

Discovering Control Symmetries via Transition Manifold Tangency

Myriam Schönenberger Kee-Eung Kim

Kim Jaechul Graduate School of AI, KAIST

mschoene.ai@gmail.com kekim@kaist.ac.kr

Abstract: Symmetries are common structural properties of robotic systems. They provide an inductive bias for sample-efficient learning and generalization, but discovering them from logged transitions requires separating the structure of the dynamics from the distribution induced by a behavior policy. We introduce *Transition-Manifold Tangency* (TMT), a geometric framework for jointly estimating linear symmetry transformations of states and actions from controlled transition data. For smooth deterministic dynamics, infinitesimal equivariance becomes a tangency condition on the transition manifold, yielding a spectral criterion that targets geometry without assuming that the behavior-induced transition distribution is invariant under the dynamics symmetries. We instantiate TMT with denoiser-based (TMT-NN) and local-PCA (TMT-KNN) estimators. Experiments demonstrate that TMT-NN accurately estimates continuous symmetries under nonlinear dynamics and a range of uneven sampling conditions, outperforming the compared methods on robotic data. Regularizing a robot dynamics model with the final discovered transformations improves prediction at orientations absent from training. We further extend TMT to discrete symmetry discovery using a manifold-consistency criterion, with diagnostic results pointing to improved search as a promising route to more effective recovery.

Keywords: Symmetry Discovery, Equivariance, Robot Learning, Transition Manifold, Score Matching

1 Introduction

Symmetries are a common source of structure in robotic systems. They relate configurations and controls across which experience can be shared, allowing equivariant models to improve sample efficiency and generalization [1, 2, 3]. Exploiting this structure typically requires a designer to specify the symmetry group and its actions on states and controls. That requirement is restrictive when learning from logged interactions whose coordinates do not make the relevant transformations apparent, or from systems whose symmetry structure is not known in advance. We ask whether the joint action of a symmetry on states and controls can instead be inferred from observed transitions.

The central difficulty is to separate symmetries of the dynamics from regularities introduced by the behavior policy. A rotationally symmetric system need not generate a rotationally invariant dataset when its controller favors particular directions or configurations. A symmetry can preserve the validity of transitions while changing their relative frequencies. Controlled dynamics also impose a coupled representation constraint: the same state transformation must act before and after a transition, while the control input may transform differently. For a planar arm with workspace-rotation symmetry, the rotation transforms state coordinates while leaving joint-torque actions unchanged. Discovery must respect this coupled transition structure while accommodating unequal sampling of symmetry-related transitions.

Existing approaches recover symmetry generators from derivatives of a learned function [4, 5] or seek transformations that preserve the observed distribution [6]. The former depend on derivative accuracy, while the latter impose a distributional condition that a dynamics symmetry need not satisfy. Manifold-based methods instead relate symmetry to the geometry of the data support [7, 8]. We develop this geometric perspective for a coupled identification problem: infer compatible state and action representations of a dynamics symmetry from a transition distribution that need not share that symmetry.

We introduce *Transition-Manifold Tangency* (TMT), a framework for discovering linear symmetry actions in the observed state and action coordinates. For smooth deterministic dynamics $s' = f(s, a)$, valid transitions lie on the graph $\mathcal{M} = \{(s, a, f(s, a))\}$. A continuous symmetry preserves this *transition manifold*, so its infinitesimal motion must lie in the tangent space at each transition. The same state generator acts on both s and s' , with a separate generator acting on a . Because this motion is linear in the unknown generators, its squared normal component defines a quadratic objective. A generalized eigenvalue problem then supplies candidate directions for joint state and action transformations.

We estimate this geometry directly from transition data. TMT-NN learns a multi-scale denoiser whose Jacobian approaches the local tangent projector at the population optimum as noise vanishes, while TMT-KNN uses local principal component analysis as a training-free reference. Neither estimator requires fitting the forward dynamics function. The geometric target does not require a symmetry-invariant transition distribution, but informative coverage is necessary to distinguish true from spurious generators. Finite-data estimates remain sensitive to sampling density and curvature. We refine and prune the resulting candidate directions before using them to regularize a learned dynamics model.

Contributions. (i) **Controlled symmetry discovery.** We formulate dynamics equivariance as transition-manifold tangency with coupled state and action generators, yielding a generalized eigenvalue problem for candidate transformations. (ii) **Geometry estimation.** We develop a multi-scale denoiser estimator and a local-PCA reference that estimate the local geometric constraints needed to assess whether candidate transformations preserve valid transitions, without fitting a forward dynamics model. (iii) **Discovery and predictive utility.** Experiments demonstrate accurate continuous-symmetry estimates across a range of nonlinear dynamics and uneven sampling conditions. After refinement and pruning, the discovered transformations improve a robot dynamics model’s predictions at orientations absent from training. (iv) **Discrete extension and diagnostics.** We extend the criterion to discrete symmetry discovery through manifold consistency. Further evaluations show that reference transformations can satisfy the TMT-NN criterion despite being missed by the search, motivating more effective search strategies.

2 Related Work

Exploiting symmetry in robot learning. Equivariant architectures exploit prescribed symmetry transformations to improve sample efficiency in visual manipulation [1] and planar grasp learning [9]. Mirroring methods encourage symmetric locomotion [10], while Mittal et al. [2] distinguish task symmetry from gait symmetry and evaluate symmetry augmentation in locomotion and manipulation. MDP homomorphic networks formalize joint state–action symmetry and construct equivariant layers from supplied transformations [3]. Residual pathway priors favor equivariant structure while allowing violations when the symmetry is approximate or misspecified [11]. These methods establish the value of known symmetry transformations. TMT addresses the preceding problem of estimating compatible state and action transformations from observed transitions.

Discovering symmetries of functions. Input–output data do not directly provide the derivatives needed to formulate symmetry constraints, motivating the use of a trained differentiable network as an intermediate model. LieGG first trains a network on data, then extracts the continuous symmetry generators encoded by the network from a polarization matrix built from its input gradients [4].

LieSD uses the Jacobian of a trained network to recover compatible input and output transformations through infinitesimal equivariance constraints [5]. Its framework supports structured representations and nonuniform sampling. Our LieSD baseline first trains a network to predict the next state from the current state and action, then uses its Jacobians to estimate a shared state transformation and a separate action transformation (Sec. 5). Discovery therefore depends on how faithfully the derivatives of the predictor represent the underlying dynamics. TMT estimates these transformations from the geometry of valid transitions without training a next-state predictor.

Geometric symmetry discovery. Lie PCA estimates linear symmetries of a data manifold through a spectral operator assembled from local tangent estimates [7]. Otto et al. [8] generalize geometric symmetry discovery to nonlinear group actions and input–output graphs, with tangency objectives defined under sampling distributions with full support on the manifold. Their framework accommodates different domain and codomain representations and does not require a symmetry-invariant sampling distribution. TMT uses this established geometric principle with the block structure of controlled transitions, sharing the state transformation between the current and successor states. TMT-KNN follows the local-PCA route, while TMT-NN estimates the required geometry with a denoiser. Beyond linear actions, Shaw et al. [12] estimate level-set functions and fit vector fields that preserve them, while Ko et al. [13] learn nonlinear infinitesimal generators by integrating their flows and optimizing task-specific validity scores. These approaches admit richer transformation families than the linear state and action transformations considered here, offering useful directions for future extensions of TMT to nonlinear symmetry actions.

Estimating manifold geometry with denoising. Stanczuk et al. [14] show that, under regularity assumptions, score directions at nearby off-manifold points converge to normal directions as noise vanishes, and use singular values of sampled score vectors to estimate intrinsic dimension. TMT-NN instead uses a denoiser’s Jacobian to estimate the projector onto directions orthogonal to the transition manifold, providing the local test for candidate infinitesimal motions. The projector interpretation is justified at the population optimum as noise vanishes (Appendix B).

Distribution-based symmetry discovery. LieGAN learns generators by adversarially matching the joint input–output distribution before and after sampled transformations [6]. A dynamics symmetry preserves which transitions are valid, but need not preserve how frequently a behavior policy visits them. Consequently, distribution matching can penalize valid dynamics symmetries when sampling frequencies change along their orbits. TMT instead tests preservation of transition geometry, an aim shared with earlier geometric discovery methods. Its target enables symmetry discovery even when symmetry-related transitions are sampled at unequal frequencies, given transition data with informative coverage.

3 Background

Controlled transitions and equivariance. We observe $\mathcal{D} = \{(s_i, a_i, s'_i)\}_{i=1}^N$ from smooth deterministic dynamics $f : \mathcal{U} \rightarrow \mathbb{R}^n$, where $\mathcal{U} \subseteq \mathbb{R}^{n+m}$ is a smooth embedded manifold of admissible state–action pairs. The transition distribution p need not be symmetry-invariant. A linear dynamics symmetry comprises a Lie group G with smooth representations $\rho_s : G \rightarrow \text{GL}(\mathbb{R}^n)$ and $\rho_a : G \rightarrow \text{GL}(\mathbb{R}^m)$ that jointly preserve $\mathcal{U}$ and satisfy the equivariance condition

$$f(\rho_s(g)s, \rho_a(g)a) = \rho_s(g)f(s, a) \quad \forall g \in G, (s, a) \in \mathcal{U}. \quad (1)$$

The transformations are assumed to be linear, but f may be nonlinear. Reward preservation for MDP symmetries is outside our scope [3].

Continuous symmetries and their generators. For a one-parameter subgroup g_t , $\rho_s(g_t) = \exp(tL_s)$ and $\rho_a(g_t) = \exp(tL_a)$, with state and action generators L_s and L_a , respectively. For open $\mathcal{U}$, differentiating (1) at $t = 0$ yields

$$J_s(s, a)L_s s + J_a(s, a)L_a a - L_s f(s, a) = 0, \quad J_s := \partial_s f, \quad J_a := \partial_a f. \quad (2)$$

This is a special case of the LieSD constraint [5] with input generator $\text{diag}(L_s, L_a)$ and output generator L_s . When admissible state-action pairs lie on a lower-dimensional manifold $\mathcal{U}$, the condition is $Df[(L_s s, L_a a)] = L_s f(s, a)$, where Df is the differential at (s, a) and $(L_s s, L_a a) \in T_{(s,a)}\mathcal{U}$.

Transition geometry and tangency. Let $x = [s; a; s'] \in \mathbb{R}^d$, $d = 2n + m$. The transition manifold is the graph of f , $\mathcal{M} = \{((s, a), f(s, a)) : (s, a) \in \mathcal{U}\}$. Let $\Pi_x^\parallel$ project orthogonally onto $T_x\mathcal{M}$ and $\Pi_x^\perp = I_d - \Pi_x^\parallel$ onto its normal space. The normal rank is $d - \dim\mathcal{U}$, equal to n for open $\mathcal{U}$ and greater than n for lower-dimensional $\mathcal{U}$. For a candidate pair $L = (L_s, L_a)$, a transition (s, a, s') is transformed into $(\exp(tL_s)s, \exp(tL_a)a, \exp(tL_s)s')$, with infinitesimal motion $X(x; L) = [L_s s; L_a a; L_s s']$.

