

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053

GEOMETRY-AWARE POLICY IMITATION

Anonymous authors

Paper under double-blind review

ABSTRACT

We propose a Geometry-aware Policy Imitation (GPI) approach that rethinks imitation learning by treating demonstrations as geometric curves rather than collections of state–action samples. From these curves, GPI derives distance fields that give rise to two complementary control primitives: a progression flow that advances along expert trajectories and an attraction flow that corrects deviations. Their combination defines a controllable, non-parametric vector field that directly guides robot behavior. This formulation decouples metric learning from policy synthesis, enabling modular adaptation across low-dimensional robot states and high-dimensional perceptual inputs. GPI naturally supports multimodality by preserving distinct demonstrations as separate models and allows efficient composition of new demonstrations through simple additions to the distance field. We evaluate GPI in simulation and on real robots across diverse tasks. Experiments show that GPI achieves higher success rates than diffusion-based policies while running 20x faster, requiring less memory, and remaining robust to perturbations. These results establish GPI as an efficient, interpretable, and scalable alternative to generative approaches for robotic imitation learning.

1 INTRODUCTION

Robots are increasingly expected to perform complex tasks in unstructured environments, ranging from dexterous manipulation to interactive collaboration. *Imitation learning* offers a promising path toward this goal, as it enables robots to acquire policies directly from expert demonstrations without relying on explicit dynamics models or simulation. Existing imitation approaches can be grouped into three families. *Explicit policies* treat imitation as supervised regression from states to actions (Calinon et al., 2007). They are fast at inference but struggle with multimodality and generalization. *Implicit policies* learn energy functions over state–action pairs (Florence et al., 2022), but are hard to train and slow to optimize at deployment. *Generative policies*, such as diffusion or flow-matching models (Chi et al., 2023; Lipman et al., 2023), excel at modeling multimodality but remain computationally heavy and brittle under distribution shifts. Despite their differences, all three approaches compress demonstrations into parametric models that must be retrained to incorporate new data and that often discard the geometric structure underlying expert behavior.

We argue that imitation learning can be made more direct, interpretable, and efficient by adopting a *geometric approach*. At its core, imitation means: (i) following the expert’s direction of motion, while (ii) approaching expert states as closely as possible. Viewed this way, a demonstration is not just a collection of samples but a *geometric curve* in state space, annotated with tangents that indicate expert actions. This perspective motivates our approach, **Geometry-Aware Policy Imitation (GPI)**. GPI represents demonstrations as *distance fields* that can be projected onto the robot’s actuated subspace, where control is applied. From these fields naturally emerge two complementary primitives: a *progression flow* that advances along expert trajectories, and an *attraction flow* that pulls current states toward them. Superimposing these flows defines a controllable vector field that drives imitation (Li & Calinon, 2025). This approach provides an approximation that reduces deviation while advancing along expert behaviors (Figure 1). In addition, the policy is guided by a distance field composition that retrieves flow fields from the most similar demonstrations, promoting coherent behavior and enabling robustness even under unknown dynamics.

A key strength of GPI is its *decoupling* of imitation into two modular components: (i) **metric learning**, which defines how states are represented and compared; and (ii) **behavior synthesis**, which constructs policies directly from distance and flow fields. This separation offers substantial flexibility: low-dimensional states can use Euclidean or geodesic distances, while high-dimensional obser-

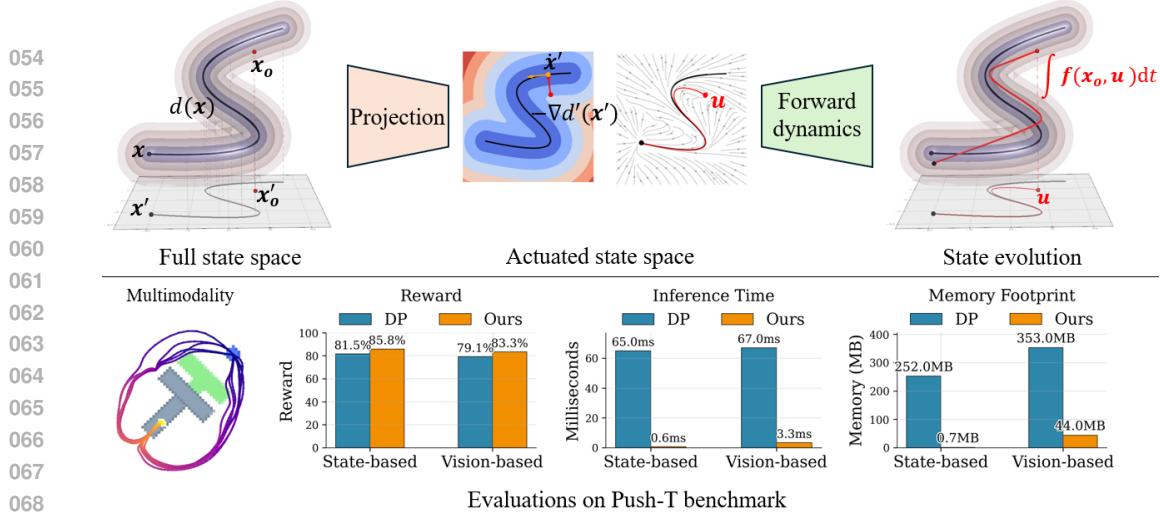


Figure 1: **Overview of Geometry-Aware Policy Imitation (GPI).** GPI treats demonstrations as geometric curves that induce distance fields in the full state space. **(Top)** The state space is projected onto the robot’s actuated subspace, where control is applied. The projected distance field gives rise to two complementary flows: an *attraction flow* from the negative gradient (red arrow) and a *progression flow* from trajectory tangents (yellow arrow). Together, they define a dynamical system that reduces the distance to demonstrations and advances along them, thus imitating expert behavior. The resulting action u is executed through the system’s dynamics, yielding state evolution $\int f(x, u) dt$ in the full state space. Multiple demonstrations can be composed naturally via Boolean operations on distance fields. Despite unknown system dynamics, the resulting trajectory aligns closely with the most similar demonstration as determined by the distance metric. **(Bottom)** On the PushT benchmark, GPI achieves multimodal imitation with a higher reward, runs 20–100× faster than diffusion policies (DDIM with 10 steps), and requires substantially less memory.

vations can rely on latent embeddings from pretrained or task-specific encoders. Policy synthesis itself is non-parametric and lightweight, enabling efficient composition of demonstrations without retraining and supporting multimodality by preserving distinct trajectories as separate flows (Pari et al., 2022). Moreover, because GPI only requires a state representation that supports distance computation, rather than directly fitting a full policy function, the learning problem is considerably simpler than in generative models. Lightweight encoders are typically sufficient, which reduces training complexity and enables fast inference at deployment.

We evaluate GPI extensively in both simulation and on real robots. In simulation, we benchmark across diverse domains—including planar pushing, 6-DoF manipulation, and dexterous hand control—with state spaces ranging from low-dimensional control vectors to raw vision inputs. For visual observations, we study multiple feature representations, from pretrained encoders to self-supervised embeddings. On real hardware, we demonstrate GPI on both a Franka arm and the Aloha bimanual system, showing that it scales robustly beyond controlled environments.

In summary, our contributions are:

- i) **Geometry-Aware Policy Imitation (GPI)**, which represents demonstrations as geometric curves that induce composable distance fields, providing a unified representation for both metric reasoning and action synthesis;
- ii) **A simple and modular formulation**, where state representation relies only on a suitable distance metric and action synthesis is realized through compositions of control primitives. Both components are lightweight, flexible, and grounded in well-studied principles;
- iii) **Extensive validation** in simulation and on real robots, showing that GPI achieves higher performance and enables efficient policy imitation—over 20× faster than state-of-the-art diffusion policies—while remaining interpretable and multimodal.

2 GEOMETRY-AWARE POLICY IMITATION

GPI constructs policies directly from demonstrations by representing them as geometric curves in state space. Each demonstration induces a distance field that encodes state similarity and gives rise to two complementary control primitives: (i) a *progression flow* that advances along demonstrated motions, and (ii) an *attraction flow* that corrects deviations by pulling states toward the trajectory.

108 Their superposition defines a dynamical system that imitates expert behavior. Local policies derived
 109 from individual demonstrations are then composed via distance-based weighting, producing a coher-
 110 ent global policy that is efficient, interpretable, and robust to perturbations. Figure 1-top illustrates
 111 these components schematically.
 112

113 **2.1 METHOD**

114 We are given N expert demonstrations $\mathcal{D} = \{\Gamma^{(i)}\}_{i=1}^N$, where each $\Gamma^{(i)}$ is a trajectory consisting of
 115 a sequence of states and actions

$$\Gamma^{(i)} = \{(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)})\}_{t=0}^{T_i}, \quad (1)$$

116 with states $\mathbf{x}_t^{(i)} \in \mathcal{X}$, actions $\mathbf{u}_t^{(i)} \in \mathcal{U}$, and horizon T_i .
 117

118 **State and actuated subspace.** A state \mathbf{x} may include both environment variables (e.g., object poses,
 119 images) that are unactuated, and robot variables that are directly actuated by control inputs. We
 120 denote by $\mathbf{x}' = P(\mathbf{x})$ the projection of \mathbf{x} onto the actuated subspace $\mathcal{X}' \subseteq \mathcal{X}$, where $P : \mathcal{X} \rightarrow \mathcal{X}'$
 121 is the projection operator. Each trajectory $\Gamma^{(i)}$ can then be viewed as a geometric curve in state
 122 space, which induces a *distance field* $d(\mathbf{x}_o \mid \Gamma^{(i)})$ measuring the proximity between a query state
 123 \mathbf{x}_o and the demonstration.
 124

125 **Action space.** We assume *velocity control* in the actuated subspace, i.e., $\mathbf{u}_t = \dot{\mathbf{x}}_t'$. Each demon-
 126 stration $\Gamma^{(i)}$ then defines a curve $\mathbf{x}_t^{(i)}$ whose actions $\mathbf{u}_t^{(i)}$ are the tangent directions in \mathcal{X}' . Velocity
 127 control is used here for clarity, but it is not a prerequisite: the formulation extends naturally to
 128 accelerations or torques, which can be executed through the robot’s kinematics or dynamics models.
 129

130 **Policy as flow field in actuated space.** From the distance field $d(\mathbf{x}_o \mid \Gamma^{(i)})$ induced by a demon-
 131 stration $\Gamma^{(i)}$, we derive two complementary flows in the actuated subspace: *Progression flow*, given
 132 by the demonstrated tangent action $\mathbf{u}_{\kappa(\mathbf{x}_o)}^{(i)} = \dot{\mathbf{x}}_{\kappa(\mathbf{x}_o)}^{(i)}$, which advances along the expert trajectory;
 133 and *Attraction flow*, obtained from the partial derivative of the distance field with respect to actuated
 134 coordinates, $-\nabla_{\mathbf{x}_o'} d(\mathbf{x}_o \mid \Gamma^{(i)})$, which corrects deviations by pulling states back toward demon-
 135 strations. Their superposition defines a policy in the actuated subspace:
 136

$$\pi_i(\mathbf{x}_o) = \lambda_1(\mathbf{x}_o) \mathbf{u}_{\kappa(\mathbf{x}_o)}^{(i)} - \lambda_2(\mathbf{x}_o) \nabla_{\mathbf{x}_o'} d(\mathbf{x}_o \mid \Gamma^{(i)}), \quad (2)$$

