DIVLAM: LATENT ACTION DISCOVERY IN WORLD MODELS THROUGH DEMONSTRATOR DIVERSITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Stochastic execution makes action discovery from video ambiguous: a demonstrator's chosen command need not match the movement executed. DivLAM separates these hidden choices in two stages. Stage A learns shared next-state predictors and demonstrator-specific executed-action probabilities in a frozen visual representation. Stage B factors these probabilities into command policies and an execution channel specifying movement probabilities given each command. Building on existing minimum-volume identification theory, we prove that policies covering every ordered pair with probabilities $2/3$ and $1/3$ satisfy the required sufficient-scattering condition for at least four commands. A matched family restricted to cyclic neighbors instead yields an absorbed channel that incorporates policy mixing. Exact-input tests recover the predicted targets, while sampled counts distinguish sampling error from persistent bias. Across five fresh TwoRoom replication datasets, all-run mean channel total-variation errors after relabeling are 0.067 for ordered pairs and 0.332 for cyclic, versus 0.188 for ignoring execution noise. This visual gap reflects both different population targets and unequal success in learning the executed basis. Discovery uses no action labels, but the frozen encoders were pretrained with actions; recovery is sensitive to encoder choice and fails on the tested PushT task. The results connect demonstrator policy coverage to channel recovery and distinguish factorization bias from errors in learning the executed basis.

1 INTRODUCTION

Latent action models (LAMs) learn hidden action variables from video for policies and world models (Edwards et al., 2019; Schmidt & Jiang, 2024; Ye et al., 2025; Bruce et al., 2024). A *command* is the action chosen; the *executed action* is the movement that occurs. Under *stochastic execution*, these can differ, so predicting observations need not recover commands.

An upward command may, for example, produce rightward motion. The *execution channel* specifies movement probabilities given commands. Even exact transition probabilities can admit multiple policies and channels—an *identifiability* failure that more data alone cannot resolve.

Demonstrator identity adds structure: different preferences mix the same execution mechanism. Policies lie in the *action simplex* of command-probability vectors. Schur (2026) show that sufficient scattering—geometric coverage formalized in Section 3.4—and suitable rank assumptions permit minimum-volume identification (Fu et al., 2018). The smallest-volume simplex of transition distributions consistent with the data recovers policies and dynamics up to relabeling. Demonstrator count and entropy alone do not certify coverage. We test whether this criterion yields recovery from pixels and predicts bias for a matched policy family violating coverage.

DivLAM implements this separation in two stages. Stage A learns an *executed basis*—one shared next-state predictor, or *branch*, per executed action—and demonstrator-specific branch probabilities in a frozen representation. Stage B factors these probabilities into command policies and an execution channel. Together, the predictors and channel define a *command-conditioned world model*. This separation distinguishes factorization bias from errors in learning the executed basis.

For four commands, we compare twelve ordered-pairs policies with four cyclic-neighbor policies repeated three times, each assigning probabilities $2/3$ and $1/3$ to two commands. The families match

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.

demonstrator count, entropy, aggregate command prior, identity-only accuracy, and execution channel, while differing in coverage and induced trajectories. Ordered pairs satisfy sufficient scattering; cyclic policies yield an *absorbed channel* incorporating policy mixing. Exact-input tests isolate these targets. Across five fresh TwoRoom datasets, all-run channel total-variation errors after relabeling are 0.067 for ordered pairs, 0.332 for cyclic, and 0.188 for ignoring noise; zero denotes exact recovery. The visual gap also reflects unequal basis-learning success.

Our contributions are a two-stage estimator, a proved policy construction with an analytically characterized failure comparison, and tests separating population targets, sampling error, and visual estimation failures (Sections 3–4). Asymmetric execution probes directional recovery; command naming and planning assess downstream use. The demonstrated setting uses a known action count, observed identities, shared dynamics, and state-independent policies and execution noise. Discovery and selection use no action labels, but the frozen encoders were pretrained with actions.

2 Background and related work

2.1 Observed transitions and the recovery target

Let $[r] = \{1, \dots, r\}$. We observe image transitions (o_t, o_{t+1}, e) from demonstrator $e \in [m]$ and encode them as latent states $z_t = \phi(o_t) \in \mathbb{R}^d$ using a frozen encoder. The command $a \in [k]$ and executed action $b \in [k]$ are hidden, with the same known number k of possibilities. Each executed action selects a *branch* of the dynamics, with next-state distribution $T_b(\cdot \mid z)$.

Demonstrator e chooses commands according to $\pi_e(a \mid z)$; the shared, state-independent channel $W[a, b] = \Pr(b \mid a)$ then selects the executed branch. Marginalizing the hidden choices gives

$$p(\cdot \mid z, e) = \sum_{a=1}^{k} \pi_e(a \mid z)\, p(\cdot \mid z, a), \qquad p(\cdot \mid z, a) = \sum_{b=1}^{k} W[a, b] T_b(\cdot \mid z). \tag{1}$$

Here $\cdot$ ranges over next states; each policy and channel row is nonnegative and sums to one. For identification, we idealize $T_b(\cdot \mid z) = \delta_{f_b^*(z)}$: branch b leads to the deterministic next state $f_b^*(z)$, and δ_x denotes a point mass at x.

We assume a Markov-sufficient encoded state and shared dynamics: given the current state and command, history and demonstrator identity provide no further information about the next state. Our main experiments use state-independent policies $\pi_e(a \mid z) = \pi_e(a)$, so each demonstrator's overall branch frequencies are mixtures of channel rows.

Recovery concerns W, the policies, and $p(\cdot \mid z, a)$ up to independent relabelings: command permutations reorder policies and rows of W, while branch permutations reorder T_b and columns of W. Overlapping command-conditioned distributions can still leave individual commands uncertain. The additional rank, continuity, and support assumptions in Appendix A.7 remain unverified for our pretrained encoders.

2.2 Related work and the role of demonstrator diversity

Latent action learning makes observation-only trajectories useful for control: LAPO learns policies through a discovered action space, LAPA uses quantized actions for vision-language-action pretraining, and Genie learns an interface for interactive video generation (Schmidt & Jiang, 2024; Ye et al., 2025; Bruce et al., 2024). Continuous models extend these approaches to richer action spaces and heterogeneous videos (Garrido et al., 2026; Ayalew et al., 2026). Analyses examine how behavior and variation unrelated to actions affect learned representations (Zhang et al., 2025; Lee et al., 2026); distractor experiments demonstrate the value of action supervision (Nikulin et al., 2025). DivLAM examines the ambiguity between command preferences and stochastic execution, which persists even with exact knowledge of executed outcomes.

Different assumptions provide different identification routes. Entropy-regularized latent action policies are identifiable under the conditions of Lachapelle (2025), while CRAFT uses policy differences across datasets to learn controllable state features (Levine et al., 2025). DivLAM adopts the minimum-volume identification principle of Schur (2026), grounded in Fu et al. (2018), and

estimates its inputs through a learned executed basis. Our ordered-pairs and cyclic comparison tests this criterion's predictions; it does not establish sufficient scattering as necessary for every action-discovery method. Appendix I expands the comparison.

3 DIVLAM

3.1 WHY SEPARATE EXECUTED MOTION FROM COMMANDS?

Under the conditions of Appendix Lemma A.1, the observable mixture determines distinct deterministic next states and their branch probabilities. These probabilities still admit $W = I$, the identity matrix representing noiseless execution, with command policies equal to branch frequencies. DivLAM therefore learns the executed basis before separating policies from execution noise (Figure 1). The second stage imposes nonnegativity and sum-to-one constraints directly, including exact zeros for unused commands. Such zeros allow the boundary coverage required by sufficient scattering. Appendix B.4 discusses the limitations of one-stage neural fitting.

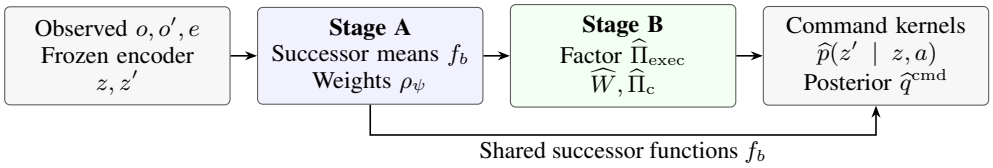

Figure 1: **DivLAM separates executed motion from intended commands.** Stage A fits encoded transitions and averages branch responsibilities by demonstrator to form $\widehat{\Pi}_{\text{exec}}$. Stage B uses minimum volume to separate these frequencies into the execution channel $\widehat{W}$ and commanded policies $\widehat{\Pi}_{\text{c}}$. Both stages supply the final model. Command a and executed branch b are hidden during discovery; physical command names use separate calibration labels.

3.2 STAGE A LEARNS A SHARED EXECUTED BASIS

For consecutive latent states z, z', each shared predictor, or *head*, $f_b : \mathbb{R}^d \to \mathbb{R}^d$ predicts the next latent state for branch b. Demonstrator-dependent probabilities $\rho_\psi(b \mid z, e)$, with learned parameters ψ, combine the k heads in a Gaussian mixture(Bishop, 1994):

$$\ell_b(z' \mid z) = \mathcal{N}(z'; f_b(z), \sigma^2 I_d), \qquad p_A(z' \mid z, e) = \sum_{b=1}^{k} \rho_\psi(b \mid z, e)\ell_b(z' \mid z). \tag{2}$$

The width σ accommodates representation and fitting error; W describes physical execution noise. The weights estimate branch probabilities $\sum_a \pi_e(a \mid z)W[a, b]$. On n training transitions, we minimize negative log-likelihood plus penalties weighted by λ_A and λ_ρ:

$$\mathcal{L}_A = -\frac{1}{n}\sum_{i=1}^{n} \log p_A(z'_i \mid z_i, e_i) + \lambda_A V_A + \lambda_\rho B_\rho. \tag{3}$$

Following Schur (2026), V_A penalizes the mean log-determinant of the radial-basis-function (RBF) similarity matrix of head predictions. Minimizing it favors coincident heads, while likelihood rewards explaining distinct branches. B_ρ discourages linear dependence between branch-weight profiles across demonstrators. These are fitting choices; the exact deterministic-basis lemma does not require them. Appendix B.3 gives their formulas and schedules.

The *responsibility* $q_i^{\text{exec}}(b)$ is the fitted probability that branch b generated transition i. Averaging within each demonstrator gives Stage B's input:

$$q_i^{\text{exec}}(b) = \frac{\rho_\psi(b \mid z_i, e_i)\ell_b(z'_i \mid z_i)}{p_A(z'_i \mid z_i, e_i)}, \qquad \widehat{\Pi}_{\text{exec}}[b, e] = \frac{1}{n_e}\sum_{i:e_i=e} q_i^{\text{exec}}(b), \tag{4}$$

where n_e counts demonstrator e's training transitions. Each column of $\widehat{\Pi}_{\text{exec}} \in \mathbb{R}^{k\times m}$ estimates branch probabilities, provided the fitted responsibilities are correct and each head represents the same branch across states and demonstrators.

3.3 Stage B separates policies from the execution channel

Write $\Pi_{\text{c}}[a, e] = \Pr(a \mid e)$ and $\Pi_{\text{exec}}[b, e] = \Pr(b \mid e)$ for our state-independent policies. Then $\Pi_{\text{exec}} = W^\top \Pi_{\text{c}}$: every branch-frequency column is a weighted average of channel rows. For full-rank W, these rows are the vertices of the *channel simplex*. Many simplices contain the same columns, including the full probability simplex $\Delta_{k-1} = \{x \geq 0 : \mathbf{1}^\top x = 1\}$ corresponding to $W = I$, where $\mathbf{1}$ is the all-ones vector. A *facet* of the channel simplex omits one vertex and contains mixtures that never select the corresponding command.

The *population criterion* uses exact underlying frequencies and selects the smallest-volume channel simplex by minimizing $\det(WW^\top)$ under exact fit and probability constraints (Schur, 2026; Fu et al., 2018). Policies avoiding different commands place frequency columns on different facets; Section 3.4 states the coverage needed for recovery. For estimated frequencies, we use

$$\begin{aligned} \min_{W,\Pi_{\text{c}}} \quad & \log\det(WW^\top + \varepsilon_B I_k) \\ \text{subject to} \quad & \|W^\top \Pi_{\text{c}} - \widehat{\Pi}_{\text{exec}}\|_F \leq \tau, \\ & W \geq 0, \quad W\mathbf{1}_k = \mathbf{1}_k, \qquad \Pi_{\text{c}} \geq 0, \quad \mathbf{1}_k^\top \Pi_{\text{c}} = \mathbf{1}_m^\top. \end{aligned} \tag{5}$$

Here $W \in \mathbb{R}^{k\times k}$ has nonnegative rows summing to one (row-stochastic), and $\Pi_{\text{c}} \in \mathbb{R}^{k\times m}$ has such columns (column-stochastic). The Frobenius norm is the square root of the sum of squared entries, and $\varepsilon_B = 10^{-9}$ stabilizes the log-determinant. We set $\tau = 0.5\widehat{\sigma}_F$, where $\widehat{\sigma}_F$ estimates the combined standard error for ideal independent branch counts; it is heuristic for learned trajectory assignments. A tighter tolerance may exclude the true channel; a looser one permits smaller simplices. The solver accepts residuals up to 1.5τ and, if needed, retries once at a larger tolerance (Appendix B.6). State-dependent policies yield only command frequencies averaged over each demonstrator's visited states.

3.4 When demonstrator diversity identifies the factors

Let $\mathcal{K} = \text{cone}(\Pi_{\text{c}})$ contain all nonnegative combinations of policy columns. Its *dual cone* is $\mathcal{K}^* = \{y : y^\top x \geq 0 \text{ for all } x \in \mathcal{K}\}$. Write ∂ for a boundary and u_j for the pure policy choosing only command j. The sufficiently scattered condition (Fu et al., 2018), building on conic-geometric uniqueness conditions for nonnegative factorization (Huang et al., 2014), requires

$$\mathcal{C}_k \subseteq \mathcal{K}, \qquad \mathcal{K}^* \cap \partial\mathcal{C}_k^* = \{\alpha u_j : \alpha \geq 0,\ j \in [k]\}, \tag{6}$$

where $\mathcal{C}_k = \{x : \mathbf{1}^\top x \geq \sqrt{k-1}\|x\|_2\}$ and $\mathcal{C}_k^* = \{y : \mathbf{1}^\top y \geq \|y\|_2\}$. The *inclusion clause* requires the policy hull—all mixtures of demonstrator policies—to contain the ball $\mathcal{C}_k \cap \Delta_{k-1}$ inscribed in the simplex, touching each facet at its center. The *boundary clause* allows only the simplex's facet planes to touch this ball while keeping the whole policy hull on one side. With full-rank W, scattering implies $\text{rank}\,\Pi_{\text{exec}} = k$, requiring $m \geq k$. For $m < k$, a singular exact fit attains zero volume (Appendix A.4).

With exact frequencies and full-rank W, every global population minimum recovers (W, Π_{c}) up to a common command permutation (Appendix Theorem A.1). Combining this with the executed-basis lemma permits independent command and branch relabelings, as in the channel metric of Equation equation 9. This argument assumes a Markov-sufficient representation, common state coverage, and continuous, distinct branches on a connected support (Appendix A.7). It does not establish recovery for neural fitting, positive ridge $\varepsilon_B > 0$, fit tolerance $\tau > 0$, or local optimization.

A construction and a comparison with predicted bias. For every $k \geq 4$, all $k(k-1)$ policies $\pi_{ij} = \frac{2}{3}u_i + \frac{1}{3}u_j$, $i \neq j$, satisfy both clauses (Appendix Proposition A.2). We use $k = 4, m = 12$. The cyclic family repeats $\frac{2}{3}u_i + \frac{1}{3}u_{i+1}$ three times per i, with indices modulo four. Its hull misses the facet center $(0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, violating inclusion. Both families match demonstrator count, policy entropy, and channel. Their *aggregate command prior*, $m^{-1}\sum_e \pi_e$, is uniform; their *identity-only accuracy*,

obtained by guessing each demonstrator's most likely command without observing a transition, is $2/3$. Their full policies and trajectory distributions still differ.

More generally, suppose $\Pi_c = S\Pi'$, where S is full-rank and doubly stochastic (nonnegative, with every row and column summing to one), and Π' is column-stochastic and sufficiently scattered. For full-rank W, the population minimum returns

$$\widetilde{W} = S^\top W, \qquad \widetilde{\Pi}_c = \Pi', \tag{7}$$

up to permutation (Appendix Proposition A.3). We call $W^\dagger = \widetilde{W} = S^\top W$ the *absorbed channel*, which incorporates command mixing. For cyclic, $S = \frac{2}{3}I + \frac{1}{3}P_{\text{cyc}}$ with $P_{\text{cyc}}u_i = u_{i+1}$, and Π' repeats each pure policy three times. Thus the recovered channel incorporates the policies' command mixing. Under *uniform replacement*, probability p redraws the executed action uniformly, including the original: $W_p = (1-p)I + (p/k)\mathbf{1}\mathbf{1}^\top$. The resulting mean row total-variation bias is $(1-p)/3$, or 0.25 at $p = 0.25$.

3.5 Recovered commands and selection protocol

The command-conditioned world model mixes learned branches using the recovered channel. Bayes' rule combines this model with a demonstrator's command prior to infer the command from a transition:

$$\widehat{p}(z' \mid z, a) = \sum_b \widehat{W}[a, b]\ell_b(z' \mid z), \qquad \widehat{q}^{\text{cmd}}(a \mid z, z', e) \propto \widehat{\Pi}_c[a, e]\widehat{p}(z' \mid z, a). \tag{8}$$

The world model is shared; identity supplies the command prior. *Grounding* assigns physical names to discovered commands using separately budgeted calibration labels after discovery (Appendix B.8).

We split complete episodes and standardize each latent coordinate using its training mean and standard deviation. An *outer run* is one complete seeded fit of the pipeline. It trains twelve Stage-A candidates (*restarts*) with different neural initializations and training randomness, selecting the highest validation mixture likelihood. Restarts share one initialization from clustering training changes $z' - z$. Every outer run and solver failure flag is retained. Ground-truth alignment is used only for evaluation, with information access specified in Appendix B.9. Appendix Algorithm 1 gives the complete procedure.

4 Experiments and analysis

Our experiments follow the identification argument: first test whether demonstrator geometry changes recovery from visual transitions, then isolate the factorization target, and finally examine when basis learning and downstream use limit the benefit.

4.1 Experimental design and evaluation

We use TwoRoom and PushT(Chi et al., 2024) with pixel observations, four directional actions, and frozen 192-dimensional visual encoders. Each dataset contains 840 transitions per demonstrator, using the twelve-demonstrator ordered-pairs or cyclic family from Section 3.4. Discovery uses no action labels; encoder pretraining uses actions. Episode splitting, train-only standardization, twelve likelihood-selected restarts, and the Stage-B tolerance rule are fixed across comparisons. Selection uses *validation* data; independent test uses new trajectories from the same demonstrators. S-fresh (generation seeds 5–9) provides the principal symmetric replication; S-init (0–2) provides the initial comparison, identity and noiseless controls, command-accuracy evaluations, and planning. A-fresh1 (13–17) and A-fresh2 (18–23) test asymmetric recovery; A-fresh2 also supports the identity-aware baseline and paired encoder/initialization comparisons. Appendix C gives the full protocol.