The following proposition specializes the tangency characterization of manifold symmetries in Otto et al. [8] to controlled transitions, with a shared state generator and a separate action generator.

Proposition 1 (Tangency condition). *If the state and action transformations preserve $\mathcal{U}$ for all t , equivariance holds if and only if*

$$\Pi_x^\perp X(x; L) = 0 \quad \text{for every } x \in \mathcal{M}. \quad (3)$$

A proof is given in Appendix A. Both continuous and discrete symmetries must keep valid transitions on the transition manifold. Continuous symmetries are characterized by the tangency condition [8], while discrete candidates are evaluated by testing manifold preservation directly.

4 Method

From tangency to a quadratic objective. Column-wise vectorization of the generator matrices L_s and L_a gives the exact representation

$$X(x; L) = \tilde{C}(x)v, \quad \tilde{C}(x) := \begin{bmatrix} s^\top \otimes I_n & 0 \\ 0 & a^\top \otimes I_m \\ s'^\top \otimes I_n & 0 \end{bmatrix}, \quad v := \begin{bmatrix} \text{vec}(L_s) \\ \text{vec}(L_a) \end{bmatrix}.$$

Here $\tilde{C}(x) \in \mathbb{R}^{d \times (n^2 + m^2)}$ and $v \in \mathbb{R}^{n^2 + m^2}$. We seek nonzero generators by minimizing the expected squared normal component:

$$\mathcal{L}(v) := \mathbb{E}_{x \sim p} \|\Pi_x^\perp \tilde{C}(x)v\|^2 = v^\top M_{\text{TMT}} v. \quad (4)$$

Here $M_{\text{TMT}} := \mathbb{E}_{x \sim p} [\tilde{C}(x)^\top \Pi_x^\perp \tilde{C}(x)] \succeq 0$. We seek its zero eigenspace, since true generators have zero loss. We normalize generators by fixing the expected squared magnitude of the induced infinitesimal motion, $v^\top \mathcal{N} v = \mathbb{E}_{x \sim p} \|X(x; L)\|^2 = 1$. Minimizing Eq. (4) under this normalization yields

$$M_{\text{TMT}} v = \lambda \mathcal{N} v, \quad \mathcal{N} := \mathbb{E}_{x \sim p} [\tilde{C}(x)^\top \tilde{C}(x)]. \quad (5)$$

At $\lambda = 0$, this reduces to the nullspace condition $M_{\text{TMT}} v = 0$. For $\lambda > 0$, the eigenvalue is the ratio of the expected squared magnitude of the normal component to that of the full induced infinitesimal motion, quantifying departure from tangency: with estimated geometry, these eigenvalues need not be exactly zero, so we select directions with small generalized eigenvalues as candidates.

In comparison, the LieSD evaluates Eq. (2) using Jacobians of a fitted dynamics model $\hat{f}$. For open $\mathcal{U}$ and exact $\hat{f} = f$, its population criterion has the same null space as Eq. (4), differing by a pointwise positive-definite weighting (Appendix C). In TMT, $\tilde{C}(x)$ is known from each transition, leaving the normal projector $\Pi_x^\perp$ and the resulting matrix M_{TMT} to be estimated.

TMT-KNN: local-PCA geometry estimation. For the K nearest neighbors x_{ij} of x_i , we form the local second-moment matrix, normal projector, and empirical tangency matrix:

$$\hat{\Sigma}_i = \frac{1}{K} \sum_{j=1}^K (x_{ij} - x_i)(x_{ij} - x_i)^\top, \quad \hat{\Pi}_{x_i}^\perp = \sum_{\ell=1}^n u_{i\ell} u_{i\ell}^\top, \quad \hat{M}_{\text{TMT}} := \frac{1}{N} \sum_{i=1}^N \tilde{C}(x_i)^\top \hat{\Pi}_{x_i}^\perp \tilde{C}(x_i).$$

Here $u_{i1}, \dots, u_{id}$ are orthonormal eigenvectors of $\widehat{\Sigma}_i$ ordered by increasing eigenvalue. Local PCA estimates tangent directions from the largest eigenvalues and normal directions from the smallest [15]. This estimator requires no training, but its accuracy depends on density, neighborhood radius, curvature, and finite-sample error. The limitation is that selecting n normal directions captures the full normal space only when state–action coordinates are locally unconstrained (Sec. 3). Additional constraints increase the codimension, so this choice captures only part of the normal space.

TMT-NN: denoiser-based geometry estimation. To reconstruct each transition x from its corrupted input $x + \sigma\xi$, we train a multi-scale denoiser $d_\phi(\cdot; \sigma)$ with parameters ϕ using the denoising score-matching objective [16], $\mathcal{L}_{\text{DSM}}(\phi) = \mathbb{E}_{x, \sigma, \xi} \|d_\phi(x + \sigma\xi; \sigma) - x\|^2$, where $\xi \sim \mathcal{N}(0, I_d)$ and $\sigma \in [\sigma_{\min}, \sigma_{\max}]$. During symmetry discovery, we evaluate its Jacobian with respect to the first input at unperturbed transitions $x \in \mathcal{M}$ to estimate geometry. For the population-optimal denoiser $d^*(\cdot; \sigma)$, Tweedie’s identity [17] gives $d^*(x; \sigma) = x + \sigma^2 \nabla \log p_\sigma(x)$ and $Jd^*(x; \sigma) = I_d + \sigma^2 \nabla^2 \log p_\sigma(x)$, where p_σ is the density of noise-perturbed transitions. Under local smoothness and positive density, $Jd^*(x; \sigma) \rightarrow \Pi_x^\parallel$ and $I_d - Jd^*(x; \sigma) \rightarrow \Pi_x^\perp$ as $\sigma \rightarrow 0$ (Appendix B). At finite noise, curvature of $\mathcal{M}$ and variation in the sampling density along $\mathcal{M}$ bias this approximation. A learned denoiser introduces estimation error, particularly in sparsely covered regions.

We approximate Eq. (4) at each noise scale by replacing the expectation with a sample average and the normal projector with $I_d - J_{i, \sigma}$, where $J_{i, \sigma} := Jd_\phi(x_i; \sigma)$. This gives

$$\widehat{\mathcal{L}}_\sigma(v) := \frac{1}{N} \sum_{i=1}^N \|(I_d - J_{i, \sigma}) \widetilde{C}(x_i) v\|^2 = v^\top \widehat{M}_\sigma v.$$

We construct these residual Gram matrices and combine them across noise scales as follows:

$$B_{i, \sigma} := (I_d - J_{i, \sigma}) \widetilde{C}(x_i), \quad \widehat{M}_\sigma := \frac{1}{N} \sum_{i=1}^N B_{i, \sigma}^\top B_{i, \sigma}, \quad \widehat{M}_{\text{TMT}} := \frac{1}{S} \sum_{\sigma} \frac{\widehat{M}_\sigma}{\text{tr}(\widehat{M}_\sigma)}. \quad (6)$$

We choose trace normalization to give each per-scale matrix unit trace, then average equally over $S = 5$ noise scales. The Gram construction ensures positive semidefiniteness even when $I_d - J_{i, \sigma}$ is not an orthogonal projector (Appendix G.9). We retain the raw learned Jacobians without projecting them onto a fixed-rank subspace.

Ground-truth-free dimension selection. We qualify eigenvalues satisfying $\lambda_j \leq 0.1\lambda_{\max}$, cut at the largest adjacent log-gap when its ratio exceeds a fixed threshold, and cap the selected count at 8 (Appendix F). This rule uses no ground-truth dimension. The eigenproblem supplies a candidate subspace, which we widen by four eigenvectors before finite-orbit refinement with random restarts and subsequent pruning for deployment (Appendix G.11). On Reacher, this reduces the 104-dimensional generator space to five to nine dimensions before refinement. Informative coverage is essential for identifiability: a spurious direction can be excluded only by transitions that violate its tangency condition. Positive density reweighting on the same support preserves the ideal zero-loss directions, whereas missing support can leave additional directions indistinguishable.

Discrete symmetries via manifold distance. A finite candidate $g = (g_s, g_a)$ acts as $g \cdot x = (g_s s, g_a a, g_s s')$. Its validity requires preservation of $\mathcal{M}$. At a fixed small noise scale, the denoiser residual provides a local proxy for manifold distance, $\|d_\phi(y; \sigma) - y\| \approx \text{dist}(y, \mathcal{M})$. We score transformed transitions through

$$D_{\text{mfd}}(g) := \mathbb{E}_{x \sim p} \|d_\phi(g \cdot x; \sigma) - g \cdot x\|^2, \quad \mathcal{L}_{\text{disc}}(g) = D_{\text{mfd}}(g) - \mu \|g - I\|_F^2, \quad (7)$$

where the second term encourages nonidentity candidates. We parameterize permutations with Gumbel–Sinkhorn relaxations [18] and learn one sign per coordinate. Several candidates are optimized in parallel with log-Frobenius repulsion, then rounded to signed permutations. The score targets transition geometry rather than invariance of the sampling distribution, so it does not require

$g \cdot p = p$. Its reliability nevertheless depends on coverage near transformed transitions and on the denoiser’s finite-noise accuracy. The exact-recovery datasets use random behavior policies. Medium and trained-policy datasets enter only the candidate-score comparison in Appendix G.6. The KNN variant replaces the denoiser residual with distance to the nearest untransformed neighbors.

5 Experimental Setup

Code and datasets: <https://github.com/mschoene/tmt>. We evaluate continuous symmetries while varying dynamics nonlinearity and behavior-policy bias, and discrete symmetries across group structures and dimensions. Appendix E specifies the state and action coordinates.

Continuous-symmetry environments. **2D Particle** is a planar point mass in a disc, with state $(x, y, \dot{x}, \dot{y}) \in \mathbb{R}^4$, force action $(F_x, F_y) \in \mathbb{R}^2$, and a one-dimensional continuous $\text{SO}(2)$ symmetry. We vary dataset size N , behavior-policy concentration κ , and drag c_d in $\dot{v} = F - c_d \|v\| v$. The nonlinear drag preserves rotation symmetry, as do the speed cap, action clip, and boundary (Appendix E). **3D Particle** extends this construction to $\mathbb{R}^3$, with three $\text{SO}(3)$ generators and linear rather than quadratic drag. **Reacher** (Minari [19]) is a two-link planar arm with $n = 10$, $m = 2$. Its observations combine joint-angle sines and cosines, joint velocities, target position, and fingertip-to-target offset. Workspace rotation about the base joint is an exact $\text{SO}(2)$ symmetry that leaves both joint torques invariant.

Discrete-symmetry environments. **CartPole** ($n = 4, m = 1$) has a $\mathbb{Z}_2$ sign flip of the full state and action. **Swimmer** ($n = 8, m = 2$, native joint-angle observation) has an exact y -reflection symmetry. The simulator audit rules out the joint-swap candidate (Appendix G.7). **TriFinger** [20] ($n = 30, m = 9$) has a $\mathbb{Z}_3$ cyclic finger permutation in position- and torque-controlled variants. Finger-cube contact makes this symmetry approximate in the default observation, which omits the cube state. We additionally evaluate a cube-free variant verified by a simulator audit (Appendix G.8).

