

Phaseformer: Minimum Requirements for Multimodal Behavior Cloning

Joseph C. Liechty¹, Tucker Hermans^{1,2}

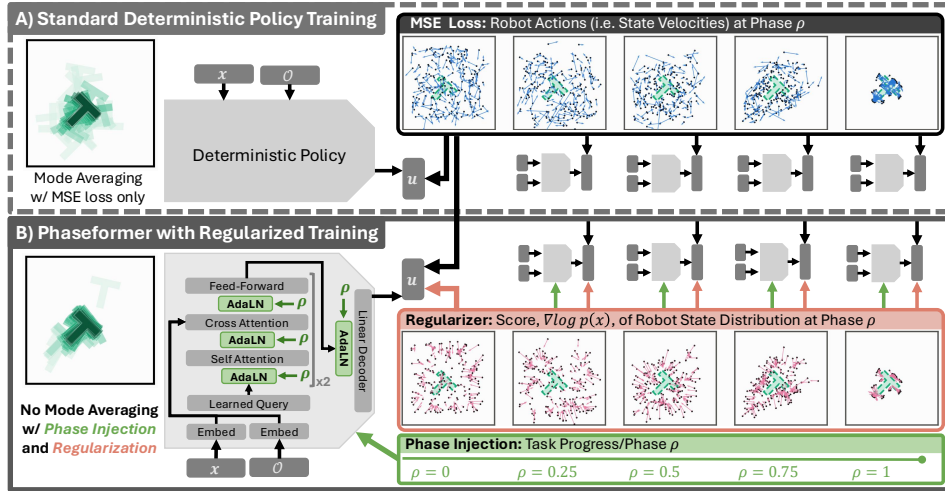


Figure 1: Phaseformer is a deterministic behavior cloning policy that injects task progress ρ through adaptive layer norms (AdaLN). A) Without phase, a deterministic policy trained with a velocity MSE loss averages modes. B) Phaseformer uses no sampling or iterative computation: the variation in the demonstrated starts, phase conditioning and a score-based penalty prevent mode averaging at decision points.

Abstract: Deterministic behavior cloning policies average the modes of multimodal demonstrations. The accepted remedies sample stochastic latent inputs during training, often combined with iterative computation. At inference however, many of these methods run deterministically. We ask what a deterministic policy minimally needs at training and inference. Reading flow matching as phase-conditioned behavior cloning on straight-line demonstrations, we show that a deterministic policy can send rollouts to different modes only through variation in its input before the split, that conditioning on task progress (phase) removes the temporal aliasing in its regression target, and that a score-based penalty pulls predictions toward the locally dominant mode where branches overlap. Phaseformer and a phase-conditioned MLP combine these with demonstrations whose start states vary, and use no latent variable, sampling or iterative computation. They are Pareto-optimal for ultra-low-latency CPU control, and in navigation, varied demonstration starts restore success where identical starts fail in every seed. See https://joeliechty.github.io/phaseformer_site/

1 Introduction

Behavior cloning in robotics has been shaped by the struggle against mode averaging, where policies average the actions of multimodal demonstrations resulting in collisions, stalls and poor perfor-

joe.liechty@utah.edu

¹Kahlert School of Computing & Robotics Center, University of Utah, Salt Lake City, UT 84112, USA.

²NVIDIA, Seattle, WA 98103

mance [1]. The converged upon solutions have evolved from fast, deterministic policies into heavily parameterized, iterative sampling methods that transform stochastic latent representations into robot actions [2, 3, 4].

Diffusion policy (DP) [2], a breakthrough result, uses stochastic sampling and iterative compute to achieve multimodal action generation. Recent works have attempted to simplify policies in order to discover what the true minimum requirements are for multimodal behavior cloning. Pan et al. [5] attribute the success of generative policies to supervised iterative computation paired with stochasticity, rather than to multimodality. Their minimal iterative policy (MIP) adds noise to a two-step regression only during training. Action chunking transformer (ACT) [3] trains a conditional variational autoencoder (CVAE) but decodes the prior mean of its latent at test time. Streaming flow policy (SFP) [6] trains on Gaussian tubes around each demonstrated action trajectory and sets their width to zero at test time, which its authors report costs no performance.

We answer the question, *what are the minimum requirements for robot policies to capture multimodal action flows?* by reading flow matching as phase-conditioned behavior cloning on straight-line demonstrations (Lemma 1) and asking which of its properties survive on real demonstrations (Sec. 3). The answer sets the minimum requirements: *a deterministic behavior cloning policy needs inputs that vary before each decision in a way the decision depends on, and the phase wherever states are revisited.* The first is necessary for any split (Lemma 2) and the second can be (Prop. 2). With phase, the MSE optimum reproduces every demonstrated marginal even where demonstrations cross (Prop. 1): averaging velocities does not average the distribution. A score-based regularizer is an optional sharpener that pulls predictions toward the locally dominant branch (Thm. 5). Unlike SFP, we take the variation from the demonstrated starts rather than synthesizing it, and our clock runs over the whole task rather than restarting with each chunk, so the policy closes the loop at every step. Table 1 shows other behavior cloning policies compared with our policy designed around our proposed minimum requirements for multimodal behavior cloning. Our policy, **Phaseformer**, is a lightweight transformer that injects phase through AdaLN which meets these requirements without iterative computation, making it a competitive, Pareto-optimal method for CPU-based tactile control.

Policy	Stochasticity in training	At test time	Clock	Feedback
DP [2]	noised action chunks	samples noise; K passes	denoising step	every T_a steps
MIP [5]	noised input to 2nd step	deterministic; 2 passes	step index	every chunk
ACT [3]	CVAE latent z	$z = 0$; 1 pass	none	every step (ensembling)
SFP [6]	tubes around demos (σ_0)	$\sigma_0 = 0$; 1 pass per action	time in chunk	every chunk
Phaseformer	demonstrated starts	deterministic; 1 pass	task progress ρ	every step

Table 1: Where each policy’s stochasticity enters. Only DP samples at test time; the others are deterministic maps of their inputs, so the input fixes the mode (Lemma 2(c)). Settings as published; App. G lists ours.

2 Related Work

Behavior cloning approximates the expert policy $\pi^*(\mathbf{u} \mid \mathbf{x})$ with a learned policy π_ϕ by supervising the conditional action distribution during offline training [7]. The weights minimize

$$\phi^* = \arg \min_{\phi} \mathbb{E}_{(\mathbf{x}, \mathbf{u}) \sim \mathcal{D}} [\ell(\pi_\phi(\mathbf{x}), \mathbf{u})] \quad (1)$$

over a dataset of state–action pairs $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{u}_i)\}_{i=1}^N$, with $\ell(\pi_\phi(\mathbf{x}), \mathbf{u}) = \|\pi_\phi(\mathbf{x}) - \mathbf{u}\|^2$ for continuous actions. With the squared loss a deterministic π_ϕ learns only the conditional mean $\mathbb{E}[\mathbf{u} \mid \mathbf{x}]$. Often $\mathbf{u} = \dot{\mathbf{x}}$ [8].

Flow matching transports samples between distributions p_0 and p_1 along a learned phase-dependent velocity field $d\mathbf{x}/d\rho = v_\psi(\mathbf{x}, \rho)$ [9]. Conditional flow matching (CFM) regresses v_ψ onto the velocities of straight paths between paired samples, $\mathbf{x}_\rho = \rho\mathbf{x}_1 + (1 - \rho)\mathbf{x}_0$ with $\rho \sim \mathcal{U}[0, 1]$,

whose velocity is $\mathbf{x}_1 - \mathbf{x}_0$. With independently drawn endpoints these paths cross, so the learned average flow is curved and needs many integration steps [10, 11]. Pairing the endpoints by optimal transport (OT), in practice on each minibatch, removes most crossings [10]. With prior $p_0 = \mathcal{N}(0, I)$, data distribution q , and γ^* their W_2 -optimal coupling, the optimal transport conditional flow matching loss is

$$\mathcal{L}_{\text{CFM}}^{\text{OT}}(\psi) = \mathbb{E}_{\rho \sim \mathcal{U}[0,1], (\mathbf{x}_0, \mathbf{x}_1) \sim \gamma^*} \|v_\psi(\mathbf{x}_\rho, \rho) - (\mathbf{x}_1 - \mathbf{x}_0)\|^2. \quad (2)$$

Inference integrates the ODE, typically by forward Euler, $\mathbf{x}_{\rho_{i+1}} = \mathbf{x}_{\rho_i} + \Delta\rho v_\psi(\mathbf{x}_{\rho_i}, \rho_i)$ [12]. Figure 1 shows why the two frameworks look alike: the demonstrated states at each phase form a distribution that evolves from a start distribution to a multimodal goal distribution, and the same picture could show the intermediate marginals of a flow.

3 Flow Matching Principles that Apply to Behavior Cloning Generally

Our method is built on similarities between behavior cloning and flow matching. Consider Fig. 2, which shows a temporally evolving distribution of robot states $\mathbf{x} = [x \ y \ \theta]$ on the SE(2) manifold for a navigation scenario (B, Sec. 5). This distribution splits from a single start distribution into two paths, arriving at two distinct goal distributions. Visually, this evolution is ambiguous: it is impossible to distinguish whether this represents a series of intermediate distributions in a denoising flow, or a time-evolving distribution of robot states from demonstrated trajectories. In both frameworks, the states at any given phase/time form a distribution where the most optimal, or most likely, states possess the highest density.

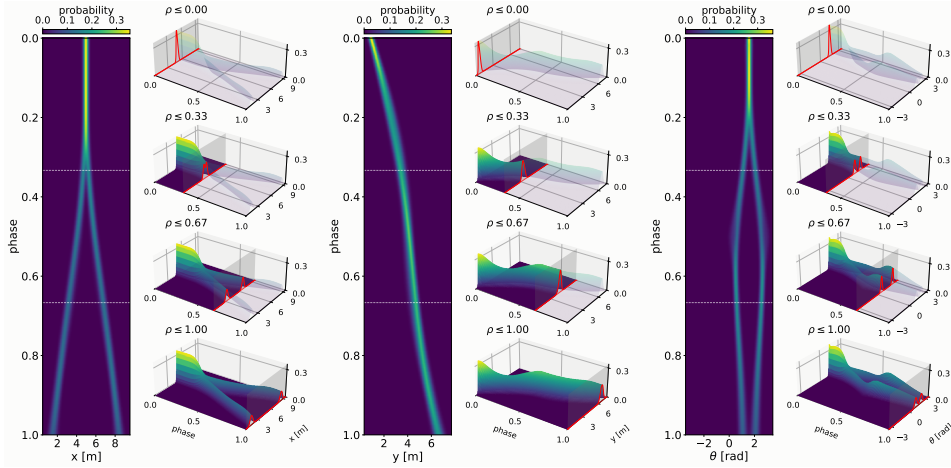


Figure 2: Evolution of the Marginal Robot State Distributions (x , y , and θ) over phase for the single-start, dual-goal, dual-path demonstrations (scenario B). Snapshots at $\rho = 0.0, 0.33, 0.67, 1.0$ show the evolution between the endpoint distributions.

Despite this structural similarity, standard approaches to behavior cloning and flow matching diverge in execution. Flow matching conditions a deterministic network on an explicit phase variable and regresses onto the average velocity at each state and phase. Velocity-based behavior cloned policies typically attempt the same task without phase. On demonstrations that visit a state at different phases or whose branches cross, they average velocities across phases and modes, hence modern policies' dependence on stochastic latent representations or iterative compute [2, 3]. Flow matching in this context is used as a multistep generative policy that denoises sampled robot actions, rather than being applied directly to predicting the flow of robot states in the demonstrated trajectories. Recognizing that flow matching is phase-conditioned behavior cloning on straight-line demonstrations provides a direct path to improve standard imitation learning.

We identify the properties that make it possible for a deterministic flow matching network to carry a unimodal start density to a multimodal target and show that each survives when the synthetic straight

paths of flow matching are replaced by demonstrations. The two methods fit the same regression target: the average velocity of a family of random paths at each state and phase. Flow matching chooses the paths, as straight lines from a chosen start density under a chosen coupling. Phase-conditioned behavior cloning receives them from the expert: curved, obstacle-avoiding or driven through contact, and paired by the expert’s own choice of goal (§A.1, Lemma 1). Properties of flow matching that use only the fact that its training paths are random paths from a start density, and not that they are straight, therefore hold for demonstrations as well. Three of those properties become the ingredients of our method.

1. **Phase** (§A.2). Flow matching conditions on the interpolation time. With phase as an input, the MSE optimum transports every demonstrated marginal, for curved and crossing demonstrations alike (Proposition 1). Without it, the optimum matches only the net flux and can overshoot or stall (Proposition 2).
2. **Input variation** (§A.3). Flow matching starts from a density, not a point. A deterministic rollout can send inputs to different modes only if they differ before the split (Lemma 2). A smooth policy that splits its starts between two modes leaves a band of undecided starts at least $g e^{-\int L}$ wide, so spreading the demonstrated starts lowers the mass in that band (Proposition 3).
3. **Label conflict** (§A.4). Optimal transport conditional flow matching chooses a coupling whose paths do not cross, so no state receives conflicting velocity labels. Behavior cloning cannot re-pair the expert’s demonstrations. Minimum-effort demonstrations in free space are the ideal case, in which behavior cloning coincides with optimal transport conditional flow matching and the conflict vanishes (Theorem 4). Where conflict remains, the score penalty moves the prediction toward the locally heavier branch (Theorem 5).

Our tasks are not the ideal case of Theorem 4, and no other result assumes it. Table 2 lists what each result assumes and which of our tasks satisfy it. All results concern population optima over all functions of the policy input, or rollouts of a given Lipschitz policy, not trained networks.

4 Methods

We predict actions that are robot-state derivatives, $\mathbf{u} = \dot{\mathbf{x}}$, as is typical in robot control [13]. A diffusion or flow-matching policy carries $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to the action distribution over a synthetic time and samples its start at test time [2, 4]. We instead let the demonstrations be the flow: the field lives in robot-state space, its time is task progress $\rho \in [0, 1]$, and its start distribution P_S is that of the demonstrated starts. Lemma 1 (App. A.1) justifies this: flow matching is phase-conditioned behavior cloning on straight-line demonstrations, so the properties of flow matching that do not rely on straight paths carry over to demonstrations. From M demonstrations we build a phase-labeled dataset $\mathcal{D}_\rho = \{(\rho_j, \mathbf{x}_{\rho_j}, \mathcal{O}_{\rho_j}, d\mathbf{x}_{\rho_j}/d\rho)\}_{j=1}^M$ with $\rho_j \sim \mathcal{U}[0, 1]$, which need not be optimal transport-paired, and train a policy, $\pi_\phi(\mathbf{x}, \rho)$, with objective

$$\phi^* = \arg \min_{\phi} \mathbb{E}_{(\rho, \mathbf{x}, \mathcal{O}, \mathbf{u}) \sim \mathcal{D}_\rho} \|\pi_\phi(\mathbf{x}, \mathcal{O}, \rho) - \mathbf{u}\|^2 + \lambda \mathcal{L}_{reg}(\phi). \quad (3)$$

Phase-Conditioned Deterministic Policy

We define a *phase-conditioned deterministic policy* as, $\mathbf{u} = \pi_\phi(\mathbf{x}, \mathcal{O}, \rho)$ where $\mathbf{u}$, $\mathbf{x}$, $\mathcal{O}$, and ρ are robot actions (state derivatives) and states, observations, and task progress respectively. We make ρ a deterministic, monotonically increasing function (like the phase in flow matching) either as a linear clock with step $\Delta\rho$ or a monotone MLP task-progress estimator on the observation $\mathcal{O}$ with minimum increment ε , both clamped at 1: $\rho_{new} = \begin{cases} \rho + \Delta\rho & \text{(Linear)} \\ \max(\text{MLP}_{TP}(\mathcal{O}), \rho + \varepsilon) & \text{(MLP)} \end{cases}$

We present two architectural implementations of our policy, 1) an MLP that takes $\mathbf{x}$, $\mathcal{O}$, and ρ as inputs and returns an action, $\mathbf{u}$, and 2) *Phaseformer*, a transformer backbone that takes $\mathbf{x}$ and $\mathcal{O}$ and returns $\mathbf{u}$. Phaseformer injects ρ into each block via Adaptive Layer Norms (AdaLN) [14], essentially learning a different policy for each step of the task (Fig. 1).

Result	What it says	Assumes	Nav.	Push-T
<i>Phase</i>				
Eqs. 9–11	phase removes the aliasing A from the loss	squared loss only	✓	✓
Proposition 1	with phase, rollouts of the optimum reproduce every marginal	$\dot{\mathbf{x}} = B(\mathbf{x})\mathbf{u}$; fixed observation; Lipschitz $\bar{v}$	✓	×
Proposition 2	without phase, rollouts can overshoot or stall	examples in $\mathbb{R}$	–	–
<i>Input Variation</i>				
Lemma 2(a)	at a shared input, the target is the branch average	squared loss only	✓	✓
Lemma 2(c)	one input follows one branch	deterministic policy and environment	✓	✓
Eq. 13, Lemma 2(b), Proposition 3	a smooth policy leaves a shell of undecided starts	Lipschitz in the robot state; same observations	✓	×
<i>Label Conflict</i>				
Theorem 4	minimum-effort demonstrations have no label conflict	(A1)–(A3)	×	×
Theorem 5	the score penalty moves the target toward the median	squared loss plus score penalty	✓	✓
<i>Link to flow matching</i>				
Lemma 1	flow matching is phase-conditioned behavior cloning	straight demonstrations	–	–

Table 2: What each result says and assumes, and whether our navigation and Push-T tasks satisfy its assumptions. Every property has a result that covers both tasks. The results that compare rollouts do not cover Push-T, whose policy observes a block that the pusher moves through contact.