Our primary outcome is mean row total variation (TV) between the estimated and target channels, allowing independent permutations of commanded rows and executed columns,

$$\mathcal{E}(\widehat{W}, T) = \min_{R,C \in \mathcal{P}_k} \frac{1}{2k} \sum_{a=1}^{k} \sum_{b=1}^{k} \left| (R\widehat{W}C^\top)_{ab} - T_{ab} \right|. \tag{9}$$

Here $\mathcal{P}_k$ contains the $k \times k$ permutation matrices, and T is the true channel W or the absorbed target $W^\dagger$. Truth is used only for evaluation; associated factors are relabeled consistently. Unless

conditioned explicitly, means retain all outer runs, with generated datasets as replication units and training seeds nested within them. A_{cmd} denotes matched commanded-action accuracy after Stage B. The diagnostic $A_{\mathrm{A}\to\mathrm{cmd}}$ instead matches Stage-A branch labels to commands; $A_{\mathrm{A}\to\mathrm{cmd}} > 0.75$ marks a good basin for analysis, never selection. For both noisy visual-experiment channels, the data-free estimate $\widehat{W} = I$ has error 0.188; this gives a reference for whether estimating noise improves on ignoring it. Appendix C.3 defines these evaluations and uncertainty.

4.2 Demonstrator geometry changes the recovered channel

On S-fresh, pairs gives mean channel error $\mathbf{0.067}$, compared with $\mathbf{0.332}$ for cyclic (Table 1). The cyclic-minus-pairs difference is positive on all five datasets, ranging from 0.184 to 0.324. Its mean is 0.266, with a nominal 95% exhaustive dataset-bootstrap interval $[0.222, 0.302]$; the exact one-sided sign test gives $p = 1/32$. These secondary calculations use the reported rounded run values and only five independent datasets (Appendix D.2). The contrast replicates S-init's all-run errors of 0.113 versus 0.349.

Table 1: **Predictions and observations on symmetric TwoRoom. Top:** S-fresh channel TV to W, all ten runs per family; "good" conditions on $A_{\mathrm{A}\to\mathrm{cmd}} > 0.75$. Predicted values are population errors, not finite-sample bounds. **Bottom:** S-init matched commanded accuracy; groups and splits are explicit. The pairs reference uses true identity and executed bin; the cyclic reference is the absorbed model's identity-only prediction. Neither reference uses learned features.

S-fresh family	Predicted TV	All-run TV	Good-basin TV	Good / all
Pairs	0	**0.067**	0.028	8/10
Cyclic	0.250	0.332	0.251	3/10
S-init group	**Models**	**Validation A_{cmd}**	**Test A_{cmd}**	**Reference**
Pairs, good basin	7	0.883	0.886	0.896
Pairs, other runs	5	0.766	0.774	0.896
Cyclic, all runs	12	0.638	0.639	0.667

The population analysis predicts different targets: pairs identifies W, while cyclic identifies $W^\dagger = S^\top W$, at distance $(1-p)/3 = 0.250$ from W for replacement probability $p = 0.25$. The S-fresh good-basin means, 0.028 and 0.251, follow this distinction (Figure 2). However, good basins occur in $8/10$ pairs runs and $3/10$ cyclic runs. The all-run gap therefore reflects both the different factorization targets and differences in basis learning; it does not isolate geometry from every consequence of the induced trajectories.

Command prediction provides a second observable. At the absorbed solution, each cyclic demonstrator has a pure *recovered* policy, so its posterior always selects that identity's primary command. Against the actual mixed policy, predicted accuracy is $2/3$. S-init cyclic models reach 0.639 on independent test trajectories. The seven good-basin pairs models instead reach 0.886, close to the 0.896 oracle that knows identity, policy, channel, and executed branch. Thus successful pairs models use transition information beyond the identity-only prediction that the cyclic population solution retains.

4.3 Population targets, sampling error, and identity controls

To remove basis learning and trajectory sampling from the comparison, we supply Stage B with exact executed frequencies. For pairs, the constrained solver reaches error below displayed precision on symmetric, asymmetric, and cyclic-slip channels; a penalized implementation gives 0.039, 0.066, and 0.247 (Appendix D.3). For cyclic, exact-input errors are 0.300, 0.250, and 0.200 at $p = 0.10, 0.25, 0.40$, matching absorption (Figure 2b). These tests isolate the target difference from Stage A; they validate the numerical solutions on these instances without certifying global convergence.

In Table 2, increasing the sample size from approximately 3,000 to 30,000 lowers pairs error from 0.039 to 0.013, while cyclic remains near 0.250. The near-pure control assigns probability $14/15$ to one command and $1/45$ to each other command; its error approaches the predicted bias 0.050. Thus more data reduce sampling error but cannot remove absorption bias. Appendix D.3 derives this target.

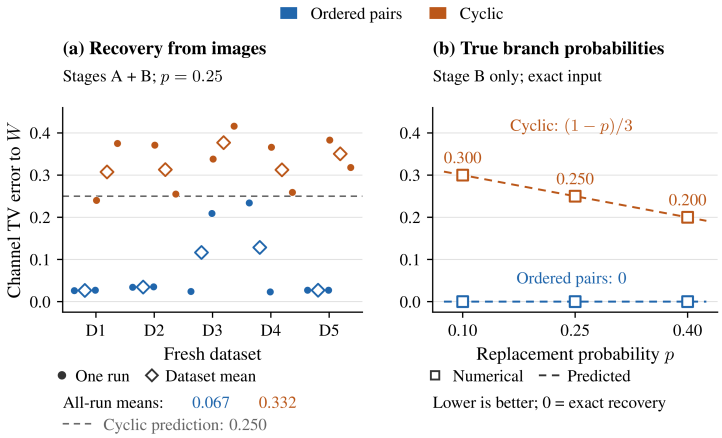

Figure 2: **Channel recovery and predicted bias.** Both panels show channel total-variation (TV) error to the true W on a shared scale; zero denotes exact recovery. **(a)** Circles show all ten S-fresh runs per family; diamonds average two runs per dataset (D1–D5, generation seeds 5–9). The gray dashed line marks the cyclic population prediction 0.250 at $p = 0.25$, not a bound on learned-model error. **(b)** Stage B uses exact branch probabilities, removing basis-learning and sampling error. Squares show numerical solutions; dashed curves give the predicted errors: zero for ordered pairs and $(1 - p)/3$ for cyclic. Cyclic bias therefore persists even with exact inputs.

Table 2: **Sampling error falls; absorption bias persists.** Stage-B channel TV to W at replacement probability $p = 0.25$. Exact-input values are numerical solutions; predictions follow the population analysis. Sampled values average ten replicates, with nominal transition total n and tolerance $0.5\widehat{\sigma}_F$, where $\widehat{\sigma}_F$ is the multinomial sampling scale. Pairs and near-pure share one study; cyclic comes from the separate noise sweep (Appendix D.3).

Family	Predicted	Exact input	$n = 3$k	10k	30k
Pairs	0	0.000	0.039	0.020	0.013
Near-pure	0.050	0.050	0.045	0.047	0.048
Cyclic	0.250	0.250	0.260	0.252	0.253

Two controls delimit the role of identity and noise. Erasing identity on S-init pairs data yields channel error 0.562 and validation accuracy 0.267, indicating failure of this decomposition when source labels are removed. With noiseless execution, error remains 0.093 and accuracy is 0.939, against a perfect-information benchmark of 1. Positive fit tolerance can itself favor a shrunken channel even when $W = I$ fits exactly; the available experiment does not separate this effect from basis error (Appendix D.4).

4.4 Compatible baselines and component ablations

An identity-aware Gaussian mixture tests whether demonstrator-specific weights suffice within a simpler transition model. It fits shared diagonal-Gaussian components to latent displacements, selects among twelve starts by validation likelihood, and uses the same Stage B. On A-fresh2, DivLAM gives error 0.074 versus 0.286 with the primary encoder, and 0.236 versus 0.449 with the second (Table 3). Only the primary-encoder DivLAM mean beats the noiseless reference 0.188. The neural estimator improves on this restricted mixture, but beating that baseline alone is insufficient evidence of useful channel recovery.

On S-init, identity-blind k-means and Gaussian mixtures give errors 0.347–0.355 and 0.321–0.399, versus DivLAM's all-run 0.113. Oracle-count tests, which sample branch counts from the true policies and channel, give 0.024–0.031 through the same Stage B. The seven good-basin pairs runs reach 0.020–0.028, showing that accurate basis learning brings the visual pipeline near this tabular benchmark. Appendix E gives baseline specifications, the Stage-A volume ablation, and VQ controls.

Table 3: **An identity-aware baseline on A-fresh2.** All-run channel error to W on the pairs family. Each of six datasets contributes two DivLAM outer runs or one twelve-start mixture fit. Both fitted methods use identity and the same Stage B. Restart counts do not imply equal computational cost.

Estimator	Primary encoder	Second encoder
DivLAM	$\mathbf{0.074}$	0.236
Identity-aware Gaussian mixture	0.286	0.449
Noiseless channel, $\widehat{W} = I$	0.188	0.188

4.5 Asymmetric recovery tests both target and structure

The asymmetric channel preserves mean diagonal probability 0.8125 while adding directional errors and a nonuniform executed prior. Across A-fresh1 and A-fresh2, pairs has mean error 0.077 to W, while cyclic has mean error 0.096 to its absorbed target $W^\dagger$; their dataset-bootstrap intervals are $[0.048, 0.106]$ and $[0.065, 0.130]$, respectively (Table 4). Together with the exact-input tests, these similar-scale errors support the predicted change of target: cyclic can approximate its absorbed channel despite failing to recover the physical one. They do not establish equal estimation difficulty for the families.

Table 4: **Fresh asymmetric recovery, primary encoder.** All-run means under the original protocol, with two runs per dataset. Errors are measured against each family's own population target. Both good-basin counts use $A_{\mathrm{A}\to\mathrm{cmd}} > 0.75$. The combined row gives equal weight to all eleven datasets.

Cohort	Datasets	Pairs to W	Good / all	Cyclic to $W^\dagger$	Good / all
A-fresh1	5	0.080	$7/10$	0.101	$5/10$
A-fresh2	6	0.074	$9/12$	0.091	$8/12$
Combined	11	0.077	$16/22$	0.096	$13/22$

The best truth-informed isotropic approximation has error 0.105. Pairs improves on it by 0.028 on average, but the improvement interval $[-0.001, 0.057]$ includes zero, and only six of eleven dataset means beat this reference. Conditional recovery is more accurate: A-fresh1's seven good-basin pairs runs average 0.027, near the oracle-count benchmark 0.022; its five good-basin cyclic runs average 0.021 to their own target. Calibration-anchored and composed-kernel checks further support recovery of directional structure in successful runs (Appendix D.5). The remaining challenge is obtaining such a basis reliably.

4.6 Basis learning limits transfer

The paired A-fresh2 study exposes this limitation (Table 5). Under the original protocol, pairs reaches the diagnostic good basin in $9/12$ runs with the primary encoder but $1/12$ with the second; cyclic shows the same encoder dependence. Supervised executed-action probes reach 0.984–0.987 on the second encoder, yet in a separate second-encoder diagnostic cohort (generation seeds 13–15), only two of 144 regenerated unsupervised restarts reach a good basin (Appendix F). Useful information is present, but the unsupervised procedure rarely accesses an accurate solution. Independently redrawing the clustering initialization at each restart does not resolve this: the second encoder's paired error change is $+0.016$, with interval $[-0.020, +0.060]$. The original protocol is retained.

PushT fails earlier in the pipeline. Mean Stage-A diagnostics are 0.302–0.390 across conditions, and pairs channel error 0.361 exceeds the noiseless reference 0.188. The near-zero cyclic-minus-pairs contrast, $+0.003$, therefore describes an end-to-end failure to recover either channel; it does not test identification conditional on an accurate basis. A supervised command probe achieves 0.570 on PushT versus 0.795 on TwoRoom, showing poorer performance of that probe without establishing an information ceiling. All fourteen exploratory runs that unfreeze the encoder collapse. The positive empirical result is recovery from TwoRoom pixels with the primary frozen encoder; reliable transfer to PushT remains unresolved (Appendix F).

Table 5: **Paired initialization confirmation on A-fresh2.** Each row has twelve runs per family. Shared initialization uses one clustering bias per outer run; independent initialization redraws it for every restart. Errors are all-run channel TV to the indicated targets. The last column counts good-basin runs for pairs (P) and cyclic (C); neither is a selection criterion.

Encoder	Initialization	Pairs to W	Cyclic to $W^\dagger$	Good P / C
Primary	Shared	0.074	0.091	$9/12$; $8/12$
Primary	Independent	0.068	0.056	$10/12$; $10/12$
Second	Shared	0.236	0.204	$1/12$; $1/12$
Second	Independent	0.253	0.216	$0/12$; $0/12$

4.7 Grounding and downstream use

Labels name the discovered commands after training. On all twelve S-init pairs models, 25 random validation labels give mean grounded test accuracy 0.834 and full-mapping probability 0.97, measured against the full-test Hungarian mapping. Mean accuracy rises from 0.701 with four labels to 0.839 with 100; this improves naming without retraining the model.

In TwoRoom planning, recovered-channel success is 61.7% versus 55.5% for an aligned channel-ignored control, with random and position-oracle references of 46.0% and 98.0%. The aggregate gain comes from the five less accurate models (59.6% versus 44.4%); the seven good-basin models give 63.1% versus 63.4%, and inserting the true channel gives 60.2% overall. With dataset differences of -6.0, $+11.5$, and $+13.0$ percentage points, this supports limited downstream use, but does not establish that channel accuracy improves planning (Appendix G).

5 Discussion

The experiments distinguish the population target from the visual estimator's access to it. Exact-input and sampled-count controls follow the predicted targets; successful visual runs approach them, while all-run errors also reflect basis learning. For cyclic policies at $p = 0.25$, even the population target is farther from the physical channel than ignoring noise (0.250 versus 0.188): minimizing the criterion more accurately need not improve physical-channel recovery.

Asymmetric experiments show directional recovery in successful runs, but the all-run advantage over the truth-informed isotropic reference remains unresolved. Encoder sensitivity and PushT failures identify executed-basis learning as a practical bottleneck without separating representation geometry from optimization as its cause.

Planning gains are concentrated in less accurate models and do not establish that improved channel recovery improves decisions. Comparing propagation of mean successors with execution branches could help isolate the planner's approximation error (Appendix G).

These results apply to a known discrete action count, observed identities, shared dynamics, state-independent policies and noise, and frozen, action-pretrained features. They motivate preserving identities and varying command supports when designing policies, while checking the full sufficient-scattering condition; demonstrator count and entropy alone do not certify coverage (Appendix J).

6 Conclusion

DivLAM separates command policies from stochastic execution through a learned executed basis and minimum-volume factorization. We prove sufficient scattering for ordered-pairs policies for every $k \geq 4$ and derive absorbed targets for specified violations. Exact-input and sampled-count tests confirm these predictions; TwoRoom experiments with an action-pretrained encoder connect policy coverage to visual channel recovery. The planning study does not establish a benefit from more accurate recovery. Policy coverage determines the factorization target in these constructions, while basis learning governs the visual estimator's access to it.

AI USE STATEMENT

We used generative AI tools to refine manuscript wording, assist with code debugging, and support selected aspects of methodological development, hypothesis refinement, proof preparation, and interpretation of results. The authors reviewed the AI-assisted material, checked the mathematical arguments and interpretations, and verified LLM-generated or modified code for correctness. All final research decisions and conclusions remained with the authors, who take responsibility for the manuscript and accompanying artifacts.

ETHICS STATEMENT

This work studies latent action discovery using simulated TwoRoom navigation and PushT object-pushing environments. Demonstrator identities denote simulated policies rather than individual people; the reported experiments do not involve human participants or personal data. The recovered commands are model variables defined up to relabeling and should not be interpreted as evidence of human intent. Although the approach may inform future control systems, the experiments do not establish the safety or reliability of deployment in physical environments.

REPRODUCIBILITY STATEMENT

Section 3 describes the two-stage estimator. Appendix A provides the theoretical assumptions and proofs, and Appendix B specifies the architecture, optimization, initialization, model selection, and complete estimation procedure. Appendix C documents data generation, frozen representations, preprocessing, evaluation metrics, and replication cohorts. Table 26 identifies the implementation entry points for the principal analyses. We will provide the codebase as supplementary material for review. Model checkpoints are omitted from the submission because of upload size restrictions; reproducing the checkpoint-dependent experiments requires these additional artifacts. Upon acceptance, we will publicly release the complete codebase and all model checkpoints used in this work.

REFERENCES

Marvin Alles, Xingyuan Zhang, Patrick van der Smagt, and Philip Becker-Ehmck. Latent action world models for control with unlabeled trajectories, 2025. URL https://arxiv.org/abs/2512.10016.

Tewodros Ayalew, Matthew Jeung, Samuel Wheeler, Xiao Zhang, Andre de la Cruz Arce, Kaylene Stocking, Michael Maire, and Matthew R. Walter. Claw: Learning continuous latent action world models via adversarial latent regularization, 2026. URL https://arxiv.org/abs/2606.04130.

Christopher M. Bishop. Mixture density networks. Technical Report NCRG/94/004, Aston University, 1994. URL https://www.microsoft.com/en-us/research/publication/mixture-density-networks/.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments, 2024. URL https://arxiv.org/abs/2402.15391.

Jiaheng Chen. Atm: Action-consistency transfer matrix for diagnosing and improving latent world models, 2026. URL https://arxiv.org/abs/2606.09028.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL https://arxiv.org/abs/2303.04137.

Ashley D. Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L. Isbell. Imitating latent policies from observation, 2019. URL https://arxiv.org/abs/1805.07914.

Xiao Fu, Kejun Huang, and Nicholas D. Sidiropoulos. On identifiability of nonnegative matrix factorization. *IEEE Signal Processing Letters*, 25(3):328–332, March 2018. ISSN 1558-2361. doi: 10.1109/lsp.2018.2789405. URL http://dx.doi.org/10.1109/LSP.2018.2789405.

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

Kejun Huang, Nicholas D. Sidiropoulos, and Ananthram Swami. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224, 2014. doi: 10.1109/TSP.2013.2285514.

David Klindt, Yann LeCun, and Randall Balestriero. When does lejepa learn a world model?, 2026. URL https://arxiv.org/abs/2605.26379.

Sébastien Lachapelle. On the identifiability of latent action policies, 2025. URL https://arxiv.org/abs/2510.01337.

Jung Min Lee, Taehyun Cho, Li Zhao, and Jungwoo Lee. Why latent actions fail, and how to prevent it, 2026. URL https://arxiv.org/abs/2605.20223.

