.268 ± 0.088	0.269 ± 0.088

Table 15: Prescribed-bracket control with decoded rollout on dSprites. Mean±sample SD over all five paired seeds on fixed development records. Both arms use 7,776 refinement updates from common P2a parents. Only prescribed-bracket weights differ; finite-action losses, rollout, anchor-refresh policy and frozen decoder buffers are shared. Paired changes are Core+R minus finite-only+R; for ratios they instead compare absolute deviations from one. Intervals are exploratory 95% paired t_4 intervals without multiplicity adjustment.

Metric	Finite-only+R	Core+R	Paired change [95% CI]
Decoded endpoint L2 ↓	8.466 ± 0.801	8.509 ± 0.867	$+0.043$ $[-0.054,\ 0.140]$
Decoded R, T_x gap cosine ↑	0.121 ± 0.074	0.146 ± 0.086	$+0.025$ $[-0.001,\ 0.050]$
Decoded R, T_y gap cosine ↑	0.125 ± 0.020	0.153 ± 0.046	$+0.028$ $[-0.008,\ 0.063]$
Decoded T_x, T_y false gap ↓	0.691 ± 0.433	0.366 ± 0.197	-0.326 $[-0.662,\ 0.011]$
Closure ↓	0.660 ± 0.047	0.523 ± 0.066	-0.136 $[-0.211,\ -0.061]$
R, T_x bracket cosine ↑	0.466 ± 0.264	0.631 ± 0.327	$+0.166$ $[0.048,\ 0.283]$
R, T_y bracket cosine ↑	0.477 ± 0.215	0.752 ± 0.126	$+0.275$ $[0.111,\ 0.439]$
R, T_x norm ratio → 1	0.166 ± 0.064	0.219 ± 0.074	-0.053 $[-0.086,\ -0.019]$
R, T_y norm ratio → 1	0.161 ± 0.073	0.208 ± 0.077	-0.047 $[-0.064,\ -0.030]$
Reconstruction BCE ↓	20.207 ± 3.359	20.217 ± 3.355	$+0.010$ $[-0.001,\ 0.021]$

Core P2b improves all six reported structural metric means over finite-only P2b. The rollout extension has lower mean decoded endpoint error and better closure and nonzero-bracket metrics than Core P2b, with a slightly larger commuting residual. Finite-only P2b still has the lowest mean decoded endpoint error; improvements are not uniform across components or metrics.

Contribution of bracket losses beyond rollout. With decoded rollout and the remaining training settings matched, adding prescribed-bracket losses improves closure, bracket direction and magnitude agreement on dSprites. Decoded order assignment and gap-magnitude errors also improve in all five seeds, while the mean commuting false gap decreases. However, mean endpoint error and reconstruction BCE increase slightly. These results identify benefits beyond rollout for algebraic and order-effect agreement, rather than uniform improvements in reconstruction or endpoint accuracy (Table 15).

F.4 Sensitivity to the grounding-label budget

We assess how the number of action-labeled grounding examples affects the reduced-label pipeline. We compare 32, 64 and 256 anchors per action on dSprites, 3DShapes and ModelNet10 over five matched seeds. Within each seed, we fix the P1 parent, anonymous discovery checkpoint, grounding and refinement update budgets, optimizer settings, loss weights and development evaluation records. Anchor subsets are deterministic, nested and drawn only from training records. The 64-anchor results reuse the original LF+64 runs.

The 32-anchor setting is the smallest common subset satisfying the grounding procedure's validity checks across all three datasets and five seeds. A proposed 16-anchor setting was excluded before training because some 3DShapes and ModelNet10 seeds had insufficient correctly assigned calibration

Table 16: Grounding-label sensitivity on development records (mean±sample SD; five seeds). Columns give action-labeled anchors per action. Bold marks the best mean, not statistical significance; norm ratios target one. Row scales apply to both means and SDs.

Metric	32/action	64/action	256/action
dSprites			
Decoded endpoint L2 ↓	**8.929**±0.375	8.977±0.448	9.032±0.420
Decoded order-gap cosine ↑	**0.116**±0.069	0.108±0.075	0.061±0.031
Decoded false gap L2 ↓ ($\times 10^{-2}$)	7.643±0.689	**5.524**±2.922	6.632±1.735
Equal-term closure ↓	**0.441**±0.153	0.454±0.110	0.605±0.035
3DShapes			
Decoded endpoint L2 ↓	10.948±1.920	**8.734**±0.114	11.365±2.276
Decoded action cosine ↑	0.010±0.037	**0.360**±0.053	0.018±0.042
Decoded false gap L2 ↓ ($\times 10^{-2}$)	7.154±4.396	**1.074**±0.072	7.952±4.032
Finite-action commutator norm ↓ ($\times 10^{-3}$)	4.161±3.927	**0.067**±0.042	4.075±3.344
ModelNet10			
Decoded endpoint L2 ↓	14.113±0.122	14.112±0.139	**14.085**±0.102
Decoded order-gap cosine ↑ ($\times 10^{-2}$)	−0.033±0.296	**0.552**±0.971	0.412±0.816
Order assignment ↑	0.496±0.005	**0.516**±0.025	0.506±0.030
Cyclic closure ↓	0.577±0.100	**0.548**±0.092	0.631±0.068
Mean bracket cosine ↑	**0.513**±0.167	0.396±0.065	0.438±0.118
Mean bracket norm ratio → 1	0.400±0.062	**0.461**±0.069	0.357±0.054

anchors. Thus, the study measures sensitivity to anchor count under a fixed training recipe, without retuning each budget.

Table 16 shows that increasing the anchor budget does not yield monotonic improvements. On 3DShapes, 64 anchors gives the best mean on every reported metric. On ModelNet10, it gives the best mean decoded order-gap cosine, order assignment, closure and bracket norm ratio; 256 anchors gives slightly lower endpoint error, while 32 gives higher bracket cosine. On dSprites, 32 anchors gives slightly better mean endpoint, direction and closure results, whereas 64 better suppresses the unwanted translation–translation order gap. Overall, 64 anchors provides an effective shared setting, while additional grounding labels alone do not consistently improve the fixed reduced-label pipeline.