Baselines. **LieSD** [5] uses a learned dynamics Jacobian. We instantiate its structured input–output formulation with a shared state representation and a separate action representation. A residual predictor $\hat{f}(s, a) = s + \Delta t \hat{F}_\theta(s, a)$ supplies autodiff Jacobians for Eq. (2). The constraints are stacked as $C^{\text{LieSD}}(x_i) v = 0$, with $v = [\text{vec}(L_s); \text{vec}(L_a)]$, and generators are extracted from the null space of $\frac{1}{N} \sum_i C^{\text{LieSD}}(x_i)^\top C^{\text{LieSD}}(x_i)$. This construction depends on derivative accuracy.

LieGAN [6] matches distributions on joint (s, a, s') observations with separate state and action representations. Transformation magnitudes are sampled from $\mathcal{U}[-\alpha, \alpha]$, with learnable α bounded away from zero. Our adaptation separately normalizes the state and action blocks to unit Frobenius norm, excluding an exactly invariant action block. A relaxed variant removes this constraint (Appendix D). Distribution matching assesses preservation of sampling frequencies, which differs from dynamics equivariance.

Metrics. *Space error* ($\mathcal{E}^{\text{space}}$) combines principal-angle misalignment with a unit penalty for each unmatched independent direction in the state or action span. The blockwise aggregation and returned counts are detailed in Appendix F. Comparisons must account for these counts. *Subspace recovery* is the angle θ between the leading direction and the ground-truth generator, in degrees. For one-dimensional spans, the squared-sine misalignment is $\sin^2 \theta$. *Closure error* ($\mathcal{E}^{\text{cl}}$) measures the pairwise Lie brackets remaining outside the discovered span [21]. It assesses algebraic consistency, not equivariance. *Exact discrete recovery: top-1* requires the highest-ranked accepted nonidentity candidate to equal a target element. *Full group* requires the accepted candidates to generate exactly the target group. All four methods share search and acceptance procedures but use different scores (Appendix G.6). *Simulator equivariance* is $E_{\text{sim}}(g) := \mathbb{E} \|g_s s' - f_{\text{sim}}(g_s s, g_a a)\|_2^2$ on fresh transitions. For a continuous candidate, we average over $g = \exp(tL)$ at $t = \pm 0.1, \pm 0.3, \pm 0.5, \pm 1$, corresponding to roughly 2° – 23° for the deployed Reacher generators.

Table 1: Reacher symmetry discovery (mean $\pm$ std over 10 seeds, lower is better). Bold marks column minima. θ and $\|L_a\|_F$ describe the leading direction selected without ground truth, with θ computed post hoc. The true action generator is zero. Closure error uses the entire returned set.

Method	θ (deg)	$\ L_a\ _F$	$\mathcal{E}^{cl}$
TMT-NN	5.8 ± 2.2	0.007 ± 0.002	0.701 ± 0.264
TMT-KNN	84.1 ± 4.2	0.081 ± 0.028	0.945 ± 0.012
LieSD	67.9 ± 4.4	0.111 ± 0.049	0.934 ± 0.030
LieGAN	89.7 ± 0.3	0.707 ± 0.000	0.942 ± 0.043

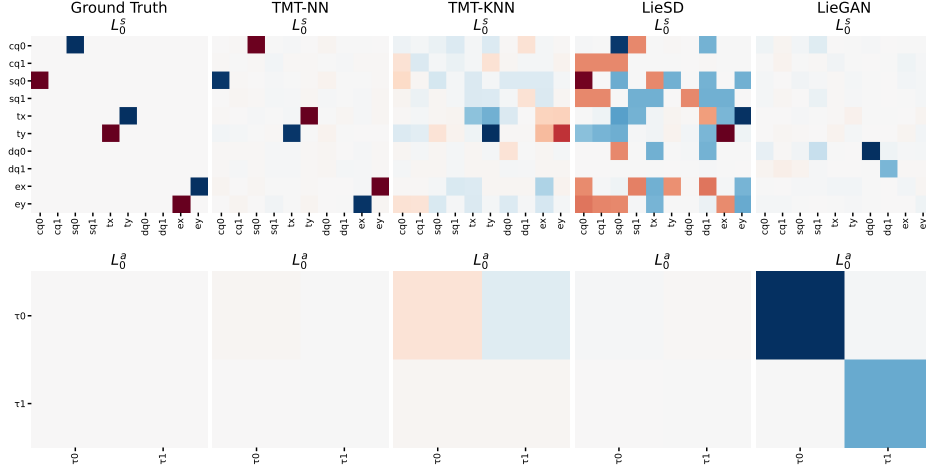


Figure 1: Representative leading continuous-generator directions on Reacher for one seed, against the ground-truth $SO(2)$ workspace rotation. The top row shows state block L_s^0 and the bottom row shows action block L_a^0 , with ground truth $L_a^* = 0$. Particle-environment counterparts are in Appendix G.1.

6 Experimental Results

We evaluate continuous discovery on Reacher (Sec. 6.1), nonlinearity and policy-bias sweeps on particle systems (Sec. 6.2), and discrete discovery (Sec. 6.3). Unless stated otherwise, each condition uses $N = 100,000$ transitions and 10 seeds.

6.1 Continuous Discovery: Comparison on Reacher

Table 1 reports Reacher discovery results. TMT-NN’s leading direction is about 6° from the true generator, compared with 68° – 90° for the baselines, and its action-block norm is an order of magnitude smaller. LieGAN remains nearly orthogonal across seeds. Its fixed $\|L_a\|_F = 1/\sqrt{2}$ is imposed by separate block normalization, although relaxing this constraint does not recover the symmetry (Appendix D). Reacher also violates the full-dimensional observation assumption of TMT-KNN: two Pythagorean identities and the fingertip-offset constraints increase the true codimension, while the estimator retains only n normal directions. Figure 1 illustrates the leading blocks for one seed.

Downstream dynamics prediction. We train a Reacher predictor with soft equivariance penalty $\lambda \|\hat{f}(g_s s, g_a a) - g_s \hat{f}(s, a)\|_2^2$, $\lambda = 3$. Training uses the default initial-pose range of ± 0.1 rad. Evaluation co-rotates the arm start and target through 12 orientations spaced 30° . Candidate selection, refinement, and pruning use no ground truth and are applied identically to all methods. The penalty weight was initially tuned with the analytic generator and checked separately on a discovered candidate (Appendix G.11).

Table 2: Reacher dynamics-prediction generalization over 10 seed-matched runs, median $[Q_1, Q_3]$ (lower is better). Angles are measured post hoc. Candidate selection, refinement, and pruning use no ground truth (Appendix G.11).

Regularizer	Angle (deg)	One-step MSE	50-step rollout MSE
None	—	.050 [.039, .088]	184.0 [159.0, 276.0]
True generator	0.0	.003 [.003, .004]	5.2 [4.5, 7.4]
TMT-NN (raw)	77.4 [74.5, 82.0]	.055 [.044, .084]	330.0 [247.0, 460.0]
TMT-NN (refined)	6.2 [5.7, 7.0]	.015 [.012, .018]	22.7 [20.6, 25.8]
TMT-NN (refined + pruned)	2.9 [1.1, 3.2]	.004 [.003, .006]	5.9 [4.7, 6.9]
TMT-KNN (refined + pruned)	82.1 [78.0, 87.6]	.012 [.006, .015]	30.0 [20.9, 43.6]
LieSD (refined + pruned)	49.1 [48.6, 49.6]	.008 [.008, .010]	20.1 [15.3, 29.0]
LieGAN (refined + pruned)	90.0 [90.0, 90.0]	.067 [.036, .094]	266.2 [205.2, 302.5]

Table 2 reports 10 seed-matched runs, with generator angles measured post hoc. This deployment pipeline is implemented separately from the discovery sweep, so its angles are not directly comparable with Table 1 (Appendix G.11). Refinement and pruning move TMT-NN’s median angle from 77° to 3° and reduce prediction errors. TMT-KNN and LieSD reduce rollout error by one to two orders of magnitude but remain above TMT-NN. LieGAN remains within half a degree of orthogonality and its prediction errors increase across stages.

Predictive improvement alone does not validate a symmetry. TMT-KNN at 82° and LieSD at 49° outperform the unregularized predictor, while raw TMT-NN at 77° does not, suggesting benefits from regularization beyond accurate symmetry recovery. TMT-NN’s one-step median remains above the true-generator result. Overlapping interquartile ranges do not establish statistical equivalence. Held-out simulator checks support eight of the ten deployed candidates, with residuals between 10^{-7} and 4×10^{-5} , but identify two failures at 0.18 and 0.49, four orders of magnitude higher (Appendix G.11). These have the largest angles, 7.4° and 17.0° , versus under 3.3° for the others. The worse candidate also gives the highest rollout error. The orbit score does not flag these failures, which require simulator access or ground-truth comparison to identify.

6.2 Controlled Stress Tests on the Particle Environments

Particle experiments vary nonlinearity and policy bias separately (Table 3). Full sweeps are in Appendix G.

Dynamics derivatives. The drag sweep preserves exact $SO(2)$ symmetry. TMT-NN returns a single direction within 3° of the generator throughout, whereas learned LieSD remains unreliable. Oracle LieSD with exact dynamics Jacobians recovers the generator at every drag value, including $c_d = 0$, where the learned version has median angle 77° . This comparison implicates derivative estimation rather than nonlinearity alone.

Asymmetric sampling. We vary von Mises concentration from unbiased behavior ($\kappa = 0$) to concentrated force directions ($\kappa = 10$). Figure 3 illustrates the resulting orbit coverage in 3D. TMT-NN’s 2D space error stays below 0.01 throughout. Its 3D error remains low through $\kappa = 5$, but increases to 3.76 at $\kappa = 10$, above both LieGAN and LieSD.

LieGAN’s full-set space error changes little across κ . At $\kappa = 0$, its leading direction has angle $18.3^\circ \pm 5.8^\circ$, an order of magnitude larger than TMT-NN’s but far from orthogonal. The full-set metric does not establish how leading-direction alignment changes under bias. Reacher provides a separate outcome: its leading direction remains nearly orthogonal under random-policy data.

6.3 Discrete Symmetry Discovery

On CartPole, LieGAN recovers the exact $\mathbb{Z}_2$ group in 9/10 seeds, TMT-NN in 4/10, and LieSD in 1/10. TMT-KNN ranks the correct reflection first in 8/10 seeds but never returns the exact group,

Table 3: 2D Particle stress tests (mean $\pm$ std, lower is better). Bold marks row minima. The shared base condition appears once. $\mathcal{E}^{\text{space}}$ measures misalignment and unmatched span dimensions, so cross-method comparisons must account for block ranks.