Alexander Levine, Peter Stone, and Amy Zhang. Offline action-free learning of ex-bmdps by comparing diverse datasets, 2025. URL https://arxiv.org/abs/2503.21018.

Chia-Hsiang Lin, Wing-Kin Ma, Wei-Chiang Li, Chong-Yung Chi, and ArulMurugan Ambikapathi. Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case. *IEEE Transactions on Geoscience and Remote Sensing*, 53(10):5530–5546, October 2015. ISSN 1558-0644. doi: 10.1109/tgrs.2015.2424719. URL http://dx.doi.org/10.1109/TGRS.2015.2424719.

Alexander Nikulin, Ilya Zisman, Denis Tarasov, Nikita Lyubaykin, Andrei Polubarov, Igor Kiselev, and Vladislav Kurenkov. Latent action learning requires supervision in the presence of distractors, 2025. URL https://arxiv.org/abs/2502.00379.

Dominik Schmidt and Minqi Jiang. Learning to act without actions, 2024. URL https://arxiv.org/abs/2312.10812.

Felix Schur. Identifying latent actions and dynamics from offline data via demonstrator diversity, 2026. URL https://arxiv.org/abs/2603.17577.

Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R. G. Lanckriet. Hilbert space embeddings and metrics on probability measures, 2010. URL https://arxiv.org/abs/0907.5309.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. URL https://arxiv.org/abs/1711.00937.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos, 2025. URL https://arxiv.org/abs/2410.11758.

Chuheng Zhang, Tim Pearce, Pushi Zhang, Kaixin Wang, Xiaoyu Chen, Wei Shen, Li Zhao, and Jiang Bian. What do latent action models actually learn?, 2025. URL https://arxiv.org/abs/2506.15691.

Xiangteng Zhang, Yang Guan, Bo Zhang, Hongyang Li, Ya-Qin Zhang, and Shengbo Eben Li. On the identifiability of controlled world models, 2026. URL https://arxiv.org/abs/2607.22430.

A IDENTIFICATION, POLICY GEOMETRY, AND REPRESENTATION ASSUMPTIONS

A.1 THE GENERATIVE MODEL AND THE QUANTITY BEING ESTIMATED

Let $[r] = \{1, \dots, r\}$. Demonstrator identity $E \in [m]$ is observed, whereas command $A \in [k]$ and executed branch $B \in [k]$ are unobserved during discovery. Equal, known alphabet sizes make W square; the determinant criterion below uses this assumption. At state z, the demonstrator chooses A according to $\pi_e(a \mid z)$. A shared channel draws $B \sim W[A, \cdot]$, after which the system follows an executed transition kernel $T_b(\cdot \mid z)$. The model assumes that this last kernel does not depend on E or A after conditioning on (z, B), and that W does not depend on state or identity. Consequently,

$$p(dz' \mid z, a) = \sum_{b=1}^{k} W[a, b] T_b(dz' \mid z), \tag{10}$$

$$p(dz' \mid z, e) = \sum_{a=1}^{k} \pi_e(a \mid z) \sum_{b=1}^{k} W[a, b] T_b(dz' \mid z). \tag{11}$$

Here a kernel assigns a next-state probability distribution to each current state. Shared dynamics and the exclusion restriction mean that identity affects this distribution only through the command policy. The state must contain enough information for this Markov description. The controlled transition basis is idealized as $T_b(\cdot \mid z) = \delta_{f_b^*(z)}$, where δ_x is a point mass at x. Stage A uses Gaussian components around learned means instead. Its positive Gaussian width is a modeling choice, and does not turn deterministic physical branches into an exact Gaussian data-generating model.

From posterior assignments to frequencies. When the fitted responsibilities equal the true executed-branch posterior, the law of total expectation gives

$$\mathbb{E}[q^{\text{exec}}(b \mid Z, Z', E) \mid E = e] = \Pr(B = b \mid E = e). \tag{12}$$

This identity explains the training average in Equation equation 4. It does not establish that an imperfect neural posterior is unbiased. A single component index must also have a consistent executed-branch meaning across states and identities before its global frequency is interpretable.

For state-independent policies $\pi_e(a \mid z) = \Pi_{\text{c}}[a, e]$, the population relation is $\Pi_{\text{exec}} = W^\top \Pi_{\text{c}}$. With state-dependent policies and constant W, let d_e denote the distribution of states visited by demonstrator e. Then

$$\Pi_{\text{exec}} = W^\top \overline{\Pi}_{\text{c}}, \qquad \overline{\Pi}_{\text{c}}[a, e] = \int \pi_e(a \mid z)\, d_e(dz). \tag{13}$$

Only these visitation-weighted marginals are identified by an aggregate factorization. Substituting them for $\pi_e(a \mid z)$ in a posterior need not give the true state-conditioned posterior. If the channel is $W(z)$, the corresponding integral contains $W(z)[a, b]\pi_e(a \mid z)$ inside it; unequal state occupancies can prevent any common effective channel from representing all identities. The reported confirmatory construction uses state-independent policies and channels, matching the simpler model.

A.2 IDENTIFYING THE EXECUTED BASIS BEFORE THE CHANNEL

The observable law first determines which successors occur and how often each occurs. Separating their frequencies into commands and execution noise is a second identification problem. The following lemma makes the first step precise.

Lemma A.1 (Executed basis from the observable mixture)**.** *Let $\mathcal{Z} \subseteq \mathbb{R}^d$ be connected, with all demonstrator-conditioned laws observable on this common support. Suppose the executed kernels are point masses $\delta_{g_b(z)}$ with continuous g_b, that $g_b(z) \neq g_c(z)$ for every $b \neq c$ and $z \in \mathcal{Z}$, and that each branch has positive probability for some identity at every z. If k continuous heads $\widetilde{g}_b$, with nonnegative weights $\widetilde{\rho}(\cdot \mid z, e)$ summing to one, reproduce every conditional law exactly as an atomic mixture, then a single permutation σ satisfies*

$$\widetilde{g}_{\sigma(b)}(z) = g_b(z), \qquad \widetilde{\rho}(\sigma(b) \mid z, e) = \Pr(B = b \mid z, e) \tag{14}$$

for all b, z, e. Identity-level branch frequencies are consequently identified up to the same permutation.

Proof. At fixed z, the equally pooled conditional law $m^{-1}\sum_e p(\cdot \mid z, e)$ has exactly k distinct atoms, each with positive mass. Equality of finite atomic measures forces the k candidate heads to occupy those atoms bijectively. Each identity's weights are then its uniquely determined masses at the atoms, defining a permutation σ_z. Near any z_0, continuity and positive separation at z_0 keep each candidate head close to its matched atom and away from the others. Thus σ_z is locally constant. Connectedness makes this permutation constant on $\mathcal{Z}$. □

The lemma identifies an exact atomic mixture, independently of the Stage-A regularizers. Branch collisions, including coincident successors at walls, violate its separation assumption. Disconnected supports can admit different permutations on different components; missing conditional laws can prevent the common-support comparison. The finite-width Gaussian objective is an estimator of this basis, not an instance covered by the lemma.

A.3 The exact population criterion

For clarity, reserve (W_*, Π_*) for the true channel and commanded policy matrix in this subsection. The unregularized population problem is

$$\begin{aligned} \min_{W,\Pi_c} \quad & \det(WW^\top) \\ \text{subject to} \quad & \Pi_{\text{exec}} = W^\top \Pi_c, \qquad W \geq 0, \quad W\mathbf{1}_k = \mathbf{1}_k, \\ & \Pi_c \geq 0, \quad \mathbf{1}_k^\top \Pi_c = \mathbf{1}_m^\top. \end{aligned} \tag{15}$$

Rows of W are probability vectors, and each column of Π_{exec} lies in their convex hull. With full rank, the determinant is proportional to the squared volume of their simplex; the proportionality constant is fixed by the probability hyperplane. This gives the minimum-volume criterion its geometric meaning.

For a cone C, its dual is $C^* = \{y : y^\top x \geq 0 \text{ for every } x \in C\}$. The notation ∂C denotes its boundary. The two clauses of Equation equation 6 require both cone inclusion and the specified boundary intersection. A large rank, determinant, or sampled coverage score does not replace these clauses.

For geometric intuition, intersect $\mathcal{C}_k$ with $\mathbf{1}^\top x = 1$. Since $\|x - \mathbf{1}/k\|_2^2 = \|x\|_2^2 - 1/k$ on this hyperplane, the intersection is the ball centered at $\mathbf{1}/k$ with radius $1/\sqrt{k(k-1)}$, inscribed in Δ_{k-1}. It touches facet $x_j = 0$ at the centroid $c_j = (\mathbf{1} - u_j)/(k-1)$. Cone inclusion is equivalent to containing this ball in the convex hull of the policy columns. Representing c_j as a convex combination can use only policies with zero jth coordinate, because all entries are nonnegative. The dual-boundary clause excludes additional tangent supporting hyperplanes; inclusion alone leaves such ambiguities possible.

Theorem A.1 (Population channel recovery, inherited)**.** *Suppose $\Pi_{\text{exec}} = W_*^\top \Pi_*$, where W_* is a full-rank, row-stochastic $k \times k$ matrix and Π_* is a column-stochastic $k \times m$ matrix satisfying Equation equation 6. Every global minimizer of Equation equation 15 has the form $(PW_*, P\Pi_*)$ for a permutation matrix P.*

This is the channel specialization of the minimum-volume result of Schur (2026), based on sufficiently scattered factorization (Fu et al., 2018). We include the argument to make its normalization and scope explicit.

Proof. The cone condition implies $\operatorname{rank}(\Pi_*) = k$; thus $\operatorname{rank}(\Pi_{\text{exec}}) = k$. Any feasible pair has invertible W. Set $A = (W^\top)^{-1} W_*^\top$, so $\Pi_c = A\Pi_*$ and $W = A^{-\top} W_*$. Column stochasticity of both policy factors and full row rank of Π_* imply $\mathbf{1}^\top A = \mathbf{1}^\top$. Each row $r_i^\top$ of A is in $\operatorname{cone}(\Pi_*)^*$ because $A\Pi_* \geq 0$. By cone inclusion, $\|r_i\|_2 \leq \mathbf{1}^\top r_i$. Hadamard's inequality and arithmetic–geometric mean therefore give

$$|\det A| \leq \prod_i \|r_i\|_2 \leq \prod_i \mathbf{1}^\top r_i \leq \left(\frac{\mathbf{1}^\top A \mathbf{1}}{k}\right)^k = 1. \tag{16}$$

It follows that $\det(WW^\top) = \det(W_* W_*^\top)/\det(A)^2$ is at least the true value. A minimizer achieves equality. Equality requires each r_i to lie on the boundary of $\mathcal{C}_k^*$; the second scattering clause makes it

an axis ray. Invertibility and the sum constraints then make A a permutation matrix, which yields the claimed common relabeling. □

This theorem selects the true factors among exact-fit explanations; it does not say that every feasible explanation is true. It also does not identify each sampled command without error. If multiple commands can generate the same executed motion, the correct posterior itself can remain uncertain. Neither the theorem nor its proof includes Stage-A estimation error, a positive ridge, a relaxed fit constraint, or a local optimizer.

The identification chain and its two permutations. Lemma A.1 first gives a consistently indexed executed basis. Equation equation 12 then gives its identity-level frequencies. Finally, Theorem A.1 separates those frequencies into commands and the channel. If P_b relabels the executed basis and P_a relabels commands, the recovered factors have the form $P_a W_* P_b^\top$ and $P_a \Pi_*$, while the observable frequencies become $P_b \Pi_{\text{exec}}$. The two permutations are independent. This explains the metric in Equation equation 41 without invoking the different statewise kernel criterion discussed in Appendix A.7.

Why the basis Gram is absent from the default Stage B. For a fixed positive-definite basis Gram matrix K, square W obeys

$$\det(WKW^\top) = \det(W)^2 \det(K). \tag{17}$$

The unregularized objectives therefore differ by a positive constant. This equivalence does not extend to $\log\det(WKW^\top + \varepsilon_B I)$. The default solver explicitly uses $K = I$, even though the extraction routine also records a basis Gram as a diagnostic. Singular K would invalidate the positive-volume identification argument.

A.4 Rank requirements and the pooled-identity control

Sufficient scattering implies full row rank of the policy matrix. With full-rank W, the frequency matrix therefore has rank k, requiring at least k identities. Having $m \geq k$ is necessary but does not itself establish scattering.

Proposition A.1 (A singular exact fit when $m < k$)**.** *For any column-stochastic matrix $F \in \mathbb{R}^{k\times m}$ with $m < k$, a row-stochastic singular W_0 and a column-stochastic Π_0 satisfy $F = W_0^\top \Pi_0$. Thus the unregularized population minimum is zero.*

Proof. Use the m columns of F, transposed, as the first m rows of W_0, and fill its remaining rows by duplicating the first. Set column e of Π_0 to coordinate vector u_e. Then $W_0^\top \Pi_0 = F$ and $\operatorname{rank} W_0 \leq m < k$. □

For one pooled column f, the collapsed channel $W_0 = \mathbf{1} f^\top$ fits with *any* command prior, has rank one, and attains zero unregularized volume. All its commanded kernels are equal, so its command posterior is constant in the observed transition and equals that arbitrary prior. This is an illustrative global minimizer, not the unique singular factorization; a positive ridge, relaxed fit, and numerical optimization can select a different solution.

For a uniform pooled frequency $f = \mathbf{1}/k$ and true $W_{\text{ur}} = (1-p)I + p\mathbf{1}\mathbf{1}^\top/k$, the error of this collapsed solution, using Equation equation 41, is

$$\text{err}_W(W_0, W_{\text{ur}}) = (1-p)(k-1)/k = 0.5625 \quad \text{at } k = 4,\ p = 0.25. \tag{18}$$

Its transition-independent command prediction has population accuracy $1/k$ when the true aggregate command prior is uniform. These are reference values for the collapsed solution, not exact predictions for every finite-sample optimizer.

Pooling identities also changes Stage A: the implemented control retrains its mixture and retains the policy barrier $\lambda_\rho = 0.1$. Appendix B.3 shows why this barrier is always active at $m = 1$. Thus the control combines loss of factorization information with a changed Stage-A optimization problem; it does not isolate the effect of identity in either stage.

A.5 A PROVED POLICY FAMILY AND A MATCHED CYCLIC CONTROL

Proposition A.2 (Ordered pairs for every $k \geq 4$)**.** *For every integer $k \geq 4$, the matrix with the $k(k-1)$ columns $\pi_{ij} = \frac{2}{3}u_i + \frac{1}{3}u_j$, $i \neq j$, satisfies both clauses of Equation equation 6.*

Proof. Let $\Delta_{k-1} = \{x \geq 0 : \mathbf{1}^\top x = 1\}$. The convex hull of these columns is $H = \{x \in \Delta_{k-1} : x_i \leq 2/3 \ \forall i\}$. At a vertex, at most one coordinate can lie strictly between its bounds; the sum constraint forces the nonzero entries to be $2/3$ and $1/3$. Thus its vertices are precisely the ordered pairs. For nonzero $x \in \mathcal{C}_k$, normalize $\mathbf{1}^\top x = 1$. Then $\|x\|_2^2 \leq 1/(k-1)$. For any coordinate $t = x_i$, Cauchy–Schwarz gives $t^2 + (1-t)^2/(k-1) \leq 1/(k-1)$, or $t(kt-2) \leq 0$. Thus every coordinate lies in $[0, 2/k] \subseteq [0, 2/3]$, and $x \in H$. Rescaling proves the first clause.

For the second clause, the dual cone is $D = \{y : 2y_i + y_j \geq 0 \text{ for all } i \neq j\}$. If $y \geq 0$, then $\mathbf{1}^\top y = \|y\|_2$ only when at most one coordinate is positive. Otherwise let $y_i = -t < 0$. The inequalities force $y_j \geq 2t$ for every $j \neq i$; there can be no other negative coordinate. Hence $y = 3tr_i + v$, with $r_i = \frac{2}{3}\mathbf{1} - u_i$ and $v \geq 0$. Since $\mathbf{1}^\top r_i = (2k-3)/3 > \sqrt{4k-3}/3 = \|r_i\|_2$ for $k \geq 4$, as $(2k-3)^2 - (4k-3) = 4(k-1)(k-3) > 0$, the triangle inequality yields $\mathbf{1}^\top y > \|y\|_2$. Only coordinate-axis rays of D can therefore lie on $\partial\mathcal{C}_k^*$, and all those rays belong to D. □

The restriction $k \geq 4$ is substantive. At $k = 3$ the first clause holds, but $(-1, 2, 2)$ is an additional dual-boundary ray; at $k = 2$, the ordered-pair hull omits the simplex vertices and inclusion fails. The experiments use the $k = 4$ member of this family.

The construction includes no pure-action demonstrator. Every column has entropy $-\frac{2}{3}\log(\frac{2}{3}) - \frac{1}{3}\log(\frac{1}{3})$, approximately 0.6365 nats, and maximum probability $2/3$. Under equal demonstrator weights, its aggregate prior is uniform. The cyclic control keeps only columns $\frac{2}{3}u_i + \frac{1}{3}u_{i+1}$, with indices modulo four, and repeats each three times. Its cone excludes $x = (0, 1, 1, 1)^\top \in \mathcal{C}_4$. Indeed, if $x = \sum_i \alpha_i(\frac{2}{3}u_i + \frac{1}{3}u_{i+1})$ with $\alpha_i \geq 0$, its zero first coordinate forces $\alpha_1 = \alpha_4 = 0$. Coordinates two and four then require $\alpha_2 = 3/2$ and $\alpha_3 = 3$, which make coordinate three $5/2$ rather than 1. Inclusion fails. Equivalently, the only cyclic policies on facet $x_1 = 0$ are $(0, 2/3, 1/3, 0)$ and $(0, 0, 2/3, 1/3)$. Their segment misses the facet centroid $(0, 1/3, 1/3, 1/3)$ where the inscribed ball touches that facet.

Both families match the scalar quantities stated in the main text, with the same channel supplied during generation. They do not hold policy support or induced state occupancy fixed. Their policies are known for constructing and analyzing this controlled comparison, but are not inputs to discovery. Testing a finite set of directions for membership in a policy cone can demonstrate a violation, subject to numerical accuracy; successful membership tests do not prove inclusion of the full cone or the dual-boundary condition.