137 where $\kappa(\mathbf{x}_o) = \arg \min_t d(\mathbf{x}_o, \mathbf{x}_t^{(i)})$ denotes the nearest demonstrated state, and $\lambda_1, \lambda_2 \geq 0$ are
 138 weights—either constant or distance-dependent chosen so that attraction dominates far from demon-
 139 strations, while progression dominates near them. This policy has been shown to yield a stable first-
 140 order dynamical system that asymptotically converges to the demonstrated trajectory if the state
 141 and action variables are continuous (Li & Calinon, 2025)¹. This can be achieved by representing
 142 a discrete trajectory with continuous functions such as splines. Thus, the robot’s behavior remains
 143 robust, predictable, and safe even under environmental changes or perturbations.
 144

145 **Composition across demonstrations.** To obtain a global policy, we compose local flow-based
 146 policies across multiple demonstrations. Given the K nearest demonstrations, the global policy is
 147

$$\pi(\mathbf{x}_o) = \sum_{i=1}^K w_i(\mathbf{x}_o) \pi_i(\mathbf{x}_o), \quad w_i(\mathbf{x}_o) = \frac{\exp(-\beta d(\mathbf{x}_o \mid \Gamma^{(i)}))}{\sum_{j=1}^K \exp(-\beta d(\mathbf{x}_o \mid \Gamma^{(j)}))}, \quad (3)$$

148 where $\pi_i(\mathbf{x}_o)$ is the local policy induced by demonstration $\Gamma^{(i)}$, $d(\mathbf{x}_o \mid \Gamma^{(i)})$ is the distance from
 149 query state \mathbf{x}_o to the trajectory $\Gamma^{(i)}$, and $\beta > 0$ is a temperature parameter controlling the sharpness
 150 of selection. This distance-based composition ensures that flows are retrieved from the most relevant
 151 demonstrations, yielding coherent behavior even under unknown dynamics. A detailed description
 152 of GPI is provided in Algorithm 1 (Appendix B).
 153

154 **2.2 CHOICE OF DISTANCE METRIC**

155 A central design choice in GPI is the distance metric $d(\mathbf{x}_o \mid \Gamma^{(i)})$, which measures the similarity
 156 between a query state and a demonstration. The state naturally consists of two complementary
 157

¹See Appendix A for the proof.

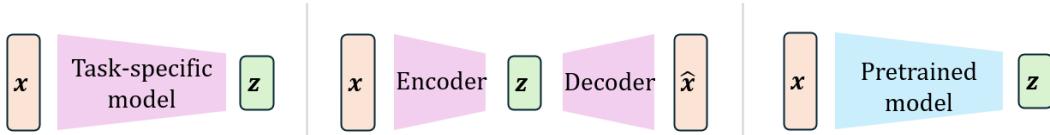


Figure 2: **Typical ways to obtain latent embedding z from raw inputs x .** (i) train a task-specific lightweight model to capture task-relevant features; (ii) use a VAE to learn task-agnostic features; or (iii) apply a pretrained model to obtain features without additional training.

parts: the robot-actuated variables (e.g., joint angles, end-effector pose) and the environment-related variables (e.g., object poses, images). Accordingly, the distance metric can be decomposed into a robot feature d_{rob} and an environment feature d_{env} , where the former also shapes the attraction flow in actuated space and the latter only influences demonstration selection and weighting.

Robot distance d_{rob} . For joint or end-effector positions $x \in \mathbb{R}^n$, Euclidean distance is standard:

$$d_{\text{Euc}}(x_1, x_2) = \|x_1 - x_2\|_2. \quad (4)$$

For end-effector orientations represented as quaternions, geodesic distances on S^3 respect rotational geometry:

$$d_{\text{quat}}(x_1, x_2) = 2 \arccos(|\langle x_1, x_2 \rangle|). \quad (5)$$

These two cases cover the most common representations in joint space and task space for robotics.

Environment distance d_{env} . This compares task-relevant but indirectly controllable variables, such as object poses or scene images. For low-dimensional object poses, d_{env} can be computed with Euclidean or geodesic distances, reusing the formulations above. For high-dimensional observations, it is common to define d_{env} in a latent space. Let $z = \Psi(x)$ denote the latent embedding of x . Then

$$d_{\text{env}}(x_1, x_2) = d_{\text{env}}(z_1, z_2), \quad (6)$$

where $z_1 = \Psi(x_1)$ and $z_2 = \Psi(x_2)$ are latent embeddings produced by a parametric model Ψ that maps raw observations to a latent space, and $d(\cdot, \cdot)$ denotes a suitable distance (e.g., Euclidean or cosine). This formulation supports multiple sources of embeddings: (i) task-specific models, where z could encode predicted object poses or desired robot actions learned via supervision; (ii) latent variables from variational autoencoders (VAEs) trained with self-supervised objectives (Kingma & Welling, 2013); and (iii) pretrained vision or multimodal encoders such as SAM (Kirillov et al., 2023), DINO (Siméoni et al., 2025), and CLIP (Radford et al., 2021), see Figure 2 for an overview. Classical dimensionality-reduction methods, such as principal component analysis (PCA), can also be used to obtain a compact latent feature (Hotelling, 1933).

While both d_{rob} and d_{env} contribute to the overall distance metric, their roles differ: d_{env} influences only the similarity ranking across demonstrations, whereas d_{rob} additionally shapes the attraction flow in the actuated subspace. This decomposition makes explicit how environmental features guide demonstration selection, while robot features govern the actual corrective control.

2.3 A 2D EXAMPLE

To illustrate GPI, we consider a simplified 2D setting where the state consists only of actuated variables x' . This abstraction is common in kinematic planning tasks, where environment dynamics are ignored. In this case, the distance field reduces to the robot-related term, $d(x_o) = d_{\text{rob}}(x'_o)$, so that state evolution and policy flows are fully contained in the same space. While prior work typically trains diffusion or flow-matching models for policy generation in this setting (Jiang et al., 2025), GPI instead addresses the problem in a fully non-parametric manner, relying directly on the distance and flow fields.

Figure 3(a) shows two demonstrations forming a Y-shaped pattern: $\Gamma^{(1)}$ (green) and $\Gamma^{(2)}$ (blue) overlap initially and then diverge into separate branches. Temporal progression is indicated by transparency from $t = 0$ to $t = 1$. Each demonstration induces a Euclidean distance field whose valleys align with its trajectory; composing them yields a global distance field (Figure 3b) visualized as an energy landscape with dense corridors along the demos and a natural decision boundary at the bifurcation. Figures 3(c,d) show the resulting flow fields: each row includes the single-demo flow (left) and the composed flow (both demos), with rollout trajectories overlaid on the energy landscape (right). Panel (c) depicts the progression flow, which follows the local tangent of the nearest

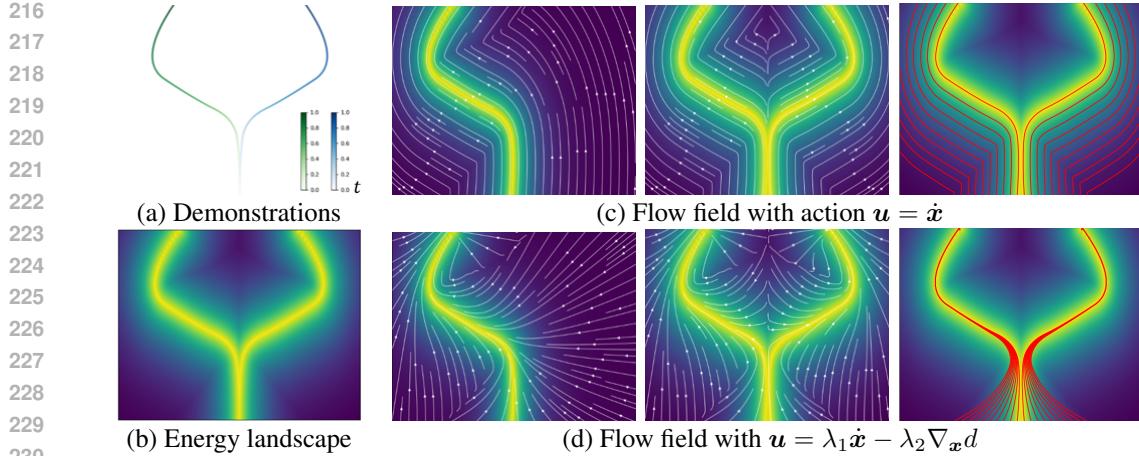


Figure 3: **From demonstrations to policy flows.** (a) Demonstrations. (b) Energy from composed distances. (c) Progression-only flow $u = \dot{x}$ may drift off the demonstrations. (d) Adding attraction $u = \lambda_1 \dot{x} - \lambda_2 \nabla_x d$ pulls states toward the demonstrations and along them, ensuring convergence.

demonstration; Panel (d) augments this with an attraction term that pulls states toward the trajectories, ensuring stable convergence. The rollout trajectories (red) show the integrated trajectories in two cases. From this perspective, diffusion policies perform well because their denoising steps implicitly induce an attraction flow toward demonstrations rather than relying solely on progression.

By representing demonstrations as distance and flow fields, policy imitation shifts from fitting a parametric model to geometric reasoning grounded in similarity, curvature, and composition, yielding several benefits: **Efficiency**—new demonstrations enrich the distance field by adding basins of attraction without retraining, and inference reduces to distance evaluations plus weighted averaging of expert actions, making it lightweight and parallelizable; **Flexibility**—decoupling similarity measurement from action synthesis keeps the framework modular, allowing task-specific distance metrics and flow compositions; **Multimodality**—each demonstration defines its own distance and flow field, preserving distinct behaviors so the policy branches smoothly toward the nearest demonstrated mode instead of averaging conflicting actions; **Interpretability**—the distance metric reveals which demonstrations influence the current action, while actions remain a linear superposition of demonstrated behaviors and corrective flows, ensuring safe, bounded outputs.

3 EXPERIMENTAL RESULTS

3.1 SIMULATION EXPERIMENTS

We first evaluate GPI on the PushT benchmark, a widely adopted task in which a robot must push a T-shaped object into a target configuration (Chi et al., 2023). This environment is particularly suitable for evaluation: it has well-established baselines for comparison, requires handling inherently multimodal pushing strategies, and involves contact-rich dynamics that cannot be solved by simple kinematic planning.