Score-Based Regularization

We add a regularizer which by Theorem 5 moves predictions from the average demonstrated velocity toward the median one along the score wherever demonstrations overlap:

$$\mathcal{L}_{reg}(\phi) = \mathbb{E}_{\rho, (\mathbf{x}_\rho, \mathbf{u}_\rho) \sim \mathcal{D}} \left| (\mathbf{u}_\rho - \pi_\phi(\mathbf{x}_\rho, \rho)) \cdot \nabla_{\mathbf{x}} \log \hat{p}_\rho(\mathbf{x}_\rho) \right| \quad (4)$$

The score is the closed-form gradient of a Gaussian KDE of the demonstrated robot states at each phase [15, 16] (App. C). Phase aligns the demonstrations over the whole task, so all of them contribute at every phase. SFP’s stabilizing gain acts as a related score term on synthetic tubes.

5 Experiments and Results

We evaluate on two SE(2) tasks, *non-holonomic navigation* and *in-plane manipulation* (Push-T), against DP [2], ACT [3] and MIP [5] (Table 1), with ten seeds of each baseline trained as reported in its original work and ten of each of our variants with and without phase and score (App. G). For *Push-T* we report score, final position and orientation error, and success rate (final coverage above 0.95). We also benchmark CPU and GPU latency (Fig. 7).

Stochastic demonstration starts with phase injection and score regularization are sufficient to resolve mode averaging. In navigation rollouts for scenario C (Fig. 3), the deterministic no phase, no score policy hits the block in 29% of time. Adding phase splits rollouts left (45%) or right (48%), cutting collisions to 7%, and score cuts them to 3%, near DP’s 1%. In Push-T rollouts, the phase variants of our deterministic policies have task scores that are consistently better than their respective non-phase variants (Fig. 4). Phase injection and score regularization also reduce orientation errors in the Push-T rollouts (Figs. 4 and 5). Only open-loop DP has significantly lower orientation error than Phaseformer+score, which is not significantly different from closed-loop DP and is significantly better than ACT, MIP, and all of the other deterministic variants.

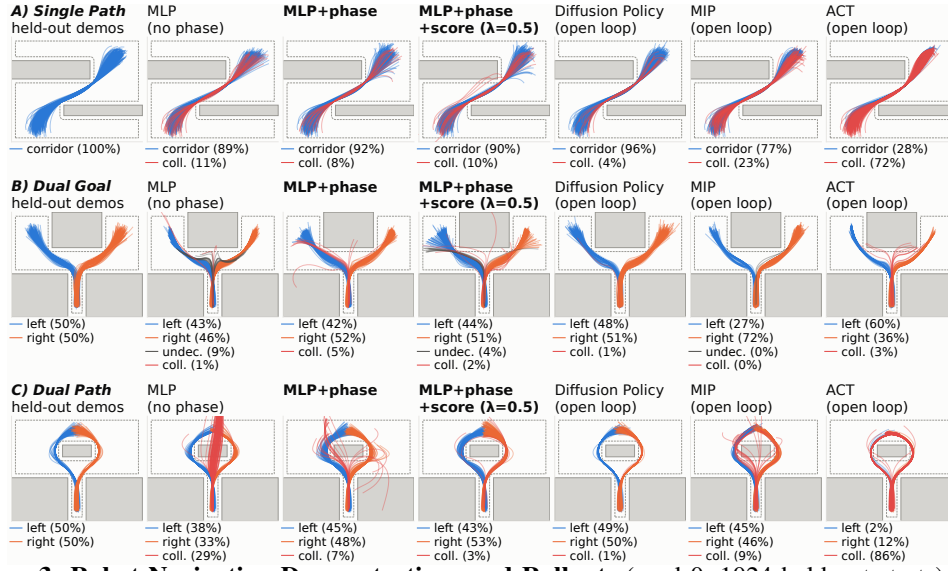


Figure 3: Robot Navigation Demonstrations and Rollouts (seed 0, 1024 held-out starts) of all policies.

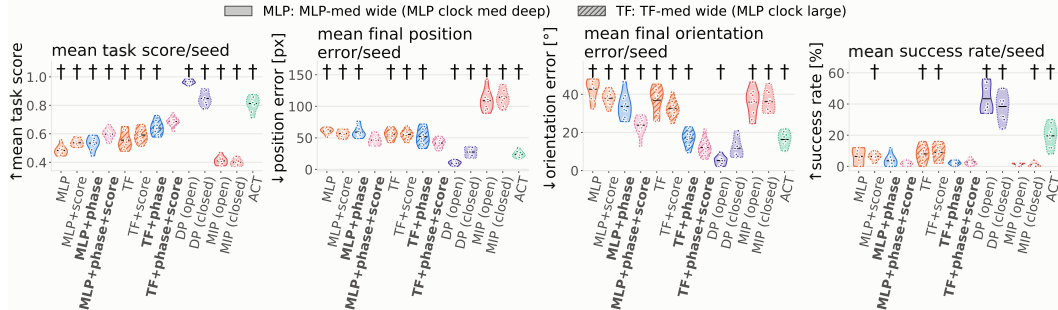


Figure 4: Push-T Episode Performance and Final Object-Pose Accuracy: mean score and final position error, orientation error, and success rate. Dots: 10 seeds; bar: mean. ACT runs closed loop. †: $p < 0.05$ vs. TF+phase+score (Welch over seeds).

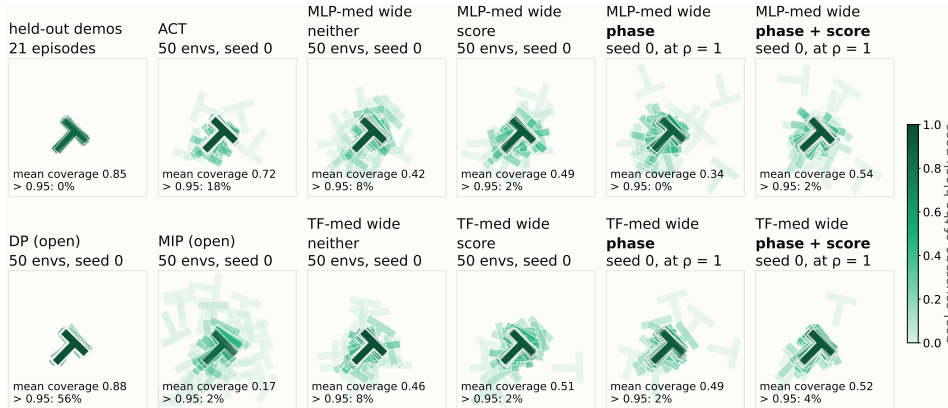


Figure 5: Final Block Poses for Push-T rollouts across 50 environments. Rollouts stop at $\rho = 1$ for phase variants or 300 steps for all other variants.

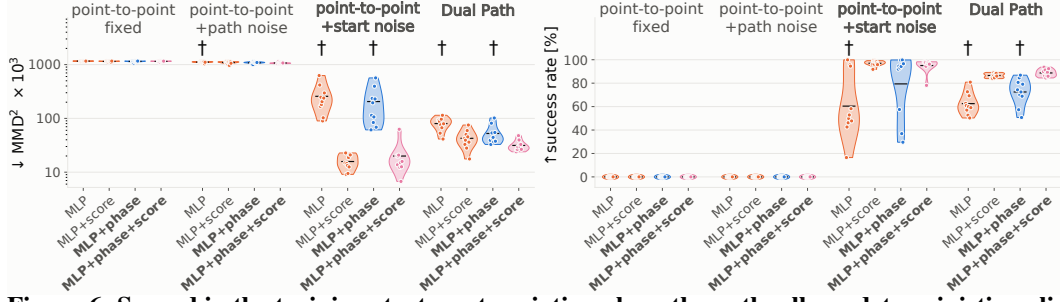


Figure 6: Spread in the training starts, not variation along the path, allows deterministic policies to split between the two routes. Four variations of our MLP policy are trained on demonstrations from four datasets that differ in one factor (named over each block). Dots: 10 seeds; Bar: mean. †: $p < 0.05$ vs. MLP+phase+score (Welch over seeds).

Modes collapse under fully deterministic training data. We rebuild the dual-path scenario (C) as deterministic point-to-point planning. The start pose $(5.0, 0.9, \pi/2)$ and the goal pose $(5.0, 7.0, \pi/2)$ are fixed, and the planner’s path waypoints are not noised. The planner then returns exactly one trajectory per homotopy class: the $n = 1024$ demonstrations are 512 copies of the left path and 512 copies of the right path. We compare this set against three others: 1) *+path noise* keeps the single start but restores C’s waypoint noising ($\sigma = 0.3$ m), so the demonstrations spread along the middle of the path but share both endpoints; 2) *+start noise* plans every demonstration from C’s start distribution ($\sigma = 8$ cm, 0.05 rad) to the same goal pose without waypoint noise; 3) *dual path*, the original scenario, combines that start distribution with waypoint point noise. Each set trains the four MLP variants (MLP, MLP+score, MLP+phase and MLP+phase+score) over 10 seeds. We evaluate every policy two ways: from the single nominal start, where a deterministic policy drives one path per seed, and from C’s 1024 held-out starts (Fig. 6). A rollout succeeds if it ends within 0.6 m of the goal without a collision. This replaces the $3\sigma_g$ criterion of App. G, which gives zero tolerance for a fixed goal pose; all numbers in this section, including those for the original C, use the 0.6 m criterion.

With a shared start state, the MSE target at the fork is the branch average, so a deterministic policy follows one path. The score penalty vanishes where demonstrations coincide and cannot change this (Lemma 2(a),(c); Theorem 5(iii)). With a fixed start and 512 copies each of a left and right path, no MLP variant reaches the goal in any seed (Fig. 6). This holds even with waypoint noise after the fork ($\rho \approx 0.15$; Lemma 2(c)). Varying the start of every demonstration (*+start*) restores 60–97% success across the four MLP variants (Fig. 6).

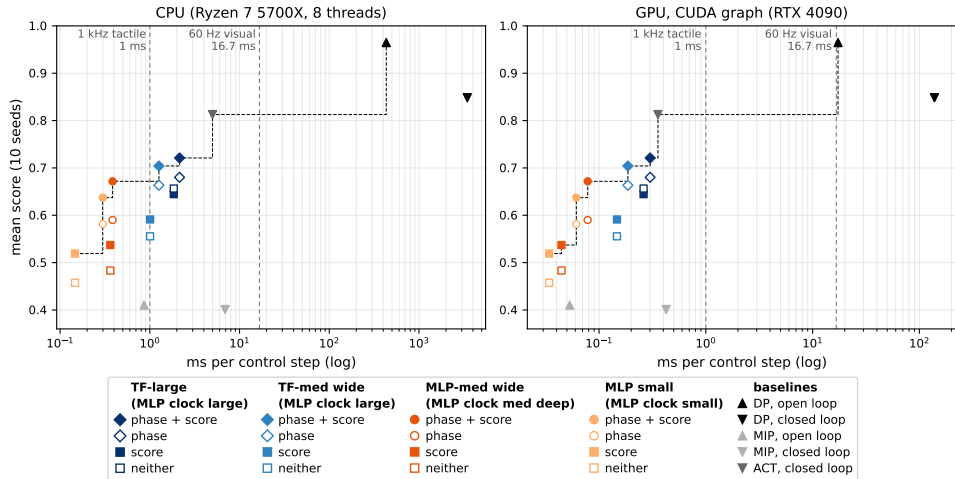


Figure 7: Task Success Versus Inference Latency Pareto Frontier for Push-T. Tactile [17] and visual [18] feedback control rates demarcated.

Score-regularized deterministic models with phase are Pareto optimal on both CPU and GPU (Fig. 7). Our deterministic Push-T policies are capable of operating closed-loop on a CPU at approximately the control rate needed for tactile control [17, 19] and are Pareto optimal. All of the Pareto-optimal policies that do not require iterative compute run closed-loop under the suggested control rate for visual control [18] on CPU.

6 Discussion and Conclusion

ACT, SFP and MIP add stochasticity only in training (Table 1), so at test time each follows the mode its input selects, as Phaseformer does; their encoders, tubes and second passes are ways to train and refine the state-action mapping, not to sample from it. Our results qualify Pan et al. [5]: on tasks that require multimodal action trajectories, neither iterative computation nor sampling at inference is needed, and the training stochasticity need not be synthetic, because varied demonstration starts supply it. A latent-variable policy must learn, at every input, a map from its latent onto all the modes demonstrated there. For example, at training ACT [3] uses a CVAE to learn a mapping from a stochastic latent representation but then at inference decodes the prior mean of that latent, so the stochasticity enters only during training and then is not used later. Instead, we add structure and simplify the learning objective by replacing the learned latent variable with measured task progress. Phaseformer obtains the same property as ACT without training a CVAE; it has no encoder, latent variable, or KL term, and the variation needed to assign starts to modes comes directly from the demonstrated start states and from temporal context. Sampling from a stochastic latent representation at inference, as in DP, remains necessary when the mode depends on nothing the policy observes, as in the single-start experiments (Lemma 2).

A second, largely orthogonal thread concerns whether a policy represents task progress at all. Dynamic Movement Primitives (DMPs) made an explicit phase variable, a canonical system that decays monotonically over one execution, the organizing structure of a long line of work in motor control [20, 21]. Their probabilistic relatives do not separate competing demonstrations on their own: a single ProMP is a unimodal Gaussian over basis-function weights [22], and GMR from a task-parameterized GMM returns a responsibility-weighted conditional mean [23], so multimodality was handled by fitting several primitives and selecting among them [e.g., 24], the same mechanism as mixture density networks [25] and CVAEs [26]. Because these models were fit over low-dimensional state and phase had to be aligned across demonstrations of different speed, the lineage scaled poorly to high-dimensional sensory input, and the shift to reactive deep policies dropped phase indexing in part for this reason, reintroducing the temporal aliasing A (Eq. 11) behind the phase-free failures of Proposition 2. Diffusion and flow-matching policies restore a phase-like variable, but it indexes denoising rather than task progress (Table 1; Sec. 3). Phaseformer is closer to the DMP lineage, but in Push-T its phase comes from a learned monotone estimator (Sec. 4; App. F), and it modulates a shared backbone via AdaLN rather than weighting a fixed kernel bank. Nor does Phaseformer select an explicit primitive: modes are resolved implicitly by phase-and-state conditioning (Proposition 1; Lemma 2(c)), with the score regularizer moving the policy toward the median demonstrated velocity where branches overlap (Theorem 5). Whether this implicit route is preferable to an explicit categorical one, rather than a less inspectable reparameterization of it, is a question these experiments do not settle.

Limitations. Phaseformer+score’s low success rate on Push-T compared to ACT and DP stems mainly from final position error since success requires 95% goal coverage, which a correctly oriented but incorrectly translated block misses. Our tasks are short-horizon and in SE(2), our policies need varied demonstration starts. Scaling to SE(3) will need longer horizons and better task-progress estimators. *Within this scope, the minimum requirements for multimodal behavior cloning are inputs that vary before each decision and a clock*, with a score penalty as an optional sharpener; deterministic single-pass policies that meet them are a viable, low-latency alternative to multi-step diffusion and flow-matching policies.

AI Use and Reproducibility

AI Use Disclosure Statement:

In this work, we used generative AI tools to help: generate synthetic data sets, design and provide feedback on research methodology or experiments, implement methods, clean and reformat datasets, support qualitative and thematic data analysis, provide critical ingredients for proving mathematical claims, create and modify scientific figures, sourcing/searching for information, refine proofs, and edit a research paper to improve readability, identify relevant literature, and format references. We did not use generative AI tools to: propose or refine hypotheses, help develop theoretical models or conceptual frameworks, formulate mathematical claims, or interpret results. Assisting with translation was not applicable to this work. We have reviewed all AI-assisted work to ensure that authors received credit where due, that our claims were mathematically sound, and our results were accurate. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility Statement:

All code is included in the supplemental materials and is available at the project page.