A.6 ABSORPTION AND CHANNEL-SPECIFIC BIAS

Proposition A.3 (Absorption)**.** *Let $\Pi_c = S\Pi'$, where Π' is column-stochastic and sufficiently scattered, and S is nonnegative, doubly stochastic, and full rank. If W is full-rank and row-stochastic, the global minimum of Equation equation 15 for $\Pi_{\text{exec}} = W^\top \Pi_c$ is $(S^\top W, \Pi')$, up to a common commanded-action permutation.*

Proof. The same frequencies satisfy $\Pi_{\text{exec}} = (S^\top W)^\top \Pi'$. The new channel is nonnegative and full rank, with $(S^\top W)\mathbf{1} = S^\top \mathbf{1} = \mathbf{1}$. Apply Theorem A.1 to this alternative generating pair. □

This is a statement about which explanation the population criterion selects, not a conclusion that every family outside the sufficient condition must fail. For the cyclic control, take $S = \frac{2}{3}I + \frac{1}{3}P_{\text{cyc}}$, where $P_{\text{cyc}}u_i = u_{i+1}$, and let Π' contain each coordinate vector three times. The latter is sufficiently scattered and S is invertible. For $W = \alpha I + \zeta \mathbf{1}\mathbf{1}^\top$,

$$S^\top W - W = \frac{\alpha}{3}(P_{\text{cyc}}^\top - I), \qquad \frac{1}{2k}\sum_{a,b} |(S^\top W - W)[a, b]| = \frac{|\alpha|}{3}. \tag{19}$$

The second expression is the average total-variation distance between corresponding channel rows. Under uniform replacement,

$$W_{\mathrm{ur}} = (1-p)I + pJ, \qquad J = \mathbf{1}\mathbf{1}^{\top}/k, \tag{20}$$

a command is replaced by a uniformly chosen action with probability p, possibly by itself. At $k = 4, p = 0.25$, its diagonal is 0.8125 and Equation equation 19 gives 0.25. Exclusive slip instead uses $W_{\mathrm{es}} = (1-p)I + \frac{p}{k-1}(\mathbf{1}\mathbf{1}^{\top} - I)$, whose diagonal is 0.75 and whose cyclic bias at these settings is $2/9$. The latent experiments use uniform replacement; exclusive slip occurs in the separate softening study.

For a softened policy family, let $S_\eta = (1-\eta)I + \eta J$, with $0 \le \eta < 1$. The general absorption formula is

$$\widetilde{W}_\eta = (1-\eta)W + \eta JW. \tag{21}$$

It reduces to $(1-\eta)W + \eta J$ when W is also column-stochastic. For the two symmetric channels above, the corresponding row-TV errors at $k = 4, p = 0.25$ are $9\eta/16$ and $\eta/2$, respectively. The full-rank premise excludes $\eta = 1$ and rank-deficient channels. These analytic targets concern exact inputs and global population solutions; empirical proximity to them is a numerical finding.

A.7 Transfer through a frozen representation

Lemma A.2 (Conditional representation transfer)**.** *Let $\mathcal{S}$ be a compact, connected state support and $g : \mathcal{S} \to \mathcal{O}$ a continuous injective observation map. Let $\chi = \phi \circ g$ be continuous and injective, including the fixed standardization in ϕ. Suppose the state process is Markov, the exclusion restriction holds, the commanded kernels $T_a^{\mathrm{cmd}} = \sum_b W[a,b]T_b$ are supported on $\mathcal{S}$ and weakly continuous, and their embedded Gram has rank k at every state. Assume sufficiently scattered policies at each state and that the demonstrator-conditioned laws are observable on the common support where the result is asserted. With a Gaussian kernel mean embedding in latent space, the same statewise minimum-volume identification result holds after encoding. If both the true and recovered kernel families are continuous, their statewise permutations agree on one global permutation.*

Proof. Compactness and injectivity make χ a homeomorphism onto its image $\mathcal{Z}$. For a Borel set $D \subseteq \mathcal{Z}$, define $T_a^{\chi}(D \mid z) = T_a^{\mathrm{cmd}}(\chi^{-1}(D) \mid \chi^{-1}(z))$. Conditioning on z is equivalent to conditioning on its state preimage; Markov sufficiency and exclusion therefore transfer. Pushforward is linear on measures, so the observable mixture retains the same policy weights evaluated at $\chi^{-1}(z)$.

To transfer rank, suppose $\sum_a c_a \Phi_\kappa(T_a^{\chi}) = 0$ for some coefficient vector c, where $\Phi_\kappa(\mu) = \int \kappa(\cdot, x)\, d\mu(x)$. The Gaussian embedding is injective on finite signed measures (Sriperumbudur et al., 2010); hence $\sum_a c_a T_a^{\chi} = 0$. Injectivity of pushforward under the homeomorphism implies $\sum_a c_a T_a^{\mathrm{cmd}} = 0$, contradicting the original Gram rank unless $c = 0$. Continuous pushforward preserves weak continuity, and $\mathcal{Z}$ is compact and connected. The statewise minimum-volume result consequently applies. Full rank makes kernels pairwise distinct. Continuity of both families makes their matching permutation locally constant; connectedness then makes it constant over $\mathcal{Z}$. □

This lemma concerns the statewise commanded-kernel criterion, whereas DivLAM factorizes aggregate executed frequencies. Its role is to describe what an information-preserving encoder retains, separately from the chain in Lemma A.1 and Theorem A.1. The local-to-global step is inherited from Schur (2026); the lemma verifies that an information- preserving representation retains its hypotheses. It requires neither linearity nor orthogonality. Optional semantic labels are not an assumption for recovery up to permutation. Fixing the remaining names requires additional information that distinguishes every command; finite random calibration sets can miss commands or produce incorrect assignments.

For deterministic physical branches, an injective continuous state encoding preserves their distinct successors; compactness makes the inverse on its image continuous. It also preserves the Markov description. These are sufficient conditions for transporting the atomic-basis assumptions, rather than necessary conditions on every useful learned representation.

Approximate representations. Suppose two encodings differ by at most δ uniformly on the same state support and κ is L-Lipschitz in each argument. Compare the pushforwards of the same physical transition kernels at the same current state. Each entry of their embedded Grams satisfies

$$\max_{a,b} |G_{ab} - \widetilde{G}_{ab}| \leq 2L\delta. \tag{22}$$

This follows by integrating the two argument-wise Lipschitz bounds. It is an entrywise, same-state bound; it neither rules out state aliasing nor establishes stability of minimum-volume optimizers. The pretrained checkpoints are not certified to satisfy injectivity, Markov sufficiency, full rank, or branch separation. These conditions delimit the theoretical interpretation of empirical recovery.

B Complete estimation and inference procedure

B.1 Notation and representation preprocessing

Table 6: Objects used by DivLAM. m is the number of demonstrators, k is the known number of commands and executed branches, and d is the latent dimension. Every stochastic matrix is normalized along the axis indicated.

Symbol	Shape or domain	Meaning
(o_i, o_i', e_i)	$e_i \in [m]$	Observed transition and demonstrator identity
z_i, z_i'	$\mathbb{R}^d$	Frozen, train-standardized features
a, b	$[k]$	Hidden command and executed branch, respectively
$f_b(z)$	$\mathbb{R}^d$	Shared successor mean for branch b
$\rho_\psi(\cdot \mid z, e)$	Δ_{k-1}	Executed-branch mixture weights
$R_\psi(z)$	$k \times m$	Columns $\rho_\psi(\cdot \mid z, e)$ for all identities
q_i^{exec}	Δ_{k-1}	Posterior branch responsibility
$\widehat{\Pi}_{\mathrm{exec}}$	$k \times m$	Training mean responsibility per identity; columns sum to one
W	$k \times k$	Execution probabilities given commands; rows sum to one
Π_{c}	$k \times m$	State-independent command policies; columns sum to one
$\sigma, \varepsilon_A, \varepsilon_B$	positive scalars	Gaussian width and Stage-A/Stage-B ridges
h, γ, τ	scalars	RBF bandwidth, policy-barrier threshold, and nominal fit tolerance

The frozen encoder produces a raw 192-dimensional ViT [CLS] feature $x^{\mathrm{raw}} = \mathrm{enc}(o)$. We use the feature before the projection MLP. Input RGB images have size 224×224 and are scaled to $[0, 1]$, then normalized with channel means $(0.485, 0.456, 0.406)$ and standard deviations $(0.229, 0.224, 0.225)$. We use a ViT-Tiny encoder with 12 layers, 3 attention heads, patch size 14, and intermediate width 768. Encoder weights remain fixed throughout both discovery stages.

Let $\mathcal{I}_{\mathrm{tr}}$ and $\mathcal{I}_{\mathrm{val}}$ be disjoint transition-index sets obtained by splitting complete episodes. We use a fixed split seed of 10,000 and holds out 20% of episodes; the confirmatory datasets have four transitions per episode. We compute each coordinate's mean μ_j and population standard deviation s_j from current-frame features in $\mathcal{I}_{\mathrm{tr}}$, then apply

$$z_j = \frac{x_j^{\mathrm{raw}} - \mu_j}{s_j + 10^{-8}}, \qquad z_j' = \frac{x_j^{\mathrm{raw}\prime} - \mu_j}{s_j + 10^{-8}} \tag{23}$$

to all splits and later inference inputs. This is diagonal standardization, rather than full covariance whitening. Each identity must have training observations to form its frequency column. The same normalization is retained for calibration and test data; it is not refitted there.

B.2 Neural parameterization

Each executed branch has a learned embedding $c_b \in \mathbb{R}^{32}$. The successor functions share two residual blocks. For block $j \in \{0, 1\}$, write $x_0 = z$ and

$$(s_j(c_b), t_j(c_b)) = M_j c_b + b_j, \tag{24}$$

$$x_{j+1} = x_j + \mathrm{MLP}_j((1 + s_j(c_b)) \odot \mathrm{LN}(x_j) + t_j(c_b)), \tag{25}$$

$$f_b(z) = z + U x_2 + b_{\mathrm{out}}. \tag{26}$$

The scale and shift vectors s_j, t_j lie in $\mathbb{R}^d$; $M_j, b_j, U, b_{\text{out}}$ are learned affine-map parameters with the dimensions implied by these expressions. Layer normalization LN has no learned affine parameters, $\odot$ denotes elementwise multiplication, and each MLP maps $d \to 256 \to d$ with GELU activation. Modulation weights and biases start at zero; the output weights start with standard deviation 0.02 and zero bias. Thus the functions initially coincide, and learned branch conditioning subsequently separates them. Neither identity nor a command label enters these functions.

For mixture weights, an identity embedding $\xi_e \in \mathbb{R}^8$ and the current state feed a separate MLP of widths $(d+8) \to 64 \to k$, with GELU activation. A learned per-identity bias $\beta_e \in \mathbb{R}^k$ is added to its logits,

$$\rho_\psi(\cdot \mid z, e) = \mathrm{softmax}(\mathrm{MLP}_\psi([z, \xi_e]) + \beta_e)\,. \tag{27}$$

This Stage-A network is state-dependent even though the controlled generating command policies are state-independent. Its flexibility helps fit encoded transitions; Stage B still receives only the aggregate matrix in Equation equation 4.

B.3 Full Stage-A objective and schedule

For a minibatch $\mathcal{B}$, the likelihood is evaluated by log-sum-exp using

$$\log \ell_b(z_i' \mid z_i) = -\frac{\|z_i' - f_b(z_i)\|_2^2}{2\sigma^2} - d\log\sigma - \frac{d}{2}\log(2\pi). \tag{28}$$

The constant terms are retained for likelihood reporting and restart selection. The regularizers in Equation equation 3 are

$$G_i[b,c] = \exp\!\left(-\frac{\|f_b(z_i) - f_c(z_i)\|_2^2}{2h^2}\right), \tag{29}$$

$$V_A = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \log\det(G_i + \varepsilon_A I_k), \tag{30}$$

$$R_\psi(z_i)[b,e] = \rho_\psi(b \mid z_i, e), \qquad B_\rho = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \left[\gamma - \log\det\!\big(R_\psi(z_i)R_\psi(z_i)^\top + \varepsilon_A I_k\big)\right]_+. \tag{31}$$

Here $[x]_+ = \max(x, 0)$, and the weight matrix is evaluated for *all* m identities at each sampled current state. The default constants are $\lambda_A = 0.05$, $\lambda_\rho = 0.1$, $\varepsilon_A = 10^{-4}$, and $\gamma = -4$.

What the regularizers select. Both terms adapt the empirical estimator of Schur (2026), with no anchor loss. When all heads coincide, $G_i = \mathbf{1}\mathbf{1}^\top$ and

$$\log\det(G_i + \varepsilon_A I) = \log(k + \varepsilon_A) + (k-1)\log\varepsilon_A. \tag{32}$$

At $k = 4$ this is -26.245, whereas well-separated heads give $G_i \approx I$ and a value near $k\log(1+\varepsilon_A) = 0.0004$. Minimizing V_A therefore favors coincidence. The likelihood rewards representing distinct successor branches, but a finite weighted sum still trades fit against volume. For the exactly specified atomic class, Lemma A.1 already identifies the basis without either penalty. The Stage-A volume term is consequently a training choice, not an additional basis-identification requirement.

The policy barrier acts on executed-branch weights, whereas sufficient scattering concerns commanded policies. For the $k = 4, m = 12$ pairs and cyclic families under uniform replacement with $p = 0.25$, the population matrix $R = W^\top \Pi_{\text{c}}$ gives ridged log determinants -0.0250 and -0.7037, respectively. Both exceed $\gamma = -4$, so the barrier is inactive at either family's true branch weights. It therefore does not distinguish their contrasting identification properties.

What changes after identity pooling. The identity-erasure control retains $\lambda_\rho = 0.1$. With one identity, write $R = \rho$; the determinant lemma gives

$$\log\det(\rho\rho^\top + \varepsilon_A I) = (k-1)\log\varepsilon_A + \log(\varepsilon_A + \|\rho\|_2^2). \tag{33}$$

For $k = 4$, this ranges from -29.017 to -27.631, always below γ. Minimizing the active barrier favors a larger $\|\rho\|_2^2$, hence more concentrated weights. More generally, for $1 \le m < k = 4$, $\mathrm{rank}(RR^\top) \le m$ and $\mathrm{tr}(RR^\top) \le m$ imply $\log\det(RR^\top + \varepsilon_A I) \le (k-m)\log\varepsilon_A + m\log(1+\varepsilon_A) < -4$. The penalty cannot make a $k \times m$ matrix full row rank in this regime. This explains the optimization confound in Appendix A.4.

Bandwidth and training schedule. For each of the first 100 training steps, compute the median of squared pairwise distances between distinct head indices, clamp it below at 10^{-12}, and take its square root. The bandwidth is the first such estimate followed by an exponential moving average with weight 0.1 on the new estimate. It is frozen after step 100. This specifies a successor-mean RBF Gram, rather than the exact mean-embedding Gram of finite-width Gaussian distributions. The coincident initialization makes the first estimate 10^{-6}, but its weight after the remaining 99 updates is only 0.9^{99}. If heads subsequently separate far beyond the frozen bandwidth, G_i approaches I and the volume gradient becomes small. This conditional mechanism alone does not establish that the term is inactive in a trained model; that requires its bandwidth and volume diagnostics.

Every restart uses Adam with learning rate 10^{-3}, 3,000 updates, minibatches of 1,024 transitions sampled with replacement from the training split, and gradient-norm clipping at 5. The Gaussian width follows

$$\sigma_t = 0.5 + \min(t/1500, 1)(\sigma_{\text{end}} - 0.5), \qquad t = 0, \ldots, 2999, \tag{34}$$

where $\sigma_{\text{end}} = 0.15$ for TwoRoom and 0.30 for PushT. Loss weights remain constant. A nonfinite loss marks the restart as diverged. There is no anchor cross-entropy or encoder update in this discovery recipe. Mixture-head and joint-encoder variants are separate experimental arms.

B.4 Why not fit commanded kernels in one neural stage?

A commanded kernel is a mixture of executed branches when W is nontrivial. Giving each commanded head its own mixture is expressive enough, but likelihood alone also admits the executed explanation $W = I$. In development experiments, curriculum-initialized mixture heads stayed near the executed-basis solution under the tested moderate volume schedules; increasing the volume weight instead degraded command recovery toward alphabet collapse. On deterministic cells, mixture heads also admitted composite actions that merged distinct branches. These observations motivate restricting Stage A to point heads and imposing explicit constraints on the remaining matrix problem. They do not rule out a better one-stage optimizer or prove global convergence of the two-stage procedure.

B.5 Initialization and restart selection

At the start of outer run s, cluster the training deltas $z'_i - z_i$ into k groups with SciPy's `kmeans2`, using k-means++ initialization and seed s. Let H_{eb} be the number of training deltas from demonstrator e assigned to cluster b. With add-one smoothing, initialize the identity bias as

$$r_{eb} = \frac{H_{eb} + 1}{\sum_c H_{ec} + k}, \qquad \beta_{eb} = 3\left(\log r_{eb} - \frac{1}{k}\sum_c \log r_{ec}\right). \tag{35}$$

The same initial bias is copied to all twelve restarts; neural initializations and minibatch streams use seeds $1000s + r$ for $r = 0, \ldots, 11$. The clustering is not redrawn within this procedure. Consequently, these restarts are dependent attempts and do not justify an independent-trial success calculation.

After training, evaluate every completed restart using the unpenalized mixture NLL on $\mathcal{I}_{\text{val}}$ at its environment's fixed σ_{end}. Select the smallest value, without consulting action accuracy or channel error, and retain every restart's status and likelihood. If all restarts diverge, record the outer run as diverged. The selected model computes training responsibilities in batches; their sums are divided by training counts and columns are renormalized to guard against numerical drift. Stage B does not refit the neural heads or mix its objective into restart selection.

B.6 Stage-B tolerance, starts, and numerical acceptance

The sampling scale and nominal fit tolerance are

$$\widehat{\sigma}_F^2 = \sum_{b=1}^{k}\sum_{e=1}^{m} \frac{\widehat{\Pi}_{\text{exec}}[b,e](1 - \widehat{\Pi}_{\text{exec}}[b,e])}{n_e}, \qquad \tau = 0.5\widehat{\sigma}_F. \tag{36}$$

For independent categorical observations with exact branch labels, the analogous expression estimates the sum of variances of the frequency entries. Here the inputs are learned soft assignments from trajectories. The expression is a multinomial-inspired scale, not a confidence bound accounting for neural estimation error, temporal dependence, or misspecification.

Why a fraction of the sampling scale? For a fixed true channel W_*, let $\mathcal{H}_* = \mathrm{conv}\{W_*[a,\cdot]^\top : a \in [k]\}$. Optimizing only the policy columns gives the best true-channel residual

$$r_*^2 = \min_{\Pi \geq 0,\ \mathbf{1}_k^\top \Pi = \mathbf{1}_m^\top} \|W_*^\top \Pi - \widehat{\Pi}_{\mathrm{exec}}\|_F^2 = \sum_e \mathrm{dist}(\widehat{\Pi}_{\mathrm{exec}}[:,e], \mathcal{H}_*)^2. \tag{37}$$

An estimated column inside $\mathcal{H}_*$ contributes zero even when it differs from its population value. Hence $r_* \leq \|\widehat{\Pi}_{\mathrm{exec}} - \Pi_{\mathrm{exec}}\|_F$, and the residual can be smaller than the total frequency-estimation error. This motivates using a fraction of the sampling scale; it does not determine a universal coefficient. The value 0.5 is fixed in the confirmatory configuration and is a heuristic, not a confidence level.