For state-based inputs, demonstrations consist of the agent position, the object position, and the object orientation. Distances are computed as a weighted combination of these components. The actuated subspace corresponds to the agent position, with its first-order derivative (velocity) serving as the action. Note that the original environment specifies actions in position control, which we adapt to velocity control for consistency with our flow-based formulation. Control policies are synthesized from the flow fields induced in the actuated subspace by corresponding demonstrations, and then executed in the environment with unknown interaction dynamics. For vision-based inputs, the state comprises the agent pose and an RGB image. Distances are computed jointly over the agent pose and an image embedding. To align with the state-based formulation, we train a lightweight task-specific model to produce the image embedding as the predicted object pose.

Experiments were conducted on an NVIDIA RTX 3090 GPU. Further details appear in Appendices C.1 and C.2. We report performance using three complementary metrics: (i) *Average / maximum reward*, evaluated over multiple random seeds and environment variations, following the same

270
271 Table 1: Performance comparison on Push-T (state-based vs. vision-based).
272
273

Method	Push-T (state-based)			Push-T (vision-based)		
	(Avg./Max.) score (%)	Training / Inference Time	Memory	(Avg./Max.) score (%)	Training / Inference Time	Memory
DDPM	82.3 / 86.3	1.0 h / 641 ms	252 MB	80.9 / 85.5	2.5 h / 647 ms	353 MB
DDIM	81.5 / 85.1	1.0 h / 65 ms	252 MB	79.1 / 83.1	2.5 h / 67 ms	353 MB
FMP	77.6 / 80.2	1.0 h / 58 ms	251 MB	75.1 / 79.3	2.5 h / 60 ms	352 MB
SFP	83.1 / 87.8	0.8 h / 51 ms	240 MB	77.5 / 81.2	2.0 h / 55 ms	341 MB
GPI (Ours)	85.8 / 89.0	0 h / 0.6 ms	0.7 MB	83.3 / 86.9	0.3 h / 3.3 ms	44 MB

274
275
276
277
278 protocol as the baselines; (ii) *time*, including training time and per-step inference time; and (iii) *memory footprint*, including memory cost for model parameters and stored demonstrations. We
279 compare GPI against Diffusion Policy (DP) (Chi et al., 2023), Flow Matching Policy (FMP) (Zhang
280 & Gienger, 2024), and Streaming Flow Policy (SFP) (Jiang et al., 2025). All baselines are
281 implemented using their official public codebases. For DP, we evaluate both a 100-step Denoising
282 Diffusion Probabilistic Model (DDPM) (Ho et al., 2020) and a 10-step Denoising Diffusion Implicit
283 Model (DDIM) (Song et al., 2021). Following standard practice, DP and FMP predict an H -step
284 action sequence ($H = 8$), whereas SFP and GPI naturally support reactive planning and operate
285 with a one-step horizon ($H = 1$). Results are summarized in Table 1. GPI achieves higher success
286 rates across all tasks than these baselines while being substantially more computationally efficient.
287

288 In the state-based setting, inference involves only low-dimensional, non-parametric distance evalua-
289 tions and flow field composition, resulting in a latency of 0.6 ms—nearly 100× faster than **diffusion- or flow matching-based baselines**. Although GPI requires storing all demonstrations for distance
290 measurement, the overall memory footprint remains lower than that of training large neural poli-
291 cies². Moreover, the underlying computations are lightweight and naturally parallelizable, further
292 contributing to its efficiency. For vision-based inputs, we employ a ResNet-18 encoder trained
293 solely for feature extraction rather than precise action prediction, which simplifies training and im-
294 proves efficiency. As a result, training completes in only 0.3 hours and inference runs at 3.3 ms per
295 step. Memory requirements are also reduced, since we store only the lightweight encoder and latent
296 embeddings of demonstrations rather than raw images or large policy networks. Additionally, this
297 modular structure allows the visual encoder to be reused across different tasks.
298

299 We further conduct a series of ablations to high-
300 light the distinctive properties of GPI:

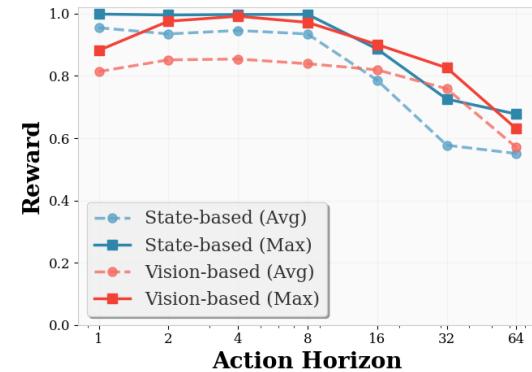
301 **Robustness.** We evaluate GPI’s robustness
302 along three complementary dimensions.

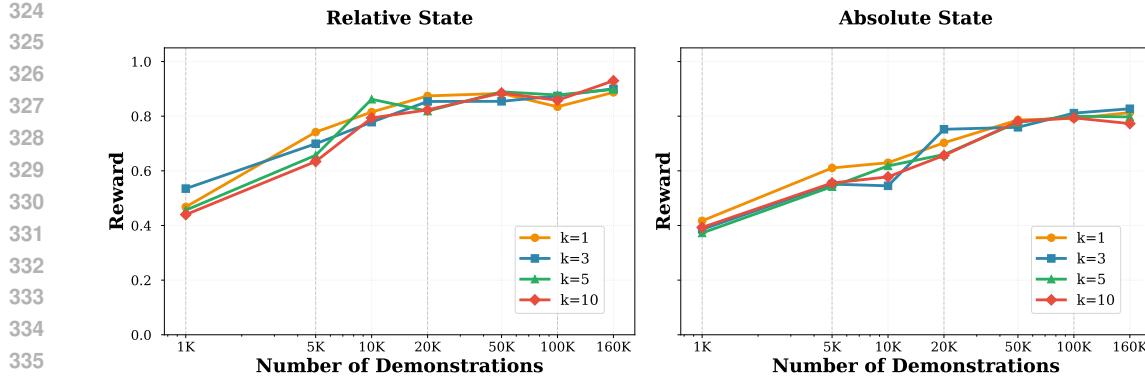
303 *Planning horizon:* GPI is reactive by default
304 ($H = 1$), but it can also be extended to a
305 receding-horizon scheme by updating the dis-
306 tance every H steps. As shown in Figure 4,
307 performance remains stable for horizons up to
308 16, showing GPI can operate either as a purely
309 reactive controller (robust to external dis-
310 turbances) or as a receding-horizon planner (with
311 improved temporal consistency).

312 *Number of neighbors:* In action composition,
313 we compare $K = 1, 3, 5, 10$. As shown in Fig-
314 ure 5, the curves are nearly overlapping in both relative and absolute state settings, confirming that
315 performance is largely insensitive to the choice of K . In addition, we also ablate the softmax tem-
316 perature β used in (3) for composing demonstra-
317 tions, and the results are reported in Appendix D.4.
318 This highlights the reliability of GPI’s local composition mechanism.

319 *State representation:* We compare object-centric (relative) and global (absolute) state formulations
320 (Figure 5). Both achieve strong performance, but relative states consistently yield slightly higher
321 scores, especially in data-scarce regimes. This suggests that GPI is robust to representation choices,
322 with relative states offering an advantage when demonstrations are limited.

323
324 ²See Appendix D.1 for a detailed explanation.

325
326 Figure 4: **Robustness to action horizons.**
327

Figure 5: Robustness of GPI with respect to demonstrations, K (neighbors), and state representations.

Distance metrics. We evaluate how the choice of distance metric and the relative weighting between robot and environment components affect performance; detailed ablations and results are reported in Appendix D.3.

Scalability with data sizes. A distinctive advantage of GPI is that, being non-parametric and training-free in the state-based setting, it enables direct study of how performance scales with the number of demonstrations, without the need for retraining. To this end, we augment the dataset with up to 160K samples regenerated from the original diffusion policy work and evaluate how performance evolves as the demonstration set grows. This setting is particularly suitable for GPI, since demonstration density directly influences both the distance query and the selection of actions in the composed policy. As shown in Figure 5, success rates increase consistently as the dataset expands from 1K to 20K demonstrations, after which performance begins to saturate. This trend reveals two key insights: (i) larger demonstration sets provide denser coverage of the state space, thereby reducing approximation errors introduced by the chosen distance metric, and (ii) our approach can serve as a practical diagnostic tool—indicating how many demonstrations are sufficient to achieve reliable policy performance before training parametric models. The method also accommodates incremental incorporation of new demonstrations, without the need for full retraining. **For a detailed analysis of latency and memory scalability as the number of demonstrations and the latent dimension increase, please refer to Appendix D.2.**

Stochasticity and multimodality. To induce stochasticity and multimodality, we inject Gaussian noise $\mathcal{N}(0, \sigma^2)$ into the query state in the actuated space (corresponding to the agent’s position). This perturbation alters the effective distance fields used in composition, thereby modifying the synthesized flow field and inducing multimodal behavior. In Figure 6, we compare the average score achieved under different noise levels. To quantify diversity, we measure the average distance among trajectories generated with different random perturbations sampled from the same noise distribution. The results show that larger noise values increase trajectory diversity but degrade performance, whereas smaller noise levels yield more deterministic behavior. Importantly, GPI exhibits multimodal behavior even under low noise (e.g., $\sigma = 0.2$), as illustrated in Figure 1 (bottom left). Beyond Gaussian perturbations, stochasticity can also be enhanced by randomly subsampling the set of demonstrations at each inference time. We found that this strategy can improve performance in practice, for instance, by helping the robot escape from regions where it would otherwise become stuck.

Natural composition of control primitives. We interpret progression and attraction as two basic control primitives that can be naturally combined within the flow field. By varying their rel-

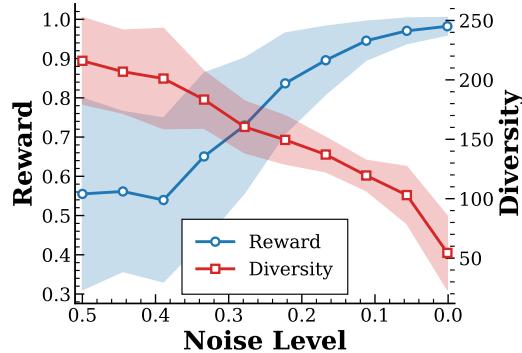


Figure 6: Noise-level ablations for score and diversity.

