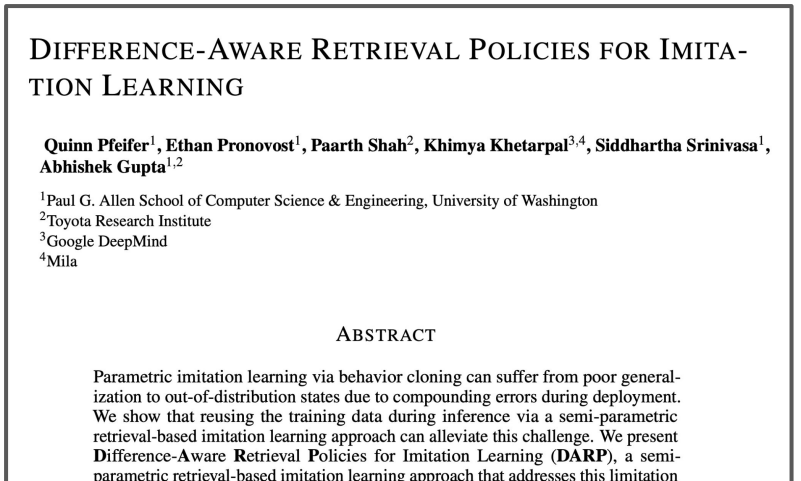
# Calibration Slide

# Difference-Aware Retrieval Policies for Imitation Learning

A2I / 1 May 2026



DIFFERENCE-AWARE RETRIEVAL POLICIES FOR IMITATION LEARNING

**Quinn Pfeifer**[1], **Ethan Pronovost**[1], **Paarth Shah**[2], **Khimya Khetarpal**[3,4], **Siddhartha Srinivasa**[1], **Abhishek Gupta**[1,2]

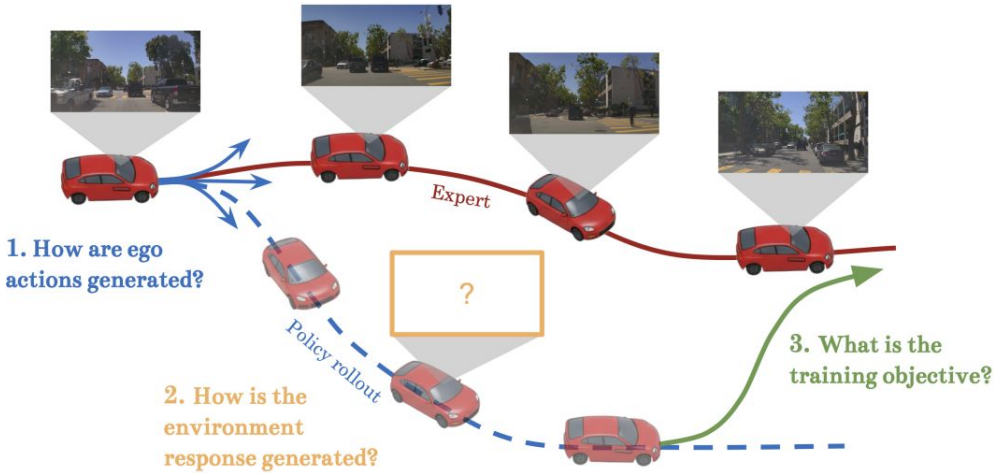
[1]Paul G. Allen School of Computer Science & Engineering, University of Washington
[2]Toyota Research Institute
[3]Google DeepMind
[4]Mila

ABSTRACT

Parametric imitation learning via behavior cloning can suffer from poor generalization to out-of-distribution states due to compounding errors during deployment. We show that reusing the training data during inference via a semi-parametric retrieval-based imitation learning approach can alleviate this challenge. We present **D**ifference-**A**ware **R**etrieval **P**olicies for Imitation Learning (**DARP**), a semi-parametric retrieval-based imitation learning approach that addresses this limitation

# Context

- **Behavior cloning** (BC) has enabled robots to learn complex behaviors from **expert demonstrations**.
- BC is brittle in practice, especially for long-horizon tasks.
- The core issue is *covariate shift*
  - Small errors accumulate during rollouts
  - Agent moves into states not well-represented in demonstration data
  - OOD! BC policies fail