References

- [1] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022.
- [2] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, Oct. 2024. ISSN 0278-3649. doi: [10.1177/02783649241273668](https://doi.org/10.1177/02783649241273668). Publisher: SAGE Publications Ltd STM.
- [3] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, 2023. doi:[10.15607/rss.2023.xix.016](https://doi.org/10.15607/rss.2023.xix.016).
- [4] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. Pi\$0\$: A Vision-Language-Action Flow Model for General Robot Control, Nov. 2024. arXiv:2410.24164 [cs].
- [5] C. Pan, G. Anantharaman, N.-C. Huang, C. Jin, D. Pfrommer, C. Yuan, F. Permenter, G. Qu, N. Boffi, G. Shi, and M. Simchowitz. Much ado about noising: Dispelling the myths of generative robotic control. *arXiv preprint arXiv:2512.01809*, 2025.
- [6] S. Jiang, X. Fang, N. Roy, T. Lozano-Pérez, L. Pack Kaelbling, and S. Ancha. Streaming Flow Policy: Simplifying diffusion/flow-matching policies by treating action trajectories as flow trajectories. *arXiv e-prints*, art. arXiv:2505.21851, May 2025. doi:[10.48550/arXiv.2505.21851](https://doi.org/10.48550/arXiv.2505.21851).
- [7] D. A. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988.
- [8] J. Urain, A. Mandlekar, Y. Du, M. Shafiullah, D. Xu, K. Fragkiadaki, G. Chalvatzaki, and J. Peters. Deep generative models in robotics: A survey on learning from multimodal demonstrations. *CoRR*, abs/2408.04380, 2024.
- [9] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

- [10] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, pages 1–34, 2024.
- [11] A.-A. Pooladian, H. Ben-Hamu, Y. Lipman, and A. Tong. Multisample flow matching: Straightening flows with minibatch optimal transport. In *International Conference on Machine Learning*, pages 28131–28156, 2023.
- [12] Y. Lipman, M. Havasi, P. Holderrieth, N. Shaul, M. Le, B. Karrer, R. T. Q. Chen, D. Lopez-Paz, H. Ben-Hamu, and I. Gat. Flow matching guide and code, 2024. arXiv:2412.06264.
- [13] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, Feb. 1987. ISSN 2374-8710. doi:10.1109/JRA.1987.1087068.
- [14] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4199–4209, 2023.
- [15] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.
- [16] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [17] L. Massari, C. M. Oddo, E. Sinibaldi, et al. Tactile sensing and control of robotic manipulator integrating fiber bragg grating strain-sensor. *Frontiers in Neurorobotics*, 13:8, 2019. doi:10.3389/fnbot.2019.00008.
- [18] A. Keipour, G. A. S. Pereira, R. Bonatti, et al. Visual servoing approach to autonomous uav landing on a moving vehicle. *Sensors*, 22(17):6549, 2022. doi:10.3390/s22176549.
- [19] J. Tidare, E. Åstrand, and M. Ekström. Evaluation of closed-loop feedback system delay. In *Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies*, pages 187–193. SciTePress, 2018. doi:10.5220/0006598301870193.
- [20] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems*, volume 15, 2002.
- [21] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.
- [22] A. Paraschos, C. Daniel, J. Peters, and G. Neumann. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems*, volume 26, 2013.
- [23] S. Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics*, 9(1):1–29, 2016.
- [24] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters. Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks. *Autonomous Robots*, 41(3):593–612, 2017. doi:10.1007/s10514-016-9556-2.
- [25] C. M. Bishop. Mixture density networks, 1994. Num Pages: 438543 Place: Birmingham Publisher: Aston University.
- [26] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, volume 28, 2015.
- [27] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

- [28] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.
- [29] M. S. Albergo and E. Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.
- [30] M. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.
- [31] I. Gyöngy. Mimicking the one-dimensional marginal distributions of processes having an Itô differential. *Probability Theory and Related Fields*, 71(4):501–516, 1986.
- [32] S. Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review, 2018. arXiv:1805.00909.
- [33] T. Mikami. Monge’s problem with a quadratic cost by the zero-noise limit of h-path processes. *Probability Theory and Related Fields*, 129(2):245–260, 2004.
- [34] C. Léonard. From the Schrödinger problem to the Monge–Kantorovich problem. *Journal of Functional Analysis*, 262(4):1879–1920, 2012.
- [35] J.-D. Benamou and Y. Brenier. A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.
- [36] R. J. McCann. A convexity principle for interacting gases. *Advances in Mathematics*, 128(1):153–179, 1997.
- [37] C. Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2009.
- [38] Y. Huang, T. Transue, S.-H. Wang, W. M. Feldman, H. Zhang, and B. Wang. Improving flow matching by aligning flow divergence. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 25813–25834. PMLR, 13–19 Jul 2025.
- [39] R. J. McCann. Polar factorization of maps on Riemannian manifolds. *Geometric and Functional Analysis*, 11(3):589–608, 2001.
- [40] R. T. Q. Chen and Y. Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, 2024.
- [41] L. Ambrosio, N. Gigli, and G. Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics ETH Zürich. Birkhäuser, second edition, 2008.
- [42] Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics*, 44(4):375–417, 1991.
- [43] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1998.
- [44] J. E. Chacón, T. Duong, and M. P. Wand. Asymptotics for general multivariate kernel density derivative estimators. *Statistica Sinica*, 21(2):807–840, 2011.
- [45] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Feb. 2000. ISBN 978-1-139-64346-7. Google-Books-ID: aXQgAwAAQBAJ.
- [46] A. Savitzky and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8):1627–1639, July 1964. ISSN 0003-2700, 1520-6882. doi:10.1021/ac60214a047.

- [47] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models, Oct. 2018. arXiv:1810.01367 [cs].
- [48] M. Hutchinson. A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines. *Communications in Statistics - Simulation and Computation*, 18(3):1059–1076, Jan. 1989. ISSN 0361-0918. doi:[10.1080/03610918908812806](https://doi.org/10.1080/03610918908812806). Publisher: Taylor & Francis .eprint: <https://doi.org/10.1080/03610918908812806>.
- [49] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962. Publisher: JSTOR.
- [50] D. O. Loftsgaarden and C. P. Quesenberry. A Nonparametric Estimate of a Multivariate Density Function. *The Annals of Mathematical Statistics*, 36(3):1049–1051, June 1965. ISSN 0003-4851, 2168-8990. doi:[10.1214/aoms/1177700079](https://doi.org/10.1214/aoms/1177700079). Publisher: Institute of Mathematical Statistics.
- [51] G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011.
- [52] L. Pineda, T. Fan, M. Monge, S. Venkataraman, P. Sodhi, R. T. Chen, J. Ortiz, D. DeTone, A. Wang, S. Anderson, J. Dong, B. Amos, and M. Mukadam. Theseus: A Library for Differentiable Nonlinear Optimization. *Advances in Neural Information Processing Systems*, 2022.

Appendices

Appendix Table of Contents

A	Theory	13
A.1	Setting: Demonstrations as Random Paths	13
A.2	Lever 1: Phase	16
A.3	Lever 2: Input Variation	17
A.4	Lever 3: Label Conflict	18
A.5	Scope	20
B	Proofs	20
B.1	Proofs for the Setting and Lemma 1	20
B.2	Proof of Proposition 1, and a Closed-Form Example	21
B.3	Proof of Proposition 2	22
B.4	Proofs for Input Variation	23
B.5	Proof of Theorem 4	24
B.6	Proof of Theorem 5, and Properties of the Score Penalty	24
C	KDE Score Calculation	25
D	Planner Details	25
E	Divergence Estimation Challenges and Regularization Ablation	26
F	Policy, Task, and Normalization Details	29
G	Training Details	31
H	Additional Results	32

A Theory

A.1 Setting: Demonstrations as Random Paths

Demonstrations. We normalize time so that the phase $\rho \in [0, 1]$ indexes progress, and treat a demonstration as a random path of robot states $(\mathbf{x}_\rho)_{\rho \in [0, 1]}$ in $\mathbb{R}^d$ with $\mathbb{E} \int_0^1 \|\dot{\mathbf{x}}_\rho\|^2 d\rho < \infty$, start distribution P_S (the law of $\mathbf{x}_0$) and goal distribution P_G (the law of $\mathbf{x}_1$). Its marginal at phase ρ is p_ρ . Nothing is assumed about the shape of the paths or about how starts are paired with goals. The action is the robot-state velocity, $\mathbf{u} = \dot{\mathbf{x}}$, with one exception. The navigation robot is a unicycle, a wheeled robot that can drive forward and turn but cannot move sideways [27, Sec. 13.1.2]. Its state is the planar pose $\mathbf{x} = (x, y, \theta)$, its action is the body twist $\mathbf{u} = (v, \omega)$ of forward speed and turn

rate, and

$$\dot{x} = v \cos \theta, \quad \dot{y} = v \sin \theta, \quad \dot{\theta} = \omega, \quad \text{i.e.} \quad \dot{\mathbf{x}} = B(\theta) \mathbf{u}, \quad B(\theta) = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix}. \quad (5)$$

The action then determines the state velocity without being equal to it; §A.5 shows which results carry over.

Notation. $\mathbb{E}[\cdot]$ averages over demonstrations, and over $\rho \sim \mathcal{U}[0, 1]$ when the phase is not fixed. $\mathbb{E}[Y \mid Z = z]$, the conditional expectation, averages Y over the demonstrations whose value of Z is z . $\Pr(E)$ is the probability of an event E : the fraction of demonstrations, or of rollouts, for which E holds. $\mathcal{U}[a, b]$ and $\mathcal{N}(m, \sigma^2)$ are the uniform and Gaussian distributions. For a map $\mathcal{T}$ and a distribution P , $\mathcal{T}_\# P$ is the distribution of $\mathcal{T}(\mathbf{x})$ when $\mathbf{x} \sim P$ (the push-forward), so $P_G = \mathcal{T}_\# P_S$ says that applying $\mathcal{T}$ to the starts gives the distribution of the goals. For a set S , $\text{dist}(\mathbf{x}, S)$ is the distance from $\mathbf{x}$ to the nearest point of S .

Observations. The action is the derivative of the robot state and of nothing else. The policy may also condition on an observation $\mathcal{O}$ of the world, such as a goal or the pose of an object, but the observation is not part of the state: the policy does not predict its derivative, and the flow the policy defines moves only the robot. Like phase, the observation is a conditioning input, $\mathbf{u} = \pi_\phi(\mathbf{x}, \mathcal{O}, \rho)$, in the way a class label or context conditions a flow-matching model. We write $\mathcal{O}_\rho$ for the observation at phase ρ along a demonstration or rollout. In our tasks, the navigation policy has no observation input, and the Push-T policy observes the block pose and predicts the pusher’s velocity (§4). How the observation enters a result depends on what the result is about.

- *Results about the loss* hold for any observation, with $(\mathbf{x}_\rho, \mathcal{O}_\rho, \rho)$ as the policy input.
- *Results about rollouts* compare robot states along rollouts that see the same observations. When the observation is a context c fixed during an episode (a goal, a static map, an initial object pose), they all hold conditionally on c , for the demonstrations that share it. When the observation changes during an episode without depending on the robot’s motion, the Lipschitz results of §A.3 hold along each observation path, but Proposition 1 does not extend in general, because the policy sees the current observation and not its path. When the robot changes what it observes, as when the pusher moves the Push-T block, nearby rollouts see different observations, and comparing them would need a model of how the world responds to the robot; the rollout results do not cover this case (Table 2).

Unless stated otherwise, we work with a fixed context and drop it from the notation. Variation of c across demonstrations is then variation in the policy input, and plays the same role as variation of the start state in §A.3.

Regression targets and label conflict. With phase as an input, the squared-loss BC objective of Eq. 1 is

$$\mathcal{L}_{\text{BC}}(\phi) = \mathbb{E}_{\rho \sim \mathcal{U}[0, 1]} \mathbb{E} \|\pi_\phi(\mathbf{x}_\rho, \rho) - \dot{\mathbf{x}}_\rho\|^2. \quad (6)$$

What its minimizer is depends on what the policy sees. Let Z be a policy input, such as $(\mathbf{x}_\rho, \rho)$, and let $Z' = h(Z)$ be the input of a policy that sees less, such as $\mathbf{x}_\rho$ for a phase-free policy. Write $m(Z) = \mathbb{E}[\dot{\mathbf{x}}_\rho \mid Z]$ and $m'(Z') = \mathbb{E}[\dot{\mathbf{x}}_\rho \mid Z']$ for the average demonstrated velocity given each input. For every policy π' of Z' with $\mathbb{E}\|\pi'(Z')\|^2 < \infty$,

$$\mathcal{L}_{\text{BC}}(\pi') = \underbrace{\mathbb{E} \|\pi'(Z') - m'(Z')\|^2}_{\text{fit error}} + \underbrace{\mathbb{E} \|\dot{\mathbf{x}}_\rho - m(Z)\|^2}_{\text{irreducible given } Z} + \underbrace{\mathbb{E} \|m(Z) - m'(Z')\|^2}_{\text{cost of not seeing } Z} \quad (7)$$

(App. B.1). With $Z' = Z$ the last term is zero.

With $Z = (\mathbf{x}_\rho, \rho)$, m is $\bar{v}(\mathbf{x}, \rho) = \mathbb{E}[\dot{\mathbf{x}}_\rho \mid \mathbf{x}_\rho = \mathbf{x}]$, the average demonstrated velocity at a state and phase. A phase-free policy sees $Z' = \mathbf{x}_\rho$ with $\rho \sim \mathcal{U}[0, 1]$ random, and m' is $\bar{v}_{\text{free}}(\mathbf{x}) = \mathbb{E}[\dot{\mathbf{x}}_\rho \mid$

$\mathbf{x}_\rho = \mathbf{x}$]. When the p_ρ have densities,

$$\bar{v}_{\text{free}}(\mathbf{x}) = \frac{\int_0^1 p_\rho(\mathbf{x}) \bar{v}(\mathbf{x}, \rho) d\rho}{\int_0^1 p_\rho(\mathbf{x}) d\rho}, \quad (8)$$

so it mixes velocities from the different phases at which demonstrations visit $\mathbf{x}$. Eq. 7 then gives, for every phase-conditioned policy π and every phase-free policy π' with finite second moments,

$$\mathcal{L}_{\text{BC}}(\pi) = \mathbb{E} \|\pi(\mathbf{x}_\rho, \rho) - \bar{v}(\mathbf{x}_\rho, \rho)\|^2 + C, \quad (9)$$

$$\mathcal{L}_{\text{BC}}(\pi') = \mathbb{E} \|\pi'(\mathbf{x}_\rho) - \bar{v}_{\text{free}}(\mathbf{x}_\rho)\|^2 + C + A, \quad (10)$$

where

$$C = \mathbb{E} \|\dot{\mathbf{x}}_\rho - \bar{v}(\mathbf{x}_\rho, \rho)\|^2, \quad A = \mathbb{E} \|\bar{v}(\mathbf{x}_\rho, \rho) - \bar{v}_{\text{free}}(\mathbf{x}_\rho)\|^2 \quad (11)$$

The optima are therefore $\bar{v}$ and $\bar{v}_{\text{free}}$. We call C the *label conflict*. It is positive exactly when, with positive probability, demonstrations that pass through the same state at the same phase do so with different velocities; no phase-conditioned policy can then fit them all. We call A the *temporal aliasing* that a phase-free policy adds by pooling phases; conditioning on phase removes it. With an observation, take $Z = (\mathbf{x}_\rho, \mathcal{O}_\rho, \rho)$ for the phase-conditioned policy and $Z' = (\mathbf{x}_\rho, \mathcal{O}_\rho)$ for the phase-free one. The target becomes $\bar{v}(\mathbf{x}, \rho) = \mathbb{E}[\dot{\mathbf{x}}_\rho \mid \mathbf{x}_\rho = \mathbf{x}, \mathcal{O}_\rho = \rho]$, C and A are defined with these inputs, and Eqs. 9–11 follow from Eq. 7 in the same way. They therefore hold for both of our tasks.

Rollouts. A rollout of a policy π solves $d\mathbf{x}/d\rho = \pi(\mathbf{x}, \rho)$, or $d\mathbf{x}/d\rho = \pi(\mathbf{x}, \mathcal{O}_\rho, \rho)$ along an observation path, and $\Phi_{\rho_0 \rightarrow \rho_1}$ denotes the map from its robot state at ρ_0 to its robot state at ρ_1 . We call π $L(\rho)$ -*Lipschitz* if $\|\pi(\mathbf{x}, \rho) - \pi(\mathbf{y}, \rho)\| \leq L(\rho) \|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y}$ and ρ , at every fixed observation when the policy has one. Thus $L(\rho)$ bounds how sharply the commanded velocity can change between nearby robot states at phase ρ ; it is unrelated to the losses $\mathcal{L}$.

Flow matching is phase-conditioned BC on straight demonstrations. CFM with a coupling γ of P_S and P_G regresses onto the velocity of the straight path between paired endpoints,

$$\mathcal{L}_{\text{CFM}}^\gamma(\phi) = \mathbb{E}_{\rho \sim \mathcal{U}[0,1], (\mathbf{x}_0, \mathbf{x}_1) \sim \gamma} \|\pi_\phi((1-\rho)\mathbf{x}_0 + \rho\mathbf{x}_1, \rho) - (\mathbf{x}_1 - \mathbf{x}_0)\|^2, \quad (12)$$

and the W_2 -optimal coupling $\gamma = \gamma^*$ gives OT-CFM (Eq. 2, with P_S and P_G in place of $\mathcal{N}(0, I)$ and q).

Lemma 1 (Flow matching is phase-conditioned BC for any coupling). *Let γ be any coupling of P_S and P_G , which have finite second moments, and let the demonstrations be the straight paths $\mathbf{x}_\rho = (1-\rho)\mathbf{x}_0 + \rho\mathbf{x}_1$ with $(\mathbf{x}_0, \mathbf{x}_1) \sim \gamma$.*

- (a) $\mathcal{L}_{\text{BC}}(\phi) = \mathcal{L}_{\text{CFM}}^\gamma(\phi)$ for every policy with $\mathbb{E} \|\pi_\phi(\mathbf{x}_\rho, \rho)\|^2 < \infty$. By Eq. 9 both are minimized by $\bar{v}$, and their minimum is the label conflict C_γ of these demonstrations.
- (b) Flow-matching inference, which integrates $d\mathbf{x}/d\rho = \pi_\phi(\mathbf{x}, \rho)$ from $\mathbf{x}_0 \sim P_S$, is the closed-loop rollout of the phase-conditioned policy; whenever the solution is unique, the two coincide.
- (c) Standard CFM (independent coupling) and OT-CFM ($\gamma = \gamma^*$) are therefore both phase-conditioned BC. The converse fails: the single demonstration $\mathbf{x}_\rho = \sin(\pi\rho)$ is BC data that no straight-line dataset produces.

Lemma 1 is the inclusion $\text{FM} \subset \text{BC}$. The rest of this appendix goes the other way. It takes the properties that let FM’s deterministic network produce multimodal outputs and asks which of them hold for demonstrations that are not straight. This is the interpolant view of flow-based generative models: rectified flow [28] and stochastic interpolants [29, 30] train on general, not necessarily straight, interpolation paths between two distributions. We apply that view with the demonstrations as the interpolant.

A.2 Lever 1: Phase

Phase removes temporal aliasing, in any task. A policy that does not see phase must fit, at each input, the average of the velocities demonstrated there at every phase, $\bar{v}_{\text{free}}$ (Eq. 8). By Eqs. 9–11, conditioning on phase changes the regression target to $\bar{v}$ and lowers the smallest achievable loss by exactly the aliasing A . This uses only the squared loss, so it holds with an observation as well, including Push-T. The next two results concern rollouts. They say what the change of target does to the rollouts when the observation is fixed.

Proposition 1 (With phase, the MSE optimum transports every demonstrated marginal). *With no observation or a fixed one, let the demonstrations be any random paths as in §A.1, curved or crossing, with any pairing. Then p_ρ solves the continuity equation $\partial_\rho p_\rho + \nabla \cdot (p_\rho \bar{v}) = 0$. If $\bar{v}$ is $L(\rho)$ -Lipschitz with $\int_{\rho_0}^{\rho_1} L(\rho) d\rho < \infty$ and $\mathbb{E}\|\mathbf{x}_{\rho_0}\| < \infty$, rolling out $\bar{v}$ from $\mathbf{x}_{\rho_0} \sim p_{\rho_0}$ reproduces p_ρ for every $\rho \in [\rho_0, \rho_1]$. The phase-free optimum satisfies only the time-integrated identity $\nabla \cdot (\mu \bar{v}_{\text{free}}) = p_0 - p_1$, where $\mu = \int_0^1 p_\rho d\rho$.*

Proposition 1 is the ODE analogue of Gyöngy’s mimicking theorem [31], which reproduces the one-time marginals of an Itô process by a Markov process with conditionally averaged coefficients, and it is the marginal-preservation property of rectified flow and stochastic interpolants [28, 30]. What it means for BC is that averaging velocities does not average the distribution. Demonstrations move p_ρ only through $\bar{v}$, so the phase-conditioned optimum reroutes crossing demonstrations instead of collapsing them, even though $C > 0$ wherever they cross. Figure 8 shows a closed-form example with conflicting labels at every state (App. B.2). Without phase, $\bar{v}_{\text{free}}$ matches only the net flux $p_0 - p_1$, which can be far from the demonstrated motion.

