Escaping the Diversity Trap in Robotic Manipulation

via Anchor-Centric Adaptation

Yanzhe Chen · Kevin Yuchen Ma · Qi Lv · Yiqi Lin · Zechen Bai · Chen Gao · Mike Zheng Shou

National University of Singapore · *ICML 2026*

"I fear not the man who has practiced 10,000 kicks once,
but I fear the man who has practiced one kick 10,000 times."

— *Bruce Lee*

Our Journey: From Hyperparameters to Data Strategy

How we arrived at the core research question

Tune Learning Rate

1e-4 → 1e-5 → 5e-5
Still inconsistent success

Tune Training Steps

5K → 20K → 50K
Overfit anchors or underfit

Tune Chunk Size

Action horizon 8, 16, 32
Minor effect on stability

Tune LR Warmup

Various warmup schedules
No consistent improvement

Shift Perspective

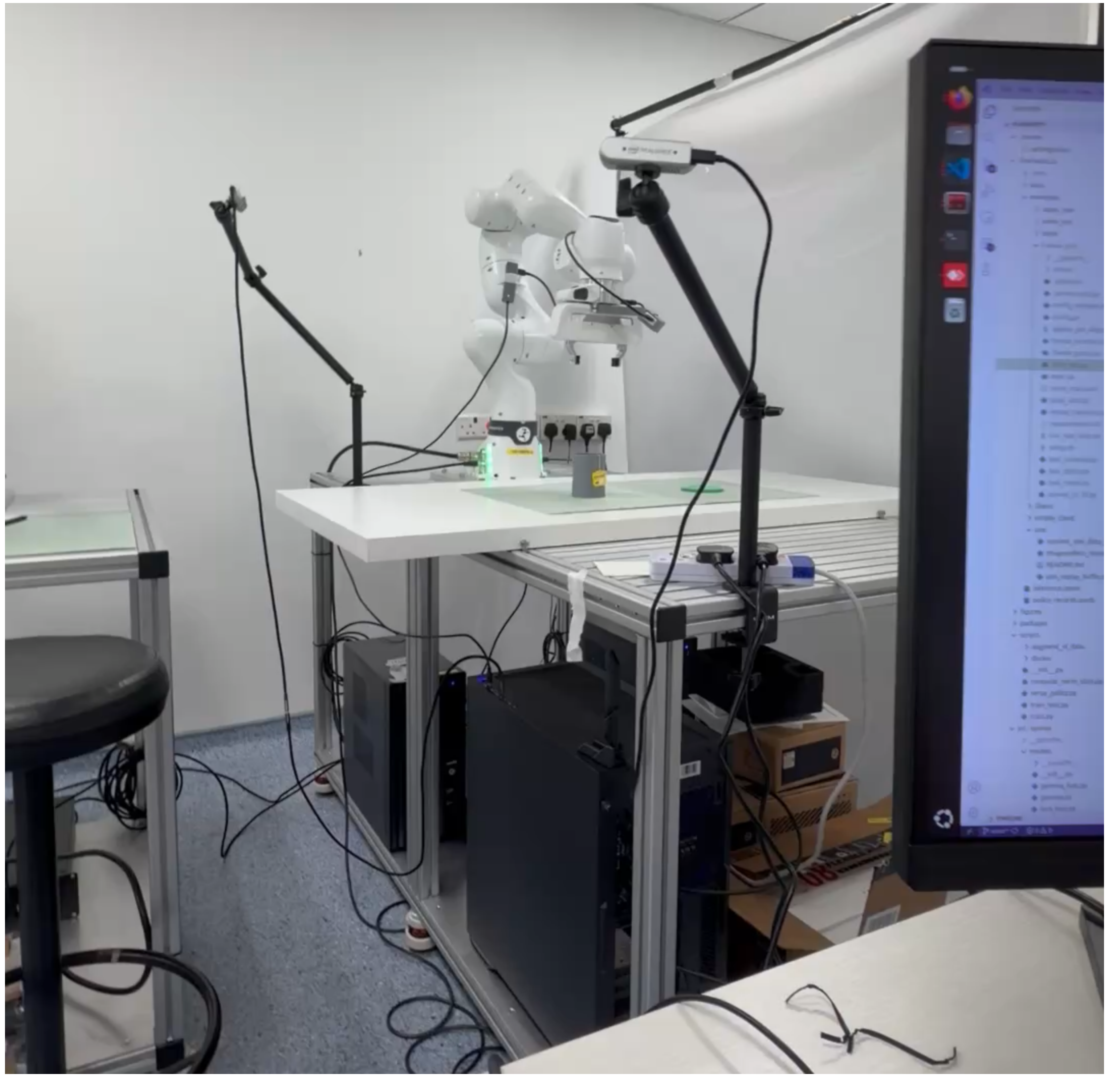
The data strategy itself
is the missing variable!