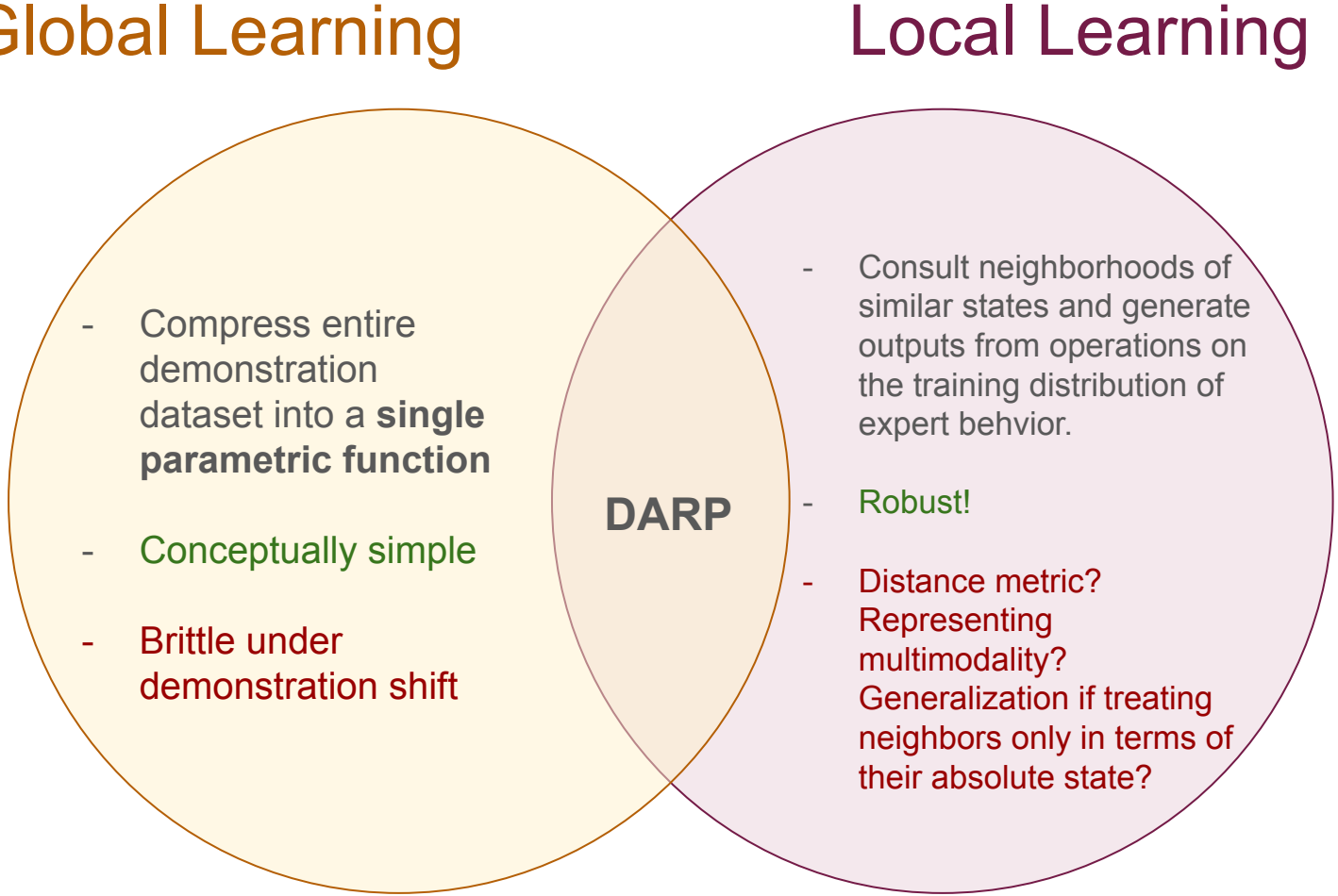
# Constraints

- This is a well recognized challenge and many approaches have been proposed to mitigate compounding error.