Proposition 2 (Phase can be necessary). *There are demonstrations in $\mathbb{R}$, with P_S absolutely continuous, whose phase-conditioned optimum fits every demonstration ($C = 0$) and whose rollouts reproduce every p_ρ , while every phase-free policy has $\mathcal{L}_{\text{BC}} > 0$ and the rollouts of $\bar{v}_{\text{free}}$ fail in one of two ways.*

- (a) **Overshoot.** Let $P_S = \frac{1}{2} \mathcal{U}[-\delta, \delta] + \frac{1}{2} \mathcal{U}[1 - \delta, 1 + \delta]$ with $0 < \delta < \frac{1}{2}$, and let the demonstrations be straight, $\mathbf{x}_\rho = \mathbf{x}_0 + \rho(\mathcal{T}(\mathbf{x}_0) - \mathbf{x}_0)$, with $\mathcal{T}(a) = a + 2$ on the first block and $a + 4$ on the second. The rollouts of $\bar{v}_{\text{free}}$ overshoot the goal of every demonstration in the first group, undershoot the goal of every demonstration in the second, and do not reach P_G . These demonstrations satisfy (A1)–(A3) of Theorem 4, so phase is needed even in the ideal case.
- (b) **Stall.** Let $P_S = \mathcal{U}[-\delta, \delta]$ with $0 < \delta < \frac{1}{2}$, and let the demonstrations go out and back, $\mathbf{x}_\rho = \mathbf{x}_0 + \sin(\pi\rho)$. Then $\bar{v}_{\text{free}} = 0$ wherever the demonstrations go, so its rollouts never move and miss p_ρ for every $\rho \in (0, 1)$, even though they end with the right distribution $P_G = P_S$.

In (a), $\bar{v}_{\text{free}} = 8/3$ exactly on the interval $(1 + \delta, 2 - \delta)$ that both groups cross. As $\delta \rightarrow 0$, the rollouts from 0 and 1 end at 2.5 and 4.5 instead of 2 and 5, and the smallest phase-free loss tends to $1/3$ (App. B.3). In one dimension $\mu \bar{v}_{\text{free}} = F_S - F_G$, where F_S and F_G are the cumulative distribution functions of P_S and P_G , so the phase-free optimum depends on the demonstrations only through their endpoint distributions and the time they spend at each state.

In our experiments. In navigation scenario C, phase splits rollouts left (45%) or right (48%) and cuts collisions from 29% to 7% (Fig. 3). The aliasing result also covers Push-T, although we have not measured how large A is there. Push-T lies outside Proposition 1 because the pusher moves the block the policy observes (Table 2).

A.3 Lever 2: Input Variation

Smooth rollouts cannot separate supports exactly. If π is $L(\rho)$ -Lipschitz, Grönwall’s inequality bounds how far apart two of its rollouts can move (App. B.4):

$$\|\Phi_{\rho_0 \rightarrow \rho_1}(\mathbf{x}) - \Phi_{\rho_0 \rightarrow \rho_1}(\mathbf{y})\| \leq K \|\mathbf{x} - \mathbf{y}\|, \quad K = e^{\int_{\rho_0}^{\rho_1} L(\rho) d\rho}. \quad (13)$$

When $K < \infty$ the rollout map is continuous, so it sends a connected start support to a connected set and cannot produce exactly separated supports. It can still produce a bimodal density whose modes are joined by a thin bridge of intermediate rollouts. A finite K does not by itself make a policy average two modes; it limits how sharply the policy can separate nearby inputs. The next two results make this precise, Lemma 2 for inputs that coincide and Proposition 3 for inputs that are close.

Lemma 2 (A split after a shared input must come from the input). *Suppose the demonstrations whose policy input at phase ρ^* is $(\mathbf{x}^*, o^*, \rho^*)$ leave $\mathbf{x}^*$ along two constant-velocity branches $\mathbf{x}^* + (\rho - \rho^*)\mathbf{u}_k$, $k = 1, 2$, with $\mathbf{u}_1 \neq \mathbf{u}_2$ and weights $w_1, w_2 > 0$. Then:*

- (a) *the regression target $\bar{v}(\mathbf{x}^*, o^*, \rho^*)$ is the average $(w_1\mathbf{u}_1 + w_2\mathbf{u}_2)/(w_1 + w_2)$;*
- (b) *any policy whose rollouts follow both branches while seeing the same observations has $L(\rho) \geq 1/(\rho - \rho^*)$ for $\rho > \rho^*$, which is not integrable at ρ^* ;*
- (c) *with a deterministic policy in a deterministic environment, rollouts that are in the same state at ρ^* , robot and world, continue identically, so they follow at most one branch. Reaching both requires variation in the robot state or the world before ρ^* , and by (a) the MSE optimum steers different inputs to different branches only where the demonstrations at those inputs take different branches.*

The $1/\rho$ bound of Theorem 4(iii) is the limiting case $\rho^* = 0$ of (b). The lemma also marks what a deterministic policy cannot do. If the mode is set by something the policy never sees, such as the demonstrator’s intent given identical inputs, no deterministic function of those inputs reproduces the split, and a stochastic policy or a latent variable is the right tool. Phase does not help there, and neither does the score penalty, which vanishes where demonstrations coincide (Theorem 5(iii)). Sec. 5 constructs this case: the demonstrations coincide until $\rho \approx 0.15$, every MLP variant fails, and varying the demonstrated starts restores success.

Proposition 3 (Bridge bound). *Let Φ be the rollout map from ρ_0 to ρ_1 of a policy with $K = e^{\int_{\rho_0}^{\rho_1} L(\rho) d\rho} < \infty$. Let A and B be closed sets of end states, such as two modes, at distance at least $g > 0$ from each other, and let S_A and S_B be the sets of starts whose rollouts end in A and in B . Call a start undecided if its rollout ends in neither.*

- (a) *Every start whose distance to S_A is positive but less than g/K is undecided. For starts drawn from any distribution, therefore,*

$$\Pr(\text{undecided}) \geq \Pr(0 < \text{dist}(\mathbf{x}_{\rho_0}, S_A) < g/K). \quad (14)$$

- (b) *Every segment from a point of S_A to a point of S_B contains a stretch of undecided starts of length at least $\ell = g/K$. In particular, let $d = 1$ and $\mathbf{x}_{\rho_0} \sim \mathcal{N}(m, \sigma^2)$. If $\ell > 4\sigma$, the starts in $[m - 2\sigma, m + 2\sigma]$, 95% of them, cannot reach both A and B . If starts in that interval do reach both, then $\Pr(\text{undecided}) \geq \varphi(2)\ell/\sigma \approx 0.054\ell/\sigma$, where φ is the standard normal density.*

Proposition 3 says what smoothness costs at a split: around the boundary between the starts sent to each mode lies a shell of undecided starts at least $g e^{-\int L}$ thick. A trained network has a finite $L(\rho)$ on the states it visits, so it has such a shell. The bound is a lower bound, so we read it as a qualitative prediction, not a rate.

- *Start variation.* For a shell of fixed width, the probability that a start lands in it falls roughly as $1/\sigma$ as the starts spread out. With a single shared start ($\sigma \rightarrow 0$), every rollout follows one path, which is Lemma 2(c).

- *Score penalty.* By Theorem 5 the penalty moves the target toward the locally heavier branch, which steepens the transition across the boundary. A steeper field has a larger L there, which is what a thinner shell requires. This is motivation, not a result about trained networks.
- *Metrics.* Apply the bound with A and B the two goal regions (scenario B), or the two corridors at the phase where the demonstrations pass the obstacle (scenario C). Rollouts in the shell reach neither, and appear among the undecided rollouts and collisions of Fig. 3. MMD and mode proportions measure the marginals of Proposition 1 instead.

The proposition needs only a Lipschitz rollout map. For closed-loop dynamics $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ with $\mathbf{u} = \pi(\mathbf{x}, \rho)$ and f L_x -Lipschitz in the robot state and L_u -Lipschitz in the action, it holds with

$$K = e^{\int_{\rho_0}^{\rho_1} (L_x + L_u L(\rho)) d\rho}, \quad \text{or} \quad K = \prod_k (1 + \Delta\rho (L_x + L_u L(\rho_k))) \quad (15)$$

for Euler rollouts with step $\Delta\rho$. It therefore covers the navigation unicycle of Eq. 5, for which $L_u = 1$ and L_x is the largest commanded forward speed (App. B.4). With a fixed context c , apply it for each c ; when the policy is also Lipschitz in c , the map from $(\mathbf{x}_{\rho_0}, c)$ to $\mathbf{x}_{\rho_1}$ is Lipschitz too, and variation of c counts as start variation. It does not cover Push-T. The pusher moves the block the policy observes, so nearby rollouts see different observations, and bounding their separation would need the block’s response to contact, which is not Lipschitz. Lemma 2(a) and (c) do not need Lipschitz rollouts and cover Push-T, where the randomized initial poses of the block and the pusher supply the variation that (c) requires.

A.4 Lever 3: Label Conflict

OT-CFM removes label conflict by choosing the coupling: under γ^* the straight paths do not cross, so $C_{\gamma^*} = 0$, while the independent coupling of standard CFM gives conflicting labels at every state (App. B.2). BC cannot re-pair the expert’s demonstrations. Theorem 4 identifies the demonstrations whose own pairing is already the OT pairing, and Theorem 5 says what the score penalty does where conflict remains.

Theorem 4 (Ideal case: BC on minimum-effort demonstrations is OT-CFM). *Assume:*

- (A1) *the dynamics are single-integrator in $\mathbb{R}^d$, $\mathbf{u} = \dot{\mathbf{x}}$;*
- (A2) *P_S and P_G have finite second moments, and P_S has a density;*
- (A3) *the demonstrations minimize the expected control effort $\mathcal{E} = \mathbb{E} \int_0^1 \frac{1}{2} \|\dot{\mathbf{x}}_\rho\|^2 d\rho$ among all random paths with these endpoint distributions.*

Then:

- (i) **Demonstrations are OT-paired straight lines.** *Almost every demonstration is $\mathbf{x}_\rho = (1 - \rho)\mathbf{x}_0 + \rho\mathcal{T}(\mathbf{x}_0)$, where $\mathcal{T} = \nabla f$ (f convex) is the Brenier map from P_S to P_G , so $(\mathbf{x}_0, \mathbf{x}_1) \sim \gamma^*$ and a phase-labeled dataset $\mathcal{D}_\rho$ (§4) built from minimum-effort demonstrations is automatically OT-paired.*
- (ii) **Same objective.** $\mathcal{L}_{\text{BC}}(\phi) = \mathcal{L}_{\text{CFM}}^{\gamma^*}(\phi)$ for every phase-conditioned policy π_ϕ .
- (iii) **No label conflict.** *Demonstrations never cross in state–phase space, so $C = 0$ and $\bar{v}$ fits every demonstration exactly: $\bar{v}(\mathbf{x}_\rho, \rho) = \mathbf{x}_1 - \mathbf{x}_0$. On $(0, 1)$, $\bar{v}$ is $L(\rho)$ -Lipschitz with $L(\rho) = \max\{1/\rho, 1/(1 - \rho)\}$, so each demonstration is the unique rollout of $\bar{v}$ on $[\rho_0, 1]$ from its state at any $\rho_0 \in (0, 1)$.*

(A3) is the zero-noise limit of planning as inference [32] with P_S and P_G imposed exactly, i.e., of the Schrödinger bridge [33, 34]. Parts (i) and (ii) are classical: (i) is the Lagrangian form of the Benamou–Brenier formula, whose minimizers are McCann’s displacement interpolation [35, 36,

37], and (ii) follows by substitution. The content we use is (iii). With minimum-effort demonstrations the expert’s pairing is the OT pairing, so label conflict vanishes and phase-conditioned BC has zero irreducible loss. The cost is the singular Lipschitz bound at $\rho = 0$: the ideal policy decides the mode at the first instant, which is how it escapes Proposition 3 ($K = \infty$ from $\rho_0 = 0$). Obstacle-avoiding and contact-rich demonstrations, including all of ours, violate (A3). An obstacle-aware version would need a constrained path cost and geodesic interpolation, and its no-crossing property would need further assumptions. We therefore use Theorem 4 as the benchmark for zero conflict, not as a model of our tasks.

The score penalty. Our regularizer adapts the divergence matching of Huang et al. [38]. If a learned field $\mathbf{v}$ moves the same densities p_ρ as a target field $\mathbf{u}$, both satisfy the continuity equation, so $\nabla \cdot (p_\rho(\mathbf{u} - \mathbf{v})) = 0$, or equivalently

$$\nabla \cdot (\mathbf{u} - \mathbf{v}) + (\mathbf{u} - \mathbf{v}) \cdot \nabla \log p_\rho = 0. \quad (16)$$

Huang et al. show that the expected absolute value of the left side bounds the total variation distance between the target and learned probability paths, and they train flow matching models with a penalty on it, using a conditional version (conditional divergence matching, CDM) that is tractable for flow matching. We keep only the second term, the velocity error along the score. Demonstrations give no closed form for the divergence of the demonstrated flow, and divergence-based regularizers built on our estimates of it were neutral or hurt performance (App. E), whereas the score of the demonstrated states can be estimated directly. With $\mathbf{u}_\rho = \dot{\mathbf{x}}_\rho$ the demonstrated velocity and $\mathbf{s}_\rho = \nabla_{\mathbf{x}} \log \hat{p}_\rho$ the score of a Gaussian kernel density estimate of the demonstrated robot states at phase ρ (App. C), the regularizer is the L_1 penalty

$$\mathcal{L}_{reg}(\phi) = \mathbb{E}_{\rho \sim \mathcal{U}[0,1]} \mathbb{E} \left| (\mathbf{u}_\rho - \pi_\phi(\mathbf{x}_\rho, \rho)) \cdot \mathbf{s}_\rho(\mathbf{x}_\rho) \right|. \quad (17)$$

Huang et al. analyze the full residual of Eq. 16; Theorem 5 characterizes what this one term does to the MSE optimum. At a state $\mathbf{x}$ and phase ρ with $\mathbf{s} = \mathbf{s}_\rho(\mathbf{x}) \neq 0$, let $\hat{\mathbf{s}} = \mathbf{s}/\|\mathbf{s}\|$ and let $b = \mathbf{u}_\rho \cdot \hat{\mathbf{s}}$ be the demonstrated velocity along the score. Over the demonstrations at $(\mathbf{x}, \rho)$, its mean is $\bar{b} = \bar{\mathbf{v}}(\mathbf{x}, \rho) \cdot \hat{\mathbf{s}}$; let m be its median (the one closest to $\bar{b}$ if the median is not unique). With an observation, read $(\mathbf{x}, \rho)$ as the full policy input $(\mathbf{x}, \mathcal{O}, \rho)$ in these averages and in Theorem 5. The score itself depends only on the robot state and phase, which are inputs of the policy, and that is all the proof uses.

Theorem 5 (Score regularization moves the policy toward the median velocity along the score). *Let π_λ minimize $\mathcal{L}_{BC} + \lambda \mathcal{L}_{reg}$ over all phase-conditioned policies. Where $\mathbf{s} = 0$, $\pi_\lambda = \bar{\mathbf{v}}$. Where $\mathbf{s} \neq 0$:*

- (i) *the component of π_λ orthogonal to $\mathbf{s}$ equals that of $\bar{\mathbf{v}}$;*
- (ii) *$a_\lambda = \pi_\lambda \cdot \hat{\mathbf{s}}$ lies between $\bar{b}$ and m , satisfies $|a_\lambda - \bar{b}| \leq \lambda \|\mathbf{s}\|/2$, and tends to m as $\lambda \|\mathbf{s}\| \rightarrow \infty$;*
- (iii) *$\pi_\lambda = \bar{\mathbf{v}}$ wherever the demonstrations at $(\mathbf{x}, \rho)$ agree along $\mathbf{s}$, in particular wherever they do not cross (as in Theorem 4), and wherever b is symmetric about $\bar{b}$.*

What the regularizer does. Across $\mathbf{s}$ the pointwise objective is the squared loss alone. Along $\hat{\mathbf{s}}$ it is the sum of $\mathbb{E}(a - b)^2$, minimized by the mean $\bar{b}$, and $\lambda \|\mathbf{s}\| \mathbb{E}|a - b|$, minimized by the median m (App. B.6). The penalty therefore acts only where the label conflict is positive along the score. It is not an OT surrogate: it uses no endpoint information, is invariant to $\mathbf{s} \rightarrow -\mathbf{s}$, is inactive where demonstrations do not cross, and cannot break a symmetric tie. Where branches overlap, the median lies in the heavier one, so the penalty is mode-seeking and steepens transitions between branches. Its cost is bias: a large $\lambda \|\mathbf{s}\|$ forfeits the marginal preservation of Proposition 1 and can suppress minority behavior (App. B.6). We apply it at splits because a smooth network effectively averages nearby branches there, the overlapping regime of Theorem 5; this too is motivation, not a result about trained networks.

A.5 Scope

Trained networks. Every result concerns a population optimum over all functions of the policy input, or the rollouts of a given Lipschitz policy. We connect the theory to trained networks only through their finite Lipschitz constants (Proposition 3).

Nonholonomic dynamics. Proposition 1 needs neither (A1) nor (A3). For $\dot{\mathbf{x}} = B(\mathbf{x})\mathbf{u}$, as for the navigation unicycle of Eq. 5, $B(\mathbf{x}) \mathbb{E}[\mathbf{u} \mid \mathbf{x}_\rho = \mathbf{x}, \rho] = \bar{\mathbf{v}}(\mathbf{x}, \rho)$, so the optimum over actions rolls out along $\bar{\mathbf{v}}$. Because the columns of $B(\theta)$ are orthonormal, the squared loss on twists also equals that on state velocities.