7

378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552
 553
 554
 555
 556
 557
 558
 559
 560
 561
 562
 563
 564
 565
 566
 567
 568
 569
 570
 571
 572
 573
 574
 575
 576
 577
 578
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 590
 591
 592
 593
 594
 595
 596
 597
 598
 599
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609
 610
 611
 612
 613
 614
 615
 616
 617
 618
 619
 620
 621
 622
 623
 624
 625
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 640
 641
 642
 643
 644
 645
 646
 647
 648
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 700
 701
 702
 703
 704
 705
 706
 707
 708
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 770
 771
 772
 773
 774
 775
 776
 777
 778
 779
 780
 781
 782
 783
 784
 785
 786
 787
 788
 789
 790
 791
 792
 793
 794
 795
 796
 797
 798
 799
 800
 801
 802
 803
 804
 805
 806
 807
 808
 809
 810
 811
 812
 813
 814
 815
 816
 817
 818
 819
 820
 821
 822
 823
 824
 825
 826
 827
 828
 829
 830
 831
 832
 833
 834
 835
 836
 837
 838
 839
 840
 841
 842
 843
 844
 845
 846
 847
 848
 849
 850
 851
 852
 853
 854
 855
 856
 857
 858
 859
 860
 861
 862
 863
 864
 865
 866
 867
 868
 869
 870
 871
 872
 873
 874
 875
 876
 877
 878
 879
 880
 881
 882
 883
 884
 885
 886
 887
 888
 889
 890
 891
 892
 893
 894
 895
 896
 897
 898
 899
 900
 901
 902
 903
 904
 905
 906
 907
 908
 909
 910
 911
 912
 913
 914
 915
 916
 917
 918
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 930
 931
 932
 933
 934
 935
 936
 937
 938
 939
 940
 941
 942
 943
 944
 945
 946
 947
 948
 949
 950
 951
 952
 953
 954
 955
 956
 957
 958
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968
 969
 970
 971
 972
 973
 974
 975
 976
 977
 978
 979
 980
 981
 982
 983
 984
 985
 986
 987
 988
 989
 990
 991
 992
 993
 994
 995
 996
 997
 998
 999
 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079
 1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1090
 1091
 1092
 1093
 1094
 1095
 1096
 1097
 1098
 1099
 1100
 1101
 1102
 1103
 1104
 1105
 1106
 1107
 1108
 1109
 1110
 1111
 1112
 1113
 1114
 1115
 1116
 1117
 1118
 1119
 1120
 1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1130
 1131
 1132
 1133
 1134
 1135
 1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1199
 1200
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1330
 1331
 1332
 1333
 1334
 1335
 1336
 1337
 1338
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1399
 1400
 1401
 1402
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1410
 1411
 1412
 1413
 1414
 1415
 1416
 1417
 1418
 1419
 1420
 1421
 1422
 1423
 1424
 1425
 1426
 1427
 1428
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447
 1448
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1560
 1561
 1562
 1563
 1564
 1565
 1566
 1567
 1568
 1569
 1570
 1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578
 1579
 1580
 1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619
 1620
 1621
 1622
 1623
 1624
 1625
 1626
 1627
 1628
 1629
 1630
 1631
 1632
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1640
 1641
 1642
 1643
 1644
 1645
 1646
 1647
 1648
 1649
 1650
 1651
 1652
 1653
 1654
 1655
 1656
 1657
 1658
 1659
 1660
 1661
 1662
 1663
 1664
 1665
 1666
 1667
 1668
 1669
 1670
 1671
 1672
 1673
 1674
 1675
 1676
 1677
 1678
 1679
 1680
 1681
 1682
 1683
 1684
 1685
 1686
 1687
 1688
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738
 1739
 1740
 1741
 1742
 1743
 1744
 1745
 1746
 1747
 1748
 1749
 1750
 1751
 1752
 1753
 1754
 1755
 1756
 1757
 1758
 1759
 1760
 1761
 1762
 1763
 1764
 1765
 1766
 1767
 1768
 1769
 1770
 1771
 1772
 1773
 1774
 1775
 1776
 1777
 1778
 1779
 1780
 1781
 1782
 1783
 1784
 1785
 1786
 1787
 1788
 1789
 1790
 1791
 1792
 1793
 1794
 1795
 1796
 1797
 1798
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1810
 1811
 1812
 1813
 1814
 1815
 1816
 1817
 1818
 1819
 1820
 1821
 1822
 1823
 1824
 1825
 1826
 1827
 1828
 1829
 1830
 1831
 1832
 1833
 1834
 1835
 1836
 1837
 1838
 1839
 1840
 1841
 1842
 1843
 1844
 1845
 1846
 1847
 1848
 1849
 1850
 1851
 1852
 1853
 1854
 1855
 1856
 1857
 1858
 1859
 1860
 1861
 1862
 1863
 1864
 1865
 1866
 1867
 1868
 1869
 1870

smoothly parameterize scene geometry, which is particularly well matched to GPI’s distance fields and flows, whereas BYOL emphasizes invariance to augmentations and may discard some of this geometric information. Finally, even an off-the-shelf pretrained SAM model within the GPI framework achieves a 41% average score without any fine-tuning. This variant underperforms our other encoders, likely due to sensitivity to segmentation quality and the downstream pose estimation module; we expect that task-specific fine-tuning would improve its performance.

3.2 ROBOT EXPERIMENTS

To further evaluate GPI, we conduct robot experiments on two challenging tasks:

(i) Box flip. The robot must flip a box by exploiting contacts among the end-effector, the box, and an aluminum crossbeam, which is challenging due to unknown, highly nonlinear dynamics. We collect 121 demonstrations on an ALOHA platform (Aladaco et al., 2024). The dataset contains over 50,000 RGB images and action pairs. A lightweight neural network takes a raw RGB image as input and predicts an action; this predicted action serves as the image embedding. Distances are computed jointly over the robot joint configuration and the action embedding to construct the distance field, from which the flow field is derived for the robot’s execution. We observe an inference time of approximately 7 ms and a memory footprint of 140 MB, comprising 139 MB for the feature-extraction model and 1 MB for storing latent features. In 50 flip trials, 39 are successful, corresponding to a 78% success rate. A trial is counted as successful if the robot flips the box to the target orientation within 500 control steps at 50 Hz. During these experiments, we also introduce occlusions and external disturbances; GPI still reliably completes the flip, indicating robustness to sensing and dynamics perturbations.

(ii) Human–robot fruit handover. A human hands fruit to the robot. The robot must execute a smooth, anticipatory interaction while synchronizing its timing with the human and remaining robust to unpredictable motions and sensing noise. This task is run on a Franka robot.

We collect a single demonstration to align the robot’s motion phase with the human hand trajectory. At execution time, a pretrained CLIP model (Radford et al., 2021) provides a fruit-detection score, which we combine with the deviation from the demonstrated hand trajectory to define the distance field. This field determines the robot’s phase and progression; the robot follows the progression flow until the desired phase is reached, yielding synchronized and fluid handovers. We observe 46 successful handovers out of 50 trials, resulting in a 92% success rate. A trial is counted as successful if the robot correctly recognizes the fruit and anticipates and adapts to the human hand motion to complete the delivery. During execution, we vary the object shapes and initial positions while still using only a single demonstration, and the policy consistently completes the task, demonstrating robustness and generalization.

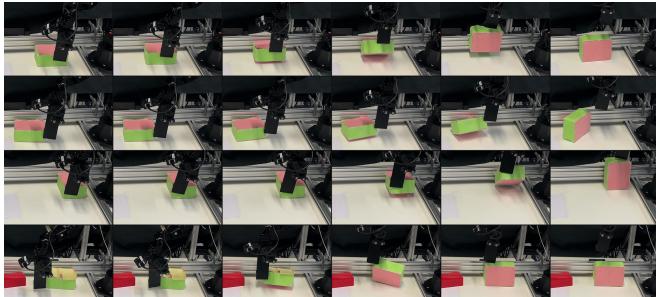


Figure 8: **Real-robot flipping task.** GPI successfully completes the task via multimodal behavior (Top 3 rows) and demonstrates robustness to visual disturbances (Bottom).

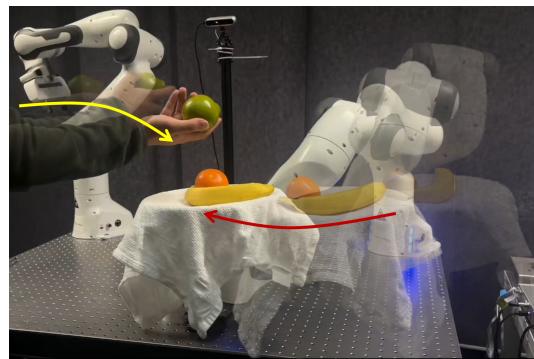


Figure 9: Real robot experiment on human-robot interaction task.

486 More details about the robot platform, experimental setup, and training details are illustrated in
 487 Appendices C.6 and C.7, respectively. The robot behavior during two tasks is shown in Figures 8, 9
 488 and the attached video.
 489

4 RELATED WORK

490 Among approaches to acquiring robotic skills—reinforcement learning (Sutton & Barto, 1998)
 491 and optimal control (Bertsekas, 1995), imitation learning (IL) (Osa et al., 2018) stands out for not
 492 requiring explicit task models or cost functions, making it especially appealing when dynamics
 493 are hard to model. Even when such models exist, demonstrations can accelerate and improve
 494 solutions (Nair et al., 2018; Razmjoo et al., 2021). Early approaches focus on time-dependent
 495 dynamical movement primitives, such as Dynamic movement primitives (DMP) (Ijspeert et al.,
 496 2013) and Probabilistic Movement Primitives (ProMP) (Paraschos et al., 2013), or time-independent
 497 dynamical systems (Khansari-Zadeh & Billard, 2011). They provide well-established approaches
 498 and efficient frameworks, but are usually limited in capturing complex, multi-modal demonstration
 499 patterns. Recent learning-based approaches, such as Implicit Behavior cloning and Diffusion policy,
 500 address this issue and have demonstrated impressive performance across a range of tasks (Florence
 501 et al., 2022; Chi et al., 2023; Zhang & Gienger, 2024). However, these methods introduce challenges
 502 such as hard to train, slow inference, and need multi-step inference (LeCun et al., 2006; Du &
 503 Mordatch, 2019; Song & Ermon, 2019; Nijkamp et al., 2020; Zhang & Gienger, 2024). GPI bridges
 504 dynamical systems and modern learning by representing demonstrations as distance fields—linking
 505 naturally to metric learning for high-level scene representations while inducing flow fields for
 506 low-level control. The closest prior, VENN (Pari et al., 2022), learns visual representations via
 507 self-supervision and retrieves policies with k NN, achieving strong visual imitation. In contrast, GPI
 508 supports diverse latent representations and synthesizes policy flows—demonstrating effectiveness
 509 on tasks with complex dynamics.
 510

5 LIMITATION AND CONCLUSION

511 We present Geometry-aware Policy Imitation (GPI), which treats demonstrations as geometric
 512 curves that induce a distance field and policy flows. This perspective yields a simple, flexible,
 513 efficient, multimodal, and interpretable policy that composes behaviors and integrates with diverse
 514 latent representations. Our approach has a few limitations that are worth exploring in future work:

515 **Choice of distance metric and representation.** The metric and visual representation are the key
 516 design levers that shape the induced flows. In this work, we rely on simple, manually specified met-
 517 rics and off-the-shelf encoders. Making these components learnable and co-optimizing them with
 518 policy synthesis—potentially conditioned on task or context—could further improve robustness and
 519 out-of-distribution generalization while preserving the geometric structure that makes GPI inter-
 520 pretable. Another promising direction is to leverage large models to provide task-relevant robotic
 521 features (Intelligence et al., 2025; Barreiros et al., 2025).