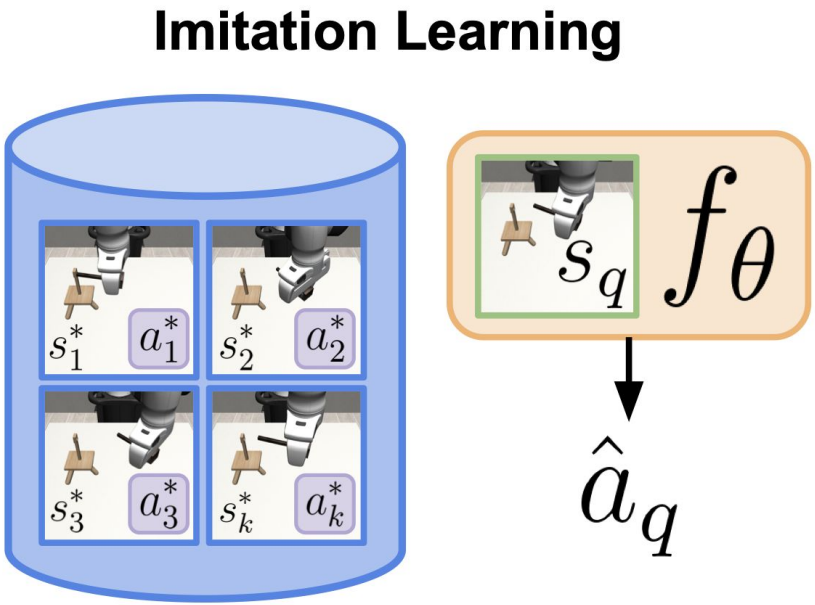
- These solutions typically go beyond **standard BC assumptions**, requiring simulators, interactive experts, large quantities of sub-optimal data, or strong task-specific structure.

- ***This paper wants to stay in the pure BC regime***
  - Learn only from expert state-action pairs, with no additional supervision or feedback.

- *Can we reduce the variance of BC policies using only the original demonstration dataset?*


Global Learning
Local Learning
- Compress entire demonstration dataset into a **single parametric function**
- Conceptually simple
- Brittle under demonstration shift
DARP
- Consult neighborhoods of similar states and generate outputs from operations on the training distribution of expert behvior.
- Robust!
- Distance metric? Representing multimodality? Generalization if treating neighbors only in terms of their absolute state?

# Method: Preliminaries




- The behavior cloning algorithm learns a policy $\pi_\theta$ from this dataset by casting imitation as a typical supervised learning problem.

- $\arg\max_\theta \mathbb{E}_{(s^*,a^*)\sim\mathscr{D}^*}[\log(\pi_\theta(a^*|s^*))]$

# Neighbor Manifold Regularized Imitation Learning (**MRIL**)

- Objective above does not control how the policy behaves on states that deviate from the manifold of expert states.

- To incorporate data manifold/neighborhood information into policy learning, let's just **modify the objective**:

normalized kernel weights
based on state differences

$$\mathscr{L}_{\mathrm{MRIL}}(f) = \underbrace{\mathbb{E}_{(s,a)\sim\mathscr{D}^*}\left[\ell\big(f(s),a\big)\right]}_{\text{supervised risk}(\mathscr{L}_{\mathrm{BC}})} + \lambda \underbrace{\mathbb{E}_{s\sim\mathscr{D}^*}\left[\sum_{i\in\mathscr{N}_k(s)} w_i(s)\left\|f(s)-f(s_i^*)\right\|_2^2\right]}_{\text{smoothness regularizer}(\mathscr{L}_{\mathrm{S}})},$$

- Smoothness penalty explicitly encourages **local consistency** of predictions: nearby states in the expert dataset should be mapped to similar actions

# MRIL Proofs: Why add this term to the loss?

- **Variance Reduction:** the Laplacian penalty in MRIL acts as a data-dependent regularizer, yielding smaller estimator variance than vanilla BC.

- **Smoothness guarantee**: minimizing loss satisfies a uniform bound on the local Lipschitz constant of f, whereas vanilla BC admits interpolants with arbitrarily large Lipschitz constants between training states.

- **Policy stability**: the deviation recursion accumulates error linearly for vanilla BC, but sublinearly for MRIL

**MRIL enjoys strictly better generalization and stability guarantees than BC**

# Can we simplify this?

What do we **not like**?

- Hyperpameter lambda needs to be tuned
- Optimizing a modified, regularized objective rather than a standard BC objective may modify the optimization landscape in adverse ways

*Can we obtain the same benefits conferred by MRIL by modifying the **policy architecture** rather than modifying the **objective**?*

# **Implicit MRIL** (iMRIL)

- Let's move the neighborhood aggregation (averaging) operation from the objective to the architecture.