Push-T. The policy observes the block, which the pusher moves through contact. Each lever has a result that covers it: phase removes temporal aliasing from the loss (§A.2); at a shared input the regression target is the branch average, and a deterministic closed loop carries one input along one branch (Lemma 2(a), (c)); and the score penalty acts as in Theorem 5. The results that compare rollouts (Proposition 1, Eq. 13, Lemma 2(b), Proposition 3) do not cover it, because nearby rollouts see different block poses.

Manifolds. Parts (i)–(iii) of Theorem 4, apart from the formula for $L(\rho)$, extend to Riemannian manifolds with geodesics in place of straight lines [39, 40]. On SE(2) and SE(3) these are geodesics of a Riemannian metric, which generally differ from group-exponential curves because SE(n) has no bi-invariant metric.

Phase. We treat ρ as the integration variable. The learned task-progress estimator of §4 makes ρ a function of the observation, which the analysis does not model.

L_1 . Neither Theorem 4 nor Proposition 1 extends to L_1 . An L_1 control cost does not determine the demonstrated velocity, being invariant to reparametrization, and an L_1 loss learns the conditional median, which need not transport the demonstrations.

B Proofs

We used Lean 4.34.1 to verify all proofs.

B.1 Proofs for the Setting and Lemma 1

The phase-free optimum. A phase-free policy sees the state but not the phase, so its squared loss is minimized by $\mathbb{E}[\dot{\mathbf{x}}_\rho \mid \mathbf{x}_\rho = \mathbf{x}]$ with $\rho \sim \mathcal{U}[0, 1]$ random. When the p_ρ have densities, Bayes’ rule gives ρ the conditional density $p_\rho(\mathbf{x}) / \int_0^1 p_{\rho'}(\mathbf{x}) d\rho'$ given $\mathbf{x}_\rho = \mathbf{x}$, which gives Eq. 8.

Proof of Eq. 7. Write $\dot{\mathbf{x}} = \dot{\mathbf{x}}_\rho$, $m = m(Z)$, $m' = m'(Z')$ and $\pi' = \pi'(Z')$. All four have finite second moments: $\dot{\mathbf{x}}$ because $\mathbb{E} \int_0^1 \|\dot{\mathbf{x}}_\rho\|^2 d\rho < \infty$, π' by hypothesis, and m and m' because, by Jensen’s inequality, $\|\mathbb{E}[\dot{\mathbf{x}} \mid Z]\|^2 \leq \mathbb{E}[\|\dot{\mathbf{x}}\|^2 \mid Z]$. Split

$$\pi' - \dot{\mathbf{x}} = (\pi' - m') + (m' - m) + (m - \dot{\mathbf{x}}).$$

Expanding the squared norm gives the three terms of Eq. 7 plus three cross terms. Each cross term is zero by the defining property of conditional expectation: for every square-integrable g ,

$$\mathbb{E}[(Y - \mathbb{E}[Y \mid W]) \cdot g(W)] = 0.$$

- $\mathbb{E}[(m - \dot{\mathbf{x}}) \cdot (\pi' - m')] = 0$: take $Y = \dot{\mathbf{x}}$ and $W = Z$. Since $Z' = h(Z)$, $\pi' - m'$ is a function of Z .
- $\mathbb{E}[(m - \dot{\mathbf{x}}) \cdot (m' - m)] = 0$: the same, since $m' - m$ is a function of Z .

- $\mathbb{E}[(m' - m) \cdot (\pi' - m')] = 0$: take $Y = m$ and $W = Z'$. Because Z' is a function of Z , the tower property gives $\mathbb{E}[m \mid Z'] = \mathbb{E}[\mathbb{E}[\dot{\mathbf{x}} \mid Z] \mid Z'] = \mathbb{E}[\dot{\mathbf{x}} \mid Z'] = m'$, and $\pi' - m'$ is a function of Z' .

□

The special cases. Eq. 9 is Eq. 7 with $Z' = Z = (\mathbf{x}_\rho, \rho)$, so $m' = m = \bar{v}$ and the last term vanishes. Eq. 10 is $Z = (\mathbf{x}_\rho, \rho)$ and $Z' = \mathbf{x}_\rho$, with $h(\mathbf{x}, \rho) = \mathbf{x}$, $m = \bar{v}$ and $m' = \bar{v}_{\text{free}}$. With an observation, $Z = (\mathbf{x}_\rho, \mathcal{O}_\rho, \rho)$ and $Z' = (\mathbf{x}_\rho, \mathcal{O}_\rho)$, with h dropping the phase. In each case the minimizer over all policies of Z' is m' , which sets the fit error to zero.

Proof of Lemma 1. (a) Substituting $\mathbf{x}_\rho = (1 - \rho)\mathbf{x}_0 + \rho\mathbf{x}_1$ and $\dot{\mathbf{x}}_\rho = \mathbf{x}_1 - \mathbf{x}_0$ into Eq. 6 gives Eq. 12 term by term; the rest is Eq. 9. When P_S has a density, Theorem 4(iii) gives $C_{\gamma^*} = 0$. (b) Both procedures solve the same initial-value problem. (c) For straight demonstrations the velocity $\mathbf{x}_1 - \mathbf{x}_0$ does not depend on ρ , so its distribution is the same at every phase. For $\mathbf{x}_\rho = \sin(\pi\rho)$ the velocity at phase ρ is $\pi \cos(\pi\rho)$, which changes with ρ . □

B.2 Proof of Proposition 1, and a Closed-Form Example

Proof. For a smooth, compactly supported h , $\frac{d}{d\rho} \mathbb{E} h(\mathbf{x}_\rho) = \mathbb{E}[\nabla h(\mathbf{x}_\rho) \cdot \dot{\mathbf{x}}_\rho] = \mathbb{E}[\nabla h(\mathbf{x}_\rho) \cdot \bar{v}(\mathbf{x}_\rho, \rho)]$. The second step averages the velocity first, which is allowed because $\nabla h(\mathbf{x}_\rho)$ is fixed once $\mathbf{x}_\rho$ is; this is the weak form of the continuity equation. Rollouts of $\bar{v}$ satisfy the same equation from the same initial distribution, and for such a Lipschitz field its solution is unique [41, Sec. 8.1]; the regularity this needs is checked below. Integrating over $\rho \in [0, 1]$ and using $\int_0^1 p_\rho \bar{v} d\rho = \mu \bar{v}_{\text{free}}$ (Eq. 8) gives the phase-free identity. □

The phase-free identity in one dimension. For $d = 1$, integrating $\frac{d}{dx}(\mu \bar{v}_{\text{free}}) = p_0 - p_1$ gives $\mu \bar{v}_{\text{free}} = F_S - F_G$, where F_S and F_G are the cumulative distribution functions of P_S and P_G . The constant of integration is zero because $\int |\mu \bar{v}_{\text{free}}| dx \leq \mathbb{E} \int_0^1 |\dot{\mathbf{x}}_\rho| d\rho < \infty$.

Regularity. The uniqueness step uses the theory of Ambrosio et al. [41, Sec. 8.1] for the Lipschitz extension of $\bar{v}$. A sufficient condition is $\int_{\rho_0}^{\rho_1} (L(\rho) + \|\bar{v}(0, \rho)\|) d\rho < \infty$, which bounds $\bar{v}$ and its Lipschitz constant on every ball by integrable functions of ρ . The second term is finite whenever $\mathbb{E}\|\mathbf{x}_{\rho_0}\| < \infty$: for every $\mathbf{x}$, $\|\bar{v}(0, \rho)\| \leq \|\bar{v}(\mathbf{x}, \rho)\| + L(\rho)\|\mathbf{x}\|$, and averaging over $\mathbf{x} \sim p_\rho$ bounds it by $\mathbb{E}\|\bar{v}(\mathbf{x}_\rho, \rho)\| + L(\rho)\mathbb{E}\|\mathbf{x}_\rho\|$, where $\mathbb{E}\|\mathbf{x}_\rho\| \leq \mathbb{E}\|\mathbf{x}_{\rho_0}\| + \mathbb{E} \int_0^1 \|\dot{\mathbf{x}}_\rho\| d\rho$.

A closed-form example. Take straight paths between independent endpoints $\mathbf{x}_0, \mathbf{x}_1 \sim \mathcal{N}(0, I)$, i.e., standard CFM (Fig. 8a). Then $\mathbf{x}_\rho \sim \mathcal{N}(0, \sigma_\rho^2 I)$ with $\sigma_\rho^2 = (1 - \rho)^2 + \rho^2$. The velocity $\mathbf{x}_1 - \mathbf{x}_0$ and the state $\mathbf{x}_\rho$ are jointly Gaussian with covariance $(\rho - (1 - \rho))I = (2\rho - 1)I$, so the average velocity at a state is

$$\bar{v}(\mathbf{x}, \rho) = \mathbb{E}[\mathbf{x}_1 - \mathbf{x}_0 \mid \mathbf{x}_\rho = \mathbf{x}] = \frac{2\rho - 1}{\sigma_\rho^2} \mathbf{x},$$

which is $L(\rho)$ -Lipschitz with $L(\rho) = |2\rho - 1|/\sigma_\rho^2 \leq 1$. Every state receives conflicting labels ($C_\gamma > 0$): many demonstrations with different velocities pass through it. Yet since $\frac{d}{d\rho} \sigma_\rho = (2\rho - 1)/\sigma_\rho$, the rollout of $\bar{v}$ from $\mathbf{x}(0)$ is $\mathbf{x}(\rho) = \sigma_\rho \mathbf{x}(0)$ (Fig. 8b). It matches p_ρ at every phase, because $\sigma_\rho \mathbf{x}(0) \sim \mathcal{N}(0, \sigma_\rho^2 I)$, and it ends at $\mathbf{x}(1) = \mathbf{x}(0)$. The rollouts never cross: they contract toward the center until $\rho = \frac{1}{2}$, where $\bar{v} = 0$ everywhere and every rollout momentarily stops, and then expand again. Averaging the velocities preserved every marginal and replaced the random pairing by the identity, which is the OT pairing of $\mathcal{N}(0, I)$ with itself.

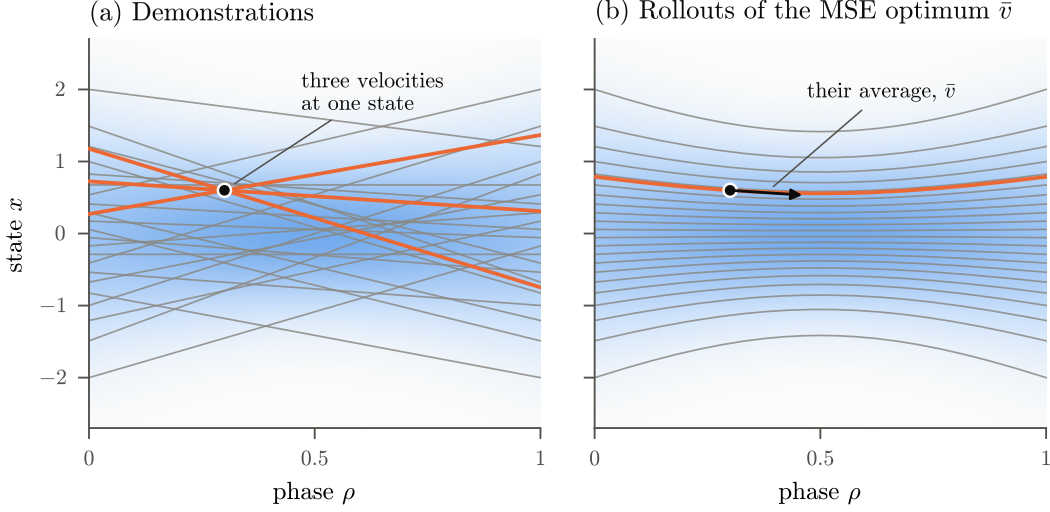


Figure 8: Averaging velocities does not average the distribution: the closed-form example of App. B.2 in one dimension. Shading is the marginal density p_ρ , the same in both panels. (a) Straight demonstrations between independently drawn standard Gaussian endpoints. They cross everywhere, so demonstrations through the same state and phase carry different velocities (orange: three through the marked point) and the label conflict is positive. (b) Rollouts of the MSE optimum $\bar{v}$ from the same starts. At each state $\bar{v}$ is the average of the demonstrated velocities there (arrow; the middle orange line in (a) has the same slope). The rollouts do not cross, yet they reproduce p_ρ at every phase (Proposition 1), and each ends where it started.

B.3 Proof of Proposition 2

Proof. (a) The map $\mathcal{T}$ sends the first block to $[2 - \delta, 2 + \delta]$ and the second to $[5 - \delta, 5 + \delta]$, and $2 + \delta < 5 - \delta$, so it is increasing and hence the Brenier map onto $P_G = \mathcal{T}_\# P_S$. The straight demonstrations are therefore minimum-effort (Theorem 4(i)), so $C = 0$ and $\bar{v}$ fits every demonstration (Theorem 4(iii)). Here $\bar{v} = 2$ on the region of state-phase space swept by the first group and $\bar{v} = 4$ on the region swept by the second, so the rollouts of $\bar{v}$ are the demonstrations themselves and reproduce every p_ρ .

The first group moves at speed 2 and occupies $[-\delta, 2 + \delta]$; the second moves at speed 4 and occupies $[1 - \delta, 5 + \delta]$. At phase ρ they lie in $[2\rho - \delta, 2\rho + \delta]$ and $[1 + 4\rho - \delta, 1 + 4\rho + \delta]$, which are disjoint, so at each state and phase only one group is present. Let μ_k be the occupation density of group k , the time its demonstrations spend near each state per unit length. By Eq. 8, $\bar{v}_{\text{free}} = (2\mu_1 + 4\mu_2)/(\mu_1 + \mu_2)$. Each μ_k is continuous and positive on the interior of its occupied interval, so $\bar{v}_{\text{free}} = 2$ on $(-\delta, 1 - \delta]$, $\bar{v}_{\text{free}} = 4$ on $[2 + \delta, 5 + \delta]$, and $2 < \bar{v}_{\text{free}} < 4$ on $(1 - \delta, 2 + \delta)$, where both groups pass. On $(1 + \delta, 2 - \delta)$ every demonstration of either group is present, $\mu_1 = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$ and $\mu_2 = \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8}$, so $\bar{v}_{\text{free}} = 8/3$. There $\bar{v} \neq \bar{v}_{\text{free}}$, so $A > 0$, and by Eq. 10 every phase-free policy has $\mathcal{L}_{\text{BC}} \geq A > 0$. As $\delta \rightarrow 0$, $A \rightarrow \frac{1}{4}(2 - \frac{8}{3})^2 + \frac{1}{8}(4 - \frac{8}{3})^2 = \frac{1}{3}$.

Because $\bar{v}_{\text{free}}$ is continuous and positive on $I = (-\delta, 5 + \delta)$, its rollouts in I are unique and increasing, and the time to travel from x to $y > x$ is $\tau(x, y) = \int_x^y dz / \bar{v}_{\text{free}}(z)$. With $G(y) = \int_0^y dz / \bar{v}_{\text{free}}(z)$, the rollout map is $\Phi(x) = G^{-1}(G(x) + 1)$, which is continuous and strictly increasing. For a start $a \in (-\delta, \delta]$ of the first group, $(a, a + 2)$ contains $(1 - \delta, a + 2)$, where $\bar{v}_{\text{free}} > 2$, and $\bar{v}_{\text{free}} \geq 2$ on the rest, so $\tau(a, a + 2) < 1$. The rollout reaches $\mathcal{T}(a) = a + 2$ before $\rho = 1$ and moves on, so $\Phi(a) > \mathcal{T}(a)$; it stays in I because its speed is at most 4. For a start $b \in [1 - \delta, 1 + \delta)$ of the second group, $(b, b + 4)$ contains $(b, 2 + \delta)$, where $\bar{v}_{\text{free}} < 4$, so $\tau(b, b + 4) > 1$ and $\Phi(b) < \mathcal{T}(b)$.

Finally, $\Phi_\# P_S \neq P_G$. Since Φ is strictly increasing, $\Pr(\Phi(\mathbf{x}_0) \leq \Phi(b)) = F_S(b)$. Since $\mathcal{T}$ is strictly increasing and pushes P_S to P_G , $F_G(\mathcal{T}(b)) = F_S(b)$. For $b \in (1 - \delta, 1 + \delta)$, $\mathcal{T}(b) \in (5 - \delta, 5 + \delta)$,

where P_G has positive density, and $\Phi(b) < \mathcal{T}(b)$, so $F_G(\Phi(b)) < F_G(\mathcal{T}(b)) = F_S(b)$. The two distribution functions differ at $\Phi(b)$.

In the limit $\delta \rightarrow 0$, $\bar{v}_{\text{free}}$ is 2 on $[0, 1)$, $8/3$ on $(1, 2)$ and 4 on $(2, 5]$. The rollout from 0 spends $1/2$ on $[0, 1]$ and $3/8$ on $[1, 2]$, and ends at $2 + 4 \cdot \frac{1}{8} = 2.5$; the rollout from 1 spends $3/8$ on $[1, 2]$ and ends at $2 + 4 \cdot \frac{5}{8} = 4.5$.