Axis	Level	TMT-NN	TMT-KNN	LieSD	LieGAN
Base	$c_d = 0, \kappa = 0$	0.001 $\pm$ 0.000	1.80 $\pm$ 1.60	6.48 $\pm$ 4.79	3.06 $\pm$ 0.02
Drag c_d (nonlinearity)	1.0	0.001 $\pm$ 0.000	11.00 $\pm$ 0.00	10.11 $\pm$ 2.83	3.12 $\pm$ 0.17
	2.0	0.002 $\pm$ 0.000	11.00 $\pm$ 0.00	10.74 $\pm$ 0.84	3.10 $\pm$ 0.17
Bias κ (asymmetry)	5	0.001 $\pm$ 0.000	2.52 $\pm$ 4.19	8.13 $\pm$ 4.62	3.01 $\pm$ 0.01
	10	0.001 $\pm$ 0.000	3.70 $\pm$ 4.95	10.74 $\pm$ 0.83	3.02 $\pm$ 0.01

because additional accepted elements enlarge its closure. On Swimmer, only TMT-KNN recovers the y -reflection, in 2/10 seeds. No method exactly recovers a nonidentity reference matrix of $\mathbb{Z}_3$ on default or cube-free TriFinger data. In the cube-free case, zero-valued coordinates make some reference permutations indistinguishable on the observed support (Appendix G.8). Post hoc scoring on Swimmer and position-controlled TriFinger separates search from acceptance. The TMT-NN criterion accepts the reference element in every evaluated seed when supplied directly, although search misses it. TMT-KNN rejects the TriFinger reference element even when supplied directly. Full recovery and diagnostic results appear in Appendix G.6 and G.10.

7 Conclusion

TMT estimates candidate linear symmetry transformations of states and actions from controlled transitions through transition-manifold tangency. The formulation couples a shared state transformation across current and successor states with a separate action transformation, allowing discovery to target transition validity without requiring a symmetry-invariant sampling distribution. Local-PCA and denoiser estimates of transition geometry lead to a quadratic criterion for candidate generators without first fitting a dynamics model. The denoiser-based estimator identifies useful continuous directions, and refined and pruned transformations improve dynamics prediction at unseen orientations. Experiments under nonlinear dynamics and uneven sampling support the geometric criterion, and the downstream evaluation distinguishes predictive utility from independently checked symmetry validity. The discrete extension provides a manifold-consistency criterion and diagnostics to guide improvements in search and acceptance. Separating search failures from acceptance failures offers a concrete direction for improving discrete recovery.

8 Limitations

Estimation and coverage. The geometric target does not require symmetry-invariant sampling, but finite-noise and finite-data estimates depend on density and curvature. TMT-KNN fixes the normal rank at n , assuming a full-dimensional admissible state-action domain. Limited orbit coverage or action excitation can admit spurious generators. Reliable algebra-dimension selection remains open.

Validation and scope. Neither orbit scores nor Lie-bracket closure certifies equivariance. Discovered transformations therefore require simulator validation before enforcement. We evaluate soft equivariance regularization of a dynamics model. Hard equivariant architectures, policy learning, and reward-preserving MDP symmetries remain untested.

Discrete search and representations. Exact recovery beyond CartPole remains unreliable. Search can miss valid elements, and acceptance rules can reject them. Discovery is restricted to linear actions in observed coordinates. Nonlinear actions, learned representations, and general stochastic or partially observed dynamics remain open.

Acknowledgments

We thank the area chair and reviewers for their valuable feedback. This work was supported by Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2024-00343989, Enhancing the Ethics of Data Characteristics and Generation AI Models for Social and Ethical Learning; No.RS-2024-00397310, Development of an AI Simulator for Creating Transparent Compounds that Can Be Altered for Tactile Sensation; No.RS-2024-00457882, AI Research Hub Project; No.RS-2020-II200940, Foundations of Safe Reinforcement Learning and Its Applications to Natural Language Processing; No.RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST); No. RS-2022-II220311, Development of Goal-Oriented Reinforcement Learning Techniques for Contact-Rich Robotic Manipulation of Everyday Objects).

References

- [1] D. Wang, R. Walters, and R. Platt. SO(2)-equivariant reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7F9c0hdvfk_.
- [2] M. Mittal, N. Rudin, V. Klemm, A. Allshire, and M. Hutter. Symmetry considerations for learning task symmetric robot policies. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.
- [3] E. van der Pol, D. Worrall, H. van Hoof, F. A. Oliehoek, and M. Welling. MDP homomorphic networks: Group symmetries in reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [4] A. Moskalev, A. Sepliarskaia, I. Sosnovik, and A. Smeulders. LieGG: Studying learned lie group generators. In *Advances in Neural Information Processing Systems*, volume 35, pages 25212–25223, 2022.
- [5] L. Hu, Y. Li, and Z. Lin. LieSD: Symmetry discovery for different data types. *Neural Networks*, 188:107481, 2025.
- [6] J. Yang, R. Walters, N. Dehmamy, and R. Yu. Generative adversarial symmetry discovery. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39488–39508, 2023.
- [7] J. Cahill, D. G. Mixon, and H. Parshall. Lie PCA: Density estimation for symmetric manifolds. *Applied and Computational Harmonic Analysis*, 65:279–295, 2023. doi:10.1016/j.acha.2023.03.001.
- [8] S. E. Otto, N. Zolman, J. N. Kutz, and S. L. Brunton. A unified framework to enforce, discover, and promote symmetry in machine learning. *Journal of Machine Learning Research*, 26(248): 1–83, 2025.
- [9] X. Zhu, D. Wang, O. Biza, G. Su, R. Walters, and R. Platt. Sample efficient grasp learning using equivariant models. In *Robotics: Science and Systems*, 2022.
- [10] F. Abdolhosseini, H. Y. Ling, Z. Xie, X. B. Peng, and M. van de Panne. On learning symmetric locomotion. In *ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2019.
- [11] M. Finzi, G. Benton, and A. G. Wilson. Residual pathway priors for soft equivariance constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.
- [12] B. Shaw, A. Magner, and K. R. Moon. Symmetry discovery beyond affine transformations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2406.03619.

- [13] G. Ko, H. Kim, and J. Lee. Learning infinitesimal generators of continuous symmetries from data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2410.21853.
- [14] J. P. Stanczuk, G. Batzolis, T. Deveney, and C.-B. Schönlieb. Diffusion models encode the intrinsic dimension of data manifolds. In *International Conference on Machine Learning*, 2024.
- [15] A. Singer and H.-T. Wu. Vector diffusion maps and the connection Laplacian. *Communications on Pure and Applied Mathematics*, 65(8):1067–1144, 2012.
- [16] P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.
- [17] B. Efron. Tweedie’s formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.
- [18] G. Mena, D. Belanger, S. Linderman, and J. Snoek. Learning latent permutations with Gumbel-Sinkhorn networks. In *International Conference on Learning Representations*, 2018.
- [19] O. G. Younis, R. Perez-Vicente, J. U. Balis, W. Dudley, A. Davey, and J. K. Terry. Minari, Sept. 2024. URL <https://doi.org/10.5281/zenodo.13767625>.
- [20] M. Wüthrich, F. Widmaier, F. Grimmering, J. Akpo, S. Joshi, V. Agrawal, B. Hammoud, M. Khadiv, M. Bogdanovic, V. Berenz, J. Viereck, M. Naveau, L. Righetti, B. Schölkopf, and S. Bauer. Trifinger: An open-source robot for learning dexterity. In *Proceedings of the 4th Conference on Robot Learning (CoRL)*, Nov. 2020. URL https://corlconf.github.io/corl2020/paper_421/.
- [21] R. T. Forestano, K. T. Matchev, K. Matcheva, A. Roman, E. B. Unlu, and S. Verner. Deep learning symmetries and their lie groups, algebras, and subalgebras from first principles. *Machine Learning: Science and Technology*, 4(2):025027, 2023. doi:10.1088/2632-2153/acd989.

A Proof of the Tangency Condition

Proof. Fix $L = (L_s, L_a)$ and $g_t = (\exp(tL_s), \exp(tL_a))$. As in Proposition 1, assume that these transformations preserve the smooth admissible domain $\mathcal{U}$ for every $t \in \mathbb{R}$. For $(s, a) \in \mathcal{U}$, set $s' = f(s, a)$ and $x = (s, a, s') \in \mathcal{M}$. The transformed transition is

$$g_t \cdot x = (\exp(tL_s)s, \exp(tL_a)a, \exp(tL_s)s'). \quad (8)$$

It belongs to $\mathcal{M}$ for every t if and only if

$$f(\exp(tL_s)s, \exp(tL_a)a) = \exp(tL_s)f(s, a) \quad \forall t \in \mathbb{R}, \quad (9)$$

which is Eq. (1) for this subgroup. Differentiation at $t = 0$ gives

$$X(x; L) = [L_s s; L_a a; L_s s']. \quad (10)$$

If Eq. (9) holds, the transformed transitions trace a curve in $\mathcal{M}$. Its derivative therefore lies in $T_x \mathcal{M}$, giving $\Pi_x^\perp X(x; L) = 0$.

Conversely, suppose this tangency condition holds at every $x \in \mathcal{M}$. The tangent space of the graph consists of vectors $(w, Df[w])$ with $w \in T_{(s,a)}\mathcal{U}$. Hence $w = (L_s s, L_a a)$ is admissible and $Df[w] = L_s f(s, a)$. Set $(s_t, a_t) = (\exp(tL_s)s, \exp(tL_a)a)$, which remains in $\mathcal{U}$ by assumption. The chain rule gives $\frac{d}{dt}[\exp(-tL_s)f(s_t, a_t)] = \exp(-tL_s)\{Df[(L_s s_t, L_a a_t)] - L_s f(s_t, a_t)\} = 0$. This quantity is therefore constant and equals $f(s, a)$ at $t = 0$, proving Eq. (9). The argument applies to open or constrained smooth $\mathcal{U}$. If tangency is known only on a restricted region, the conclusion holds on parameter intervals during which the corresponding graph points remain in that region. $\square$

B Tweedie's Identity and the Tangent Projector

Let Z be supported on the smooth embedded manifold $\mathcal{M}$, with a smooth density relative to its volume measure that is positive near the point under consideration. Let $Y = Z + \sigma\xi$, where $\xi \sim \mathcal{N}(0, I_d)$ is independent of Z , and denote the ambient density of Y by p_σ . The population-optimal squared-error denoiser is $d^*(y; \sigma) = \mathbb{E}[Z \mid Y = y]$. Tweedie's identity [17] gives

$$d^*(y; \sigma) = y + \sigma^2 \nabla_y \log p_\sigma(y), \quad Jd^*(y; \sigma) = I_d + \sigma^2 \nabla_y^2 \log p_\sigma(y). \quad (11)$$

The Jacobian limit can be seen directly from the posterior covariance. Differentiating the Gaussian posterior yields $Jd^*(y; \sigma) = \sigma^{-2} \text{Cov}(Z \mid Y = y)$. Fix $x \in \mathcal{M}$ and use a local graph chart $z = x + u + h(u)$, where $u \in T_x \mathcal{M}$, $h(u) \in N_x \mathcal{M}$, and $h(0) = Dh(0) = 0$. Smoothness gives $h(u) = O(\|u\|^2)$. Conditional on $Y = x$, the posterior concentrates at $u = O(\sigma)$, and the rescaled tangent coordinates u/σ converge to a standard Gaussian on $T_x \mathcal{M}$. The normal displacement is of smaller order. Consequently, the posterior covariance divided by σ^2 converges to the orthogonal tangent projector, giving $Jd^*(x; \sigma) \rightarrow \Pi_x^\parallel$ and $I_d - Jd^*(x; \sigma) \rightarrow \Pi_x^\perp$.