Key Realization: Under a tight data budget, what matters most is *how you allocate demonstrations*, not which hyperparameters you use to train them.

→ This motivated us to study the data collection strategy itself as a first-class research problem.

Our Journey: From Hyperparameters to Data Strategy

How we arrived at the core research question



The Core Problem: Budget-Efficient VLA Adaptation

Deploying pretrained VLAs on physical hardware under data constraints

VLAs offer broad capability from pretraining (π0, π0.5, OpenVLA⋯)

Real-world deployment requires fine-tuning to bridge the embodiment gap

Robot demonstrations are expensive — typical budgets: 50-150 trajectories

Standard practice: collect diverse single-shot demos to maximize coverage

The Pivotal Question

Under a limited budget,
what is the most effective
data collection strategy
to learn a robust policy?

The Diversity Trap

Spreading scarce samples too thin leaves each condition under-represented, inflating estimation variance and destabilizing the policy.

Our Answer: The Coverage-Density Trade-off → Anchor-Centric Adaptation (ACA)

Empirical Evidence: The Inverted-U Phenomenon

More diversity is not always better — there exists an interior optimum

01

Single-shot coverage fails

With one demo per condition, policy estimation is dominated by noise — performance collapses at boundaries.

02

Repetition at anchors stabilizes

Concentrating repeated demos on a sparse set of anchors builds a reliable 'policy skeleton'.

03

Inverted-U trend

As # anchors K increases beyond an optimum, success rate declines — excessive diversity dilutes per- condition density.

04

Budget-dependent optimum

Larger budgets support higher diversity without sacrificing stability — consistent with theory.

Before the Math: Two Paradigms of Theoretical Research

How theory and experiment inform each other in ML research

① Inductive (Empirical → Theory)

1. Observe interesting phenomenon in experiments
2. Identify the underlying pattern or mechanism
3. Construct a formal mathematical model to explain it
4. Use theory to predict new phenomena

Example: observing inverted-U → formalizing Coverage-Density Trade-off

ACA
uses both

② Deductive (Theory → Method)

1. Derive formal bounds or principles from first principles
2. Identify what the theory prescribes as optimal
3. Design a method that operationalizes the theoretical insight
4. Validate empirically that the method works as predicted

Example: Corollary 3.3 (interior K) → ACA's two-stage design*

Math Toolkit: Reading Theoretical Proofs

Understanding the vocabulary of formal ML theory

A

Assumption

Definition

A condition we take as given. Bounds the scope of validity. E.g., 'A1: f* is L- Lipschitz in p'.

In ACA

Defines what the world must look like for our proof to hold.

L

Lemma

Definition

A 'helper' result. Proved independently, then used as a building block inside the main proof.

In ACA

Think of it as a sub- routine in a program.

P

Proposition

Definition

A self- contained result with full proof. Often a key step, but not the main theorem.

In ACA

Prop. 3.1 is our coverage- density decomposition — the core analytical result.

C

Corollary

Definition

A result that follows directly from a proposition/theorem with little additional proof.

In ACA

Cor. 3.3 derives K — the optimal anchor count — from Lemma 3.2 by calculus.*

R

Remark

Definition

An informal observation or clarification about a result. Not proved, but insightful.

In ACA

Used to discuss tightness, practical implications, or scope limitations.

Formalizing the Problem: Coverage-Density Trade-off

Section 3 — Modeling real-robot adaptation as learning a conditional vector field

Setup & Notation

$p \in P \subset$	Geometric condition (e.g., object pose)
$t \in [0, 1]$	Flow matching time
$z \in Z$	Action-state variable
$f^*(z, p, t)$	Ground-truth conditional vector field
N	Fixed total demo budget
K	Number of distinct conditions (anchors)
n	Repeats at anchor p , Σn = N
$h = \sup \min \| p \;\; p \|$	Fill distance (worst-case coverage gap)

Three Assumptions

A1

Smoothness in Condition

$$\| f^*(z,p,t) \;\; f^*(z,p',t) \| \le L \| p \;\; p' \|$$

The target vector field changes smoothly with condition p. Nearby poses → similar actions.