If $\tau < r_*$, the true channel is excluded. Under symmetric replacement, expanding the candidate simplex toward $W = I$ can reduce this residual and corresponds to less mixing. Conversely, increasing τ expands the feasible set and can allow a lower-volume simplex inside the estimated columns, corresponding in that family to more mixing. A larger tolerance never increases the global minimum objective, but arbitrary channels and local solvers need not exhibit monotone entrywise noise bias.

An ideal-multinomial diagnostic. To examine the residual scale independently of neural estimation, we sampled 10,000 frequency matrices from the twelve $k = 4$ ordered-pair policies under uniform replacement at $p = 0.25$, with $n_e = 672$ independent categorical draws per identity. The true channel simplex is $\{x : \mathbf{1}^\top x = 1,\ x_b \geq p/k\}$, so Euclidean projection computes r_* exactly. With a fixed NumPy seed recorded in the verification script, the mean $r_*/\widehat{\sigma}_F$ was 0.3837; 9.50% of draws exceeded $0.5\widehat{\sigma}_F$ and 0.02% exceeded $0.75\widehat{\sigma}_F = 1.5\tau$, the first-call acceptance threshold. For ideal counts, the true channel therefore satisfies the nominal constraint in about 90% of draws and the acceptance rule in nearly all. This diagnostic does not calibrate coverage for learned trajectory assignments. The accompanying verification script specifies every sampling and projection step.

The solver packs $k^2 + km$ raw entries of (W, Π_{c}) into a single vector, bounds each entry in $[0,1]$, imposes row-sum and column-sum equalities, and adds the inequality $\tau - \|W^\top \Pi_{\mathrm{c}} - \widehat{\Pi}_{\mathrm{exec}}\|_F \geq 0$. It uses SLSQP with numerical derivatives, objective ridge $\varepsilon_B = 10^{-9}$, `ftol`$= 10^{-12}$, and eight starts. The first start is

$$W_0 = 0.9 I_k + 0.1 J, \qquad \Pi_{\mathrm{c},0} = \widehat{\Pi}_{\mathrm{exec}}. \tag{38}$$

It is near identity and need not satisfy the fit constraint. The other starts independently sample each channel row and each policy column from a Dirichlet distribution with all concentration parameters equal to one. The outer-run seed seeds this generator. We allow at most 500 SLSQP iterations per start in the confirmatory procedure; sampled tabular sweeps can explicitly override this cap.

Actual acceptance and fallback. Let r_j and L_j be a start's fit residual and ridged log determinant evaluated on the raw optimizer output. We mark the candidate accepted when $r_j \leq 1.5\tau$. Thus an acceptance flag is weaker than satisfaction of the displayed nominal constraint. Optimizer success is recorded separately and is not required by this flag; the flag itself checks fit residual, while the optimizer is responsible for the bounds and simplex equalities. Among accepted candidates, selection uses the smallest L_j. If none is accepted, the routine returns the candidate with smallest r_j, flagged as unaccepted.

We retry once with $\tau = \widehat{\sigma}_F$ if the first call returns no accepted candidate. This retry uses the same seed and eight-start rule and is recorded by its tolerance multiplier. If it also fails acceptance, the returned minimum-residual candidate remains an unresolved fit; it is not evidence of successful recovery. All outer-run records, including these flags, must remain visible in aggregation.

Before returning matrices, the routine clips negative entries and renormalizes channel rows and policy columns. Recorded residuals and objectives, including the acceptance decision, refer to the raw optimizer output before this normalization. Recomputed consistency of returned matrices is a distinct diagnostic. Neither numerical acceptance nor optimizer termination certifies a global minimum.

B.7 Pseudocode for the two-stage procedure

No true commands, executed labels, channel entries, or generating policy vectors are inputs to this algorithm. Optional post-discovery grounding never changes the selected discovery model. An unresolved Stage-B acceptance flag remains attached to any returned factors and predictions.

Algorithm 1 DivLAM for one outer run

Input Episode-grouped (o_i, o'_i, e_i), frozen encoder, known k, outer seed s, and frozen environment configuration.

Output Successor heads, $\widehat{W}, \widehat{\Pi}_{\mathrm{c}}$, and diagnostics, or a recorded Stage-A failure.

Split episodes into $\mathcal{I}_{\mathrm{tr}}, \mathcal{I}_{\mathrm{val}}$ using the canonical seed.
Encode observations; fit training standardization and apply Equation equation 23.
Compute one training-delta clustering bias β with seed s (Equation equation 35).
for $r = 0, \ldots, 11$ **do**
    Initialize (f, ρ) with seed $1000s + r$ and the shared bias β.
    Train with Equation equation 3; record completion or divergence.
    If completed, record validation mixture NLL at σ_{end}.
end for; if all restarts diverged, **return** the outer-run failure.
Select the completed restart with smallest validation mixture NLL.
Compute training responsibilities and $\widehat{\Pi}_{\mathrm{exec}}$ by Equation equation 4.
Compute $\widehat{\sigma}_F$ by Equation equation 36.
for $c \in (0.5, 1.0)$, in this order, **do**
    Set $\tau = c\widehat{\sigma}_F$ and run eight-start SLSQP as in Appendix B.6.
    Record raw residuals r_j, objectives, convergence, and acceptance $r_j \le 1.5\tau$.
    If any candidate is accepted, select the minimum-objective accepted candidate and **break**.
    If $c = 1.0$ and neither tolerance produced an accepted candidate, select the minimum-residual candidate from this second call and retain its failed-acceptance flag.
end for
Clip and normalize the selected factors, preserving the raw diagnostics.
Form command kernels and posteriors with Equations equation 8 and equation 39.
return the heads, $\widehat{W}, \widehat{\Pi}_{\mathrm{c}}$, and all restart and solver records.

Optional grounding Fit semantic names with separately budgeted calibration labels; freeze the map for test evaluation.

Table 7: Information access. "Labels" distinguishes any prior supervision from the absence of command labels in discovery.

Operation	Information used	Role of action information
Encoder pretraining	Upstream trajectory data and action-conditioned prediction	Labels influence the frozen encoder through pretraining gradients
Discovery training	Training observation pairs and identities; known k	No true commands, executed labels, true W, or true policies
Stage-A selection	Validation observation pairs and identities	Mixture NLL only; no accuracy-based selection
Stage-B selection	Estimated training frequency matrix and training counts	Objective and numerical fit acceptance only
Controlled design and diagnostics	Simulator policies, channel, and optional replayed branches	Privileged information for analysis, not discovery inputs
Semantic grounding	A separately counted labeled calibration set	Fits the command-name permutation after discovery
Independent test	Fixed model, fixed preprocessing, fixed calibration map	Labels score the declared test predictions

B.8 Posterior normalization, grounding, and relabeling

With the selected heads and σ_{end}, define the command kernel as in Equation equation 8. Its normalized posterior under the state-independent policy model is

$$\widehat{q}^{\mathrm{cmd}}(a \mid z, z', e) = \frac{\widehat{\Pi}_{\mathrm{c}}[a, e] \sum_b \widehat{W}[a, b] \ell_b(z' \mid z)}{\sum_{c=1}^{k} \widehat{\Pi}_{\mathrm{c}}[c, e] \sum_b \widehat{W}[c, b] \ell_b(z' \mid z)}. \tag{39}$$

The normalizer is positive because the policy column and channel rows are probability vectors and each Gaussian density is positive. We evaluate these sums in log space, adding 10^{-12} inside logarithms of channel and policy entries as a numerical guard, then takes the maximum a posteriori command. The returned channel also defines an identity-independent forward model for a chosen latent command. A posterior for a new demonstrator requires that demonstrator's policy prior; it is not supplied by the existing m policy columns.

Semantic grounding. For L separately budgeted labeled calibration transitions, let C_{uv} count how often latent command prediction u occurs with true command name v. Hungarian assignment chooses a permutation $\widehat{\pi} = \arg\max_\pi \sum_u C_{u,\pi(u)}$. This mapping is then frozen and applied to predictions on a disjoint test set. It changes names, not channel entries or neural parameters. Finite samples, ties, and imperfect predictions can produce incorrect maps; grounding cannot repair an incorrect discovered factorization. Independent-test grounding uses this separation.

Two latent index spaces. Channel rows index commanded actions and columns index executed branches, so structural comparison permits independent permutations. For permutation matrices P_a, P_b, relabel all associated factors as

$$W^{\mathrm{al}} = P_a \widehat{W} P_b^\top, \qquad \Pi_{\mathrm{c}}^{\mathrm{al}} = P_a \widehat{\Pi}_{\mathrm{c}}, \qquad \Pi_{\mathrm{exec}}^{\mathrm{al}} = P_b \widehat{\Pi}_{\mathrm{exec}}. \tag{40}$$

Reorder the successor list by the same P_b. These operations preserve the factorization and the composed command model. The structural channel metric, with W° the target channel, is

$$\mathrm{err}_W(\widehat{W}, W^\circ) = \min_{P_a, P_b} \frac{1}{2k} \sum_{a,b} \left| (P_a \widehat{W} P_b^\top - W^\circ)[a,b] \right|. \tag{41}$$

For $k = 4$, structural comparison enumerates all 24×24 assignments. It uses the true channel for evaluation, separately from the calibration-fixed semantic map. Both index spaces must be aligned, including when channels are asymmetric.

B.9 Information access and evaluation populations

Table 7 summarizes access at each stage. The frozen LeWM encoders were pretrained with actions supplied to an upstream predictor, whose loss updates the image encoder. Discarding that predictor and freezing the features does not erase this supervision history. The label-free claim applies to the subsequent discovery and selection procedure. Encoder sufficiency and injectivity remain assumptions, rather than certified consequences of its pretraining objective.

The confirmatory experiment selects restarts on its 20% validation split and reports action accuracy on that split. Although selection never accesses its action labels, those reported accuracies are validation results. Independent-test experiments use newly generated trajectories and separate calibration pools. We distinguish these evaluation populations from the validation results.

The stored Stage-A field `acc_matched` in the confirmatory experiment compares executed-head predictions to *commanded* labels. The supervised branch probe instead uses *executed* labels. They measure different targets; a common percentage scale does not make them directly comparable. Likewise, ground-truth "good basin" labels are diagnostic summaries, not restart-selection or run-exclusion rules. Dataset-level repetitions and optimization seeds have different roles: seed variation measures optimization on fixed data, whereas uncertainty over generated datasets requires aggregation at the dataset level.

C Data, representations, and evaluation

C.1 Environments and controlled policy variation

TwoRoom and PushT observations are 224×224 pixels. TwoRoom uses four unit directional commands (east, north, west, south), speed 5, and frame skip 5. PushT uses four directional bins scaled to 0.3. Generation uses four macro-steps per episode and randomized initial states. The short episodes were chosen during development to reduce wall-pinning and no-op transitions. Each confirmatory dataset has $m = 12$ demonstrators and 840 transitions per demonstrator, totaling 10,080

transitions. Dynamics configurations, including execution noise, are hash-checked for equality across identities. This verifies shared simulator configuration, not the Markov sufficiency of the visual features.

Let u_i be coordinate vector i in $\mathbb{R}^4$. The pairs family contains all twelve policies $\pi_{ij} = \frac{2}{3}u_i + \frac{1}{3}u_j$, $i \neq j$. The cyclic family repeats each policy $\frac{2}{3}u_i + \frac{1}{3}u_{i+1}$ three times, with indices interpreted modulo four. Identity-to-policy assignments are shuffled by generation seed. Both constructions have entropy $-\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = 0.63651$ nats per identity, a uniform aggregate command prior, and identity-only accuracy $2/3$. The pairs construction satisfies the sufficient-scattering condition, whereas the cyclic construction fails cone inclusion (Proposition A.2). Neither demonstrator count nor these scalar summaries is itself a compliance certificate.

Table 8: **Realized policy summaries in the initial cohorts.** Entries list generation seeds 0, 1, and 2. Entropy is averaged over identities. The final column is the maximum deviation of the empirical aggregate command prior from $1/4$; finite sampling prevents exact empirical matching.

Family	Entropy (nats)	Identity-only accuracy	$\max_a \lvert\widehat{\pi}(a) - 1/4\rvert$
Pairs	0.6401/0.6396/0.6416	0.660/0.661/0.658	0.0060/0.0058/0.0035
Cyclic	0.6385/0.6407/0.6386	0.663/0.660/0.663	0.0061/0.0056/0.0064

The main symmetric channel uses uniform replacement,

$$W_p = (1-p)I_k + \frac{p}{k}\mathbf{1}_k\mathbf{1}_k^\top. \tag{42}$$

Here p is the probability of redrawing the executed bin uniformly, which can redraw the commanded bin. Thus the probability of execution differing from the command is $p(1 - 1/k)$, not p. At $k = 4, p = 0.25$, the diagonal is 0.8125. The separate softening experiment instead used exclusive slip, with diagonal 0.75 and off-diagonal entries $0.25/3$. We keep those conventions separate in Appendix D.3. They also differ from the Gaussian width σ in the learned transition likelihood.

C.2 Frozen representations and data access

The primary TwoRoom encoder is a ViT-Tiny with a 192-dimensional raw CLS representation, pretrained in the LeWM/innovation-SIGReg setting. The second encoder uses an independent pretraining seed. Their pretraining predictor received actions and backpropagated into the encoder. The upstream action modules are discarded during discovery, but this does not remove the features' action-conditioned pretraining history.

The fixed episode split reserves 20% for validation, using split seed 10,000. Coordinatewise means and standard deviations are estimated on the training split and reused for validation and test. This is diagonal standardization, not full-covariance whitening. The original protocol draws one delta-clustering initialization bias per outer run and shares it across twelve restarts. Validation mixture likelihood selects a restart without labels. Stage B uses training responsibilities. Its detailed objective, acceptance criteria, and fallback appear in Appendix B.6. The fitted Gaussian width ends at 0.15 for TwoRoom and 0.30 for PushT; the latter was calibrated using validation fit during development.

The initial pairs/cyclic cohort began with two outer seeds per dataset; a declared extension added two more. These twelve-run summaries are not mixed with the six-run control summaries. For independent-test evaluations, new rollout randomness preserves the same demonstrators and identity assignments. This tests new trajectories from known demonstrators, not unseen-demonstrator generalization. Appendix G describes the separate use of validation labels for semantic calibration. Simulator policies, true channels, executed-label replay, and counterfactual branch queries enter only the stated privileged diagnostics and structural scoring.

C.3 Channel alignment, accuracy, and uncertainty

For row-stochastic matrices $A, B \in \mathbb{R}^{k\times k}$, define $d_{\mathrm{TV}}(A, B) = \frac{1}{2k}\sum_{a,b}|A_{ab} - B_{ab}|$. Equation equation 9 minimizes this distance over the two latent index spaces independently. At $k = 4$, all $24^2 = 576$ pairs of permutations are enumerated. If R, C are the minimizing permutations, the

Table 9: **Cohorts and replication units.** An outer run contains twelve Stage-A restarts. Development, original confirmation, and fresh confirmation are kept distinct. The final paired study contains $6 \times 2 \times 2 \times 2 \times 2 = 96$ outer runs across datasets, seeds, families, encoders, and initialization recipes.

Cohort	Purpose	Generation seeds	Runs/dataset/cell
S-init	Pairs/cyclic; identity/no-noise controls	0–2	4; 2 controls
S-fresh	Symmetric replication; initialization comparison	5–9	2
Asymmetric development	Metric and mechanism development	10–12	2
A-fresh1	Asymmetric primary-encoder confirmation	13–17	2
Second-encoder diagnosis	Earlier asymmetric transfer check	13–15	2
A-fresh2	Asymmetric; both encoders and recipes	18–23	2

aligned factors are

$$W^{\mathrm{al}} = R\widehat{W}C^{\top}, \qquad \Pi_{\mathrm{c}}^{\mathrm{al}} = R\widehat{\Pi}_{\mathrm{c}}, \qquad \Pi_{\mathrm{exec}}^{\mathrm{al}} = C\widehat{\Pi}_{\mathrm{exec}}. \tag{43}$$

The successor heads receive the same column relabeling. Thus $(W^{\mathrm{al}})^{\top}\Pi_{\mathrm{c}}^{\mathrm{al}} = C\widehat{W}^{\top}\widehat{\Pi}_{\mathrm{c}}$, preserving the fitted relation. Separate matches to W and $W^{\dagger}$ are separate diagnostics; their minimizing permutations need not agree. The actual learned posterior is always formed in the model's own consistent coordinates.

Matrices are aligned per run before any averaging. For a fixed target T, convexity requires

$$d_{\mathrm{TV}}\left(\frac{1}{N}\sum_{r=1}^{N} W_r^{\mathrm{al}}, T\right) \le \frac{1}{N}\sum_{r=1}^{N} d_{\mathrm{TV}}(W_r^{\mathrm{al}}, T). \tag{44}$$

Here N is the number of models in the displayed population. The mean-matrix error can be smaller than mean per-run error and must not replace it as the headline. The final asymmetric analyses use this check and a permutation-invariance test. For the isotropic W_p, independent column matching does not change the minimum true-channel error because W_p commutes with every permutation. That invariance does not generally hold for the absorbed target.

Let $\widehat{a}_i$ be the Stage-B posterior's predicted command. Its matched accuracy is

$$A_{\mathrm{cmd}} = \max_{\pi \in \mathfrak{S}_k} \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\pi(\widehat{a}_i) = a_i\}, \tag{45}$$

where $\mathfrak{S}_k$ is the permutation group, a_i is the evaluation command, and n counts scored transitions. This matching is an oracle evaluation operation. Grounded accuracy uses a mapping fixed by a paid calibration set instead. The runner's Stage-A score, denoted $A_{\mathrm{A}\to\mathrm{cmd}}$, substitutes $\arg\max_b q_i^{\mathrm{exec}}(b)$ for $\widehat{a}_i$ but still compares with *commanded* labels `a_true`. The diagnostic $A_{\mathrm{A}\to\mathrm{cmd}} > 0.75$ defines a good basin in the confirmatory analyses. It does not measure direct accuracy against true executed labels and never selects a model. The supervised executed-action probe uses `a_exec`; its percentages answer a different question.

For the pairs and cyclic policies at $p = 0.25$, an oracle that uses identity alone achieves $2/3$. If identity e has weight ω_e and policy π_e, an oracle that additionally observes the true executed bin b achieves

$$A_{e,b}^{\star} = \sum_{e} \omega_e \sum_{b} \max_{a} \pi_e(a) W[a, b] = 0.8958. \tag{46}$$

The weights are uniform here. This is a privileged identity-and-executed-bin benchmark, not a measured representation ceiling. Simply equating the executed bin with the command gives 0.8125. These references explain why good branch prediction and intended-action inference need different evaluations.