This is a local population limit under smoothness and positive density, not a finite-noise identity. Density variation and curvature affect the posterior covariance at finite σ , and learning the denoiser from finitely many transitions adds estimation error. Related small-noise results describe normal alignment of scores at nearby off-manifold points [14]. Neither limiting argument supplies tangent directions in regions absent from the observed support.

C Relationship to LieSD: Mahalanobis-Reweighted Equivalence

We compare LieSD with the ideal orthogonal normal-residual criterion on the same input–output graph. This comparison concerns exact graph geometry and is distinct from the learned, multiscale estimator in Eq. (6).

Setup. Let $\mathcal{U} \subseteq \mathbb{R}^{n+m}$ be open and $\hat{f} : \mathcal{U} \rightarrow \mathbb{R}^n$ be smooth. Consider $x = (s, a, s')$ on $\widehat{\mathcal{M}} = \{(s, a, \hat{f}(s, a)) : (s, a) \in \mathcal{U}\}$, so $s' = \hat{f}(s, a)$. Write $J_s = \partial_s \hat{f}$ and $J_a = \partial_a \hat{f}$. The columns of $V(x) := \begin{bmatrix} I_n & 0 \\ 0 & I_m \\ J_s & J_a \end{bmatrix}$ span $T_x \widehat{\mathcal{M}}$, and its normal space is spanned by

$$U(x) := \begin{bmatrix} -J_s^\top \\ -J_a^\top \\ I_n \end{bmatrix} \in \mathbb{R}^{d \times n}, \quad V^\top U = 0. \quad (12)$$

Since $U^\top U = I_n + J_s J_s^\top + J_a J_a^\top \succ 0$, the orthogonal normal projector of this graph is

$$\widehat{\Pi}_x^\perp = U(x)W(x)U(x)^\top, \quad W(x) := (I_n + J_s J_s^\top + J_a J_a^\top)^{-1} \succ 0. \quad (13)$$

The open joint state–action domain ensures that the graph has dimension $n + m$ and normal rank n . On a constrained domain, additional admissibility constraints change the normal space, so this comparison must be reformulated using intrinsic differentials.

LieSD coefficient matrix. On $\widehat{\mathcal{M}}$, Eq. (2) becomes $C^{\text{LieSD}}(x)v = 0$, where

$$C^{\text{LieSD}}(x) = \begin{bmatrix} s^\top \otimes J_s - s'^\top \otimes I_n & a^\top \otimes J_a \end{bmatrix} \in \mathbb{R}^{n \times (n^2 + m^2)}. \quad (14)$$

For points $x_i \in \widehat{\mathcal{M}}$, write $M^{\text{LieSD}} := N^{-1} \sum_i C^{\text{LieSD}}(x_i)^\top C^{\text{LieSD}}(x_i)$.

Algebraic identity. Using $\widetilde{C}(x)$ from Sec. 4,

$$\begin{aligned} U(x)^\top \widetilde{C}(x) &= \begin{bmatrix} -J_s & -J_a & I_n \end{bmatrix} \begin{bmatrix} s^\top \otimes I_n & 0 \\ 0 & a^\top \otimes I_m \\ s'^\top \otimes I_n & 0 \end{bmatrix} \\ &= \begin{bmatrix} -(s^\top \otimes J_s) + s'^\top \otimes I_n & -(a^\top \otimes J_a) \end{bmatrix} = -C^{\text{LieSD}}(x). \end{aligned} \quad (15)$$

Equations (13) and (15) imply

$$\tilde{C}(x)^\top \widehat{\Pi}_x^\perp \tilde{C}(x) = C^{\text{LieSD}}(x)^\top W(x) C^{\text{LieSD}}(x). \quad (16)$$

The orthogonal normal-residual Gram on this synthetic graph is therefore

$$\widehat{M}_{\text{TMT}}^{\text{synth}} := \frac{1}{N} \sum_{i=1}^N C^{\text{LieSD}}(x_i)^\top W(x_i) C^{\text{LieSD}}(x_i), \quad (17)$$

a pointwise positive-definite reweighting of the LieSD residuals.

Equality of null spaces. Because $W(x_i) \succ 0$, a weighted residual has zero quadratic loss if and only if $C^{\text{LieSD}}(x_i)v = 0$. Thus

$$\ker \widehat{M}_{\text{TMT}}^{\text{synth}} = \bigcap_i \ker C^{\text{LieSD}}(x_i) = \ker M^{\text{LieSD}}. \quad (18)$$

The same argument applies to expectations under a common sampling distribution when the quadratic forms are finite. When $\hat{f} = f$, this establishes equality of the null spaces of the ideal TMT and LieSD population criteria. It does not equate their nonzero spectra or the learned finite-data estimators. In particular, Eq. (6) uses residual Grams of raw denoiser Jacobians, with per-scale trace normalization, rather than the exact orthogonal projector used here. $\square$

D LieGAN Implementation

Our LieGAN adaptation uses joint (s, a, s') observations with separate state and action representations. The generator outputs Lie-algebra coefficients, and each block is normalized to unit Frobenius norm. Magnitudes are sampled from $\mathcal{U}[-\alpha, \alpha]$ with learnable α bounded away from zero to exclude identity collapse. Discrete search is a separate phase after adversarial training. Candidates start from random signed permutations and use the same Gumbel-Sinkhorn relaxation as the other methods, with the trained discriminator replacing the manifold score (Appendix G.6).

Separate block normalization fixes $\|L_a\|_F = 1/\sqrt{2}$ after joint rescaling and excludes an exactly invariant action block. This is a confound on Reacher, where the true action generator is zero. Table 4 compares the constrained variant with a rerun allowing the action norm to vanish. Relaxation improves the prediction errors by less than a factor of two but does not close the gap to TMT-NN. The reported pruning comparison reduces rollout standard deviation from $233.7 \rightarrow 48.3$ while leaving the error approximately unchanged, as also observed for the constrained pipeline.

Table 4: LieGAN with and without the action-block norm constraint, on the Reacher downstream task of Table 2. Medians over 10 seeds after refinement and pruning, lower is better.

Regularizer	One-step MSE	50-step rollout MSE
LieGAN, blocks normalized separately	.067	266
LieGAN, action-block norm free	.038	248
TMT-NN	.004	5.9
True generator	.003	5.2

LieGAN fixes its generator count K_{gen} before training. TMT-NN, TMT-KNN, and LieSD instead select candidates from spectral gaps and complete the resulting sets under Lie-bracket closure, subject to a cap (Eq. (5), Appendix F). Full-set metrics therefore compare sets with different sizes and selection procedures. Table 1 reports each method’s leading direction to separate alignment from returned count.

E Environment State and Action Index Conventions

The following conventions specify the signed permutations evaluated in Sec. 6.3.

Idx	State	Idx	Action
0	x position	0	F_x force
1	y position	1	F_y force
2	$\dot{x}$ velocity		
3	$\dot{y}$ velocity		

E.1 2D Particle ($n = 4, m = 2$)

The continuous ground truth is $\text{SO}(2)$ acting jointly on (x, y) , $(\dot{x}, \dot{y})$, and (F_x, F_y) , with one generator. Without norm constraints, the free-particle limit admits any $P \in \mathfrak{gl}(2)$ acting as $L_s = \text{diag}(P, P)$, $L_a = P$, giving a four-dimensional algebra. The speed cap, ℓ_2 action-norm clip, and circular boundary remove scaling and shear symmetries while preserving rotations. On 1000 stored transitions at $c_d = 0$, rotation satisfies Eq. (1) to 2×10^{-9} , while scaling and shear residuals exceed 2. The circular boundary uses specular reflection, whose inward normal rotates with the particle, preserving $\text{SO}(2)$ even at reflection events. A box boundary would not preserve this symmetry.

E.2 3D Particle ($n = 6, m = 3$)

States 0–2 give position (x, y, z) and 3–5 give velocity. Actions 0–2 give force. The continuous ground truth is $\text{SO}(3)$ with three generators.

E.3 Reacher / Minari ($n = 10, m = 2$)

Idx	State	Idx	Action
0	$\cos(\theta_1)$	0	joint-1 torque
1	$\cos(\theta_2)$	1	joint-2 torque
2	$\sin(\theta_1)$		
3	$\sin(\theta_2)$		
4	target x		
5	target y		
6	$\dot{\theta}_1$		
7	$\dot{\theta}_2$		
8	fingertip–target x		
9	fingertip–target y		

The ground truth is $\text{SO}(2)$ workspace rotation about the base joint, with invariant torques ($L_a^* = 0$). The second angle θ_2 is relative to the first link and remains invariant. Accordingly, L_s^* is zero on the $(\cos \theta_2, \sin \theta_2)$ coordinates (indices 1, 3). The observations are redundant: both $(\cos, \sin)$ pairs satisfy Pythagorean identities, and indices 8–9 are determined by the joint angles and target position to within 2×10^{-5} on stored data. The intrinsic state dimension is approximately six rather than ten. The resulting codimension exceeds n , violating the assumption behind TMT-KNN’s fixed normal rank (Sec. 4).

E.4 CartPole ($n = 4, m = 1$)

States: cart position, cart velocity, pole angle, pole angular velocity. Action: horizontal force. Ground truth: $\mathbb{Z}_2$ sign-flip on all components.

E.5 Swimmer ($n = 8, m = 2$)

The experiments use the native angle representation. An embedded representation is listed for comparison. **Angle representation:** states 0–2 are body and joint angles, 3–5 are linear and angular body velocities, and 6–7 are joint angular velocities. Actions are joint torques. The exact $\mathbb{Z}_2$ symmetry is y -reflection, with sign changes in the appropriate angle, velocity, and action coordinates. It

does not exchange the joints. **Embedded representation:** replacing each angle θ by $(\cos \theta, \sin \theta)$ increases the state dimension. Reflection induces the corresponding sign changes without exchanging joint pairs.

E.6 TriFinger ($n = 30, m = 9$)

Three identical fingers (A, B, C) have three joints each. States 0–8 are joint positions (A: 0–2, B: 3–5, C: 6–8), 9–17 are joint velocities, 18–26 are measured torque readouts, and 27–29 are scalar fingertip contact-force magnitudes for A, B, C. All joint-indexed blocks use the same finger ordering. The *position* actions are target joint positions and the *torque* actions are joint torques, both using blocks 0–2, 3–5, 6–8 for A, B, C. The target $\mathbb{Z}_3$ cyclic permutation ($A \rightarrow B \rightarrow C \rightarrow A$) acts on the finger-local triples in each joint-indexed state block, the three force entries, and the three action triples. It requires no additional spatial rotation in these coordinates. Both variants share this state representation and permutation structure.