A2

Effective Local Estimation

$$E \| f (z,p ,t) \;\; f^*(z,p ,t) \| \le C\sigma / \sqrt{n}$$

With more repeated demos at an anchor, estimation error decreases at rate $1/\sqrt{n}$.

A3

Nearest-Anchor Surrogate

$$f (z,p,t) := f (z, p_{i(p)}, t)$$

Conservative surrogate: any unseen point queries its nearest anchor. Neural nets do better ($O(h^2)$).

Proposition 3.1: The Coverage-Density Decomposition

The core error bound that motivates everything downstream

Proposition 3.1 (Coverage-Density Bound)

$$\text{Under A1-A3:}\quad \sup_{p,z,t}\ E \| f \quad f^* \| \ \le\ \max_i E \| f \quad f^* \| \ +\ L \cdot h$$

Term 1: Estimation (Density) Error

$$\max_i\ E \| f\,(z, p\,,\, t) \quad f^*(z, p\,,\, t) \|$$

Controlled by: sample density n per anchor.
Reduces as 1/√n with more repeated demos.
High when each condition is seen only once.

Term 2: Extrapolation (Coverage) Bias

$$L \cdot h \ = \ L \cdot \sup_p\ \min_i\ \| p \quad p \|$$

Controlled by: fill distance h between anchors.
Reduces as K increases (denser anchor placement).
High when workspace has large uncovered regions.

The Tension: Increasing K (more conditions) reduces extrapolation bias but increases estimation error per condition.

Proof sketch: Triangle inequality + A1 (Lipschitz) + A3 (nearest-anchor surrogate) → complete proof in §A.1.2

Lemma 3.2 & Corollary 3.3: The Interior Optimal K*

Under non-negligible noise, the optimal anchor count is strictly less than N

Lemma 3.2 (Uniform Allocation + Quasi-Uniform Anchors)

Assume: uniform repeats $n = N/K$, quasi-uniform placement $h \le c \cdot K^{-1/d}$

$$E(K) \le C\sigma\sqrt{K/N} + Lc \cdot K^{-1/d}$$

↑ grows with K (more anchors = less data per anchor = more noise)

↓ shrinks with K (more anchors = better coverage = less extrapolation)

Corollary 3.3 (Interior Optimal Allocation)

Under non-negligible noise $\sigma > 0$, minimizing $E(K)$ gives:

$$K^* \propto (L^2 N / \sigma^2)^{d/(d+2)} < N$$

The key result: K* is strictly less than N.
Fully diverse sampling ($K=N$, $n=1$) leaves estimation noise $C\sigma$ as a constant floor.

Comparing K = N vs. K = K*

Max Diversity ($K = N$, $n = 1$)

- → $C\sigma + Lc \cdot N^{-1/d}$
- → Constant noise floor $C\sigma$ never goes away
- → Can't improve no matter how large N gets

Optimal Allocation ($K = K^*$)

- → $(\sigma^{2/(d+2)} \cdot L^{d/(d+2)} \cdot N^{-1/(d+2)})$
- → Decays with N — estimation noise is controlled
- → Repetition at anchors drives error to zero

Proof: Differentiate $E(K)$ w.r.t. K and set to zero. The estimation term grows as $\sqrt{K}$ while coverage decays as $K^{-1/d}$, yielding the interior minimum.

From Vector Field Error to Task Failure: Motivating Two Stages

Section 3.5 — Grönwall's inequality connects field error to open-loop trajectory deviation

Trajectory Propagation (Grönwall's Lemma)

Let (T) be the policy trajectory and z*(T) the ground truth. If both vector fields are Λ-Lipschitz in z:

$$\| \ (T) \quad z^*(T) \| \ \leq \ (e^{\Lambda T} \quad 1)/\Lambda \cdot \delta(p)$$

where $\delta(p) = \sup_{z,t} \| f\,(z,p,t) \quad f^*(z,p,t) \|$ *is the field error at condition p.*

Key implication: Field errors are exponentially amplified as T grows.