(b) At phase ρ every demonstration has velocity $\pi \cos(\pi\rho)$ and is shifted by the same $\sin(\pi\rho)$, so distinct starts never meet at the same phase. Thus $C = 0$, $\bar{v}(\mathbf{x}, \rho) = \pi \cos(\pi\rho)$, which is 0-Lipschitz, and its rollouts reproduce every demonstration. Since $P_G = P_S$, the one-dimensional identity of App. B.2 gives $\mu \bar{v}_{\text{free}} = F_S - F_G = 0$, so $\bar{v}_{\text{free}} = 0$ on $(-\delta, 1 + \delta)$, where $\mu > 0$. The same follows directly from Eq. 8: a pass at speed c spends time proportional to $1/c$ near each state, so it contributes $\pm c \cdot \frac{1}{c} = \pm 1$ to the numerator, and the outward and return passes cancel. The rollouts from $\mathbf{x}_0 \in (-\delta, \delta)$ therefore stay at $\mathbf{x}_0$, while for $\rho \in (0, 1)$ the demonstrations are shifted by $\sin(\pi\rho) > 0$, so $p_\rho \neq P_S$; at $\rho = \frac{1}{2}$ the two supports are disjoint. Every occupied state is visited at phases ρ and $1 - \rho$ with opposite velocities, so $A > 0$ and every phase-free policy has $\mathcal{L}_{\text{BC}} > 0$. $\square$

B.4 Proofs for Input Variation

Grönwall. If π is $L(\rho)$ -Lipschitz, two rollouts $\mathbf{z}_\rho, \mathbf{z}'_\rho$ satisfy $\frac{d}{d\rho} \|\mathbf{z}_\rho - \mathbf{z}'_\rho\| \leq \|\pi(\mathbf{z}_\rho, \rho) - \pi(\mathbf{z}'_\rho, \rho)\| \leq L(\rho) \|\mathbf{z}_\rho - \mathbf{z}'_\rho\|$ wherever they differ, and Grönwall's inequality gives Eq. 13. For closed-loop dynamics $\dot{\mathbf{x}} = f(\mathbf{x}, \pi(\mathbf{x}, \rho))$, $\|f(\mathbf{x}, \pi(\mathbf{x}, \rho)) - f(\mathbf{y}, \pi(\mathbf{y}, \rho))\| \leq (L_x + L_u L(\rho)) \|\mathbf{x} - \mathbf{y}\|$, and the same argument applies. An Euler step $\mathbf{x} \mapsto \mathbf{x} + \Delta\rho f(\mathbf{x}, \pi(\mathbf{x}, \rho_k))$ is $(1 + \Delta\rho (L_x + L_u L(\rho_k)))$ -Lipschitz, and the rollout map is their composition. For the unicycle of Eq. 5, $f(\mathbf{x}, \mathbf{u}) = B(\theta)\mathbf{u}$. The columns of $B(\theta)$ are orthonormal, so $\|B(\theta)(\mathbf{u} - \mathbf{u}')\| = \|\mathbf{u} - \mathbf{u}'\|$ and $L_u = 1$. Only the first column depends on the state, and $\|B(\theta)\mathbf{u} - B(\theta')\mathbf{u}\| = |\mathbf{v}| \|(\cos \theta - \cos \theta', \sin \theta - \sin \theta')\| \leq |\mathbf{v}| |\theta - \theta'|$, so L_x is the largest commanded forward speed $|\mathbf{v}|$.

Proof of Lemma 2. (a) is the conditional mean over the demonstrations at $(\mathbf{x}^*, o^*, \rho^*)$. (b) At phase ρ the two branches are $(\rho - \rho^*) \|\mathbf{u}_1 - \mathbf{u}_2\|$ apart and their velocities differ by $\|\mathbf{u}_1 - \mathbf{u}_2\|$. Both rollouts see the same observation, so the ratio bounds the Lipschitz constant in the robot state; it is exactly $1/(\rho - \rho^*)$, and $\int_{\rho^*}^1 d\rho/(\rho - \rho^*) = \infty$. (c) A deterministic policy in a deterministic environment makes the closed loop a deterministic system, so with unique solutions its future is a function of its state at ρ^* ; the last sentence is (a) applied at each input. $\square$

Proof of Proposition 3. (a) Φ is continuous, so S_A and S_B are closed. For $\mathbf{x} \in S_A$ and $\mathbf{y} \in S_B$, $g \leq \|\Phi(\mathbf{x}) - \Phi(\mathbf{y})\| \leq K \|\mathbf{x} - \mathbf{y}\|$ by Eq. 13, so every point of S_B is at least g/K from S_A . A start with $0 < \text{dist}(\mathbf{x}_{\rho_0}, S_A) < g/K$ is therefore in neither S_A nor S_B .

(b) Parametrize the segment from $\mathbf{a} \in S_A$ to $\mathbf{b} \in S_B$ by arc length $t \in [0, \|\mathbf{b} - \mathbf{a}\|]$. Let t_1 be the largest t whose point is in S_A , and t_2 the smallest $t \geq t_1$ whose point is in S_B ; both exist because the sets are closed. The open sub-segment between them meets neither set, and its length is at least the distance between its endpoints, which is at least g/K by (a). For $d = 1$, let $W = [m - 2\sigma, m + 2\sigma]$. If starts in W reach both A and B , the undecided sub-interval between two of them lies in W and has length at least ℓ . This is impossible when $\ell > 4\sigma$. When $\ell \leq 4\sigma$, the density of $\mathcal{N}(m, \sigma^2)$ is at least $\varphi(2)/\sigma$ on W , so the sub-interval has probability at least $\varphi(2)\ell/\sigma$, with $\varphi(2) = e^{-2}/\sqrt{2\pi} \approx 0.054$. $\square$

For $\mathbf{x}_{\rho_0} \sim \mathcal{N}(\mathbf{m}, \sigma^2 I)$ in $\mathbb{R}^d$, apply (b) on each line parallel to a fixed unit vector $\mathbf{e}$: given the other coordinates, the coordinate along $\mathbf{e}$ is $\mathcal{N}(\mathbf{m} \cdot \mathbf{e}, \sigma^2)$. The probability of an undecided start is then at least $0.054 \ell/\sigma$ times the probability of drawing a line whose starts within 2σ of its center reach both A and B .

B.5 Proof of Theorem 4

Proof. (i) For each path, Jensen’s inequality gives

$$\frac{1}{2}\|\mathbf{x}_1 - \mathbf{x}_0\|^2 = \frac{1}{2}\left\|\int_0^1 \dot{\mathbf{x}}_\rho d\rho\right\|^2 \leq \int_0^1 \frac{1}{2}\|\dot{\mathbf{x}}_\rho\|^2 d\rho, \quad (18)$$

with equality if and only if the velocity is constant. Since $(\mathbf{x}_0, \mathbf{x}_1)$ is a coupling of P_S and P_G , averaging gives $\mathcal{E} \geq \frac{1}{2}\mathbb{E}\|\mathbf{x}_1 - \mathbf{x}_0\|^2 \geq \frac{1}{2}W_2^2(P_S, P_G)$. Straight lines between γ^* -paired endpoints attain both bounds, so every minimizer does: its paths are straight and its endpoints are W_2 -optimally paired. By (A2) and Brenier’s theorem, γ^* is unique and supported on the graph of $\mathcal{T} = \nabla f$ [42, 37].

(ii) Substitute $\mathbf{x}_\rho = (1 - \rho)\mathbf{x}_0 + \rho\mathbf{x}_1$ and $\dot{\mathbf{x}}_\rho = \mathbf{x}_1 - \mathbf{x}_0$ from (i) into Eq. 6. Lemma 1 extends this identity to every coupling.

(iii) Take two starts $\mathbf{a}, \mathbf{b}$ and let $\Delta_0 = \mathbf{a} - \mathbf{b}$ and $\Delta_1 = \mathcal{T}(\mathbf{a}) - \mathcal{T}(\mathbf{b})$. Because f is convex, $\mathcal{T}$ is monotone, $\langle \Delta_0, \Delta_1 \rangle \geq 0$, so the two demonstrations’ states at phase ρ satisfy

$$\|\mathbf{x}_\rho - \mathbf{x}'_\rho\|^2 = \|(1 - \rho)\Delta_0 + \rho\Delta_1\|^2 \geq (1 - \rho)^2\|\Delta_0\|^2 + \rho^2\|\Delta_1\|^2. \quad (19)$$

For $\rho < 1$, equal states force $\mathbf{a} = \mathbf{b}$: demonstrations never cross, so the phase- ρ state determines the start and hence the velocity, and $C = \mathcal{L}_{\text{BC}}(\bar{v}) = 0$. The velocities differ by $\Delta_1 - \Delta_0$, and by Eq. 19, $\|\Delta_1 - \Delta_0\|^2 \leq \|\Delta_0\|^2 + \|\Delta_1\|^2 \leq \|\mathbf{x}_\rho - \mathbf{x}'_\rho\|^2 / \min(\rho, 1 - \rho)^2$, which is the bound on $L(\rho)$. Every $\mathbf{x} \in \mathbb{R}^d$ can be written in exactly one way as $(1 - \rho)\mathbf{a} + \rho\mathbf{t}$ with $\mathbf{t} \in \partial f(\mathbf{a})$, because $\frac{1-\rho}{2}\|\cdot\|^2 + \rho f$ is strongly convex [43, Prop. 12.60]; setting $\bar{v}(\mathbf{x}, \rho) = \mathbf{t} - \mathbf{a}$ extends the bound to all of $\mathbb{R}^d$, since ∂f is monotone too. Each demonstration is a rollout of $\bar{v}$, and it is the only one from its phase- ρ_0 state: two rollouts $\mathbf{z}_\rho, \mathbf{z}'_\rho$ satisfy $\frac{d}{d\rho}\|\mathbf{z}_\rho - \mathbf{z}'_\rho\| \leq L(\rho)\|\mathbf{z}_\rho - \mathbf{z}'_\rho\|$, and $L(\rho)$ is bounded on every $[\rho_0, \rho_1] \subset (0, 1)$, so by Grönwall’s inequality rollouts that agree at ρ_0 agree on $[\rho_0, \rho_1]$. $\square$

B.6 Proof of Theorem 5, and Properties of the Score Penalty

Proof. Both losses average over the demonstrations at each policy input $(\mathbf{x}, \rho)$, or $(\mathbf{x}, \mathcal{O}, \rho)$ with an observation, and $\mathbf{s}$ is a function of that input, so $\pi_\lambda(\mathbf{x}, \rho)$ minimizes $J(\mathbf{v}) = \mathbb{E}\|\mathbf{v} - \mathbf{u}_\rho\|^2 + \lambda \mathbb{E}[(\mathbf{u}_\rho - \mathbf{v}) \cdot \mathbf{s}]$ over those demonstrations. Where $\mathbf{s} = 0$, J is the squared loss alone. Otherwise, writing $\mathbf{v} = a\hat{\mathbf{s}} + \mathbf{v}_\perp$ and $\mathbf{u}_\rho = b\hat{\mathbf{s}} + \mathbf{u}_\perp$ splits J into $\mathbb{E}\|\mathbf{v}_\perp - \mathbf{u}_\perp\|^2$, which is minimized by $\mathbb{E}\mathbf{u}_\perp$ and gives (i), and $g(a) = \mathbb{E}(a - b)^2 + \lambda\|\mathbf{s}\| \mathbb{E}|a - b|$. The function g is strictly convex, with subgradient $2(a - \bar{b}) + \lambda\|\mathbf{s}\| h(a)$, where $h(a)$ is the subgradient of $\mathbb{E}|a - b|$. Where b has no atom at a , $h(a) = \Pr(b < a) - \Pr(b > a)$, which gives the stationarity condition $\mathbf{v} = \bar{v} + \frac{\lambda}{2} \mathbb{E}[\text{sgn}((\mathbf{u}_\rho - \mathbf{v}) \cdot \mathbf{s})] \mathbf{s}$. In general, $h(a) \subseteq [0, 1]$ above the smallest median of b , $h(a) \subseteq [-1, 0]$ below the largest, and $0 \in h(a)$ exactly at the medians. At a_λ the subgradient contains 0, so $|a_\lambda - \bar{b}| \leq \lambda\|\mathbf{s}\|/2$. Because $2(a - \bar{b})$ has the sign of $a - \bar{b}$, the subgradient is negative when a is below both $\bar{b}$ and the largest median, and positive when a is above both $\bar{b}$ and the smallest median, so a_λ lies between $\bar{b}$ and m . As $\lambda\|\mathbf{s}\| \rightarrow \infty$, the minimizers of $g/(\lambda\|\mathbf{s}\|)$ approach the medians of b , and the quadratic term selects the one closest to $\bar{b}$; this proves (ii). Finally, 0 is a subgradient at $a = \bar{b}$, i.e., $a_\lambda = \bar{b}$, exactly when $\bar{b}$ is a median of b , which holds if b is constant or symmetric about $\bar{b}$; with (i) this gives (iii). $\square$

Size and direction of the shift. Where b has no atom at a_λ , the stationarity condition reads $a_\lambda = \bar{b} + \frac{\lambda\|\mathbf{s}\|}{2} (\Pr(b > a_\lambda) - \Pr(b < a_\lambda))$: the prediction moves in proportion to how unevenly the demonstrations split around it, toward the side with more of them. The penalty depends on $\mathbf{s}$ only through its axis, since replacing $\mathbf{s}$ by $-\mathbf{s}$ leaves $\mathcal{L}_{\text{reg}}$ unchanged, so it cannot prefer moving up the density over moving down it; the direction of the shift is set by where the median lies relative to the mean. With velocities ± 1 along the score, equally likely, $a_\lambda = 0$ for every λ , so an exact symmetric tie is not broken. With velocity 0 with probability $\frac{3}{4}$ and 4 with probability $\frac{1}{4}$, and $\lambda\|\mathbf{s}\| = 2$, the mean is 1, the median is 0 and $a_\lambda = \frac{1}{2}$, a shift against $\mathbf{s}$; reflecting the law gives a shift along $\mathbf{s}$.

Bias. If a third of the demonstrations at a state move along $\hat{\mathbf{s}}$ at speed 3 and the rest do not, then $\bar{b} = 1$, $m = 0$ and $a_\lambda = \max\{1 - \lambda\|\mathbf{s}\|/6, 0\}$, so for $\lambda\|\mathbf{s}\| \geq 6$ the regularized policy ignores the

moving third. More generally, $\pi_\lambda \neq \bar{v}$ wherever $s \neq 0$ and $\bar{b}$ is not a median of b , so the rollouts of π_λ need not reproduce p_ρ (Proposition 1), and minority behavior can be suppressed.

Realizability. These pointwise laws arise from genuine demonstration distributions. At an interior phase ρ^* , draw $\mathbf{x}_{\rho^*} = Z \sim \mathcal{N}(1, 1)$ independently of the velocity u and set $\mathbf{x}_\rho = Z + (\rho - \rho^*)u$. The demonstrations at $x = 0$ then have the distribution of u , and the score of p_{ρ^*} at $x = 0$ is 1, or $1/(1 + h^2)$ for a Gaussian kernel estimate with bandwidth h .

C KDE Score Calculation

The kernel score $\nabla \log \hat{p}_\rho$ is the gradient of a Gaussian kernel density estimate, built at each phase from the demonstrations at that phase and expressed in each pose’s body tangent space through the log map,

$$\hat{p}_\rho(\mathbf{x}) = \frac{1}{M} \sum_{j=1}^M \mathcal{N}(\mathbf{x}; \mathbf{x}_\rho^{(j)}, h^2 I), \quad \nabla_{\mathbf{x}} \log \hat{p}_\rho(\mathbf{x}) = \frac{1}{h^2} \sum_{j=1}^M w_j(\mathbf{x}) (\mathbf{x}_\rho^{(j)} - \mathbf{x}), \quad (20)$$

with $w_j(\mathbf{x}) \propto \exp(-\|\mathbf{x}_\rho^{(j)} - \mathbf{x}\|^2/2h^2)$ normalized to sum to one.

Push-T. For Push-T the kernel score is computed over the pusher position only, in pixel coordinates ($\mathbb{R}^2$, bandwidth $h = 32$ px), from the training episodes’ pusher positions at each point of a 128-point phase grid. It is a direction in the same space as the pusher phase velocity that the policy predicts, and it depends only on inputs of the policy, as Theorem 5 requires.

Bandwidth. A fixed bandwidth attenuates the score where the demonstrations are narrow: for a Gaussian marginal of spread σ , the kernel score is $\sigma^2/(\sigma^2 + h^2)$ times the true score (Realizability, in the proof of Theorem 5). The score enters $\mathcal{L}_{reg}$ only through its axis and the weight $\lambda\|\mathbf{s}\|$ (Theorem 5), so for a locally Gaussian marginal this attenuation lowers the effective weight in narrow regions without changing the axis of the correction. This is why we accept a fixed-bandwidth KDE for the score but not for the divergence targets of App. E, which enter the loss as regression targets.

Extending the Score to SE(3). For M demonstrations and N phase steps it costs $\mathcal{O}(NM^2)$ pairwise log-map evaluations. It is computed once before training, and no policy needs it at inference. The estimator carries over to SE(3) unchanged, with six-dimensional twists in place of three-dimensional ones. A direct implementation costs a small constant factor more per pair: $1.4\text{--}1.7\times$ on a GPU and $2.6\text{--}3.8\times$ on a CPU (Figure 9). The larger cost is statistical. Even with the best bandwidth, the mean squared error of a kernel estimate of a density gradient decays as $M^{-4/(D+6)}$ in D dimensions [44]. Matching the accuracy that 10^3 demonstrations give on SE(2) ($D = 3$) therefore takes on the order of 10^4 on SE(3) ($D = 6$), and the number of pairs grows about a hundredfold. Even so, the precompute stays under a second on a single GPU (Figure 9). How much of this statistical cost applies depends on how many of the added degrees of freedom the demonstrations actually vary in. It can be reduced by summing only over the k nearest neighbors, which brings the kernel evaluations down to $\mathcal{O}(NMk)$, or by replacing the kernel estimate with a learned score model.