E.7 Walker2D / Minari ($n = 17, m = 6$)

Idx	State	Idx	Action
0	root z height	0–2	right thigh / leg / foot torque
1	root angle	3–5	left thigh / leg / foot torque
2–4	right thigh / leg / foot angle		
5–7	left thigh / leg / foot angle		
8	root x velocity		
9	root z velocity		
10	root angular velocity		
11–13	right thigh / leg / foot angular velocity		
14–16	left thigh / leg / foot angular velocity		

Ground truth: $\mathbb{Z}_2$ bilateral mirror swapping left and right leg triples in both state and action.

F Dimension-Selection Details

Space error and effective rank. For each state or action block, we form orthonormal bases of the discovered and reference generator spans. With effective ranks r, r^* and principal angles θ_j , the block error is $|r - r^*| + \sum_{j=1}^{\min(r, r^*)} \sin^2 \theta_j$. The state and action errors are averaged with weights $n/(n + m)$ and $m/(n + m)$. Missing and additional independent directions incur unit penalties, whereas redundant generators do not increase the dimension penalty. Returned generator counts must therefore be distinguished from effective block ranks.

Jitter. Let $\hat{\mathcal{N}} := \frac{1}{N} \sum_{i=1}^N \tilde{C}(x_i)^\top \tilde{C}(x_i)$ denote the sample estimate of $\mathcal{N}$. For numerical solution of the empirical version of Eq. (5), we add diagonal jitter to $\hat{\mathcal{N}}$ when $\text{tr}(\hat{\mathcal{N}}) > 0$, obtaining the positive-definite matrix

$$\hat{\mathcal{N}} + 10^{-6} \frac{\text{tr}(\hat{\mathcal{N}})}{n^2 + m^2} I \succ 0. \quad (19)$$

The positive diagonal term makes the generalized eigenproblem well posed. Its scale is 10^{-6} times the mean diagonal of $\hat{\mathcal{N}}$.

Selection rule. Given the generalized eigenvalues $\lambda_1 \leq \dots \leq \lambda_{n^2+m^2}$ of the empirical version of Eq. (5), the number of retained candidates k is determined as follows.

1. Qualifying eigenvalues are those with $\lambda_j \leq 0.1 \lambda_{\max}$. If none qualifies, $k = 0$. If exactly one does, $k = 1$.
2. Otherwise, find the largest adjacent log-gap $\log \lambda_{j+1} - \log \lambda_j$ among qualifying eigenvalues. Cut there only when the eigenvalue ratio exceeds 5.0 for TMT-NN, 1.5 for TMT-KNN,

or 1.2 for LieSD. These method-specific ratios are scale-invariant and fixed in the selection rule, which does not use the ground-truth algebra dimension.

3. If no gap clears the threshold, all qualifying eigenvalues are retained, with k capped at 8.

The ratio thresholds filter small spectral gaps. Restricted coverage may prevent a separating gap from appearing, causing selection to reach the cap. Threshold adjustment cannot replace informative coverage (Sec. 4).

How often the selected dimension is correct. On the 2D Particle, where the true algebra is one-dimensional, raw selection returns $k = 4$ and refinement with ranking yields a single final candidate in 10/10 seeds. On Reacher, also one-dimensional, raw selection is exact in 4/10 seeds ($k = 1$ in four seeds, $k = 2$ in four, $k = 5$ in two); refinement and ranking leave a singleton in 8/10 seeds and a five-element set in 2/10, after which the downstream model uses the lowest-orbit-score candidate. Finite-data noise smears the zero eigenspace, so exact dimension recovery needs a gap above the estimation error and unstable gaps remain ambiguous.

Three counts, not one. The pipeline produces three distinct counts and they should not be confused. The *selected* count k is what the rule above returns. Refinement then searches the span of those k eigenvectors widened by four more and deduplicates survivors (Appendix G.11), so the *refined* count can exceed k , and a direction selected at $k = 1$ can move substantially because the search space is five-dimensional rather than one. The refined set is finally completed under Lie-bracket closure, capped at 16, giving the *returned* count. On Reacher the selected count is 1 in four seeds, 2 in four and 5 in two, while the returned count runs from 5 to 16 with mean 6.6 for TMT-NN; TMT-KNN and LieSD saturate the closure cap of 16 in every seed, and LieGAN returns its fixed configuration value of 4 with no refinement or closure step at all (Appendix D).

Within each block, each excess subspace dimension adds one unit to the space error. Returned counts can therefore dominate the aggregate. On Reacher, the three baselines have leading-direction errors between 0.86 and 1.00, yet full-set scores are 13.93 for TMT-KNN, 13.83 for LieSD, and 4.00 for LieGAN. The first two are within 0.1 of their dimension-dependent ceiling, $(10 \cdot 16 + 2 \cdot 4)/12 = 14.0$. LieGAN, which returns four directions, cannot exceed 4.0. Since $L_a^* = 0$, the action-block error equals the rank of the estimated action span, reported as 4.0 for all three baselines. Table 1 therefore reports leading-direction metrics. Full-set particle results should be interpreted alongside returned counts.

G Additional Results and Ablations

G.1 Generator Visualizations on the 2D Particle

The low-dimensional generator of the 2D Particle allows direct inspection of the learned matrix structure. Figure 2 complements the Reacher visualization in Fig. 1.

G.2 Dataset Size Scaling (2D Particle)

Table 5 evaluates sensitivity to dataset size. Errors vary substantially across seeds, with standard deviations exceeding the means in several conditions. TMT-NN achieves near-zero reported space error from 25k transitions onward, but its error and variability increase at 10k. Each column denotes the requested dataset size, of which 10% is held out for validation. Thus, the 100k condition fits the estimators on 90k transitions, unlike the $N = 100,000$ base condition in Table 3, which uses the full dataset. LieSD is particularly sensitive to this protocol difference.

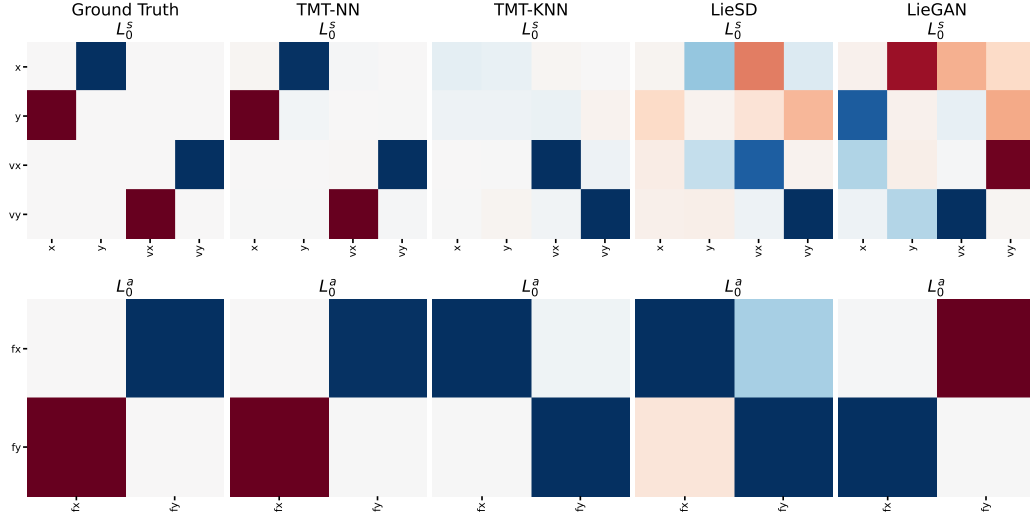


Figure 2: Leading continuous-generator directions on the 2D Particle for one seed, compared with the ground-truth $\text{SO}(2)$ generator. Top: state block L_s^0 over $(x, y, \dot{x}, \dot{y})$. Bottom: action block L_a^0 over (F_x, F_y) . The target applies the same antisymmetric rotation to position, velocity, and force. TMT-NN recovers this structure in both blocks. TMT-KNN and LieSD recover the velocity sub-block but deviate in the position sub-block. LieGAN produces a noisy state block and an incorrect sign pattern in the action block. These single-seed visualizations complement the aggregate results in Sec. 6.2.

Table 5: Space error over the full returned generator set on 2D Particle as dataset size varies. Values are mean $\pm$ standard deviation over 10 seeds. Lower is better, with column minima in bold. Interpret the results together with the returned counts in Appendix F, since each excess independent direction incurs a unit penalty.

Method	10k	25k	50k	100k
TMT-NN	3.90 ± 5.04	0.001 ± 0.000	0.001 ± 0.000	0.000 ± 0.000
TMT-KNN	1.84 ± 1.59	3.17 ± 2.59	1.68 ± 1.34	1.28 ± 0.64
LieSD	11.00 ± 0.00	6.89 ± 4.19	7.88 ± 3.79	9.55 ± 1.57
LieGAN	3.03 ± 0.02	3.05 ± 0.02	3.08 ± 0.03	3.07 ± 0.03

G.3 Behavior-Policy Bias: 2D and 3D Particle

TMT-NN maintains low space error across all tested bias levels in 2D. In 3D, the error remains low through $\kappa = 5$ and increases substantially at $\kappa = 10$, where its reported space error ranks third among the four methods.

G.4 Dynamics Nonlinearity Sweep (2D Particle)

Table 7 completes the drag sweep in Sec. 6.2. The error of learned LieSD is not monotonic in c_d , motivating the exact-Jacobian comparison below. The simulator is piecewise smooth at every drag value because boundary reflection, the speed cap, and action clipping introduce branches. The reflection branch is activated on approximately 1.4% of transitions even at $c_d = 0$.

G.5 Oracle Evaluation of LieSD on the 2D Particle

At $c_d = 0$, 1.447% of transitions execute the hard boundary-reflection branch. We evaluate four drag values, each with and without reflection-branch transitions, over three seeds, for 24 combinations. Before refinement, LieSD with exact dynamics Jacobians selects $k = 1$ in all 24 tests and achieves a median principal angle of 2.30×10^{-6} degrees. TMT with exact manifold normals independently

Table 6: Space error over the full returned set on 2D and 3D Particle under von Mises behavior-policy bias κ . Values are mean $\pm$ standard deviation over 10 seeds, with row minima in bold. The unbiased 2D condition appears in Table 3. In 2D, four independent directions against a one-dimensional reference incur a dimension penalty of 3. Its excess above 3 equals $\frac{2}{3} \sin^2$ of the angle between the target generator and the span of its four state blocks, rather than the angle to its leading direction.