522 **Scene dynamics and stability.** Our current results follow the standard imitation learning paradigm:
 523 environment dynamics and unactuated components are treated as unknown, and policies are learned
 524 purely from data rather than from a full dynamics model. A natural extension is to incorporate known
 525 or learned dynamics models into the flow construction and analyze when the resulting closed loop
 526 is provably stable and robust, for example via Lyapunov or contraction certificates with perturbation
 527 and model-mismatch bounds. This could provide stronger guarantees in safety-critical settings.
 528 Our Lyapunov-style analysis assumes a smooth distance field in the actuated subspace, so it does
 529 not formally extend to discretized or non-smooth demonstrations, even though our time-discretized
 530 benchmarks remain empirically robust.
 531

532 **Scalability of demonstrations.** GPI stores only latent features and distances are computed in a
 533 single batched operation, leading to favorable latency and memory scaling in our empirical study.
 534 However, the memory footprint still grows linearly with the number of stored states. Future work
 535 could reduce this dependence via compact implicit distance parameterizations while preserving ge-
 536 ometric fidelity and fast retrieval.
 537

538
 539

540 REFERENCES
541

542 Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sankh Chan, Kenneth Draper, De-
543 bidatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, et al. Aloha 2: An enhanced
544 low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024. 9

545 Jose Barreiros, Andrew Beaulieu, Aditya Bhat, Rick Cory, Eric Cousineau, Hongkai Dai, Ching-
546 Hsin Fang, Kuniyuki Hashimoto, Muhammad Zubair Irshad, Masha Itkina, et al. A care-
547 ful examination of large behavior models for multitask dexterous manipulation. *arXiv preprint*
548 *arXiv:2507.05331*, 2025. 10

549 Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control, Volumes I and II*. Athena Sci-
550 entific, Belmont, MA, 1st edition, 1995. 10

552 Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a
553 task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cyber-
554 netics)*, 37(2):286–298, 2007. 1

555 Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitz-
556 mann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in
557 Neural Information Processing Systems*, 37:24081–24125, 2025. 8, 22

559 Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake,
560 and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The Inter-
561 national Journal of Robotics Research*, pp. 02783649241273668, 2023. 1, 5, 6, 10

563 Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. In
564 *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 10

565 Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian
566 Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In
567 *Conference on robot learning*, pp. 158–168. PMLR, 2022. 1, 10

569 Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena
570 Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar,
571 et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural
572 information processing systems*, 33:21271–21284, 2020. 8

573 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recog-
574 nition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.
575 770–778, 2016. 8

577 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in
578 neural information processing systems*, 33:6840–6851, 2020. 6

579 Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal
580 of Educational Psychology*, 24(6):417–441, 1933. 4

582 Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical
583 movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):
584 328–373, 2013. 10

586 Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess,
587 Adnan Esmail, Michael Equi, Chelsea Finn, Niccolò Fusai, et al. \pi_{\{0.5\}} a vision-language-
588 action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. 10

589 Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for
590 flexible behavior synthesis. In *International Conference on Machine Learning*, 2022. 8, 22

592 Sunshine Jiang, Xiaolin Fang, Nicholas Roy, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Sid-
593 dharth Ancha. Streaming flow policy: Simplifying diffusion / flow-matching policies by treating
action trajectories as flow trajectories. *arXiv preprint arXiv:2505.21851*, 2025. 4, 6

594 S Mohammad Khansari-Zadeh and Aude Billard. Learning stable nonlinear dynamical systems with
 595 Gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011. 10
 596

597 Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*
 598 *arXiv:1312.6114*, 2013. 4
 599

600 Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete
 601 Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023*
 602 *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3992–4003. IEEE Com-
 603 puter Society, 2023. 4, 8
 604

605 Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-
 606 based learning. *Predicting structured data*, 1(0), 2006. 10
 607

608 Y. Li and S. Calinon. From movement primitives to distance fields to dynamical systems. *IEEE*
 609 *Robotics and Automation Letters (RA-L)*, 2025. 1, 3
 610

611 Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching
 612 for generative modeling. In *11th International Conference on Learning Representations, ICLR*
 2023, 2023. 1
 613

614 Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-
 615 Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline
 616 human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.
 617 8
 618

619 Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and P. Abbeel. Overcoming
 620 exploration in reinforcement learning with demonstrations. *International Conference on Robotics*
 621 and *Automation (ICRA)*, pp. 6292–6299, 2018. 10
 622

623 Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent
 624 non-persistent short-run MCMC toward energy-based model. In *Advances in Neural Information*
 625 *Processing Systems (NeurIPS)*, volume 33, pp. 11588–11600, 2020. 10
 626

627 Takayuki Osa, Fabio Pardo, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters.
 628 An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):
 1–179, 2018. doi: 10.1561/2300000053. 10
 629

630 Alexandros Paraschos, Christian Daniel, Jan R Peters, and Gerhard Neumann. Probabilistic move-
 631 ment primitives. *Advances in neural information processing systems*, 26, 2013. 10
 632

633 Jyothish Pari, Nur Muhammad (Mahi) Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto.
 634 The Surprising Effectiveness of Representation Learning for Visual Imitation. In *Proceedings of*
 635 *Robotics: Science and Systems*, New York City, NY, USA, June 2022. doi: 10.15607/RSS.2022.
 636 XVIII.010. 2, 8, 10
 637

638 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agar-
 639 wal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya
 640 Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings*
 641 *of the 38th International Conference on Machine Learning*, volume 139 of *ICML*, pp. 8748–8763,
 2021. 4, 9
 642

643 Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel
 644 Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement
 645 learning and demonstrations. *Robotics: Science and Systems XIV*, 2018. 8
 646

647 A. Razmjoo, T. S. Lembono, and S. Calinon. Optimal control combining emulation and imitation
 648 to acquire physical assistance skills. In *20th International Conference on Advanced Robotics*
 (ICAR), pp. 338–343. IEEE, 2021. 10

648 Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose,
649 Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel
650 Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana,
651 Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé
652 Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL <https://arxiv.org/abs/2508.10104>. 4

653

654 Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 6

655

656 Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution.
657 In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 10

658

659 Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press,
660 Cambridge, MA, 1st edition, 1998. 10

661

662 Fan Zhang and Michael Gienger. Affordance-based robot manipulation with flow matching. *arXiv
663 preprint arXiv:2409.01083*, 2024. 6, 10

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

APPENDIX

A CONVERGENCE OF THE FLOW POLICY

We prove convergence of the policy introduced in Section 2.1, which combines progression and attraction flows to form a stable dynamical system in the actuated subspace. For clarity, we rewrite the flow policy (equation 2) as

$$\dot{\mathbf{x}} = \lambda_1 \dot{\mathbf{x}}_{t^*} - \lambda_2 \nabla d(\mathbf{x}), \quad (7)$$

where $d(\mathbf{x})$ is the distance to the demonstration, $\nabla d(\mathbf{x})$ its gradient, $\dot{\mathbf{x}}_{t^*}$ the tangent velocity at the projection point \mathbf{x}_{t^*} , and $\lambda_1, \lambda_2 \geq 0$ weight progression and attraction.

We analyze stability using the Lyapunov function

$$V(\mathbf{x}) = \frac{1}{2} d^2(\mathbf{x}) \geq 0, \quad (8)$$

which vanishes only on the demonstration. Its time derivative is

$$\dot{V}(\mathbf{x}) = d(\mathbf{x}) \nabla d(\mathbf{x})^\top \dot{\mathbf{x}}. \quad (9)$$

Substituting the dynamics gives

$$\dot{V}(\mathbf{x}) = d(\mathbf{x}) \nabla d(\mathbf{x})^\top (\lambda_1 \dot{\mathbf{x}}_{t^*} - \lambda_2 \nabla d(\mathbf{x})). \quad (10)$$

To simplify this expression, we use the fact that the projection point \mathbf{x}_{t^*} is defined as the minimizer of the squared distance

$$\|\mathbf{x}_{t^*} - \mathbf{x}\|^2. \quad (11)$$

At this minimizer, the derivative with respect to t must vanish:

$$(\mathbf{x}_{t^*} - \mathbf{x})^\top \dot{\mathbf{x}}_{t^*} = 0. \quad (12)$$

This condition implies that the displacement vector $\mathbf{x}_{t^*} - \mathbf{x}$, and therefore the gradient $\nabla d(\mathbf{x})$, is orthogonal to the trajectory tangent $\dot{\mathbf{x}}_{t^*}$:

$$\nabla d(\mathbf{x})^\top \dot{\mathbf{x}}_{t^*} = 0. \quad (13)$$

With this orthogonality property, the Lyapunov derivative reduces to

$$\dot{V}(\mathbf{x}) = -\lambda_2 d(\mathbf{x}) \|\nabla d(\mathbf{x})\|^2 \leq 0, \quad (14)$$

with equality only if $d(\mathbf{x}) = 0$. This shows that the system is globally stable and asymptotically converges to the demonstrated trajectory in the actuated space.

Assumptions and scope. This analysis is carried out for a continuous-time system in the actuated subspace, assuming a smooth demonstration trajectory and a well-defined, differentiable distance field around it. The result should therefore be interpreted as a convergence guarantee for this idealized setting. In practice, GPI is implemented in discrete time and with time-discretized demonstrations; the same flow construction is used, and we empirically observe stable rollouts in all benchmarks.