D Planner Details

The planner solves for the optimal trajectory using Levenberg–Marquardt optimization with the following objective:

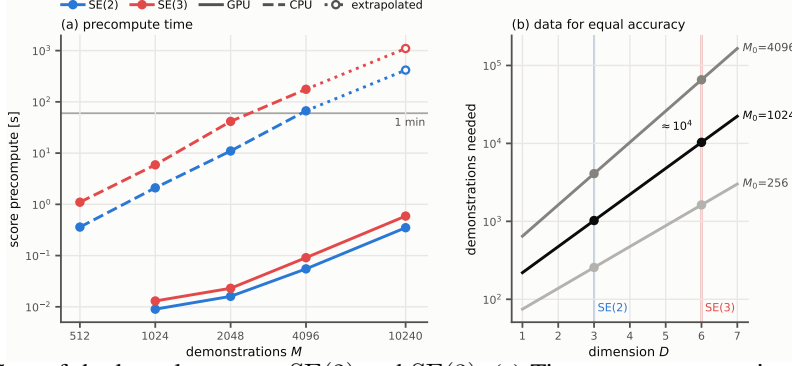


Figure 9: Cost of the kernel score on SE(2) and SE(3). (a) Time to precompute it over 51 phase steps against the number of demonstrations M (GPU: RTX 4090, float32, fused kernels; CPU: 16 threads, float64). Extrapolated points use the per-pair cost at $M = 4096$ and are lower bounds. (b) Demonstrations needed in D dimensions to match the accuracy of M_0 demonstrations on SE(2), $M_0^{(D+6)/9}$, from the $M^{-4/(D+6)}$ rate of kernel density gradient estimates [44]; an order of magnitude, not a prediction.

$$\begin{aligned}
\min_{\{T_i\}_0^N, \{u_i\}_0^{N-1}} & \frac{1}{2} \left[w_d \sum_{i=0}^{N-1} \left\| \text{Log}(T_i^{-1} T_{i+1}) - \Delta t [v_i, 0, \omega_i]^\top \right\|^2 \right. \\
& + w_c \sum_{i=1}^N \max(0, \epsilon - d(\mathbf{p}_i))^2 + w_l \sum_i \|h(u_i)\|^2 + w_s \sum_{i=0}^{N-2} \|u_{i+1} - u_i\|^2 \\
& \left. + w_e \sum_i \|u_i\|^2 + w_b (\|\text{Log}(T_s^{-1} T_0)\|^2 + \|\mathbf{p}_N - \mathbf{p}_g\|^2) \right] \quad (21)
\end{aligned}$$

Weights were configured as $w_d = 2000$, $w_c = 60$, $w_l = 60$, $w_s = 2$, $w_e = 0.05$, and $w_b = 10^4$. The collision margin ϵ is 0.45 m (comprising a 0.3 m robot radius and 0.15 m clearance), and d is a 0.1 m grid signed distance field (SDF) of the obstacles. The control penalty h acts as a hinge on the velocity limits $v \in [0, 2]$ m/s and $|\omega| \leq 2$ rad/s, and lateral slip is strictly prevented by the zero in the twist vector $[v, 0, \omega]^\top$.

E Divergence Estimation Challenges and Regularization Ablation

We found that divergence-based regularization hurt model performance rather than improving it, while score-based regularization significantly improved it (Fig. 10). We theorize this is because divergence is difficult to compute from sparse training data, especially on higher dimensional manifolds like SE(2) and SE(3). Huang et al. [38] regularize flow matching with conditional divergence matching (CDM), which penalizes the mismatch between the divergence of the conditional target flow and that of the learned field, together with a score-alignment term. Their target divergence is available in closed form, which is not the case for demonstration data. In our ablations, we tested three versions of this loss:

$$\mathcal{L} = \mathcal{L}_{\text{BC}} + \lambda \mathcal{L}_{\text{reg}}, \quad \mathcal{L}_{\text{reg}} \in \{\mathbb{E}|e_{\text{div}} + e_{\text{score}}| \text{ (CDM)}, \mathbb{E}|e_{\text{div}}| \text{ (div)}, \mathbb{E}|e_{\text{score}}| \text{ (score)}\} \quad (22)$$

where $e_{\text{div}} = \widehat{\nabla \cdot \mathbf{u}_\rho}(\mathbf{x}_\rho) - \nabla \cdot \pi_\phi(\mathbf{x}_\rho, \rho)$, with $\widehat{\nabla \cdot \mathbf{u}_\rho}$ the kNN estimate of the demonstrations' divergence described below, and $e_{\text{score}} = (\mathbf{u}_\rho - \pi_\phi(\mathbf{x}_\rho, \rho)) \cdot \nabla_{\mathbf{x}} \log \hat{p}_\rho(\mathbf{x}_\rho)$ is the integrand of Eq. 4 (the units align without rescaling: twists are in m/s and rad/s, divergence in 1/s, and the score in 1/m and 1/rad).

The demonstration data clearly shows that the score (Figs. 11 and 12) and the divergence (Fig. 11) evolve over phase. While accurately estimating these quantities improves flow matching perfor-

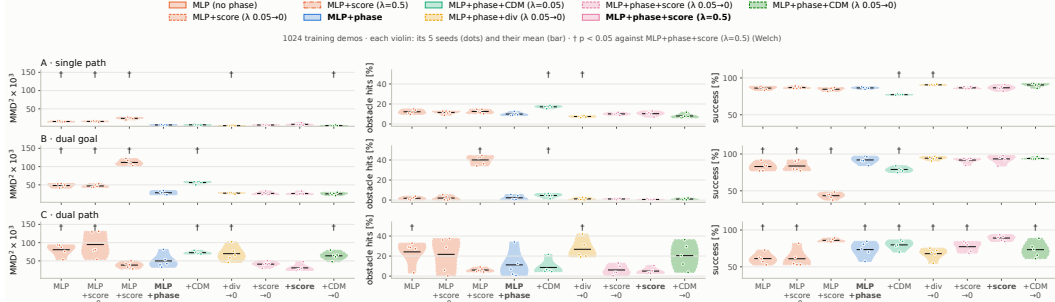


Figure 10: Regularizer ablation on the three navigation scenarios (1024 training demonstrations, 5 seeds per variant; † marks $p < 0.05$ against MLP+phase+score, $\lambda = 0.5$, Welch over seeds). Score-based regularizers are the only ones that improve performance significantly; divergence-based regularizers are neutral or significantly hurt performance. Labels such as $\lambda 0.05 \rightarrow 0$ denote a weight annealed to zero over training.

mance [38], obtaining an unbiased divergence estimate—which measures the rate of particle expansion and contraction in the flow field [45]—proved challenging.

For the *demonstration data*, we locally approximated divergence by tracking the distance r_k to the 16th nearest neighbor (kNN) among all demonstrations at the same phase step. Distances were measured using a left-invariant $SE(2)$ metric, defined as $d = |\text{Log}(T_a^{-1}T_b)|$ at 1 m/rad. Empirically, this kNN approach significantly outperformed a fixed-bandwidth Kernel Density Estimator (KDE) by avoiding the attenuation of narrow distributions. We estimated the log-volume derivative as $\nabla \cdot u \approx D \frac{d}{dt} \log r_k$ (with $D = 3$), computing the derivative via a Savitzky–Golay filter [46]. The filter used a 9-sample (1.35 s) window and a 2nd-order polynomial, evaluated at $dt = 0.15$ s to match the planner’s 6.67 Hz control step. For the demonstration score, the Gaussian KDE bandwidth was set to $h = 0.2$, calculated in each pose’s body tangent space.

Conversely, for the *policy divergence*, $\nabla \cdot \pi_\phi(\mathbf{x}_\rho, \rho)$, we avoided stochastic trace estimators [47, 48] altogether. We efficiently computed the exact trace using reverse-mode vector-Jacobian products, summing the partial derivatives $\cos \theta \partial_x v + \sin \theta \partial_y v + \partial_\theta \omega$ evaluated at a zero perturbation $T \text{Exp}(\epsilon)$.

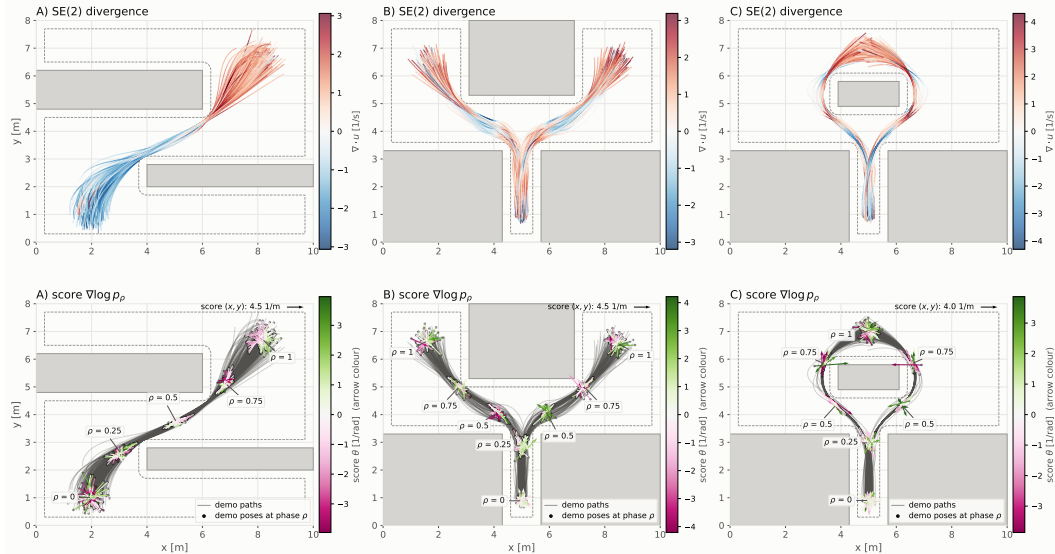
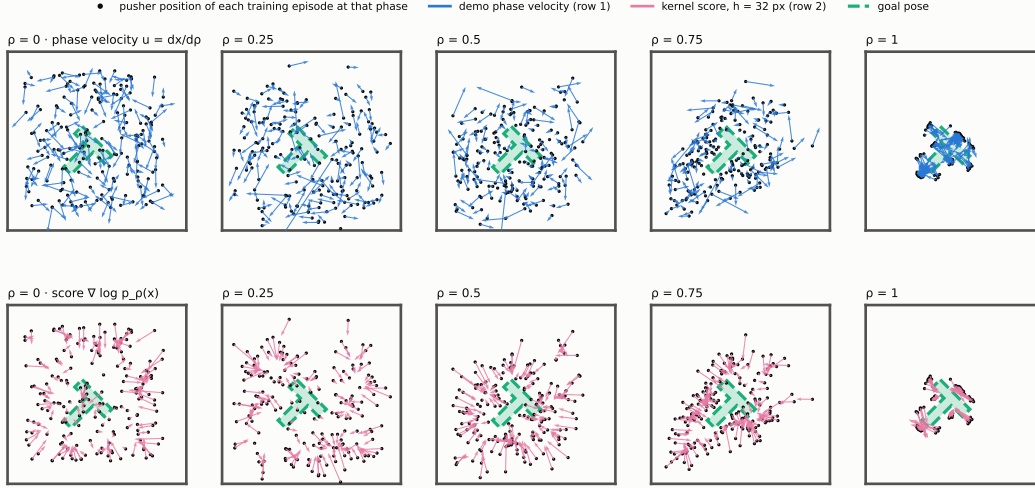


Figure 11: Estimated SE(2) divergence (top; kNN, $k = 16$) and kernel score (bottom; Gaussian KDE, $h = 0.2$) of the robot navigation demonstrations for scenarios A, B, and C, illustrating how the flow and density evolve over phase.



Arrow lengths are scaled per panel (the 90th-percentile arrow is 60 px), so they show direction and relative size within a panel only.

Figure 12: Demonstration phase velocities and corresponding kernel scores for the Push-T dataset. Snapshots at distinct temporal phases ($\rho \in \{0, 0.25, 0.5, 0.75, 1\}$) show the local modes of the spatial pusher distribution converging toward the final goal pose.

Both the *kNN* and *KDE divergence estimators* derive from the continuity equation [45]. Along any particle trajectory x_t in a flow u , the divergence equals the negative rate of change of the log-density: $\nabla \cdot u(x_t, t) = -\frac{d}{dt} \log p(x_t, t)$. We compare two density estimates evaluated across demonstrations at the same phase step, using a left-invariant SE(2) metric ($d(T_a, T_b) = \|\text{Log}(T_a^{-1}T_b)\|$, at 1 m/rad):

- **KDE:** A fixed-bandwidth Gaussian kernel density estimate [49] (leave-one-out), yielding $\widehat{\nabla \cdot u} = -\frac{d}{dt} \log \hat{p}_h$.
- **kNN:** The k -nearest-neighbor density [50] $\hat{p} \propto k/(Mr_k^D)$, where r_k is the distance to the k -th neighbor and $D = 3$. This yields $\widehat{\nabla \cdot u} = D \frac{d}{dt} \log r_k$.

Both apply the same Savitzky–Golay derivative [46] (window 9, order 2), isolating the density estimator as the sole variable.

Synthetic flows of known divergence. To assess accuracy against ground truth, we generate SE(2) particle flows that are linear in exponential coordinates: $T_i(t) = c \text{Exp}(\xi_i(t))$ with $\xi_i(t) = e^{Rt} \xi_i(0)$, where $R = \text{diag}(r_x, r_y, r_\theta)$. The divergence of this flow is exactly $\text{tr } R$ at every particle and time, neglecting a minor $O(\theta^2/12)$ Haar-measure correction [51]. We use $M = 1024$ particles with initial spreads (0.3 m, 0.3 m, 0.15 rad), matching the start distribution of scenario A.

Fig. 13(a) tracks a single contracting direction against its spread σ . A fixed-bandwidth KDE mathematically attenuates the signal; for a Gaussian marginal, it recovers only a fraction $(1 + 2u^2)/(1 + u^2)^2$ of the true divergence, where $u = h/\sigma$. Our measured KDE curves perfectly match this closed-form attenuation. This shrinkage severely impacts our data: at scenario A’s narrowest point (minor-axis spread 0.025 m, at $\rho = 0.64$), KDE with $h = 0.2$ captures just 3% of the true compression. Conversely, kNN adapts its length scale to local sample spacing. With $k = 16$, kNN recovers 61% at this narrowest point and reaches 95% by $\sigma = 0.08$ m (the start spread of scenarios B and C, where KDE $h = 0.2$ only recovers 26%).

Fig. 13(b) plots root-mean-square bias against per-sample noise over four flow types: isotropic expansion ($+1.2 \text{ s}^{-1}$), y -compression (-0.4), heading compression (-0.4), and area-preserving strain (0). Every $k \in \{2, \dots, 64\}$ yields a bias below 0.06 s^{-1} , where increasing k smoothly trades slight bias for lower noise. KDE cannot match this: its lowest bias (0.085 s^{-1} at $h = 0.1$) suffers twice the noise of kNN at $k = 16$. At $h = 0.2$, KDE bias jumps to 0.155 s^{-1} versus kNN’s 0.018 s^{-1} for

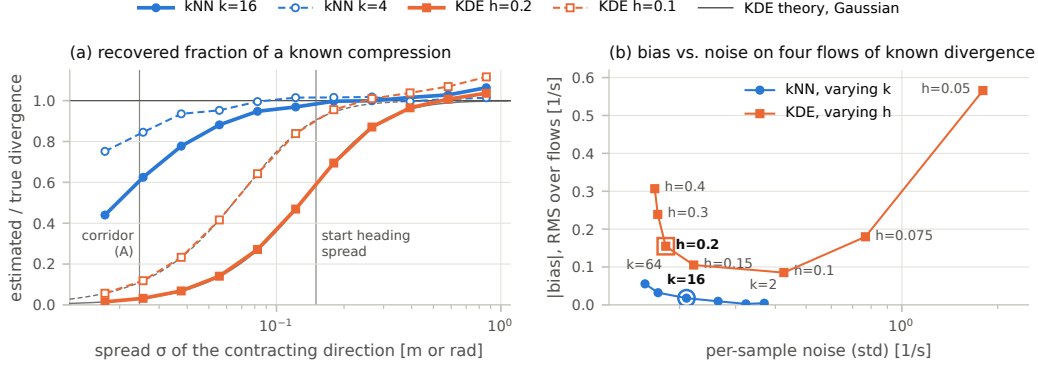


Figure 13: Divergence estimators on synthetic SE(2) flows of known divergence. (a) Fraction of a known compression recovered, against the spread of the contracting direction. Gray lines show the closed-form attenuation of a Gaussian KDE; vertical lines mark scenario A’s narrowest corridor and its start-heading spread. (b) RMS bias over four flows against per-sample noise, sweeping k (kNN) and h (KDE); the circled points are the settings shown in (a) and in Fig. 14.

similar noise levels. Bias is a critical flaw for a regression target: while zero-mean noise averages out over a minibatch, bias permanently shifts the target.

We select $k = 16$ as the optimal balance. Smaller values ($k = 4$) reduce attenuation but increase per-sample noise by 50% (0.32 vs. 0.21 s^{-1}). Furthermore, k must remain well below the per-mode demonstration count so that neighbors belong to the same mode; with $n = 1024$ demonstrations (512 per mode in B and C), $k = 16$ is far below that limit.

Estimates on the navigation datasets. Fig. 14 confirms these behaviors on the navigation datasets. The top row tracks mean divergence over phase, while the bottom row shows the spread across demonstrations within a mode. Both kNN settings agree closely, whereas KDE estimates artificially flatten toward zero.

This attenuation is most severe in scenarios B and C, where the initial distribution (0.08 m , 0.05 rad) is much narrower than the KDE kernel. In scenario C ($\rho = 0.3$), kNN detects a strong compression (-2.2 s^{-1}) that KDE $h = 0.2$ almost entirely misses ($+0.07 \text{ s}^{-1}$). At $\rho = 0.2$, the estimators even disagree on sign (-0.86 vs. $+0.77 \text{ s}^{-1}$). KDE’s artificially small spread across demonstrations reflects this shrinkage toward zero, not statistical precision. Relying on the synthetic ground-truth benchmark, we confidently adopt kNN ($k = 16$) for all divergence targets.

Even with the lower-bias kNN targets, divergence regularization failed to improve model performance and often degraded it. The main difference between our divergence regularization, which fails, and that of Huang et al. [38], which succeeds, is how the target is obtained (kNN estimation versus a closed-form expression), so we hypothesize that the per-sample noise of the estimated targets degrades the supervisory signal enough to hurt performance. We therefore regularize our models with the score term only, which significantly improved performance (Fig. 10).

F Policy, Task, and Normalization Details

Overview: For *robot navigation*, we evaluate a 134K-parameter version of our MLP policy on three scenarios: A) single start, single goal, single path, B) single start, dual goal, dual path, and C) single start, single goal, dual path (Fig. 3). The policy maps an augmented state (including phase) to a body twist, strictly satisfying non-holonomic kinematic constraints. Collision-free demonstrations were generated using Theseus [52]. During rollouts, phase came from the linear clock of Sec. 4. We simulated real-world execution by holding controls constant over integration steps and clamping outputs to the planner’s operational bounds. Planner details are in App. D.