The cyclic population solution makes a more specific prediction. For an identity e whose dominant physical command is $i(e)$, the absorbed policy is $\pi'_e = u_{i(e)}$. The command posterior formed from this recovered policy is therefore

$$q^{\mathrm{cmd}}(a \mid z, z', e) = \mathbf{1}\{a = i(e)\}, \qquad \Pr\{\widehat{a} = a \mid e\} = \pi_e(i(e)) = \frac{2}{3}. \tag{47}$$

Here the indices are aligned with the primary physical commands, and π_e is the *actual* mixed demonstrator policy. The posterior is a point mass in the recovered coordinates; it is not equal to the original mixed prior. This gives the cyclic accuracy reference in Table 1. Its proximity to the observed test accuracy is a consequence predicted by absorption, not an additional identification theorem.

For the symmetric and fixed asymmetric visual-experiment channels, the noiseless estimate I_k supplies a second reference:

$$\mathcal{E}(I_k, W) = 1 - \frac{\mathrm{tr}(W)}{k} = 0.1875. \tag{48}$$

In general, permutation-invariant scoring replaces the trace by the maximum assignment sum. For these two channels, each diagonal entry is the largest in its row, so the diagonal assignment is optimal and the displayed expression follows. This reference uses no training data. It is distinct from the truth-informed best isotropic approximation for the asymmetric channel.

Initial cell intervals use 10,000 hierarchical bootstrap draws: resample generation datasets, then outer seeds within each selected dataset. The initial family contrast bootstraps paired dataset-mean differences. The final initialization comparison exhaustively resamples the six paired dataset differences ($6^6 = 46{,}656$ draws). Combined asymmetric intervals use 200,000 resamples of eleven dataset means. Unless otherwise specified, intervals use the 2.5th and 97.5th percentiles and are nominal 95% summaries, not guaranteed-coverage intervals. In particular, the three-dataset initial and PushT cohorts provide little information about between-dataset variation; their bootstrap endpoints should be read descriptively. Neither nested seeds nor additional bootstrap draws increase that replication count. The fresh symmetric $\pm$ summaries in the experiment record are across-run standard deviations. The dataset-level analysis is specified in Appendix D.2; it does not treat training seeds or calibration draws as new datasets.

D Expanded recovery results

D.1 Symmetric recovery and the initial controls

The fresh symmetric cohort uses the unchanged primary-encoder protocol on five generated datasets. Table 10 reports every outer run and the Stage-A diagnostic. The pairs-minus-cyclic separation is not driven by excluding poor solutions: all ten runs per family contribute. The pairs family has eight good-basin runs, compared with three for cyclic. Their conditional mean errors to W are 0.028 and 0.251. The two weaker pairs runs explain much of its across-run dispersion, while the unequal success frequencies show that basis-learning difficulty also differs between families.

Table 10: **Fresh symmetric per-run outcomes.** Within each entry, the two numbers are outer seeds 0 and 1. Both channel columns target the true W. $A_{\mathrm{A}\to\mathrm{cmd}}$ is the Stage-A matched diagnostic against commanded labels. Datasets D1–D5 in Figure 2 correspond to generation seeds 5–9. All entries retain the source table's three-decimal precision.

Dataset	Pairs error	Pairs $A_{\mathrm{A}\to\mathrm{cmd}}$	Cyclic error	Cyclic $A_{\mathrm{A}\to\mathrm{cmd}}$
g5	0.026/0.027	0.804/0.802	0.240/0.375	0.795/0.448
g6	0.034/0.035	0.805/0.804	0.371/0.255	0.438/0.810
g7	0.024/0.209	0.810/0.618	0.338/0.416	0.620/0.449
g8	0.234/0.023	0.630/0.808	0.366/0.259	0.443/0.807
g9	0.027/0.027	0.820/0.818	0.383/0.318	0.453/0.630

In the initial twelve-run pairs cohort, seven runs met the good-basin criterion, with channel errors 0.020–0.028. Their incidence varied by dataset ($4/4$, $1/4$, and $2/4$). Four initial cyclic runs with a good Stage-A diagnostic had true-channel errors 0.253, 0.258, 0.268, and 0.271, close to the 0.25 population prediction. Their low error to the absorbed target is conceptually different from recovering the true channel. Initial Stage-B commanded accuracies were 0.834 $[0.781, 0.883]$ for pairs, 0.638 $[0.612, 0.658]$ for cyclic, and 0.939 $[0.850, 0.994]$ for noiseless pairs. These are validation outcomes under the selection protocol, with the hierarchical intervals defined in Appendix C.3.

Table 11: **S-init geometry and control outcomes.** All outer runs are retained. $G \times R$ gives generated datasets and outer runs per dataset. Brackets are the recorded nominal 95% hierarchical bootstrap intervals; with only three datasets, their endpoints are descriptive. Accuracy is matched commanded accuracy on the validation split.

Condition	$G \times R$	Channel TV to W	Validation A_{cmd}
Pairs	3×4	$0.113\ [0.027, 0.207]$	$0.834\ [0.781, 0.883]$
Cyclic	3×4	$0.349\ [0.308, 0.390]$	$0.638\ [0.612, 0.658]$
Identity erased	3×2	0.562	0.267
Pairs, no noise	3×2	$0.093\ [0.022, 0.196]$	$0.939\ [0.850, 0.994]$

The initial paired cyclic-minus-pairs contrast is 0.236, with recorded dataset-bootstrap interval $[0.145, 0.320]$. The main uncertainty analysis uses the separate five-dataset S-fresh cohort rather than relying on this three-dataset interval.

Pooling all identities into one group preserves the pairs transitions but removes their source labels. The resulting channel error 0.562 and commanded accuracy 0.267 show that this pipeline does not recover the intended decomposition in that control. They should not be read as a universal accuracy bound for unidentifiable problems. A prior can make command prediction easy without making the channel unique.

D.2 Dataset-level uncertainty in the fresh symmetric contrast

This secondary analysis uses the three-decimal run entries in Table 10, with no model retraining. For generation dataset g, first average its two outer-run errors separately for each family, giving $\bar{\varepsilon}_{g,\mathrm{P}}$ and $\bar{\varepsilon}_{g,\mathrm{C}}$. Define the paired difference $\Delta_g = \bar{\varepsilon}_{g,\mathrm{C}} - \bar{\varepsilon}_{g,\mathrm{P}}$, so positive values favor pairs. The five differences are

$$(\Delta_5, \ldots, \Delta_9) = (0.2810, 0.2785, 0.2605, 0.1840, 0.3235), \qquad \bar{\Delta} = 0.2655. \tag{49}$$

Their minimum is 0.1840. Under independent dataset signs with probability $1/2$ of a positive difference, five positives give an exact one-sided sign-test value $2^{-5} = 0.03125$; the two-sided value is 0.0625. The direction follows the identification hypothesis, but this calculation is a secondary analysis of the reported results, not a newly preregistered confirmatory test. The test concerns the sign probability, not the magnitude of the mean improvement.

We also enumerate all $5^5 = 3{,}125$ ordered resamples of five dataset differences with replacement, assigning equal weight to each, and take the 2.5th and 97.5th percentiles of their means. This gives $[0.2218, 0.3019]$, reported as $[0.222, 0.302]$ in the main text. Exhaustive enumeration removes bootstrap Monte Carlo error, not the uncertainty from having only five datasets. The interval is nominal and has no finite-sample coverage guarantee. If every reported run error is rounded to the nearest 0.001, each difference, its average, and either bootstrap endpoint can shift by at most 0.001; none of the observed signs can change. The included analysis script reproduces these calculations.

D.3 Exact input, sample size, and tolerance

Exact-input tests provide the known matrix $\Pi_{\mathrm{exec}} = W^\top \Pi_{\mathrm{c}}$ directly to Stage B. This removes Stage-A estimation and trajectory sampling from the comparison. Table 12 gives the logged solver comparison. The constrained implementation uses raw simplex variables and an explicit fit tolerance, whereas the earlier solver uses a softmax parameterization and an annealed fit penalty. Exact zeros, the optimization landscape, and fit enforcement change together. Consequently, these tests validate the combined implementation choices on specific instances, not a causal decomposition of their effects.

For the cyclic policy family, define $S = \frac{2}{3} I + \frac{1}{3} P$, where $P u_i = u_{i+1}$ is the cyclic permutation. The policies factor as $\Pi_{\mathrm{c}} = S\Pi'$ with Π' consisting of repeated simplex vertices. The population minimum-volume target is then $W^\dagger = S^\top W$, as in Proposition A.3. Under uniform replacement, its true-channel error is $(1 - p)/3$. Thus decreasing error to $W^\dagger$ is evidence for the predicted biased factorization, not successful recovery of W.

Table 12: **Stage B on exact pairs-family inputs.** Channel TV in the known executed basis, with commanded rows matched. The numerical tolerance is 10^{-6} for the constrained comparison. 0.0000 means zero at displayed precision. The asymmetric synthetic channel here is distinct from the fixed asymmetric channel used in the visual experiments.

Channel	Constrained solver	Penalized solver
Symmetric replacement, $p = 0.25$	0.0000	0.0388
Asymmetric Dirichlet-perturbed	0.0000	0.0662
Cyclic slip, $(1-p)I + pP$, $p = 0.25$	0.0000	0.2466

For finite samples, the scripts draw commands and executed bins from the known policies and channel and form multinomial counts. Each nominal total n allocates $\lfloor n/m \rfloor$ samples per demonstrator, so the actual total is $m\lfloor n/m \rfloor$. The frozen tolerance is $\tau = 0.5\widehat{\sigma}_F$, where

$$\widehat{\sigma}_F^2 = \sum_{b,e} \frac{\widehat{\Pi}_{\text{exec}}[b,e](1 - \widehat{\Pi}_{\text{exec}}[b,e])}{n_e}. \tag{50}$$

Here n_e is the sampled count for identity e. This expression is the multinomial Frobenius sampling scale; for learned soft responsibilities it does not automatically account for Stage-A error or episode dependence. Sampled-input tabular comparisons use four solver starts and 200 iterations per start; the population sweep uses eight starts and 500 iterations. The neural confirmatory runner uses eight starts and inherits the 500-iteration default. These budgets should not be conflated.

Table 13: **Finite-sample solver and tolerance comparison.** Mean channel TV over ten replicates, uniform replacement $p = 0.25$. n is the nominal total number of transitions. The two policy families have different population targets. These rows come from one regenerated finite-sample study.

Family	Solver / tolerance	$n = 1$k	3k	10k	30k
Pairs	Constrained, $0.5\widehat{\sigma}_F$	.069	.039	.020	.013
Pairs	Constrained, $2\widehat{\sigma}_F$	.243	.142	.078	.046
Pairs	Penalized	.085	.068	.054	.047
Near-pure	Constrained, $0.5\widehat{\sigma}_F$	.051	.045	.047	.048
Near-pure	Penalized	.070	.040	.029	.027

The pairs-family error decreases across the sampled sizes. The looser tolerance allows substantially larger errors, even though fit remains acceptable under that rule. Neither the finite grid nor its monotone pairs curve establishes an asymptotic rate. The near-pure plateau has a different origin, which can be determined analytically.

The near-pure population target. Each near-pure column is $\pi_i = (1-\alpha)u_i + \frac{\alpha}{k-1}(\mathbf{1} - u_i)$, with $\alpha = 0.2/3$, and each of the $k = 4$ columns is repeated three times. Let $J = \mathbf{1}\mathbf{1}^\top/k$ and $\eta = k\alpha/(k-1) = 4/45$. Direct substitution gives

$$\pi_i = S_\eta u_i, \qquad S_\eta = (1-\eta)I + \eta J. \tag{51}$$

The matrix S_η is nonnegative, doubly stochastic, and invertible. The repeated vertices u_i form a sufficiently scattered policy matrix. Proposition A.3 therefore identifies the *global population* minimum as $(S_\eta^\top W, \Pi')$, up to commanded permutation, where Π' contains those vertices. Under uniform replacement, $JW = J$, hence

$$W_\eta = (1-\eta)W + \eta J, \qquad \mathcal{E}(W_\eta, W) = \eta(1-p)\frac{k-1}{k} = \frac{9\eta}{16} = 0.050. \tag{52}$$

The last equality uses $k = 4$ and $p = 0.25$. The eigenvalues of S_η are 1 and $1-\eta$ with multiplicity $k-1$, so the determinant-volume ratio is

$$\frac{\det(W_\eta W_\eta^\top)}{\det(WW^\top)} = \det(S_\eta)^2 = (1-\eta)^{2(k-1)} \simeq 0.572. \tag{53}$$

This establishes the target independently of numerical optimization. The exact-input solution has error 0.050, consistent with that target, while sampled errors at 3k–30k remain 0.045–0.048. For comparison, write $S_x = (1-x)I + xJ$. The channel $S_\alpha W = (1-\alpha)W + \alpha J$, with error 0.0375, also fits exactly with policy factor $S_\beta \Pi'$, where $\beta = (\eta - \alpha)/(1-\alpha) = 1/42$, because $S_\alpha S_\beta = S_\eta$. Its volume ratio $(1-\alpha)^{2(k-1)} \simeq 0.661$ exceeds 0.572, so it is not the minimizer. Agreement in error does not, by itself, certify that every numerical solution attains the population optimum.

The penalized solver's smaller near-pure error at 30k, 0.027 versus 0.048, is better recovery of W in that cell. It is not evidence of more faithful minimization of the exact population volume criterion, whose target is 0.050 from W. Finite-sample fit penalties and incomplete optimization can both leave a solution nearer to truth. The available summaries do not distinguish those mechanisms. This example separates fidelity to an estimation criterion from fidelity to the physical channel when identification fails.

Table 14: **Separate noise and sample-size sweep.** Channel TV to W_p, ten replicates per sampled cell. $\pm$ entries are replicate standard deviations; only means are tabulated for the cyclic family. This sweep uses a different sampling stream from Table 13, so overlapping cells need not coincide.

Family	p	Exact input	$n =$ 3k	10k	30k
Pairs	.10	.0000	$.030 \pm .004$	$.018 \pm .003$	$.010 \pm .002$
Pairs	.25	.0000	$.033 \pm .007$	$.022 \pm .005$	$.013 \pm .002$
Pairs	.40	.0000	$.046 \pm .022$	$.026 \pm .005$	$.014 \pm .004$
Cyclic	.10	.3000	.306	.301	.301
Cyclic	.25	.2500	.260	.252	.253
Cyclic	.40	.2000	.215	.204	.204

Softening the pairs matrix to $\Pi_\eta = (1-\eta)\Pi_c + \frac{\eta}{k}\mathbf{1}_k\mathbf{1}_m^\top$ provides a second controlled violation. Here $\eta \in [0,1)$ is the added uniform policy mass. For the doubly stochastic channels in that experiment, absorption predicts $W_\eta = (1-\eta)W + \frac{\eta}{k}\mathbf{1}_k\mathbf{1}_k^\top$. The exclusive-slip sweep at $\eta = 0, 0.02, 0.05, 0.10$ reports agreement with error $\eta/2$. Under replacement noise at $p = 0.25$, the corresponding coefficient would instead be $9\eta/16$. The coefficients must not be interchanged.

State-dependent noise tests further delimit the model. With replacement probabilities 0.4 and 0.1 in two equally visited regions and a common occupancy distribution across demonstrators, the exact-input solver recovers the occupancy-weighted channel at displayed error 0.0000, with error 0.1125 to either region-specific channel. Alternating demonstrator occupancies of 80/20 and 20/80 instead give a fitted channel 0.0675 from that common 50/50 average despite exact fit. In general,

$$\Pi_{\text{exec}}[:, e] = \sum_s d_e(s) W(s)^\top \pi_e \tag{54}$$

for state-independent command policy π_e, demonstrator occupancy $d_e(s)$, and state-dependent channel $W(s)$. Different d_e need not admit one shared effective channel. Feasibility alone cannot establish correct recovery: for any column-stochastic empirical matrix, $W = I$ and $\Pi_c = \widehat{\Pi}_{\text{exec}}$ already give an exact fit.

D.4 What identity removal and noiseless execution establish

With identities pooled, Stage B receives a single frequency column. If that column is exactly uniform, the rank-one channel $J = \mathbf{1}\mathbf{1}^\top/k$ is an exact-fit, zero-volume solution, with error $(1-p)(k-1)/k = 0.5625$ to W_p at $k = 4, p = 0.25$. This explains why a collapsed uniform solution is consistent with the observed 0.562 error. It is not a unique consequence of the criterion: other rank-deficient channels can also fit one column with zero volume, and the command posterior depends on the associated policy factor. The observed 0.267 accuracy therefore remains a pipeline outcome, not a universal chance bound or a uniquely predicted constant posterior.

For noiseless execution, positive tolerance introduces a different ambiguity even when Stage-A frequencies are exact. Let $\widehat{\Pi}_{\text{exec}}$ be any column-stochastic empirical frequency matrix. The true channel I_k fits it with zero residual by choosing the policy factor $\widehat{\Pi}_c = \widehat{\Pi}_{\text{exec}}$. For $0 < \lambda < 1$, keep

this factor and choose $W_\lambda = (1-\lambda)I_k + \lambda J$. Then

$$\|W_\lambda^\top \widehat{\Pi}_{\text{exec}} - \widehat{\Pi}_{\text{exec}}\|_F = \lambda \|(I_k - J)\widehat{\Pi}_{\text{exec}}\|_F, \qquad \det(W_\lambda W_\lambda^\top) = (1-\lambda)^{2(k-1)} < 1. \quad (55)$$

Thus any positive tolerance admits a sufficiently small $\lambda > 0$ with lower volume than I_k. If the norm vanishes, the residual is already zero for every such λ. The shrunken channel has spurious-noise error $\mathcal{E}(W_\lambda, I_k) = \lambda(k-1)/k$. This construction shows that residual noiseless error need not originate entirely in Stage A; it does not predict the implemented solver's error magnitude.

The observed noiseless result is channel error 0.093 and validation commanded accuracy 0.939. The appropriate perfect-information accuracy reference is 1, whereas the noisy pairs reference is 0.8958 in Equation equation 46. We do not report a matched noiseless oracle-count ablation, so these results cannot quantify the relative sources of channel error. Moreover, subtracting a tabular error from the end-to-end TV error would not yield an additive Stage-A contribution: basis estimation, the fit region, and optimization interact. The analytical construction identifies a source of bias without assigning unsupported numerical shares to those effects.

D.5 Asymmetric channels and composed transitions

The visual experiments use the fixed row-stochastic channel

$$W_{\text{asym}} = \begin{pmatrix} 0.90 & 0.08 & 0.01 & 0.01 \\ 0.01 & 0.85 & 0.12 & 0.02 \\ 0.02 & 0.02 & 0.80 & 0.16 \\ 0.20 & 0.04 & 0.06 & 0.70 \end{pmatrix}. \quad (56)$$

Its mean diagonal is 0.8125 and its condition number is approximately 1.66. Under a uniform command prior, the executed prior is $(0.2825, 0.2475, 0.2475, 0.2225)$. The truth-informed isotropic reference minimizes channel TV over $W_q = (1-q)I + \frac{q}{4}\mathbf{1}\mathbf{1}^\top$, $q \in [0,1]$. Its minimum error is 0.105; $q = 0.20$ is one minimizer. This reference uses the true channel and is not a deployable data-fitted baseline.