- iMRIL optimizes at training time:

$$\arg\min_{\theta} \mathbb{E}_{(s^*,a^*)\sim\mathscr{D}^*} \Big[ \Big\| \underbrace{\left( \frac{1}{k} \sum_{i \in \mathcal{N}_k(s^*)} f_\theta(s_i^*) \right)}_{\hat{f}(s^*)} - a^* \Big\|_2 \Big]$$

- At test time, inference can be performed on a new state $s_q$ simply by retrieving the *k*-NN of $s_q$ and performing neighborhood aggregation.

# **Implicit MRIL** (iMRIL)

- Equivalent to the laplacian regularization of MRIL.

- This implicit smoothing view constrains the class of functions that can be represented after aggregation.
  - The neighbor-conditioned network learns how expert actions vary under local perturbations, proposing locally adapted actions for each neighbor.
  - The aggregation operator then enforces variance reduction by smoothing these proposals across the neighborhood.

- In this way, learning provides accuracy by **correcting local bias**, while aggregation provides stability by **controlling variance**.

# DARP

- Because the neighborhood aggregation should learn how expert actions vary under local perturbations, so $f_\theta$ needs more context:
  - Like knowledge of differences between a query and a neighbor state

$$\hat{a}_q = f_{\text{DARP}}(s_q) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(s_q)} a'_i$$

$$= \frac{1}{k} \sum_{i \in \mathcal{N}_k(s_q)} f_\theta(s^*_i, a^*_i, \Delta s_i = s^*_i - s_q).$$

- At training time, this would form the action prediction used in the BC loss.

# Note on Neighborhood Aggregation Step

- **Averaging** over neighborhood predictions is just one permutation-invariant aggregation function.

- For example, DARP can predict the parameters of an action distribution (like means, covariance, weights of a Gaussian mixture model) to train **multimodal** action distributions instead of only **unimodal**.

- They show this works for the Push-T task

# Evaluation

- **MuJoCo Tasks:**
  - Hopper (single-legged hopping robot)
  - Walker (bipedal humanoid)
  - Ant (quadruped)
  - HalfCheetah (biped)

- **Robosuite Tasks:**
  - Single robotic arm manipulating objects

- **Robocasa Tasks:**
  - A single robotic arm manipulating objects in a randomized kitchen setting

# Can DARP Consistently Outperform Standard BC?

Taking action of NN

Locally weighted regression on NNs

Transformer-based ICL method conditioned on NN

Explicitly smoothed version

| Method | Hopper | Ant | Walker | HalfCheetah |
|---|---|---|---|---|
| R&P (Sridhar et al., 2025) | 711.82 ± 85.63 | -305.97 ± 76.42 | 419.18 ± 50.21 | -178.64 ± 29.75 |
| LWR (Pari et al., 2022) | 1703.78 ± 245.95 | 846.59 ± 216.06 | 1484.91 ± 356.54 | 1945.82 ± 567.26 |
| BC | 2313.65 ± 203.75 | 2376.20 ± 339.43 | 2658.40 ± 274.08 | 1063.23 ± 371.08 |
| REGENT (Sridhar et al., 2025) | 1819.39 ± 186.24 | -302.10 ± 146.67 | 507.01 ± 76.10 | 169.85 ± 63.10 |
| MRIL | 2793.63 ± 156.41 | 3869.08 ± 241.00 | 4370.96 ± 168.13 | 701.58 ± 195.08 |
| **DARP** | **3545.57 ± 3.54** | **4383.28 ± 266.37** | **4894.01 ± 75.12** | **5515.41 ± 841.33** |
| **DARP Set Transformer** | 2965.86 ± 103.08 | **4063.79 ± 218.80** | **4752.42 ± 109.23** | 3417.85 ± 764.57 |

Table 1: **Both DARP and DARP Set Transformer outperform other approaches across all domains.** Performance Comparison of DARP vs. BC and other baselines across MuJoCo Environments Using Low-Dimensional State. Scores reported are averaged across 100 independent trials with 95% confidence intervals.

(Euclidean distance for NN retrieval)

# Can DARP Handle More Complex State Representation and Action Distributions?

- Use R3M image embeddings instead of low-dimensional state features.

- *What happens if states are represented as high-dimensional feature vectors ex- tracted from visual observations?*

| Method | Stack | Thrd. | Peg |
|---|---|---|---|
| BC | 44 | 38 | 17 |
| **DARP** | **75** | **76** | **52** |

Table 4: Success rates (%) on vision-based Robosuite tasks.