756 **B GPI ALGORITHM**
757758 **Algorithm 1** Geometry-Aware Policy Imitation

760 **Require:** $\mathcal{D} = \{\Gamma^{(i)}\}_{i=1}^N$, each $\Gamma^{(i)} = \{(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)})\}_{t=0}^{T_i}$; projection P ; encoder Ψ ; robot/environment
761 distances $d_{\text{rob}}, d_{\text{env}}$; mixing $\alpha_{\text{rob}}, \alpha_{\text{env}} > 0$; weights $\lambda_1(\cdot), \lambda_2(\cdot)$; temperature β ; top- K
762
763 **Ensure:** Control $\mathbf{u} \in \mathcal{X}'$ at query \mathbf{x}_o
764 1: $\mathbf{x}'_o \leftarrow P(\mathbf{x}_o), \mathbf{z}_o \leftarrow \Psi(\mathbf{x}_o)$
765 2: **for all** $i \in \{1, \dots, N\}$ (**parallel over demonstrations**) **do**
766 3: **Per-time distances**
767 $\mathbf{d}_{\text{rob}}^{(i)} \leftarrow (d_{\text{rob}}(\mathbf{x}'_o, \mathbf{x}'_t^{(i)}))_t, \mathbf{d}_{\text{env}}^{(i)} \leftarrow (d_{\text{env}}(\mathbf{z}_o, \Psi(\mathbf{x}_t^{(i)})))_t$
768 4: **Combined distance:** $\mathbf{d}^{(i)} \leftarrow \alpha_{\text{rob}} \mathbf{d}_{\text{rob}}^{(i)} + \alpha_{\text{env}} \mathbf{d}_{\text{env}}^{(i)}$
769 5: **Nearest time index and scalar distance:**
770 $\kappa^{(i)}(\mathbf{x}_o) \leftarrow \arg \min_t \mathbf{d}_t^{(i)}, \quad d(\mathbf{x}_o \mid \Gamma^{(i)}) \leftarrow \min_t \mathbf{d}_t^{(i)}$
771 6: **Progression flow:** $\mathbf{u}_{\kappa}^{(i)} \leftarrow \mathbf{u}_{\kappa^{(i)}(\mathbf{x}_o)}^{(i)} = \dot{\mathbf{x}}_{\kappa^{(i)}(\mathbf{x}_o)}^{(i)}$
772 7: **Attraction flow:** $\mathbf{u}_{\text{att}}^{(i)} \leftarrow -\nabla_{\mathbf{x}'_o} d_{\text{rob}}(\mathbf{x}'_o, \mathbf{x}'_{\kappa^{(i)}(\mathbf{x}_o)}^{(i)})$
773 8: **Local policy:**
774 $\pi_i(\mathbf{x}_o) \leftarrow \lambda_1(d(\mathbf{x}_o \mid \Gamma^{(i)})) \mathbf{u}_{\kappa}^{(i)} + \lambda_2(d(\mathbf{x}_o \mid \Gamma^{(i)})) \mathbf{u}_{\text{att}}^{(i)}$
775 9: **Top- K selection by demonstration distance:** $I_K \leftarrow$ indices of the K smallest $d(\mathbf{x}_o \mid \Gamma^{(i)})$
776 10: **Softmax weights over selected demos:** $w_i(\mathbf{x}_o) \leftarrow \frac{\exp(-\beta d(\mathbf{x}_o \mid \Gamma^{(i)}))}{\sum_{j \in I_K} \exp(-\beta d(\mathbf{x}_o \mid \Gamma^{(j)}))} \quad (i \in I_K)$
777 11: **Global policy:** $\mathbf{u} = \pi(\mathbf{x}_o) = \sum_{i \in I_K} w_i(\mathbf{x}_o) \pi_i(\mathbf{x}_o)$
778 12: **return** \mathbf{u}

785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

810 **C IMPLEMENTATION DETAILS**
 811

812 **C.1 PUSH T TASK WITH STATE-BASED INPUTS**
 813

814 For low-dimensional states, each demonstration is represented as

815 $\mathbf{x}_t^{(i)} = [x_a, y_a, x_b, y_b, \theta_b] \in \mathbb{R}^5$,
 816

817 where (x_a, y_a) denote the agent position, (x_b, y_b) the block position, and θ_b the block orientation.
 818 The associated action specifies the target location for a low-level controller:

819 $\mathbf{u}_t^{(i)} = [x_{\text{target}}, y_{\text{target}}]$,
 820

821 which we rewrite for velocity control as the relative displacement:
 822

823 $\mathbf{u}_t^{(i)} = [x_{\text{target}} - x_a, y_{\text{target}} - y_a]$.
 824

825 All state variables are normalized to $[0, 1]$ before computing distances. The distance field $d(\mathbf{x}, \Gamma^{(i)})$
 826 is defined as the weighted sum of three components:
 827

828 $d(\mathbf{x}, \mathbf{x}_t^{(i)}) = w_{\text{obj}} \|(x_b, y_b) - (x_b^{(i)}, y_b^{(i)})\|_2 + w_{\text{agt}} \|(x_a, y_a) - (x_a^{(i)}, y_a^{(i)})\|_2 + w_{\theta} \text{ang}(\theta_b, \theta_b^{(i)})$, (15)
 829 where $\text{ang}(\cdot, \cdot)$ denotes angular distance. Unless otherwise stated, the weights are set to $w_{\text{obj}} = w_{\text{agt}} = w_{\theta} = 1.0$.
 830

831 Each demonstration induces a distance field and an associated flow policy. At inference time, the
 832 global policy is formed by composing the K nearest demonstration policies, with $\lambda_1 = \lambda_2 = 1.0$.
 833 Evaluation is performed on environment seeds 500–510 using three distinct policy seeds.
 834

835 We further explore several variants to improve the flexibility of GPI:

836 **Relative vs. absolute state representation.** The PushT task involves nonlinear contact dynamics,
 837 so the choice of state representation is important. In the *relative* variant, the agent position is
 838 expressed in the object’s coordinate frame:
 839

840 $\tilde{\mathbf{p}}_a = R(-\theta_b) ((x_a, y_a) - (x_b, y_b))$, (16)
 841

842 where $R(-\theta_b)$ is the SE(2) rotation matrix aligning the block’s orientation to the x -axis. The
 843 demonstrated action \mathbf{u}_t is similarly transformed. During execution, the predicted action $\tilde{\mathbf{u}}$ is mapped
 844 back to global coordinates via the inverse transformation:
 845

846 $\mathbf{u} = R(\theta_b) \tilde{\mathbf{u}} + (x_b, y_b)$. (17)
 847

848 **Smooth flow fields.** When the action horizon is set to 1, the controller is highly reactive and may
 849 produce abrupt changes whenever the nearest demonstration switches. To mitigate this, we apply
 850 first-order smoothing to the action sequence:
 851

852 $\mathbf{u}_t^{\text{smooth}} = \alpha \mathbf{u}_t + (1 - \alpha) \mathbf{u}_{t-1}^{\text{smooth}}$, (18)
 853

854 where $\alpha \in [0, 1]$ is a smoothing parameter.
 855

856 **Recent-action suppression.** To mitigate oscillatory behavior arising from repeatedly selecting near-
 857 identical actions, we maintain a sliding-window memory \mathcal{M} of the most recent M actions. During
 858 action selection, if the candidate \mathbf{u}_t lies within a tolerance ϵ of any element in \mathcal{M} , it is suppressed
 859 and the next-best candidate from the composed policy is chosen. This mechanism enforces diversity
 860 over short horizons, prevents immediate backtracking to previously executed actions, and ensures
 861 the policy explores novel trajectories while preserving responsiveness.
 862

863 **Perturbed query states.** To evaluate robustness, we perturb the query agent position with additive
 864 Gaussian noise:
 865

866 $\tilde{\mathbf{x}}' = \mathbf{x}' + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I)$, (19)
 867

868 where $\mathbf{x}' = (x_a, y_a)$ is the agent substate. The noise variance σ^2 is annealed over time, decaying
 869 from $\sigma = 0.1$ at the beginning of execution to $\sigma = 0.001$ at later steps. This perturbation injects
 870 stochasticity into the query states, which increases variability in the retrieved flows and can induce
 871 multimodal behaviors.
 872

873 **Subsampled demonstrations.** For efficiency and robustness, instead of using all demonstrations,
 874 we randomly sample a subset $\Gamma_{\text{sub}} \subset \Gamma$ at each query. The global policy is then composed over Γ_{sub} .
 875 Empirically, we find that subsampling does not reduce performance; in some cases, the induced
 876 stochasticity even helps the agent escape undesirable cycles or “stacked” behaviors.
 877

864 C.2 PUSHT TASK WITH VISION-BASED INPUTS
865866 In the PushT environment, observations consist of an RGB image \mathbf{I} together with agent positions
867 (x_a, y_a) . Each demonstration state is represented as

868
$$\mathbf{x}_t^{(i)} = [x_a, y_a, \mathbf{I}].$$

869

870 **Vision encoder.** To obtain compact image features, we use an encoder ψ with a ResNet-18 backbone
871 (group normalization) and a projection head (MLP with sizes [512, 256, 128, 3]). The encoder is
872 trained with a mean squared error (MSE) loss to predict the object position and orientation:
873

874
$$\psi(\mathbf{I}) \approx [x_o, y_o, \theta_o], \quad \mathcal{L}_{\text{MSE}} = \frac{1}{B} \sum_{i=1}^B \|\mathbf{x}_{\text{pred}}^{(i)} - \mathbf{x}_{\text{target}}^{(i)}\|_2^2.$$

875

876 Training is performed for 200 epochs using the Adam optimizer with a learning rate of 0.001.
877878 **Distance metric and policy synthesis.** After training, each demonstration image is embedded as
879

880
$$\mathbf{z}_t^{(i)} = \psi(\mathbf{I}_t^{(i)}),$$

881

882 and for a query state $\mathbf{x}_o = [x_a, y_a, \mathbf{I}]$,

883
$$\mathbf{z}_o = \psi(\mathbf{I}).$$

884 Distances are defined in this learned feature space and policy synthesis then proceeds identically to
885 the state-based inputs.
886887 C.3 PUSHT TASK WITH RESNET-18 ENCODER AND PCA
888889 We construct a compact observation embedding by reusing the same ResNet-18 encoder from
890 the Diffusion Policy implementation (task-pretrained on PUSHT). At inference, this encoder is
891 frozen and used as a fixed feature extractor. We aggregate features over a short temporal
892 window (`obs_horizon` = 2), apply PCA for dimensionality reduction on the image features, and
893 concatenate with the last two agent positions (normalized and reweighted to balance scale). Each
894 demonstration is thus represented in this joint embedding space. At test time, the current observation
895 is embedded in the same way, and the closest demonstration under cosine similarity is identified. The
896 policy then follows the flow induced by this demonstration, with progression and attraction weights
897 set to $\lambda_1 = \lambda_2 = 1.0$.898 **Per-timestep features.** Given an image \mathbf{I} and agent position $[x_a, y_a]$, we extract a 512-D descriptor
899 $\psi(\mathbf{I})$ with the frozen ResNet-18 backbone (final FC removed; BatchNorm \rightarrow GroupNorm as in the
900 diffusion policy).901 **Temporal windowing and dimensionality reduction.** With `obs_horizon` $T = 2$, we flatten the
902 last T descriptors and apply IncrementalPCA to project them to 16 principal components:

903
$$\mathbf{z}_t = \text{PCA}_{16}([\psi(\mathbf{I}_{t-1}), \psi(\mathbf{I}_t)]) \in \mathbb{R}^{16}.$$

904

905 **Concatenation with agent positions.** To balance image and agent information, we concatenate the
906 PCA embedding \mathbf{z}_t with the normalized agent positions from the last two steps. All embeddings are
907 L2-normalized before similarity computations.908 **Policy selection.** At test time, the query embedding is compared to the demonstration database using
909 cosine similarity, and the flow is executed with $\lambda_1 = \lambda_2 = 1.0$. To prevent degenerate repeats, the
910 selected pair is removed from the database at the next step.
911912 C.4 PUSHT TASK WITH VAE
913914 We construct a compact observation embedding using a convolutional variational autoencoder
915 (VAE) trained directly on PUSHT images. At inference, we discard the decoder and use only the
916 encoder to produce latent codes, which are concatenated with scaled agent positions to form the fi-
917 nal embedding. The global policy then follows the flow induced by the closest demonstration under
918 cosine similarity, with progression and attraction weights set to $\lambda_1 = \lambda_2 = 1.0$.