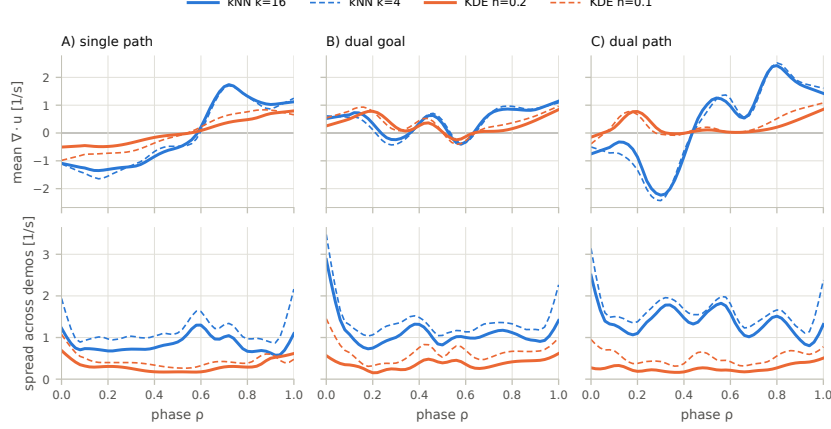


Figure 14: Divergence estimators on the navigation demonstrations. Top: mean demonstration divergence over phase (per-mode means, averaged). Bottom: spread across demonstrations within a mode. KDE estimates are flattened toward zero wherever the demonstration distribution is narrower than the kernel, most visibly in B and C.

For *Push-T*, we evaluate four policies trained on the human demonstrations of Chi et al. [2]: two MLPs, MLP small (0.79M params) and MLP-med wide (1.78M), and two Phaseformers, TF-med wide (1.79M) and TF-large (3.42M). Each is trained as four variants that toggle the phase input and the score regularizer (phase + score, phase, score, neither). The policies map the joint object-effector state and, in the phase variants, the current phase to the effector’s *phase velocity*; the MLPs take phase as an input and the Phaseformers inject it via AdaLN. Phase comes from an MLP estimator whose size depends on the policy. See below for network, estimator, normalization, and control details.

Robot Navigation: Demonstration trajectories consist of $N = 50$ steps with a time step of $\Delta t = 0.15$ s, yielding a 7.5 s horizon. The policy network is a 3×256 SiLU MLP (133,634 parameters). The input is formulated as $[(x - c_x)/s_x, (y - c_y)/s_y, \cos \theta, \sin \theta, 2\rho - 1]$. Positions are normalized to $[-1, 1]$ using a fixed affine map based on a 10×8 m workspace center $(c_x, c_y) = (5, 4)$ m and half-extents $(s_x, s_y) = (5, 4)$ m. Because this scaling is tied to the physical workspace, it remains consistent across all scenarios and seeds. Heading is parameterized as $(\cos \theta, \sin \theta)$ to avoid discontinuities at $\theta = \pm\pi$. The network outputs $[v, \omega]$ remain unscaled in m/s and rad/s, allowing the policy divergence $\nabla \cdot \pi_\phi$ to be evaluated directly in s^{-1} to match the divergence targets of App. E.

Push-T: Demonstrations consist of Diffusion Policy’s 206 human Push-T episodes [2] (25,650 steps recorded at 10 Hz, $\Delta t = 0.1$ s; episode lengths 49–246 steps, median 123), split 90/10 into 185 training and 21 held-out episodes. The state comprises the end-effector $\mathbf{x}_{ee} = [x_{ee} \ y_{ee}]$ and the object $\mathbf{x}_{obj} = [x_{obj} \ y_{obj} \ \theta_{obj}]$, min-max normalized over the training set, with θ_{obj} passed as $[\cos \theta_{obj} \ \sin \theta_{obj}]$.

The MLP backbones are SiLU networks with four hidden layers taking the concatenated input $[\mathbf{x}_{obj} \ \mathbf{x}_{ee} \ 2\rho - 1]$: 4×512 for MLP small (793,090 parameters) and 4×768 for MLP-med wide (1,779,458). The Phaseformer backbones have width 180 and four heads, with two blocks for TF-med wide (1,791,002 parameters) and four for TF-large (3,416,762). Each block applies self-attention, cross-attention, and a feed-forward layer, each preceded by an AdaLN layer whose shift, scale, and gate come from a sinusoidal embedding of ρ (Fig. 1). $\mathbf{x}_{ee}$ and $\mathbf{x}_{obj}$ are embedded as separate tokens that a single learned query reads via cross-attention, and an AdaLN-modulated linear decoder maps the query to the action. MLP-med wide and TF-med wide are parameter-matched, so comparing them isolates the backbone. The phase-free variants drop $2\rho - 1$ from the MLP input and drive the Phaseformer’s AdaLN with a constant embedding; parameter counts above are for the phase variants.

Both networks are trained via MSE against the demonstrations’ forward difference $\mathbf{u}_k = (L - 1)(\mathbf{x}_{ee,k+1} - \mathbf{x}_{ee,k})$ for an episode of L steps, i.e. the pusher’s phase velocity with $\rho_k = k/(L - 1)$. During rollouts, the commanded displacement is executed through the inverse of the environment’s PD controller and clipped to the simulator’s $[0, 512]^2$ limits; App. G gives the phase increment and integration step.

Phase estimator. MLP_{TP} is a ReLU MLP with a sigmoid output trained with an MSE loss to predict each demonstration’s normalized phase from the current state and the episode’s initial block pose, one estimator per policy seed. We use three sizes: 2×512 (0.27M parameters) for MLP small, 3×512 (0.53M) for MLP-med wide, and 3×724 (1.06M) for both Phaseformers. MLP small keeps the original 2×512 estimator; each larger policy uses the size, among 2×512 , 2×724 , 3×512 and 3×724 , under which its phase + score variant scored highest in closed loop. All sizes are trained identically. The estimator’s cost is included in the reported latency of every variant that uses phase; the phase-free variants run without it. We fit it on all demonstrations, including the validation episodes. The estimator is a way to give the policy a task-progress signal, not a claim that the phase regressor generalizes, so we let it fit the demonstrations closely. Every policy is evaluated in closed-loop simulator rollouts, not against held-out demonstrations.

Table 3: Push-T policies in Fig. 7 and their phase estimators. Parameter counts are for the phase variants; the phase-free variants run without an estimator. Each MLP_{TP} is a ReLU MLP with a sigmoid output, and the Phaseformers use four attention heads.

Model	Backbone	Params	MLP_{TP}	Params
MLP small	4×512 SiLU MLP	793,090	2×512	268,801
MLP-med wide	4×768 SiLU MLP	1,779,458	3×512	531,457
TF-med wide	Phaseformer, width 180, 2 blocks	1,791,002	3×724	1,058,489
TF-large	Phaseformer, width 180, 4 blocks	3,416,762	3×724	1,058,489

G Training Details

Robot Navigation: We train for 20,000 steps using Adam (no weight decay) with minibatches of 2048 samples drawn uniformly with replacement, cosine-annealing the learning rate from 10^{-3} to 10^{-5} . The regularizer weight is fixed at $\lambda = 0.5$. Final weights are evaluated directly without checkpoint selection.

Rollouts are performed from 1024 held-out start poses. MMD is computed using a Gaussian kernel ($\sigma = 0.3$) on $[x, y, \cos \theta, \sin \theta]$ across all $N + 1$ steps. Collisions are registered if the exact interpolated arc between steps breaches the 0.3 m robot radius against obstacles or boundaries. A success is recorded if a collision-free rollout terminates within $3\sigma_g$ of a goal mean. Methods are compared against MLP+phase+score ($\lambda = 0.5$) using Welch’s t -test across seeds ($p < 0.05$).

Push-T: We follow Diffusion Policy’s low-dimensional Push-T recipe [2] rather than the navigation step budget: 500 epochs over the 185 training episodes (23,034 windows, 90 minibatches of 256 per epoch, 45,000 steps) with AdamW ($\text{lr} = 10^{-4}$, $\beta = (0.95, 0.999)$, $\epsilon = 10^{-8}$, weight decay 10^{-6}), a cosine schedule with 500 warmup steps stepped every minibatch, and an exponential moving average of the weights (power 0.75, maximum decay 0.9999) updated every step. The regularizer weight is held constant at $\lambda = 2.5$ with a kernel bandwidth of 32 px; the score is precomputed once per training row as the kernel score of the training episodes’ pusher positions at that row’s phase on a 128-point phase grid. Both were selected on the validation loss of MLP small over $\lambda \in \{0.5, 2.5, 10\}$, anneal fraction $\in \{0.5, 1\}$ and bandwidth $\in \{16, 32, 64\}$ px, and $\lambda = 2.5$ was confirmed by a rollout sweep over $\lambda \in \{0.5, 2.5, 5, 10, 25\}$ on a smaller Phaseformer (two blocks of width 128, 0.91M parameters). The larger backbones reuse these settings without retuning. Unlike the navigation task, the evaluated weights are the EMA model at the lowest EMA validation loss; the validation loss is the phase-velocity MSE alone, so every variant is selected on the same quantity. Each variant is trained for 10 seeds.

Rollouts follow Diffusion Policy’s test protocol: 50 environments per seed (seeds 100000–100049), at most 300 steps each, giving 500 episodes per variant. The oracle phase is unavailable at test time, so ρ comes from the phase estimator (App. F): $\rho_t = \min\{1, \max(\rho_{t-1} + \varepsilon, \text{MLP}_{\text{TP}}(\mathcal{O}_t))\}$ with minimum increment $\varepsilon = 1/245$ (one step of the longest demonstration), while the integration step is held at the median demonstration’s $\Delta\rho = 1/(L_{\text{med}} - 1)$, $L_{\text{med}} = 123$. The commanded displacement $\mathbf{u} \Delta\rho$ is executed through the inverse of the environment’s PD controller ($k_p = 100$, $k_v = 20$) and clipped to the simulator’s $[0, 512]^2$ action space. We report the episode score $\max_t \text{clip}(\text{coverage}_t/0.95, 0, 1)$, the success rate (final goal coverage above 0.95, an IoU of about 0.90 against the goal pose), and the terminal position and orientation error of the block averaged within a seed. Methods are compared against TF-med wide’s phase + score variant using Welch’s t -test across seeds ($p < 0.05$).

Training Results: The training and evaluation losses as well as model selection epochs for robot navigation and for Push-T are shown in Figs. 15 and 16, respectively.

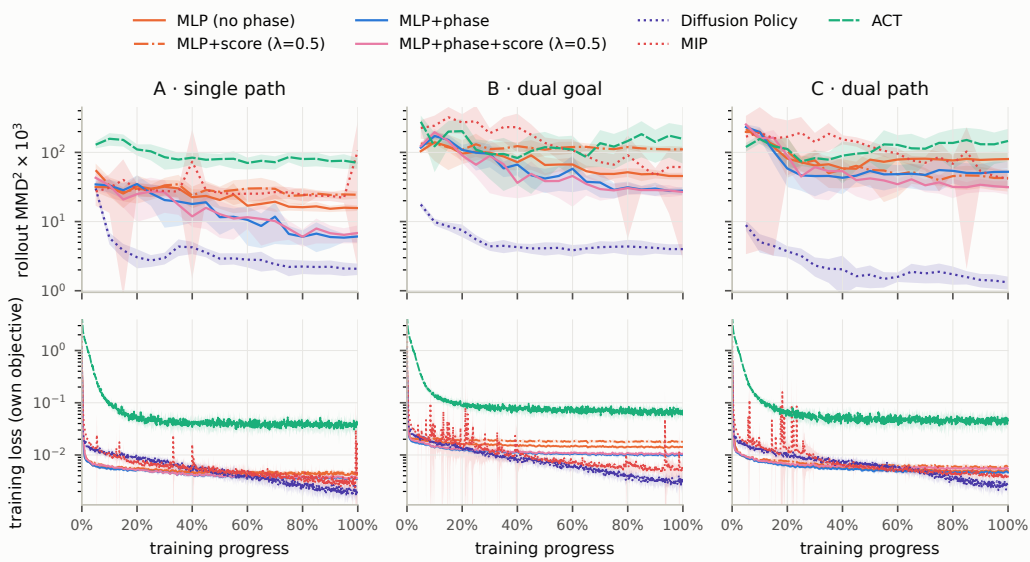


Figure 15: Robot Navigation Training: $n = 1024$ demonstrations, mean over 10 seeds (band: ± 1 s.d.). **Top:** squared MMD of the 1024 held-out rollouts against the held-out demonstrations, at the 20 evaluation points of each run’s schedule. Log scale on both rows. **Bottom:** each model’s own training objective: the variants’ velocity MSE (the regression term alone, without the regularizer), Diffusion Policy denoising MSE, MIP regression MSE, ACT L1 + 10 KL. The four objectives are in different units and are not comparable by height; only their shape is. The variants train for 20k steps and the generative baselines for 100k, so both axes are the fraction of each model’s own budget; the twenty evaluation points fall at the same fractions either way.

H Additional Results

Figure 17 shows more evaluations of the robot navigation task, specifically the MMD², % obstacle hits, and % success of the deterministic MLP policies compared to the stochastic baselines.

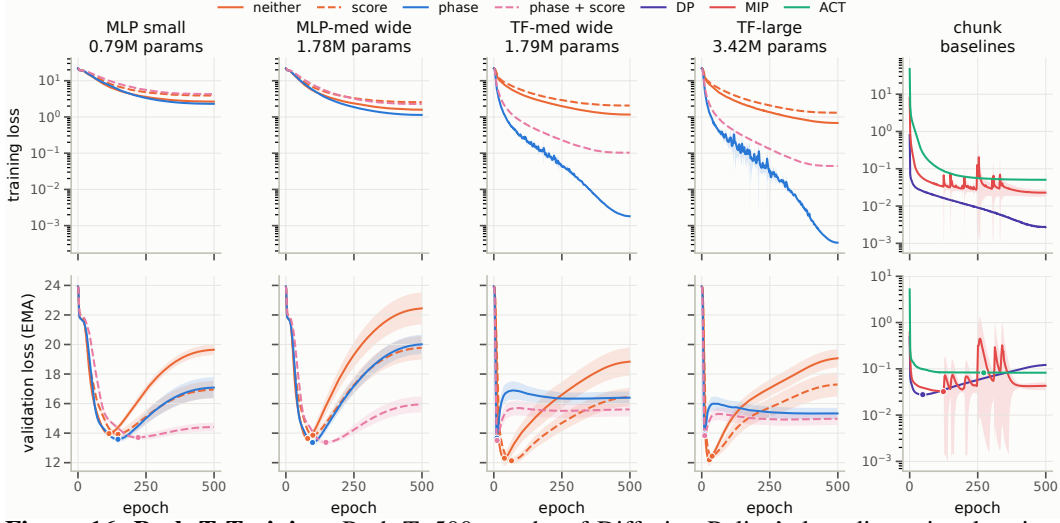


Figure 16: Push-T Training: Push-T, 500 epochs of Diffusion Policy’s low-dimensional recipe, mean over 10 seeds (band: ± 1 s.d.). **Top:** training loss. For the eight policy variants this is the phase-velocity MSE alone, so a score variant is comparable with the variant whose network it shares; for the chunk baselines it is their own objective (DP’s denoising MSE, MIP’s regression MSE, ACT’s L1 + 10 KL), which is neither the variants’ nor each other’s. **Bottom:** the EMA model’s validation loss, the quantity every variant’s checkpoint is selected on; the dot marks the selected epoch. The training row is log scale, as is the baselines’ validation panel; the variants’ validation loss is linear. The two backbone columns share their axes, so an variant and its transformer twin are read straight across; the baselines keep their own.

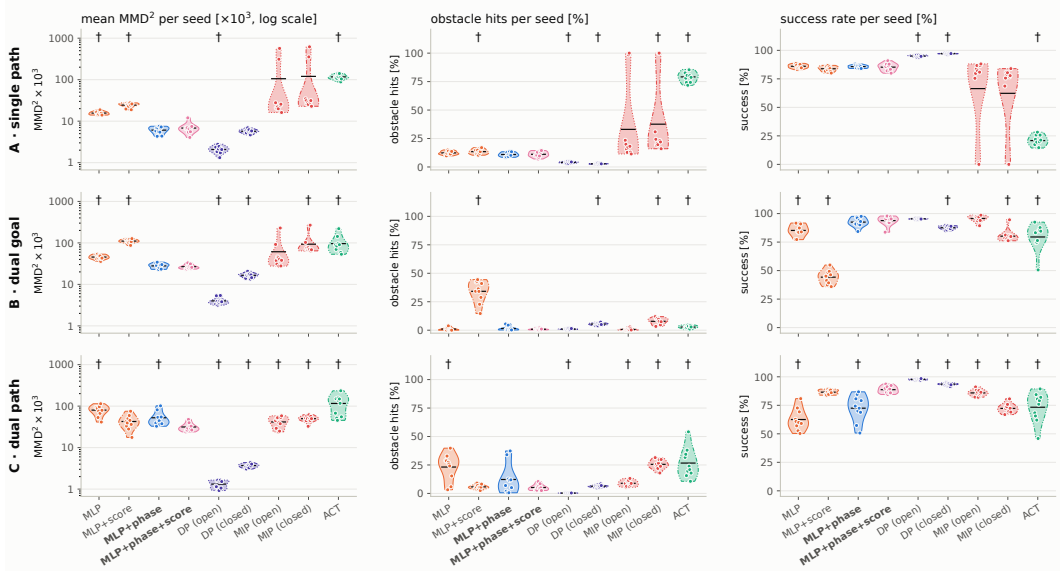


Figure 17: Robot Navigation Model Performance measured by MMD^2 , % obstacle hits, and % success. 1024 training demos; each violin shows the model’s 10 seeds (dots) and their mean (bar). Regularizer variants: $\lambda = 0.5$ held constant. The DP and MIP run open loop, as trained, and closed loop (temporal ensembling). ACT runs closed loop with temporal ensembling. † marks $p < 0.05$ against MLP+phase+score ($\lambda=0.5$) (Welch over seeds).