	Dataset	TMT-NN	TMT-KNN	LieSD	LieGAN
2D	$\kappa = 1$	0.001 $\pm$ 0.000	7.13 $\pm$ 4.78	2.22 $\pm$ 1.62	3.06 $\pm$ 0.03
	$\kappa = 5$	0.001 $\pm$ 0.000	2.52 $\pm$ 4.19	8.13 $\pm$ 4.62	3.01 $\pm$ 0.01
	$\kappa = 10$	0.001 $\pm$ 0.000	3.70 $\pm$ 4.95	10.74 $\pm$ 0.83	3.02 $\pm$ 0.01
	unbiased	0.004 $\pm$ 0.001	11.05 $\pm$ 0.09	11.11 $\pm$ 0.15	1.64 $\pm$ 0.27
3D	$\kappa = 1$	0.004 $\pm$ 0.001	11.12 $\pm$ 0.09	11.06 $\pm$ 0.11	2.16 $\pm$ 0.17
	$\kappa = 5$	0.011 $\pm$ 0.006	4.89 $\pm$ 4.33	5.97 $\pm$ 4.74	1.94 $\pm$ 0.12
	$\kappa = 10$	3.76 $\pm$ 3.71	4.87 $\pm$ 4.34	2.30 $\pm$ 0.10	2.10 $\pm$ 0.18

Table 7: Space error over the full returned set on 2D Particle as velocity-dependent drag varies. Values are mean $\pm$ standard deviation over 10 seeds, with row minima in bold. The dynamics retain exact SO(2) symmetry at every drag value. The $c_d = 0$ condition appears in Table 3. LieGAN remains near its dimension penalty of 3. At $c_d \geq 1$, TMT-KNN saturates the closure cap and attains the dimension-determined error 11.00. LieSD also saturates in all but one seed per condition, producing the reported variation.

Drag	TMT-NN	TMT-KNN	LieSD	LieGAN
0.1	0.000 $\pm$ 0.000	3.68 $\pm$ 3.58	3.89 $\pm$ 4.01	3.06 $\pm$ 0.03
0.5	0.001 $\pm$ 0.000	6.58 $\pm$ 4.76	2.78 $\pm$ 3.21	3.08 $\pm$ 0.08
1.0	0.001 $\pm$ 0.000	11.00 $\pm$ 0.00	10.11 $\pm$ 2.83	3.12 $\pm$ 0.17
2.0	0.002 $\pm$ 0.000	11.00 $\pm$ 0.00	10.74 $\pm$ 0.84	3.10 $\pm$ 0.17

selects $k = 1$ in all 24 tests and achieves a median angle of 2.89×10^{-6} degrees. These are distinct oracle constructions with the same recovered null space in this setting.

For learned LieSD, excluding reflection transitions reduces the $c_d = 0$ median angle from 76.9 degrees to 14.6 degrees but leaves a nonzero error. Its error remains non-monotonic in drag both with and without reflection transitions. Refinement is excluded to evaluate the initial estimators directly, since it can perturb an already accurate oracle direction.

Coverage under bias. Figure 3 illustrates how concentration of the applied-force direction changes the observed coverage of rotation orbits in 3D. This relates to the coverage requirement in Sec. 4. Restricted coverage can make true and spurious generators difficult to distinguish, although the visualization alone does not measure the excitation gap.

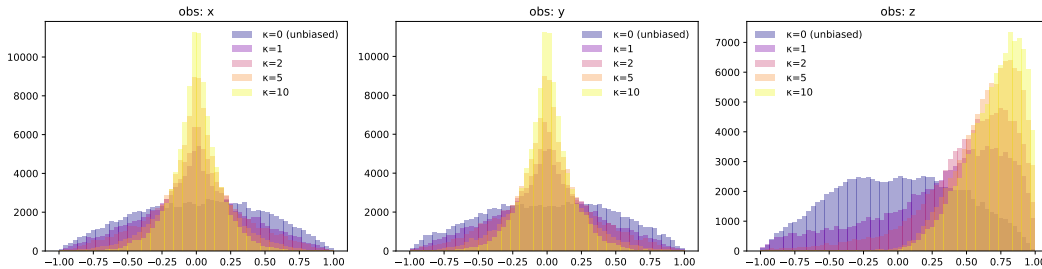


Figure 3: Positions observed in the 3D Particle as the von Mises concentration κ of the applied-force direction increases. The data occupy a narrower portion of the rotation orbit, illustrating a potential coverage limitation. The visualization alone does not establish a causal effect attributable only to sampling density.

G.6 Discrete Discovery: Full Results

Table 8 reports exact CartPole recovery. LieGAN achieves the highest top-1 and full-group recovery rates. The accepted TMT-KNN set contains the target reflection in every seed and ranks it first in eight, but also contains additional matrices at Frobenius distances 2.0–5.5 from the target. Closing these accepted sets under composition produces groups of 4–64 elements, preventing exact full-group recovery. Every exactly recovered candidate has simulator equivariance error $E_{\text{sim}} \approx 3 \times 10^{-7}$ on 500 fresh transitions.

Because many signed permutations are involutions, the order of a returned element does not establish CartPole recovery. The nonidentity element must equal the target reflection. We also report $\|g - I\|_F$ and simulator equivariance E_{sim} as defined in Sec. 5.

Table 8: Exact CartPole recovery over 10 seeds. Top-1 evaluates the highest-ranked accepted non-identity element. Full-group recovery evaluates the group generated by all accepted elements.

Method	Exact top-1	Exact full group
TMT-NN	4/10	4/10
TMT-KNN	8/10	0/10
LieSD	1/10	1/10
LieGAN	9/10	9/10

TMT-KNN recovers the Swimmer y -reflection in 2/10 seeds, compared with 0/10 for TMT-NN, LieSD, and LieGAN. On TriFinger, no method recovers either nonidentity element of $\mathbb{Z}_3$ on the default or cube-free datasets. The cube-free results and their identifiability limitation are detailed in Appendix G.8.

Scoring the analytic target g^* with each trained scorer distinguishes failures of search from failures of acceptance (Appendix G.10). For TMT-NN, the target lies within the acceptance region in every seed on both environments. On TriFinger, for example, $d(g^*) \approx 0.008$ while $\tau \approx 0.013$, indicating that search failed to propose an acceptable target. For TMT-KNN on TriFinger, the target score is approximately three times the threshold and would be rejected even if proposed. These diagnostics use ground truth after discovery. Identity-based threshold calibration alone does not reveal which failure occurred.

Candidate scores. All four methods use the same Gumbel–Sinkhorn search [18], rounding to signed permutations, held-out scoring, and acceptance rule $\tau = \min(\tau_{\text{max}}, \beta d(I))$. The threshold is calibrated separately for each run from the identity baseline without ground truth. The methods differ in the score d . TMT-NN uses the held-out denoiser residual $d_{\text{mfd}}(g) := (\mathbb{E}_{x \sim p} \|d_\phi(g \cdot x; \sigma) - g \cdot x\|^2)^{1/2}$, TMT-KNN uses distance to the untransformed data, LieSD evaluates consistency with its learned dynamics $\hat{f}$, and LieGAN uses its trained discriminator. The candidate initialization procedure is shared across methods and is independent of the learned LieGAN continuous generators.

Table 9 reports the method-specific scores $d(g)$ of rounded candidates, and Table 10 reports their identity-normalized ratios $\rho(g) = d(g)/d(I)$. The proxies have different meanings and scales. Their absolute values cannot rank methods, and normalization does not make them equivalent recovery metrics. The ratio is 1 at the identity by construction, so a small ratio alone does not establish recovery of a nonidentity symmetry. For TMT-NN on CartPole, the mean ratio 12.5 ± 30.5 is driven by a single seed, whereas the median is 2.6.

Identity reconstruction diagnostic. For TMT-NN, the identity residual is $d_{\text{mfd}}(I) = 0.00062 \pm 0.00012$ on CartPole and 0.003 on Swimmer (random). These values quantify reconstruction of held-out data by the denoiser. They do not establish accurate geometry away from the data or exclude denoiser error as a contributor to candidate scores. On CartPole, the LieSD identity baseline is approximately 0.0005 and the LieGAN baseline is 0.011. These values belong to different proxies and are not directly comparable measures of fit.

Table 9: Scores of rounded discrete candidates, mean $\pm$ standard deviation over 10 seeds. Each method uses a distinct proxy and scale. Absolute values therefore support neither cross-method ranking nor direct comparisons across environments. The scores diagnose consistency rather than exact recovery. Walker2D is included only for this diagnostic, since exact recovery was not evaluated there.

Environment / Dataset	TMT-NN	TMT-KNN	LieSD	LieGAN
CartPole (random)	0.0072 $\pm$ 0.0169	0.0486 $\pm$ 0.0105	0.0013 $\pm$ 0.0006	0.0177 $\pm$ 0.0179
Swimmer (random)	0.230 $\pm$ 0.049	0.390 $\pm$ 0.126	1.072 $\pm$ 0.212	0.984 $\pm$ 0.010
TriFinger position (random)	0.285 $\pm$ 0.066	2.660 $\pm$ 0.303	6.114 $\pm$ 0.560	0.951 $\pm$ 0.049
TriFinger torque (random)	0.633 $\pm$ 0.168	2.334 $\pm$ 0.226	2.855 $\pm$ 0.258	0.984 $\pm$ 0.009
Swimmer (medium)	0.395 $\pm$ 0.101	0.703 $\pm$ 0.204	1.664 $\pm$ 0.191	0.993 $\pm$ 0.008
Swimmer (trained)	0.505 $\pm$ 0.176	0.242 $\pm$ 0.032	0.411 $\pm$ 0.095	0.659 $\pm$ 0.155
Walker2D (random)	0.324 $\pm$ 0.116	6.322 $\pm$ 0.370	19.803 $\pm$ 1.567	0.970 $\pm$ 0.018
Walker2D (medium)	0.154 $\pm$ 0.127	2.973 $\pm$ 0.145	19.035 $\pm$ 1.903	0.800 $\pm$ 0.120

Table 10: Identity-normalized scores ρ of rounded discrete candidates, mean $\pm$ standard deviation over 10 seeds. The ratio equals 1 at the identity and does not establish recovery of a nonidentity target.

Environment	Group	TMT-NN	TMT-KNN	LieSD	LieGAN
CartPole (random)	$\mathbb{Z}_2$	12.52 $\pm$ 30.45	0.94 $\pm$ 0.14	2.77 $\pm$ 1.62	1.83 $\pm$ 2.30
Swimmer (random)	$\mathbb{Z}_2$	75.62 $\pm$ 17.86	4.33 $\pm$ 1.42	12.58 $\pm$ 1.93	2.25 $\pm$ 0.07
TriFinger (position, random)	$\mathbb{Z}_3$	42.56 $\pm$ 10.26	4.98 $\pm$ 0.56	14.73 $\pm$ 1.06	1.91 $\pm$ 0.10
TriFinger (torque, random)	$\mathbb{Z}_3$	61.05 $\pm$ 15.72	9.00 $\pm$ 0.84	8.45 $\pm$ 0.83	1.99 $\pm$ 0.10

Scaling discrete discovery. The Walker2D results provide candidate-score diagnostics but no evidence of exact recovery, which was not evaluated in that environment. The TriFinger results directly demonstrate the gap between an acceptable target score and successful identification. Structured, compositional, or hierarchical search over signed permutations is a direction for addressing this gap, alongside calibration of acceptance without ground truth (Sec. 8).

G.7 Swimmer: Why the Joint Swap Is Not a Symmetry

Among the tested transformations, only the y -reflection satisfies Swimmer equivariance to the numerical floor. Swapping the joints and links is not a relabeling automorphism. The proximal joint rotates the downstream subtree, whereas the distal joint moves only the terminal link relative to its parent. Reversing the chain would require exchanging the root and tail bodies, which a coordinate permutation does not implement. A paired simulator evaluation over 20,000 transitions supports this distinction.