918 **Per-timestep features.** Given an image \mathbf{I}_t with pixel values normalized to $[0, 1]$, the VAE encoder
 919 outputs a Gaussian posterior

$$920 \quad \mathbf{z}_t \sim q_\phi(\mathbf{z} \mid \mathbf{I}_t), \quad \mathbf{z}_t \in \mathbb{R}^d,$$

921 with diagonal covariance. At inference, we use only the posterior mean μ_t as the latent feature.
 922

923 **Retrieval.** At test time, we encode the current observation window to obtain \mathbf{z}_t , normalize it, and
 924 compute cosine similarity against the stored database features. The demonstration with the highest
 925 similarity is selected, and its associated action sequence defines the flow. Cosine similarity achieved
 926 slightly higher performance (average return ≈ 0.88) compared to Euclidean distance (≈ 0.85).
 927

928 **Training Setup.** We train the VAE with a standard Gaussian prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a Gaussian
 929 reconstruction likelihood $p(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}(\hat{\mathbf{x}}(\mathbf{z}), \tau^2 \mathbf{I})$ with fixed $\tau = 2 \times 10^{-1}$. This choice of τ
 930 balanced the reconstruction and KL terms: with $\tau = 0.2$ both the reconstruction loss and the KL
 931 divergence decreased steadily, whereas using smaller τ values led to optimization stalling (neither
 932 term decreased). Training was performed for 25 epochs with the Adam optimizer (learning rate
 933 1×10^{-4}). At inference, we discard the decoder and use only the encoder’s posterior mean.
 934

935 C.5 PUSH-T TASK WITH SAM-BASED POSE EMBEDDING

936 We estimate object pose directly from images using a pretrained SAM/SAM2 pipeline (no fine-
 937 tuning). From each frame we obtain a binary mask of the T-block, from which we extract its centroid
 938 (x_b, y_b) and axial orientation θ_b (defined modulo π). Combined with the agent position (x_a, y_a), this
 939 yields the state

$$940 \quad \mathbf{x}_t = [x_a, y_a, x_b, y_b, \theta_b] \in \mathbb{R}^5.$$

941 All variables are normalized to $[0, 1]$ before distance computations; angular differences use the same
 942 axial angular distance as in the state-based setup. Distances and policy composition follow the same
 943 formulation, with weights $w_{\text{obj}} = w_{\text{agt}} = w_\theta = 1.0$ and flow execution with $\lambda_1 = \lambda_2 = 1.0$.
 944

945 **Per-timestep pose extraction.** Given a SAM mask, the centroid is

$$946 \quad (x_b, y_b) = \text{centroid}(\text{mask}),$$

947 and the orientation is computed from second-order moments of foreground pixels. Let μ_{pq} denote
 948 centralized moments; the principal axis corresponding to the largest covariance eigenvalue indicates
 949 the elongation direction. We define

$$950 \quad \theta_b = \frac{1}{2} \text{atan2}(2\mu_{11}, \mu_{20} - \mu_{02} + \varepsilon),$$

951 wrap θ_b to $(-\pi, \pi]$, and treat it as axial (modulo π) for angular distance.
 952

953 **Retrieval and policy selection.** At test time, we form $\mathbf{x}_t = [x_a, y_a, x_b, y_b, \theta_b]$, apply the same
 954 normalization as above, and compute distances to all stored demonstration states using the state-
 955 based metric. We retrieve the K nearest neighbors (default $K = 1$) and execute the composed flow
 956 with $\lambda_1 = \lambda_2 = 1.0$.
 957

958 **Tracking and prompting details.** We use SAM2’s video predictor (sam2.1_hiera_tiny) to
 959 track the T-block across frames, re-prompting each step with a skeletal outline derived from the
 960 most recent pose estimate to stabilize mask propagation. To compensate for a small systematic bias
 961 in predicted centroids, we apply a constant offset correction to (x_b, y_b) , calibrated on seeds 500–700.
 962

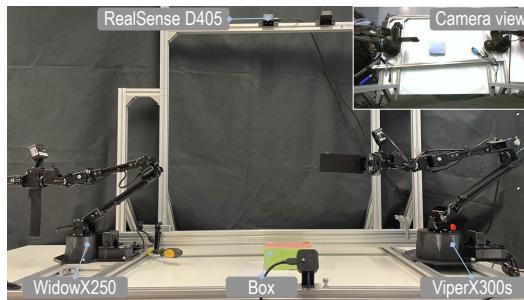
963 **Limitations.** Performance depends on segmentation quality; occlusions and viewpoint changes can
 964 induce drift in the estimated pose, which in turn affects retrieval and control.
 965

966 C.6 ROBOT-FLIP TASK

967 **Robot teleoperation:** We utilized a bimanual robotic system configured with a ViperX300s (follower)
 968 and a WidowX250 (leader), along with a RealSense D405 camera from a top-down view. The
 969 system is built on an open-source platform. By using robot teleoperation, we collected 121 demon-
 970 strations, each contains 200 to 1000 timesteps to complete the flip task. The dataset is structured
 971 in an HDF5 format and includes robot actions and observations, where observations are composed
 972 of effort, images, joint angles, and joint velocities. Specifically, we teleoperated the leader robot
 973 (WidowX250) to control the follower (ViperX300s) robot for manipulation tasks (flip the box). The

972 camera records images at an 848×480 resolution with a 30 Hz frequency, and then crops them to a
 973 320×240 resolution for policy training.
 974

975 **Policy imitation.** The policy imitation process is similar to the pushT task with vision-based inputs.
 976 Specifically, we use a vision encoder that takes RGB images as input and predicts the desired robot
 977 action as a latent embedding using an MSE loss. Training is performed for 100 epochs using the
 978 Adam optimizer with a learning rate of 0.0001. After training, we calculate the latent feature of
 979 each demonstrated image as a feature database. The online inference involves the computation of a
 980 distance field that includes both distance measurement in this latent space and an additional distance
 981 metric for joint position displacement, guiding the flow field and policy composition. Both attraction
 982 and progression parameters are set to 1.0 during execution. To ensure the temporal consistency, the
 983 task is run with horizon=100.



984
 985
 986
 987
 988
 989
 990
 991
 992
 993
 994
 995
 996
 997
 998
 999
 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1020
 1021
 1022
 1023
 1024
 1025

Figure 10: ALOHA teleoperation platform.

C.7 HUMAN-ROBOT INTERACTION TASK

998 We use the `openai/clip-vit-base-patch32` CLIP model for vision-language grounding.
 999 Positive and negative text prompts for hand-held object detection are listed below.
 1000

Text prompts.

```
1001 pos_prompts = [
1002     "a photo of a hand holding a banana",
1003     "a hand holding an apple",
1004     "a human hand holding an orange",
1005     "a hand holding a pear",
1006     "a hand holding a strawberry",
1007     "a hand holding grapes",
1008     "a hand holding a piece of fruit",
1009     "a person's hand holding a fruit",
1010     "close-up of a hand holding a fruit",
1011 ]
1012 neg_prompts = [
1013     "an empty hand",
1014     "a hand with nothing in it",
1015     "a hand holding a baseball",
1016     "a hand holding a black ball",
1017     "a hand holding a blue cup",
1018     "a hand holding a plastic cup",
1019     "a hand holding adhesive tape",
1020     "a hand holding a tape roll",
1021     "a hand holding a screwdriver",
1022     "a hand holding a tool",
1023     "a hand holding a non-fruit object",
1024 ]
1025
```

1026 **D ADDITIONAL EXPERIMENTAL RESULTS**
10271028 **D.1 MEMORY COST**
10291030 The state-based PUSHT dataset has $25,000 \times 7 = 175,000$ elements, requiring $175,000 \times 4 \approx 0.67$
1031 MB with `float32`, consistent with the observed 0.7 MB. For comparison, an MLP with layers
1032 [7, 512, 256, 128, 1] has 168,449 parameters (≈ 0.64 MB), which is at a similar scale. However,
1033 typical models are far larger than simple MLP; e.g., a state-based diffusion policy exceeds 200 MB.1034 Although GPI’s memory grows linearly with the number of demonstrations, this is practical in our
1035 setting: robot actions are low-dimensional, and high-dimensional observations are stored as compact
1036 latent features. Inference is lightweight, parallelizable, and can use subsampling or approximate
1037 nearest-neighbor search to bound latency. As we demonstrated in the paper, GPI achieves orders-of-
1038 magnitude gains in efficiency over standard baselines in common imitation-learning settings.
10391040 **D.2 COMPLEXITY AND SCALABILITY**
10411042 **Complexity of one control step.** All demonstrations are stored in a tensor of shape (NT, D) , where
1043 N is the number of demonstrations, T is the trajectory length, and D is the state / feature dimension.
1044 Given the current observation of shape $(1, D)$, we compute its distance to all stored states in a single
1045 batched operation. The complexity of this retrieval step is therefore $O(NTD)$, implemented in
1046 parallel on GPU. Once the top- K neighbors are selected, combining their flows to compute the final
1047 control command is $O(KD)$, which is negligible for small K compared to the retrieval cost.1048 **Empirical scalability.** To make the scaling explicit, we report retrieval latency and memory usage
1049 as we vary the number of stored states from 10^2 to 10^6 and the feature dimension D from 5 to
1050 512. Latency is measured in milliseconds (ms) on a single GPU, and memory usage is reported in
1051 megabytes (MB). We measure the cost of computing Euclidean distances between a single query
1052 state and all stored features, which is the component that grows with the database size:1053 Table 4: Retrieval latency and memory usage for a single query as a function of the number of stored states and
1054 feature dimension D . Each entry reports *latency (ms) / memory (MB)*.
1055

# states	$D = 5$	$D = 32$	$D = 128$	$D = 512$
1×10^2	0.039 / 0.00	0.033 / 0.01	0.034 / 0.05	0.033 / 0.20
1×10^3	0.039 / 0.02	0.036 / 0.12	0.035 / 0.49	0.039 / 1.96
1×10^4	0.033 / 0.19	0.037 / 1.22	0.068 / 4.88	0.217 / 19.6
1×10^5	0.051 / 1.91	0.221 / 12.2	0.517 / 48.8	1.973 / 196
1×10^6	0.291 / 19.1	1.920 / 122	4.891 / 488	19.26 / 1955

1063 Even with very large databases (10^6 states) and high-dimensional features ($D = 512$), retrieval
1064 remains below 20 ms with about 2 GB (approximately 1955 MB) of memory, which is compatible
1065 with typical real-time manipulation settings.
10661067 **D.3 ABLATIONS ON DISTANCE METRICS**
10681069 In all experiments, the metric used by GPI is constructed from simple and physically motivated
1070 components: Euclidean distance for robot states (e.g., joint angles, end-effector positions), geodesic
1071 distance for quaternions, and cosine similarity for latent visual embeddings. These choices follow
1072 the geometry and physical meaning of each state component. All state dimensions are normalized
1073 before distance computation, which further reduces sensitivity to manual tuning.
10741075 To assess sensitivity to the specific form of the metric, we perform an ablation on the PushT task
1076 comparing L1 distance, L2 distance, and cosine similarity, both in the original state space and in a
1077 VAE-based latent space. The average rewards are summarized in Table 5.
10781079 In the state space, both L1 and L2 distances perform well, indicating that GPI is not sensitive to
1080 the exact choice of norm as long as the metric is consistent with the underlying state geometry. In
1081 the latent space, cosine similarity performs best, with L1 and L2 still competitive, consistent with

1080
1081
1082 Table 5: Average reward for different distance metrics on the pushT task.
1083
1084
1085
1086
1087

Metric	State space	Latent space (VAE)
L1 norm distance	88	81
L2 norm distance	86	85
Cosine similarity	88	87

1088 standard practice for feature embeddings. Even when directly using cosine similarity in the state
1089 space, performance remains reasonable. Overall, these results indicate that GPI does not rely on
1090 fragile, hand-tuned metrics and works reliably with simple, task-aligned distances.