Table 15: **Primary-encoder asymmetric per-run outcomes.** Pairs errors target W; cyclic errors target $W^\dagger$. Slash-separated values are outer seeds 0 and 1. The accuracy columns are the validation Stage-A commanded-label diagnostic, not the final commanded posterior. Development runs are separated from both fresh batches. Values are reported to three decimals.

Dataset	Pairs error	Pairs $A_{\text{A}\to\text{cmd}}$	Cyclic error	Cyclic $A_{\text{A}\to\text{cmd}}$
Development				
g10	0.246/0.028	0.551/0.783	0.014/0.203	0.789/0.612
g11	0.016/0.260	0.809/0.444	0.021/0.024	0.802/0.801
g12	0.027/0.026	0.806/0.807	0.316/0.243	0.641/0.601
First fresh batch				
g13	0.238/0.022	0.612/0.814	0.029/0.208	0.806/0.596
g14	0.028/0.031	0.793/0.792	0.022/0.145	0.792/0.652
g15	0.084/0.025	0.740/0.790	0.021/0.019	0.793/0.795
g16	0.027/0.024	0.827/0.830	0.015/0.111	0.821/0.661
g17	0.029/0.291	0.792/0.534	0.226/0.211	0.529/0.534
Second fresh batch				
g18	0.022/0.022	0.811/0.811	0.024/0.256	0.796/0.591
g19	0.040/0.041	0.813/0.814	0.233/0.025	0.606/0.806
g20	0.223/0.018	0.668/0.803	0.051/0.049	0.808/0.810
g21	0.032/0.029	0.809/0.812	0.243/0.014	0.478/0.804
g22	0.194/0.027	0.657/0.818	0.018/0.122	0.805/0.628
g23	0.020/0.217	0.805/0.604	0.030/0.032	0.817/0.818

The development cohort (generation seeds 10–12) gives all-run pairs error about 0.100; four of six runs give 0.016–0.028. It is not pooled into the fresh estimate. The first fresh batch gives 0.0800, with

improvement over the isotropic reference 0.0250 and 95% dataset-bootstrap interval $[-0.023, 0.072]$. The second gives 0.0737. With equal weight for each of the eleven datasets, the combined mean is 0.0766 with interval $[0.048, 0.106]$. The cyclic family's combined error to its *absorbed* target is 0.0956 with interval $[0.0645, 0.1299]$. These targets answer different recovery questions. The per-run tables are rounded, so recomputed display means can differ from archived summaries in the last digit.

Structural matching is permissive about names. A separate privileged diagnostic therefore fixes commanded-row and executed-head mappings from training-split calibration labels, then tests the composed model on 24 simulator states with counterfactual branch queries. This diagnostic uses substantially more information than the small command-label budgets in Appendix G. For good-basin models, its anchored channel error equals the structural error at displayed precision. The same equality is recorded for all twenty good-basin pairs models across the four arms of the final paired confirmation.

For each queried state and command, the learned mixture $P = \sum_b \widehat{W}[a, b]\mathcal{N}(f_b(z), \sigma^2 I)$ is compared with the true encoded branch distribution $Q = \sum_b W[a, b]\delta_{\phi(g_b(s))}$. Here $g_b(s)$ is the simulator successor after executing branch b, and δ_x is a point mass at x. The squared maximum mean discrepancy is

$$\mathrm{MMD}_h^2(P, Q) = \mathbb{E}_{X,X' \sim P}\kappa_h(X, X') + \mathbb{E}_{Y,Y' \sim Q}\kappa_h(Y, Y') - 2\mathbb{E}_{X \sim P, Y \sim Q}\kappa_h(X, Y), \qquad (57)$$

where draws in each expectation are independent and $\kappa_h(x, y) = \exp[-\|x - y\|_2^2/(2h^2)]$ is the RBF kernel. The implementation takes h from the median pairwise distance among the first 256 standardized training successors in its bandwidth pool. Gaussian-mixture expectations are computed in closed form; scores are averaged over states and commands.

In the earlier diagnostic cohort, good-basin means are about 0.0004 for learned-channel composition, 0.0003 using the same heads with the true channel, and 0.004 when using the identity channel. These are descriptive comparisons, not a reported equivalence test. Less accurate models can have anchored channel error up to 0.09 above structural error. Some also compose better with their learned channel than with the true one, consistent with channel estimates compensating for basis errors. Such compensation does not constitute recovery of the physical channel.

E Baselines and ablations

E.1 Compatible executed-frequency estimators

The compatible baselines replace Stage A while retaining the frozen representation, training split, and Stage-B implementation. They therefore test whether simpler executed-frequency estimates support the same recovery. They do not compare complete systems with different encoders or additional supervision.

For the identity-blind baselines, the input is the standardized latent displacement $d_i = z_i' - z_i$. Four-component k-means uses twelve starts and selects by training inertia. A diagonal-covariance Gaussian mixture uses twelve initializations and selects by training likelihood. Hard cluster assignments are counted separately within each demonstrator to construct $\widehat{\Pi}_{\mathrm{exec}}$. Thus identity is absent during clustering but is used during aggregation and Stage B.

The privileged oracle-count control constructs $W^\top \Pi_{\mathrm{c}}$ from the simulator policies and true channel, then samples multinomial counts at the actual training count for each identity. It removes learned basis error while retaining count noise and numerical factorization. It is a reference for those effects, not a lower bound on all attainable errors or an implementable label-free method.

The stronger identity-aware baseline models

$$p(d \mid e) = \sum_{b=1}^{k} r_{be}\mathcal{N}(d; \mu_b, \mathrm{diag}(v_b)) \,. \qquad (58)$$

Here μ_b and the coordinate-variance vector v_b are shared across demonstrators; $r_{be} \geq 0$ and $\sum_b r_{be} = 1$ are identity-specific component weights. Expectation maximization uses twelve k-means++ starts, at most 500 iterations, covariance floor 10^{-6}, and weight floor 10^{-8} followed by renormalization. The stopping threshold is 10^{-7} change in per-sample training negative log likelihood. Its implementation

Table 16: **Initial symmetric pairs cohort.** Channel TV to W on generation seeds 0–2. The first three rows fit each dataset once after their stated internal initialization budget; DivLAM includes four outer runs per dataset. Cohort size and information access differ, so no equal-cost claim is implied.

Estimator	Dataset 0	Dataset 1	Dataset 2
k-means + Stage B	.355	.347	.350
Diagonal Gaussian mixture + Stage B	.321	.397	.399
Oracle executed counts + Stage B	.025	.031	.024
DivLAM, all twelve runs	0.113 overall mean		
DivLAM, seven good-basin runs	0.020–0.028 range		

includes a reseeding safeguard for components with negligible mass. Validation mixture likelihood selects within this method. The fitted matrix r becomes the Stage-B input, with the same tolerance and fallback rule used for DivLAM.

On the six final fresh datasets, the all-dataset baseline means are 0.286 and 0.449 for the primary and second encoders, respectively (Table 3). DivLAM is better on the aggregate comparison for both. However, Equation equation 58 is a restricted displacement model: DivLAM's successor functions depend on the current state and its mixture weights can depend on both state and identity. The comparison supports the full neural estimator over this specific baseline; it cannot attribute the gain solely to identity or rule out richer mixture models.

E.2 Vector-quantized controls

The local VQ control uses a learned discrete codebook (van den Oord et al., 2018) within an inverse/forward latent model, with four entries and the same frozen features and splits. An identity-conditioned variant adds a demonstrator-conditioned code prior. These are implemented comparison arms, not full reproductions of every published latent-action system. Their outputs support matched action accuracy but do not specify a commanded/executed channel factorization.

Table 17: **VQ controls on initial TwoRoom data.** Matched commanded-label accuracy on the canonical held-out validation split, six outer runs per family. The plain-control $\pm$ values are across-run standard deviations; the identity-conditioned summary reports means only. These accuracies are not channel TV errors.

Control	Pairs	Cyclic
VQ	0.603 ± 0.136	0.614 ± 0.111
VQ with identity-conditioned prior	0.423	0.481

The observed VQ accuracy difference between families is small relative to run dispersion. The identity-conditioned arm does not improve these means. Neither observation shows that no alternative VQ architecture could benefit from demonstrator information.

E.3 Stage-A regularization and Stage-B design

The frozen-protocol volume ablation changes only the Stage-A coefficient λ_A in Equation equation 3 to zero. The reported pairs results are shown below; each dataset contributes one outer seed rather than the four in the initial full-method cohort.

Two of three pairs runs recover a useful basis without the volume term, and the experiment record also notes one such cyclic run. The limited cohort gives no evidence that this term is essential for these successful solutions. It does not establish equal reliability or general irrelevance. In particular, this ablation does not remove Stage B's minimum-volume criterion, whose population role concerns the factorization ambiguity. The exact-input solver comparison in Table 12 changes several design choices together and is not a volume-only ablation.

Table 18: **Removing the Stage-A volume term.** Initial pairs datasets, one outer run per dataset, all retained. $A_{\mathrm{A}\to\mathrm{cmd}}$ is the validation Stage-A diagnostic scored against commands. The corresponding all-run mean channel error is approximately 0.114; this smaller cohort is not a paired estimate against the full twelve-run mean.

	Dataset 0	Dataset 1	Dataset 2
Channel error	0.027	0.022	0.294
$A_{\mathrm{A}\to\mathrm{cmd}}$	0.798	0.802	0.526

The exploratory deterministic loss study is kept separate in Appendix H. Its null regularizer effects cannot establish that each loss term is necessary, and its results use a different policy family and selection protocol.

F Optimization reliability and transfer limits

F.1 Restart dependence and selection

Good conditional recovery and reliable access to that solution are different outcomes. In development experiments, delta-based initialization produced seven good candidates among 32 restarts, compared with two among 32 for the alternative initialization. Because the twelve restarts of the frozen protocol share one clustering-derived bias within an outer run, they are not independent draws, and per-restart rates do not translate into outer-run success probabilities. Outer-run success frequencies are therefore reported per cohort in Tables 1 and 4. In a sixteen-restart development study, validation likelihood selected a good solution in all four pools containing one. This conditional selection result does not establish that good candidates are commonly available.

F.2 Second-encoder diagnosis

The second-encoder diagnostic cohort (generation seeds 13–15; Table 9) has all-run pairs error 0.1924, with only one of six runs in the good basin; that run has error 0.0223. Regenerating the twelve restart pools from both policy families gives 144 candidates. The regenerated pools reproduce the stored validation likelihoods exactly. Only two candidates exceed the diagnostic $A_{\mathrm{A}\to\mathrm{cmd}} > 0.75$, in two different pools. Likelihood selection chooses one and misses the other by about 0.03 negative log likelihood. This shows both rare access and imperfect selection, with candidate scarcity the larger observed problem in these pools.

The diagnostic score in this pool code compares predicted branches with *commanded* labels. The separate supervised probe instead uses replayed *executed* labels and the same successor-head architecture, trained for 4,000 steps. It reaches executed accuracy 0.984–0.987 on the second encoder and 0.985–0.988 on the primary encoder. Supervision can therefore extract an accurate branch model from either representation. This does not certify an identifiable unsupervised objective or show that feature geometry is irrelevant to optimization.

Additional diagnostics are consistent with that distinction. Delta-cluster purity against true executed labels is 0.40–0.46 for the second encoder versus 0.45–0.61 for the primary one in the measured development cohort. Full-covariance whitening reduces this purity to about 0.32 on both. Purity is privileged diagnostic information, not an available model-selection statistic. A post-hoc silhouette-based predictor does not reliably predict per-dataset recovery. A proposed shared-confusion model also fits soft marginal frequencies better than it predicts recovered channels. These checks do not justify a unique causal explanation for every failed run.

F.3 Independent initialization does not resolve the limitation

The candidate protocol draws a new clustering bias independently for each Stage-A restart. On the five fresh symmetric datasets, it gives pairs error 0.074 ± 0.083 versus 0.067 ± 0.078 under the original protocol, with eight good runs out of ten for each. A development pilot on the second encoder met a prespecified validation-likelihood criterion for proceeding to confirmation, but the held-out confirmation did not establish an improvement.

That confirmation crosses six new datasets, two outer seeds, two families, two encoders, and both initialization recipes, for 96 outer runs. No run failed, and none required the tolerance fallback. All runs remain in the analysis. Figure 3 plots pairs-family dataset means; Table 5 gives aggregate outcomes.

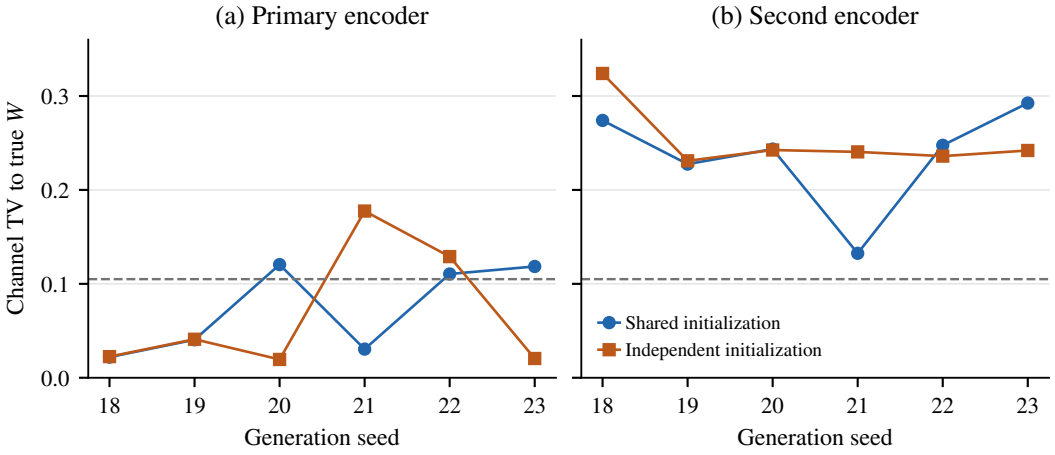

Figure 3: **Changing initialization across two encoders.** Each point is the mean of two outer runs on one pairs-family dataset. The horizontal line is the truth-informed isotropic reference 0.105. The second encoder stays above that reference in every dataset mean, and independently redrawing initialization does not establish a paired improvement. Table 19 gives all individual runs.

For the second encoder, the independent-minus-shared changes are approximately $+0.050$, $+0.003$, -0.001, $+0.108$, -0.011, and -0.051 across the six datasets. The paired mean is $+0.0163$ with exhaustive dataset-bootstrap interval $[-0.0203, +0.0602]$. For the primary encoder it is -0.0054 with interval $[-0.0662, +0.0603]$. The original protocol is retained. These intervals support neither a reliable benefit nor equivalence of the recipes. The failed intervention also limits any claim that increasing the number of independent initialization draws is sufficient to remedy the observed candidate scarcity.

Table 19: **Final paired study, all outer runs.** Each slash-separated entry is outer seed 0/1. Columns give pairs channel error to W and cyclic error to $W^\dagger$ for each initialization recipe; all 96 models are represented. Main-paper summaries use unrounded archived aggregates; these entries retain the source table's three-decimal precision.

Encoder	Dataset	Shared initialization		Independent initialization	
		Pairs	Cyclic	Pairs	Cyclic
Primary	g18	0.022/0.022	0.024/0.256	0.022/0.023	0.023/0.023
	g19	0.040/0.041	0.233/0.025	0.040/0.042	0.253/0.021
	g20	0.223/0.018	0.051/0.049	0.020/0.019	0.048/0.053
	g21	0.032/0.029	0.243/0.014	0.032/0.323	0.144/0.016
	g22	0.194/0.027	0.018/0.122	0.027/0.231	0.015/0.017
	g23	0.020/0.217	0.030/0.032	0.021/0.020	0.029/0.032
Second	g18	0.287/0.261	0.222/0.253	0.295/0.353	0.215/0.206
	g19	0.230/0.225	0.222/0.245	0.231/0.231	0.235/0.241
	g20	0.243/0.244	0.209/0.210	0.243/0.242	0.209/0.215
	g21	0.030/0.235	0.254/0.238	0.239/0.242	0.253/0.256
	g22	0.244/0.251	0.128/0.229	0.242/0.230	0.128/0.231
	g23	0.299/0.286	0.203/0.032	0.226/0.258	0.203/0.204

F.4 ALTERNATIVE FEATURES, PUSHT, AND JOINT TRAINING

Table 20: **Additional frozen-encoder controls.** Initial symmetric datasets 0–2, one outer run per cell and dataset. Positive separation means cyclic error minus pairs error. These small cohorts are kept separate from the primary twelve-run comparison.

Encoder	Pairs errors	Cyclic errors	Mean separation
Second pretraining seed	.259/.296/.201	.313/.368/.333	+.086
Marginal-SIGReg	.194/.284/.202	.320/.454/.371	+.155

Both alternative encoders retain a positive contrast on all three initial symmetric datasets, but neither reaches the good Stage-A basin in these small cohorts. This establishes a contrast in the measured samples, not broad reliable channel recovery. The projector-output representation also performs poorly, with mean Stage-A diagnostic accuracy about 0.45 in the recorded exploratory test.

Table 21: **PushT under the fixed protocol.** Six outer runs per cell. Channel intervals are hierarchical nominal 95% bootstrap intervals over only three generated datasets and nested outer seeds. Both accuracy columns use validation commanded labels; $A_{\mathrm{A}\to\mathrm{cmd}}$ scores Stage-A branch labels. Small-cluster intervals are descriptive.

Condition	Channel error	$A_{\mathrm{A}\to\mathrm{cmd}}$	A_{cmd}
Pairs	.361 [.335, .389]	.365	.471
Cyclic	.364 [.336, .389]	.372	.507
Identity erased	.576 [.562, .594]	.302	.267
Pairs, no noise	.486 [.461, .511]	.390	.482

PushT's paired cyclic-minus-pairs error is $+0.003$ with interval $[-0.023, +0.053]$. Mean Stage-A diagnostics remain 0.302–0.390, and both noisy-family channel errors exceed the noiseless-channel reference 0.1875. These outcomes locate a failure before accurate basis recovery. They provide no positive end-to-end evidence for the geometry advantage on PushT, while leaving identification conditional on an accurate basis unassessed in that environment. Aggregate diagnostic means alone do not establish that every individual run fails.

The PushT supervised diagnostic uses an MLP on $[z, z' - z]$, without identity, to predict commanded actions. It reaches 0.570, versus 0.795 on TwoRoom with the same probe family. This indicates lower achieved predictive performance, but cannot order the information contents of the representations or identify their Bayes accuracies. In particular, the 0.8125 reference for predicting the command by the executed bin is not a universal ceiling for this probe: its state input may also inform the demonstrator or command prior. This diagnostic is distinct from the supervised executed-label probe above.

Finally, fourteen exploratory joint-training runs unfreeze the encoder while varying anti-collapse strength, stop-gradient handling, and the number of demonstrators. All collapse: representation effective rank falls to approximately 1–4, responsibility mass concentrates on one action, and accuracy approaches the majority-class level. The outcome does not prove that joint learning is impossible. It establishes that the tested joint recipe does not provide a successful extension of the frozen-feature method.