G.8 TriFinger: Simulator Symmetry Evaluation

The default TriFinger platform includes a cube whose state is absent from the 30-D observation. Finger-cube contact can therefore violate the target $\mathbb{Z}_3$ relation in the observed coordinates. In paired simulator evaluations over 20,000 transitions per nonidentity element, 36–40% violate cyclic-permutation equivariance ($E_{\text{sim}} > 10^{-6}$). Every violation involves contact, while the contact-free subset agrees to the numerical floor.

We also generate a cube-free dataset with `object_type=NONE` and a 5 mm contact-clearance rejection rule. The same simulator check finds no violations (0/20,000 per element for both action modalities). Nevertheless, none of the four methods recovers a nonidentity target element over 10 seeds. Contact-induced violations alone therefore do not explain the discovery failure.

The cube-free variant also introduces an identifiability limitation. Its three fingertip contact-force coordinates are identically zero, so permutations of those coordinates are indistinguishable on the data. Exact matching to a single target matrix is consequently stricter than identification of the

induced transformation on the observed support. This ambiguity limits the interpretation of exact-match failures on the cube-free dataset. It does not apply to the default dataset, where those entries vary.

G.9 Hyperparameters and Estimator Details

Data are standardized per coordinate before the estimators are formed. Equation (6) trace-normalizes the Gram matrix at each noise scale before averaging. It uses the raw denoiser Jacobian, without projecting it onto a fixed-rank subspace. For any real $J_{i,\sigma}$, $B^\top B$ is positive semidefinite, so the resulting operator defines a quadratic objective without requiring the Jacobian to be symmetric or idempotent. This algebraic property does not by itself guarantee accurate normal-space estimation.

Table 11: Hyperparameters shared across environments unless otherwise specified.

<i>Denoiser (TMT-NN)</i>	
Noise range $[\sigma_{\min}, \sigma_{\max}]$	$[10^{-4}, 10^{-2}]$, $S = 5$ log-spaced scales
Evaluation scale	$\sigma = 10^{-2}$
Architecture	MLP, hidden [256, 256]
Training	8000 epochs, Adam 10^{-3} , cosine decay, weight decay 10^{-6} , batch 2048
<i>Local PCA (TMT-KNN)</i>	
Neighbors K	40
Anchor points	8000
<i>Continuous selection and refinement</i>	
Qualifying threshold	$\lambda_j \leq 0.1 \lambda_{\max}$
Gap ratio	5.0 (TMT-NN), 1.5 (TMT-KNN), 1.2 (LieSD)
Candidate cap / closure cap	8 / 16
Refinement subspace	$k + 4$ eigenvectors, $\max(2(k+4), 16)$ random restarts
Transformation times	$t \in \{0.25, 0.5, 1.0\}$
Deduplication / pruning	cosine > 0.9 / entries below $0.1 \max(\ L_s\ _\infty, \ L_a\ _\infty)$
<i>Discrete search</i>	
Parallel candidates C	64
Gumbel–Sinkhorn temperature	annealed $5.0 \rightarrow 0.05$
Signs	continuous pre-activation per coordinate, hardened by $\text{sign}(\cdot)$ at readout
Acceptance	$\tau = \min(\tau_{\max}, \beta d(I))$ with $\beta = 2.0$
<i>Data</i>	
Transitions per dataset	100,000 (particle sweeps vary N where stated)
Preprocessing	per-coordinate standardization of $[s; a; s']$
Seeds	42–51

G.10 Scoring the True Element

For each TriFinger and Swimmer checkpoint, we evaluate the analytic element g^* using the trained scorer and the same transformation path used during search. The score is compared with the original acceptance threshold, which remains calibrated from the identity baseline. Ground truth is used only for this post hoc diagnostic. Table 12 distinguishes three outcomes. *Search-limited* means $d(g^*) < \tau$, so the target would be accepted if proposed. *Threshold-limited* means $d(g^*) \geq \tau$, so it would be rejected even if proposed. *Recovered* means search returned the target. LieGAN is omitted because its checkpoints do not retain candidate scores.

The score margins clarify these outcomes. For TMT-NN, $d(g^*)$ is below the threshold on both TriFinger (0.008 versus 0.013) and Swimmer (0.003 versus 0.006). For TMT-KNN on TriFinger, it is approximately three times the threshold (0.56 versus 0.20). LieSD on TriFinger is sensitive to the threshold, with both quantities near 0.45 and seeds on either side.

Table 12: Post hoc evaluation of the analytic target g^* with each trained scorer and its original acceptance threshold. Counts are over 10 seeds, except for TMT-KNN on TriFinger, where one checkpoint is missing.

Method	TriFinger (position)	Swimmer
TMT-NN	10/10 search-limited	10/10 search-limited
TMT-KNN	9/9 threshold-limited	8/10 search-limited, 2/10 recovered
LieSD	8/10 threshold-limited, 2/10 search-limited	10/10 search-limited

G.11 Deployment Pipeline for Downstream Regularization

Candidate selection, refinement, and pruning use no ground-truth symmetry. The downstream penalty weight is a separate choice and was tuned using the analytic generator, as documented below.

Normalization. We jointly scale (L_s, L_a) so that $\|L_s\|_F^2 + \|L_a\|_F^2 = 1$. Separate block normalization would alter the relative state and action transformations and eliminate information carried by a near-zero action block. Joint normalization preserves the generator direction and keeps its magnitude within the working range of the matrix-exponential approximation used downstream.

Orbit score and refinement. For a generator vector v , the orbit score averages the denoiser residual over held-out data transformed by $(\exp(tL_s), \exp(tL_a))$ at $t \in \{0.25, 0.5, 1.0\}$. Projected gradient descent minimizes this score over unit-norm coefficients in the span of the k selected eigenvectors and four additional eigenvectors from the same ordering. Optimization starts from $\max(2(k+4), 16)$ random unit vectors. Thus, the eigenproblem defines the search subspace and the orbit score selects a direction within it. Refinement can substantially change the direction even when $k = 1$.

Deduplication and pruning. Candidates with $|v_i \cdot v_j| > 0.9$ are treated as duplicates. We retain them greedily in ascending orbit-score order, then prune entries in either block below $0.1 \max(\|L_s\|_\infty, \|L_a\|_\infty)$. If several candidates remain, we deploy the one with the lowest orbit score.

Downstream training and evaluation. The dynamics model is trained with $\mathcal{L}_{\text{reg}} + \lambda \mathcal{L}_{\text{eq}}$, where $\lambda = 3$ and $\mathcal{L}_{\text{eq}} = \|\hat{f}(g_s s, g_a a) - g_s \hat{f}(s, a)\|_2^2$. Training uses the default initial-pose range of ± 0.1 rad. Evaluation uses a rotated-environment wrapper to co-rotate the initial arm pose and target through 12 orientations at 30° intervals over the full circle.

Penalty weight. A single-seed sweep with the analytic generator selects $\lambda = 3$ by validation MSE. The tested values are $\{0, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000\}$. The selected MSE is 0.00104, compared with 0.00155 without regularization. This tuning therefore uses ground-truth information. A separate check using the refined and pruned TMT-NN candidate also selects $\lambda = 3$. It yields one-step MSE 0.00537 and rollout MSE 5.76, compared with 0.00660 and 30.66 at $\lambda = 0.1$.

Held-out simulator check. Each deployed generator is evaluated on 300 fresh Reacher-v5 transitions collected under a uniform random policy with a seed disjoint from discovery, training, and downstream evaluation. We average E_{sim} across $t = \pm 0.1, \pm 0.3, \pm 0.5, \pm 1$. The unit-norm generators have three antisymmetric state blocks with angular frequencies $\omega \approx 0.40\text{--}0.41$, giving rotations of approximately 2.4° at the smallest $|t|$ and 23° at the largest.

The analytic generator achieves 9.9×10^{-16} . Eight deployed candidates yield errors from 6×10^{-7} to 3.6×10^{-5} , while seeds 45 and 51 yield 0.18 and 0.49. A random skew-symmetric generator scaled to the norm of the analytic generator yields 17.9. The two high-residual candidates also have the largest angles to the target, 7.35° and 17.02° , compared with at most 3.22° for the other eight. These finite simulator checks identify the same two failures as the ground-truth angle diagnostic.

They support the remaining candidates on the tested transitions and transformation magnitudes, without certifying global equivariance. Both diagnostics require information beyond the orbit score.

Per-stage results. Table 13 reports all three deployment stages for the baselines. The discovery sweep in Table 1 uses a separate implementation of selection and refinement, so the two tables do not describe successive stages of a common run. The 5.8° discovery result for TMT-NN and the 77.4° raw deployment angle in Table 2 therefore refer to different selection and refinement implementations.

Refinement accounts for most of the improvement for TMT-KNN and LieSD. Their median angles change from approximately 88° to 81° and 87° to 50° , respectively, while rollout errors fall from approximately 1210 to 53 and 2015 to 32. Pruning yields further reductions in prediction error. LieGAN rollout error increases after both refinement and pruning.

Two raw LieSD seeds have rollout errors orders of magnitude above the others. The target $SO(2)$ generator is skew-symmetric and its exponentials preserve Euclidean norm, whereas approximately 72% of the raw-candidate norm lies in the symmetric part. This fraction alone does not distinguish the two unstable seeds or establish the cause of their large errors. After refinement, the symmetric fraction is approximately 42% across seeds and no refined seed exhibits the same instability.

Table 13: Baseline results at each stage of candidate selection, refinement, and pruning. Values are median $[Q_1, Q_3]$ over 10 matched seeds, with the evaluation protocol of Table 2. Candidate processing uses no ground-truth symmetry, while the shared penalty weight was tuned with the analytic generator. Two raw LieSD seeds have extreme rollout errors, reflected in the upper quartile.

Method	Stage	Angle (deg)	One-step MSE	50-step rollout MSE
TMT-KNN	raw	87.6 [84.9, 88.3]	.161 [.124, .276]	1209.5 [892.4, 1771.5]
	refined	81.4 [79.0, 88.4]	.019 [.015, .036]	53.4 [34.7, 94.1]
	pruned	82.1 [78.0, 87.6]	.012 [.006, .015]	30.0 [20.9, 43.6]
LieSD	raw	86.9 [85.4, 88.9]	.128 [.115, .266]	2015.0 [607.6, 34519]
	refined	50.2 [49.1, 50.8]	.011 [.009, .012]	31.6 [28.3, 36.1]
	pruned	49.1 [48.6, 49.6]	.008 [.008, .010]	20.1 [15.3, 29.0]
LieGAN	raw	89.7 [89.5, 89.8]	.036 [.028, .047]	156.2 [145.9, 180.6]
	refined	89.5 [89.5, 89.7]	.045 [.037, .052]	234.8 [190.4, 256.2]
	pruned	90.0 [90.0, 90.0]	.067 [.036, .094]	266.2 [205.2, 302.5]