1091 We also study how the relative weighting between the environment distance d_{env} and the robot dis-
1092 tance d_{rob} influences performance on PushT and Robomimic tasks. We define the combined metric
1093

$$d = w_{\text{rob}} d_{\text{rob}} + w_{\text{env}} d_{\text{env}}, \quad (20)$$

1094 and vary the ratio $w_{\text{env}}/w_{\text{rob}}$. The results are given in Table 6. Performance is clearly degraded when
1095 the environment weight is extremely small or large (e.g., $w_{\text{env}}/w_{\text{rob}} = 0.01$ or 100), but remains
1096 high over a broad intermediate range (approximately $w_{\text{env}}/w_{\text{rob}} \in [0.1, 10]$), with the best results
1097 typically obtained near equal weighting (around 0.5 – 1.0). These trends confirm that the metric and
1098 its weighting are important design choices, but also show that GPI remains robust to a wide range
1099 of relative weightings between robot and environment distances.
1100

1101
1102 Table 6: Evaluation of robot–environment distance weighting on various benchmarks.
1103

$w_{\text{env}}/w_{\text{rob}}$	0.01	0.1	0.5	1.0	5.0	10.0	100.0
PushT	0.38	0.76	0.83	0.87	0.80	0.80	0.54
Lift	0.68	0.85	0.98	1.00	0.83	0.78	0.52
Can	0.61	0.73	0.88	0.96	0.79	0.58	0.18
Square	0.23	0.55	0.72	0.82	0.63	0.51	0.16

1110 D.4 ABLATION ON COMPOSITION HYPERPARAMETERS

1111 We ablate the softmax temperature β used in (3) for composing the top- K demonstrations on PushT
1112 (state-based). As shown in Table 7, performance remains stable over several orders of magnitude of
1113 β .
1114

1115
1116 Table 7: Effect of the softmax temperature β on PushT.
1117

β	10^{-4}	10^{-3}	10^{-2}	10^{-1}
Avg. Score	85.1 ± 1.2	84.8 ± 0.9	85.8 ± 1.2	85.6 ± 1.4

1122 D.5 ROBOMIMIC AND ADROIT HAND TASKS

1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

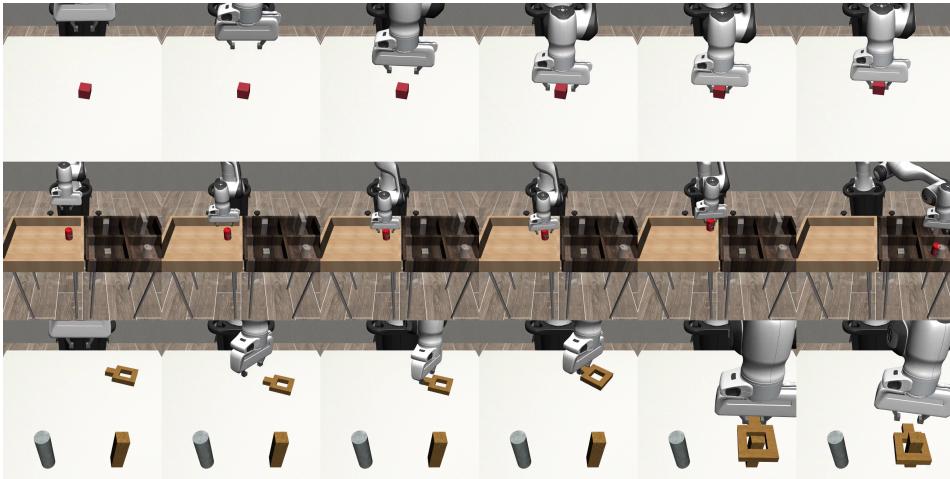


Figure 11: Snapshots of experimental results for Lift, Can, and Square tasks on Robomimic environments.

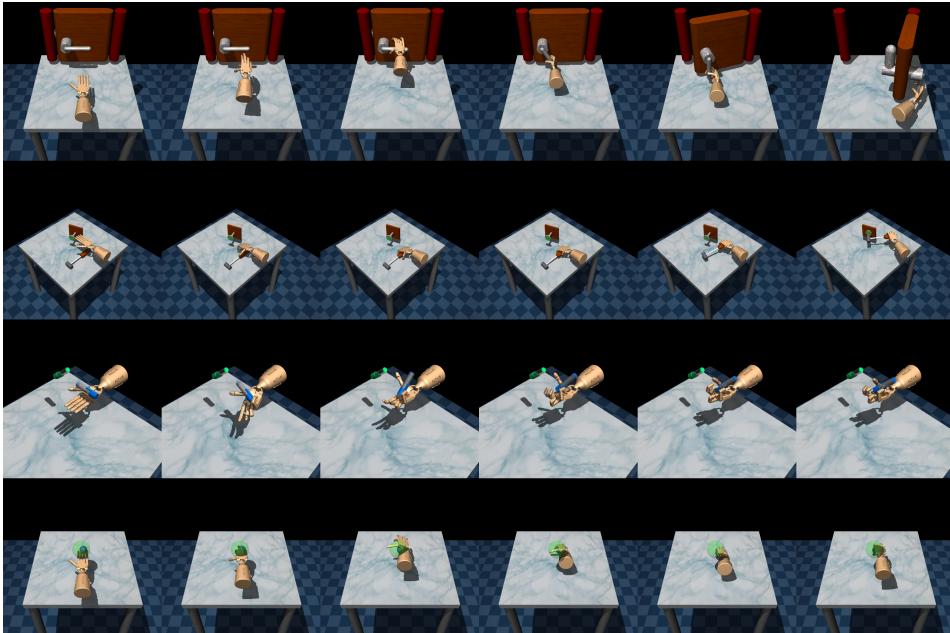


Figure 12: Snapshots of experimental results for Door, Hammer, Pen, and Relocate on Adroit hand tasks.

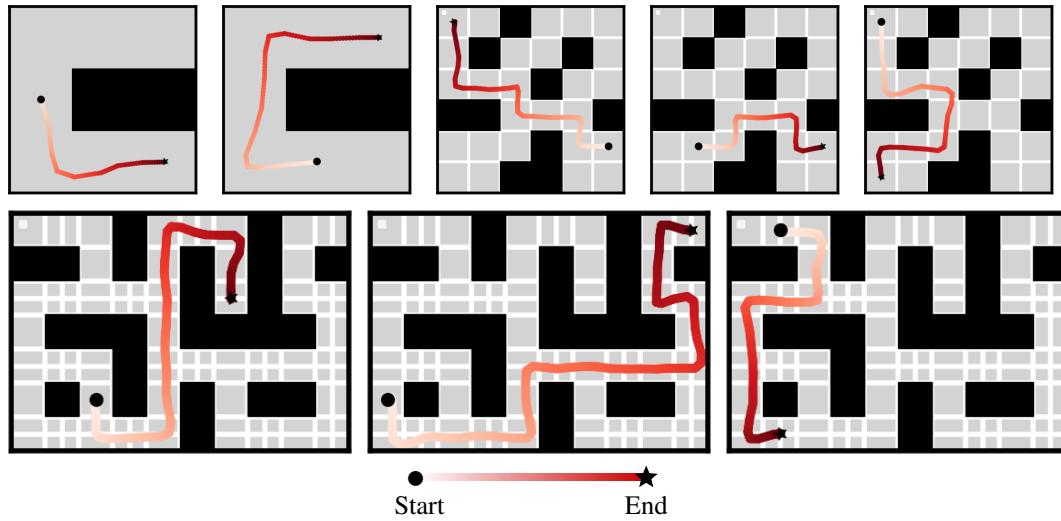
D.6 2D MAZE

We evaluate our approach on the 2D Maze benchmark, previously used by (Chen et al., 2025; Janner et al., 2022). Unlike these methods, our approach is *training-free*: at test time we select a suffix of a single demonstration using a simple distance metric and execute it. Concretely, for demonstration i of length H and timestep k , we minimize

$$D(i, k) = 10 \|\mathbf{x}_0 - \mathbf{x}_k^{(i)}\|_2 + 5 \|\mathbf{x}_g - \mathbf{x}_g^{(i)}\|_2 + 0.1 (H - k),$$

where \mathbf{x}_0 is the initial state, $\mathbf{x}_k^{(i)}$ is the k -th state of demonstration i , \mathbf{x}_g is the task goal, and $\mathbf{x}_g^{(i)}$ is the goal state associated with demonstration i . The final term penalizes long remaining horizons; since 2D Maze demonstrations can include detours, this bias favors suffixes that proceed more directly to the goal. After selecting (i^*, k^*) , we execute the suffix $\{\mathbf{x}_{k^*:H}^{(i^*)}\}$ as the plan. In doing so, our method also recovers the effective task horizon $H - k^*$, something most alternative approaches cannot determine directly. Instead, they must either: (i) assume a long horizon and truncate once

1188 the task is completed, (ii) assume a short horizon and repeat until completion, or (iii) try multiple
 1189 horizons and select the smallest successful one.
 1190



1208 Figure 13: Results on 2D Maze using our method. Without any training, a simple distance-based criterion
 1209 achieves a 100% success rate across all tasks, with an average inference time of 0.08 seconds.
 1210
 1211

1212 E REPRODUCIBILITY STATEMENT

1213 We will release our code, configuration files, and evaluation scripts upon publication. Key imple-
 1214 mentation details and protocols are documented in the main text and appendix to facilitate reproduc-
 1215 tion in the interim.
 1216

1217 F USE OF LARGE LANGUAGE MODELS (LLMs)

1218 We used LLMs (e.g., ChatGPT and Claude) to rephrase and polish the manuscript and to assist with
 1219 coding tasks. All LLM-generated code was reviewed, edited, and integrated by the authors; the
 1220 LLM did not design algorithms or produce experimental results.
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241