G GROUNDING, INDEPENDENT TEST, AND PLANNING

G.1 WHAT THE LABEL BUDGET MEASURES

For each frozen model, the calibration experiment draws n_ℓ command labels without replacement from its validation pool and solves a Hungarian assignment between predicted command indices and those labels. The resulting permutation is fixed before scoring all independent-test transitions. Each budget uses thirty draws per model. Model parameters, selected restart, and channel remain fixed.

Two outcomes separate naming reliability from prediction quality. Mean grounded accuracy averages test accuracy over calibration draws and models. Full-mapping probability is the fraction of calibration

mappings equal to the Hungarian assignment computed from all labeled test predictions. That full-test assignment is an oracle *reference for scoring*, not the deployed mapping. It need not certify a unique physical naming for a poor model, particularly if its confusion matrix is ambiguous.

Table 22: **Random command-label budgets with independent test.** Means over all models and thirty draws per model. The symmetric cohort has twelve initial pairs models; the first fresh asymmetric cohort has ten pairs models. Values are rounded; 1.00 does not imply a mathematical guarantee of correct naming.

Cohort	Outcome	$n_\ell = 4$	10	25	100
Symmetric	Mean grounded accuracy	.701	.799	.834	.839
	Full-mapping probability	.64	.85	.97	1.00
First fresh asymmetric	Mean grounded accuracy	.725	.853	.874	.877
	Full-mapping probability	.64	.93	.98	1.00

The mean improves with budget because fewer mappings misname predicted commands, not because discovery is retrained. In the initial symmetric cohort, the reported pooled median remains near 0.882 across these budgets, while the mean rises substantially. That median hides meaningful tails: at four labels, per-model medians range from 0.481 to 0.896, the worst model's full-mapping probability is 0.23, and the worst observed draw has test accuracy 0.038. The worst model's mapping probability rises to 0.87 at 25 labels and 0.97 at 100. Four labels suffice when they happen to cover every command, but four randomly drawn labels do not reliably do so. The 50-label symmetric mean, not shown in the compact table, is 0.837.

In the final asymmetric confirmation, the original protocol at 25 labels gives mean grounded test accuracy 0.867 with the primary encoder and 0.712 with the second encoder, with mapping probabilities 0.98 and 0.81. The independent-initialization candidate gives $0.884/0.665$ and $0.99/0.71$, respectively. Better naming cannot in general repair an inaccurate latent basis or channel.

G.2 Validation and new trajectories

Table 23: **Initial symmetric models on independent test trajectories.** Matched commanded accuracy after Stage B; the matching in each scored split uses evaluation labels. Grounded accuracy in Table 22 instead fixes the mapping from its limited calibration sample.

Group	Models	Validation	Test
Pairs, good basin	7	.883	.886
Pairs, other runs	5	.766	.774
Cyclic, all runs	12	.638	.639

Observed validation-to-test changes are small for these frozen models. In the first fresh asymmetric cohort, pairs accuracy changes from 0.878 to 0.877 and cyclic accuracy from 0.652 to 0.653. In the final original-protocol cohort, pairs accuracy changes from 0.873 to 0.870 on the primary encoder and from 0.728 to 0.727 on the second. These observations show no material observed gap on the tested trajectories. They do not establish absence of overfitting in general, or transfer to new policies and demonstrators.

G.3 Planning with an aligned comparator

The TwoRoom planning evaluation covers all twelve initial pairs models. It uses exhaustive 4^6 latent action sequences with replanning, 50 paired episodes per model and arm, and 25 random command labels for grounding. The simulator applies the true replacement channel at $p = 0.25$ during execution. Paired arms share episode randomness.

The learned-channel arm propagates the predicted successor mean for a commanded action,

$$\bar{f}_a(z) = \sum_b \widehat{W}[a, b] f_b(z), \tag{59}$$

where $f_b(z)$ is the learned successor for executed branch b. Applying subsequent nonlinear predictions to this mean approximates multi-step stochastic prediction; the planner does not enumerate all possible execution-noise branches. The channel-ignored control uses the dominant learned branch

$$b^*(a) = \arg\max_b \widehat{W}[a, b]. \tag{60}$$

The map is bijective for all twelve evaluated models. This aligns commanded and executed indices before removing the stochastic mixture. A privileged arm uses the true channel in the learned heads' coordinate system. The position oracle and random policy are additional reference arms, not equally informed estimators.

Table 24: **TwoRoom planning.** Mean success rates over all twelve models and conditional basin groups. Each learned-model arm has 50 paired episodes per model. The oracle and random entries are reference-policy results.

Arm	All models	Good basin (7)	Other runs (5)
Recovered channel	.617	.631	.596
Aligned channel-ignored	.555	.634	.444
True channel, privileged	.602	—	—
Position oracle, privileged	.980	—	—
Random policy	.460	—	—

The recovered-minus-ignored difference is positive in eight models, zero in two, and negative in two, with range -0.24 to $+0.28$. Dataset-mean differences are $-0.060, +0.115, +0.130$, making the three generated datasets the appropriate small replication set. The episode-level discordant counts are 143 versus 106. We report these counts descriptively, without a significance test. Models share generation datasets and the evaluation reuses episode seeds across models, so the 600 model–episode pairs are not independent replicates for population inference. Conditioning on the fitted models alone does not remove this shared-episode dependence.

The improvement is concentrated in the five less accurate models, which gain about 15.2 percentage points, while the good-basin means differ by only -0.3 points. The other-run channel-ignored mean, 0.444, is also below the random-policy reference 0.460; the recovered-channel mean is 0.596. These comparisons do not establish that more accurate channels cause better planning. Averaging imperfect successor predictions is one possible explanation, but no planner-side intervention isolates it. Recovered-minus-true-channel success is $+0.015$, with dataset-bootstrap interval $[-0.062, +0.093]$. The interval supports neither equivalence nor a resolved advantage.

H Exploratory evidence and provenance

H.1 Earlier studies and their scope

The early grid comprises 52 cells with three training seeds on one generated dataset, using anchored or Dirichlet policy families and a preliminary selection protocol. Its sampled conic-coverage statistic uses known simulator policies and finite membership tests. It is a diagnostic, not proof of sufficient scattering. The later pairs construction and frozen cohorts therefore carry the principal recovery claims.

In the early stochastic TwoRoom comparison at $p = 0.25$, the anchored eight-demonstrator family achieves matched commanded accuracy 0.887 ± 0.013 versus 0.299 ± 0.000 for the single-demonstrator cell. The anchored family's identity-only benchmark is already 0.8843. High accuracy in that cohort is consequently weak evidence for discovering the physical channel, which motivates using channel TV as the primary outcome in the controlled pairs/cyclic experiment.

The deterministic loss study shows little regularizer effect on the anchored cell; it does not support an assertion that every term is empirically necessary. Alphabet-size misspecification also shows merged actions when too few branches are provided and starved components when too many are provided. At sixteen modeled branches for four true actions, seven components are nearly unused, and purity about 0.96. These are diagnostics under a known true alphabet, not a validated procedure for selecting k. The joint-learning negative is reported separately in Appendix F.

Table 25: **Exploratory deterministic loss ablation.** Matched accuracy, mean and standard deviation over three training seeds on one generated dataset. These use a preliminary protocol and are separate from the frozen-protocol ablation in Table 18.

Objective	Anchored, $m = 8$	Single demonstrator
Fit only	$.832 \pm .230$	$.730 \pm .088$
Fit + volume	$.833 \pm .232$	$.685 \pm .129$
Fit + policy regularizer	$.833 \pm .232$	$.633 \pm .113$
Full objective	$.833 \pm .232$	$.683 \pm .132$

H.2 Code entry points

All numbers above use the cohort definitions in Table 9. The table below identifies the implementation entry points needed to reconstruct the principal analyses when their referenced data and model artifacts are available.

Table 26: **Analysis entry points.** Paths are relative to `scripts/` in the supplementary code unless otherwise stated. Reproduction also requires the run outputs, datasets, and pretrained checkpoints read by these scripts.

Evidence	Entry point
Fixed neural protocol	`run_confirmatory.py`
Structural alignment	`../synthetic/alignment.py`
Initial matrix and uncertainty	`analyze_t6.py`
Finite samples; noise grid	`c3_finite_sample_regen.py`; `t6_tabular_sweep.py`
Identity-blind; identity-aware baselines	`t3_baselines.py`; `g_idmix_baseline.py`
Fresh asymmetric analyses	`analyze_f.py`; `analyze_g.py`
Pool diagnosis; supervised probe	`g_pool_diag.py`; `g_sup_probe.py`
Composed-kernel check	`g_joint_kernel.py`
Grounding and independent test	`t1_test_eval.py`; `t1_budget_means.py`; `f_test_eval.py`; `g_test_eval.py`
Aligned planning	`plan_t5_audit.py`

Reproduction should preserve the distinction between raw experiment outputs and tables rounded for display. In particular, confidence intervals and combined means should be regenerated from the recorded per-run outputs, with the same alignment and dataset-level aggregation, before rounding.

I Extended related work

From video to an action interface. Latent action methods make unannotated trajectories available for learning behavior. LAPO learns an action space together with policies and dynamics, then adapts to executable actions (Schmidt & Jiang, 2024). LAPA uses quantized latent actions as targets for vision-language-action pretraining before adaptation with robot data (Ye et al., 2025). Genie uses a learned action interface to control generated environments (Bruce et al., 2024). These systems establish the usefulness of actions inferred from observations. DivLAM asks which structure such an interface recovers when execution is stochastic: it estimates command preferences and an execution channel, with semantic names assigned afterward using a separate calibration budget.

Prediction and the content of latent actions. The information needed to explain video changes can exceed the information needed for control. Zhang et al. (2025) analyze this distinction in a linear model, connecting latent action learning to principal component analysis and characterizing the roles of behavior and exogenous noise. Lee et al. (2026) extend the analysis to exogenous state and explain how representations and auxiliary objectives can reduce its influence. Experiments with distractors further demonstrate benefits from action supervision (Nikulin et al., 2025). These findings address what enters a learned action representation. DivLAM examines a remaining ambiguity even when executed outcomes are known: different policies and channels can generate the same outcome

frequencies. Its absorption analysis concerns that decomposition and does not establish robustness to arbitrary visual distractors.

Identifying commands from policy variation. Minimum-volume recovery need not require pure observations: in hyperspectral unmixing, suitable geometric coverage can identify the underlying simplex even when every observed pixel is mixed (Lin et al., 2015). Schur (2026) use demonstrator identity to express observed transitions as mixtures of shared latent transition kernels. Sufficiently scattered policies and rank conditions identify the factors through minimum-volume estimation, drawing on Fu et al. (2018). DivLAM inherits this principle. It introduces a concrete estimator in an executed-transition basis, constructs policies with provable geometric compliance, and derives the alternative channel selected for specified violations. Exact frequencies, sampled counts, and visual trajectories then expose different sources of recovery error. This progression connects the population result to estimation without treating the neural pipeline as covered by the exact identification theorem.

Other sources of identifying information. Policy diversity is one route to identification. Lachapelle (2025) establish identification for an entropy-regularized latent action policy objective under assumptions including continuous deterministic dynamics with distinct action effects and suitable support. These restrictions differ from a stochastic command-to-execution channel. CRAFT also uses differences between agents, but targets controllable state representations in exogenous block Markov decision processes (Levine et al., 2025). Its treatment of temporally correlated nuisance features suggests a possible precursor to channel recovery. Combining the methods would require assumptions linking state and action identification; the present experiments do not establish that composition.

Continuous actions and heterogeneous videos. Continuous latent actions broaden the interface beyond a known finite alphabet. Garrido et al. (2026) study constrained continuous actions in diverse videos, including the difficulties of noise and heterogeneous embodiment. CLAW learns continuous actions with a world model and adversarial latent regularization (Ayalew et al., 2026), while Alles et al. (2025) combine observed controls and inferred actions in a shared space for offline control. These settings also clarify the limits of using identity solely as a policy label: if embodiment or physics changes with the demonstrator, a shared execution mechanism may no longer explain the data. DivLAM's controlled experiments do not evaluate recovery in that setting.

State representations and models with observed actions. Identifying an action interface also depends on the representation in which dynamics are modeled. Klindt et al. (2026) analyze latent state identification under a predictive, Gaussian-regularized objective. Zhang et al. (2026) study representation and controlled transition identification with observed actions, and ATM uses action information to diagnose latent world models (Chen, 2026). These questions concern the state or a supplied action interface. DivLAM instead estimates hidden commands within a fixed representation. Our encoder probes establish that supervised prediction can extract action information; they do not certify the representation conditions in Lemma A.2 or guarantee access to a consistent executed basis during discovery.

J Extended discussion, limitations, and implications

J.1 What the geometry comparison establishes

The ordered-pairs and cyclic families have the same number of demonstrators, per-policy entropy, aggregate command prior, identity-only command accuracy, and execution channel. Their recovery difference therefore cannot be attributed solely to those matched quantities. Their complete policies and induced state occupancies do differ. The visual comparison does not isolate geometric compliance while holding every feature of the trajectory distribution fixed. The exact-frequency experiment supplies the complementary evidence: after bypassing visual basis learning, the population factorization follows the predicted true or absorbed channel for the two constructions.

Sufficient scattering is a sufficient condition for the inherited minimum-volume result. Failure of that condition alone does not prove that every estimator must fail, or even that the same criterion fails for every noncompliant family. Our negative prediction follows from the specified policy mixing in

Proposition A.3. Likewise, identity removal measures the behavior of the implemented estimator on the studied data. It does not turn a population non-identifiability statement into a universal chance-accuracy bound.

J.2 A CHANNEL CAN BE IDENTIFIABLE WHILE COMMANDS REMAIN UNCERTAIN

Factor recovery and transition-level command classification measure different objects. Even with the correct channel and policies, an executed rightward move can have positive probability under multiple commands. A posterior over those commands is then intrinsically uncertain. This explains why the privileged identity-and-executed-branch benchmark in the symmetric construction is 0.896, below perfect accuracy, despite knowing the generating mechanism. Conversely, a command classifier can exploit demonstrator preferences without recovering the channel. Reporting channel error and command accuracy together helps distinguish these cases.

Structural alignment and semantic grounding also answer different questions. Independent row and column permutations compare the estimated channel with the true channel without penalizing arbitrary latent indices. Grounding uses a limited calibration set to attach command names and fixes that mapping before independent test evaluation. An accurate structural match does not guarantee a reliable mapping from a small random sample, and relabeling cannot repair incorrectly estimated factors.

J.3 POPULATION STRUCTURE DOES NOT REMOVE ESTIMATION ERROR

Stage B receives learned soft assignments from dependent trajectory samples. Its tolerance is based on a multinomial-inspired scale, which does not account for all representation, dependence, and modeling errors. A positive determinant ridge, a relaxed fit constraint, and local multistart optimization further distinguish the implemented problem from the exact population criterion. The exact-input results validate the reported numerical behavior on the tested instances. They supply neither a finite-sample recovery theorem nor a global-convergence guarantee.

The oracle-count controls show what the same factorization can achieve with accurate executed frequencies. Together with the neural-versus-mixture comparisons, they locate a substantial source of error in estimation of the transition basis and its frequencies. However, the identity-aware baseline is a restricted diagonal-Gaussian mixture of latent displacements. Its gap does not rule out more expressive non-neural estimators. The small Stage-A volume ablation also leaves the broader role of that regularizer unresolved; it does not remove the distinct Stage-B minimum-volume criterion.

The near-pure control further distinguishes statistical error from a misspecified recovery target. Its exact minimum-volume solution is 0.050 from the true channel, while the penalized solver reaches error 0.027 at 30,000 sampled transitions (Appendix D.3). A lower error to truth in this comparison does not imply better minimization of the population criterion. Incomplete optimization and finite-sample penalties can leave an estimate closer to the physical channel when the exact criterion selects a biased one. The recorded results do not isolate which mechanism accounts for the penalized solver's advantage.

J.4 REPRESENTATIONS AFFECT ACCESS TO A USEFUL SOLUTION

The second encoder retains information sufficient for accurate supervised executed-action prediction, yet useful solutions are rare in its unsupervised restart pools. This combination supports a limitation in the fitted estimator's access to a useful solution. It does not separate representation geometry from optimization: the geometry can influence which solutions the objective and initialization make accessible. The tested independent-initialization change has paired error difference $+0.016$ with interval $[-0.020, +0.060]$, leaving an improvement unestablished.

Freezing the encoder makes the estimation space common to all demonstrators. It does not establish Markov sufficiency, injectivity, or the continuity and rank conditions used in the representation argument. Moreover, the encoders were pretrained through an action-conditioned predictor. The absence of action labels from discovery and restart selection must be interpreted at that stage. All fourteen tested joint-training runs collapsed, so the results do not support an end-to-end discovery claim.

J.5 Generalization and downstream utility remain limited

The positive result is concentrated in controlled TwoRoom experiments. Both policy families fail to yield accurate recovery on PushT, where Stage A does not learn a useful basis. The absence of a geometry contrast there cannot test the population prediction conditional on an accurate basis. On TwoRoom, the combined asymmetric-channel advantage over the truth-informed isotropic reference remains unresolved at the reported uncertainty level. Dataset generation and training randomness have distinct roles: more seeds on one dataset characterize optimization without providing equally many independent data replications. The reported uncertainty therefore uses generated datasets as the relevant replication units.

The planning comparison evaluates every model in its cohort and aligns command and executed-head coordinates before removing the channel. The mean gain is modest and varies across the three generated datasets. It is concentrated among less accurate discovery models, while the diagnostic good-basin models perform similarly under the two planners. Consequently, these observations do not establish a monotone relationship between structural channel accuracy and planning success. Supplying the true channel also leaves an unresolved comparison: recovered-minus-true-channel success is $+0.015$, with dataset-bootstrap interval $[-0.062, +0.093]$. The planner propagates expected latent successors, which approximates multi-step stochastic prediction through nonlinear dynamics. Understanding when channel recovery improves decisions therefore requires separating channel error, successor-model error, and the planning approximation.

J.6 Implications for collecting and using demonstrations

Demonstrator labels preserve variation that pooling can obscure, but more identities need not add new policy directions. In our controlled construction, the generating policies are known and compliance can be proved. In an observational corpus, those policies are hidden. Finite directional tests can reveal a violation when applied to known or estimated policies, but passing those tests cannot certify the full sufficient condition. Thus the experiments motivate preserving identity metadata and studying policy coverage, without offering a general test that certifies an unlabeled corpus before training.

Further extensions must address state dependence and changing dynamics explicitly. With a constant channel, aggregating a state-dependent policy produces a visitation-weighted marginal, which need not be the policy prior at a particular state. With state-dependent channels, unequal state occupancies may prevent a single shared aggregate channel altogether. Changes of embodiment or physics introduce an additional source of demonstrator-dependent transitions. Extending DivLAM to these settings requires identifying and estimating that structure before interpreting recovered factors as a common command interface.