# Ablations: What Part of DARP Makes the Most Difference?

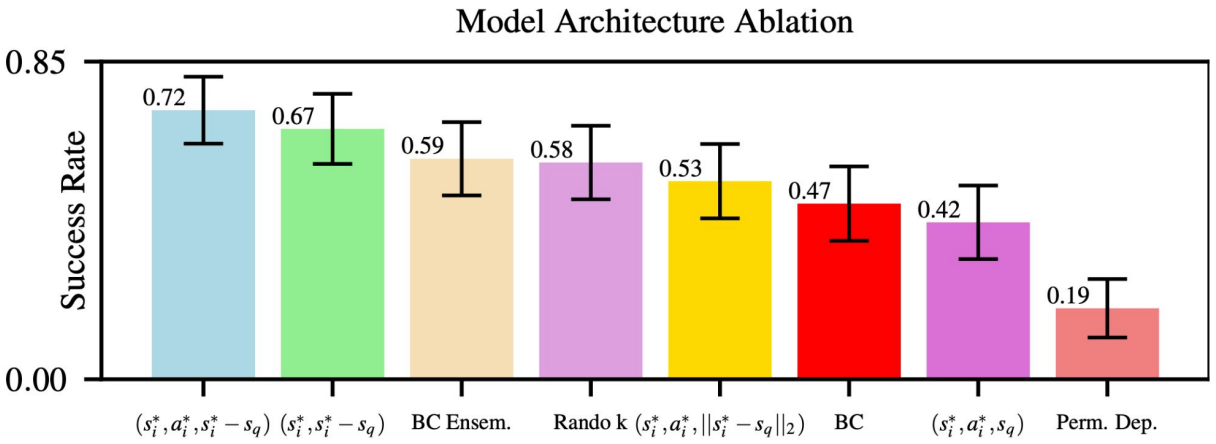



Figure 3: **Distance vectors and permutation invariance contribute heavily to DARP's success.** Exploration of how the performance of a DARP agent is impacted as various changes are made to the core architecture demonstrates that DARP success is most attributed to the distance vectors $(s_i^*, a_i^*, s_i^* - s_q)$. Success rate is averaged across 100 trials on the Robosuite Stack environment with 95% confidence intervals.

# What happens when we enter OOD state with DARP vs BC?

Reward splits at
**State of Divergence**

T_s is value below which
1% of training data lies

Delta likelihood:
Nearby states differ
in familiar ways
(small shifts)

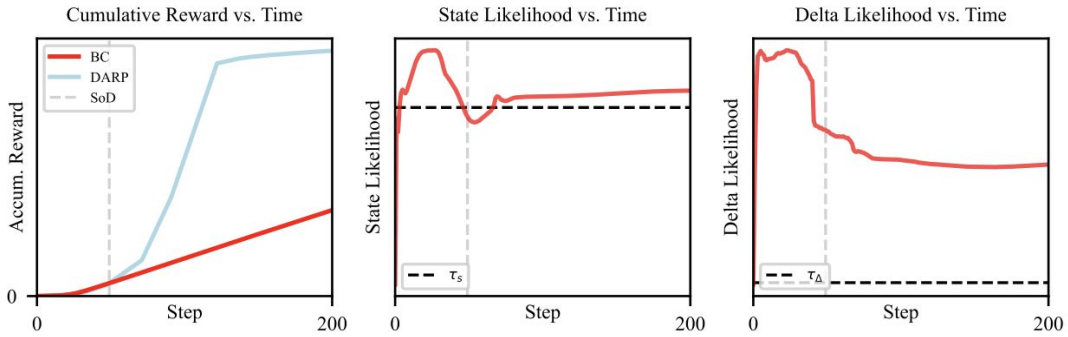



Figure 4: **Cumulative rewards for BC and DARP on the Robosuite stack task illustrate initially identical rollouts that diverge as BC fails the task and DARP succeeds.** A vertical dashed line indicates the step in which the two diverge, labeled "SoD". At the SoD, the state likelihood is $< \tau_s$ (OOD), but the delta likelihood is $> \tau_\Delta$ (in distribution).

# Overall Thoughts

- Expensive computationally as nonparametric method
- Clear Intuitive Method
  - Experiments could be better discussed (selecting *k* is not clear from the paper)
- Does it help with long horizon tasks?