Two Sources of Field Error → Two Stages

Stage 1 — Problem:

High estimation error at core conditions (low density)

→ Concentrate budget on repeated anchor demos → stabilize the policy skeleton

Stage 2 — Problem:

High extrapolation bias at boundary conditions (large hbd)

→ Add targeted boundary demos where deviation score e(p) is highest → patch coverage gaps

Deviation Score as Proxy for Boundary Risk

$$e(p) = (1/T) \cdot \Sigma \quad \| \quad (p) \quad a\,(p) \|_1$$

Teacher- forced deviation e(p) is computed by running the Stage- 1 policy on demo observations (no state compounding) and comparing predicted vs. demonstrated actions. High e(p) ↔ high extrapolation risk at condition p.

Anchor-Centric Adaptation (ACA): Framework Overview

A two-stage pipeline operationalizing the Coverage-Density trade-off

Budget Partition: $N = N$ (anchoring) + N_{probe} (screening) + N_{bd} (boundary expansion)

Stage 1: Anchor-Centric Stabilization (Densification)

Reduce estimation variance by concentrating demos at a sparse set of core anchors

Select Anchors {p }

Form a coarse, quasi- uniform cover of the reachable workspace. K anchors balancing coverage and density (guided by K*).

Collect Repeated Demos

Collect n > 1 demonstrations per anchor via leader- follower teleoperation. Concentrated repetition suppresses estimation noise.

Train Action Expert θ

Minimize flow-matching objective on anchor dataset . Freeze the pretrained VLM to preserve multimodal representations.

Training Objective

θ = argmin_θ E_{(a,p)~ } [_FM(θ; a, p)]

VLM: Frozen → preserve pretrained multimodal features
Action Expert: Trained → adapts to target embodiment
Budget: ~80% of total N allocated here

What Stage 1 Delivers

✓ Low-variance, stable policy skeleton at anchors
✓ Reliable performance in well-supported regions
May still fail at un-covered boundary conditions

Boundary Mining via Teacher-Forced Deviation

Efficiently identifying high-risk regions where Stage-1 policy will fail

Probe Collection

Collect N_probe demo trajectories at conditions sampled across the workspace (one trajectory per candidate condition). Forms candidate set _cand.

Teacher-Forced Deviation Score

$$e(p) = (1/T)\ \Sigma \quad \| \quad (p) \quad a\ (p) \|_1$$

Decode actions using θ on the demonstration observation sequence. Teacher- forcing prevents state-drift compounding, making e(p) a clean proxy for action prediction error.

Select Top-k Boundaries

Rank all probe conditions by e(p). Select the k highest-scoring conditions as boundary set _bd. The k probe trajectories are reused as boundary supervision — no extra collection cost.

Local Boundary Expansion

Around each selected p^(j), collect additional demos via random perturbations within a small neighborhood. Targeted coverage reduction in the most dangerous regions.

Key advantage: teacher- forcing avoids compounding error, giving a clean, unbiased signal for boundary identification.

Stage 2: Constrained Residual Adaptation (Expansion)

Patch boundary behavior without drifting the stable core from Stage 1

Residual Architecture

$$f_{final}(z,p,\tau) = f_{\theta}(z,p,\tau) + \Delta\varphi(z,p,\tau)$$

f_{θ}: Frozen Stage-1 base policy
$\Delta\varphi$: Trainable residual branch (LoRA)

Zero-init: $\Delta\varphi$ starts at 0 → optimization begins exactly at θ

LoRA rank = 32, α = 32
Applied to selected linear layers of Action Expert
Flow-time modulation τ → residual specializes per denoising step

Why Residual, not Full Fine-tuning?

- → Full fine-tuning on boundary data drifts the consolidated core → Stage-1 stability is lost
- → Residual architecture constrains updates to a low-capacity subspace
- → Ablation (Table 2): full FT yields ↑ S@3 but ↓ S@1/S@2 — exactly this trade-off

Why LoRA?

- → Parameter-efficient: only ~0.5% of parameters trained
- → Low-rank constraint naturally limits drift
- → Compatible with VLA Action Expert architecture

Theoretical Connection

- → Prop. 3.1: boundary error ≈ $L \cdot h_{bd}$ (extrapolation dominated)
- → Stage 2 reduces h_{bd} by adding K2 boundary conditions
- → Residual keeps estimation error on anchors unchanged

Experimental Setup

Real-robot evaluation designed to capture non-vanishing estimation noise

Hardware

Robot:
7-DoF Franka Panda

Cameras:
2× RealSense D435 + wrist D405

Demos:
Leader-follower teleoperation

GPUs:
8× NVIDIA H200

Backbone:
π0 and π0.5 (flow-matching VLA)

4 Real-Robot Tasks

Block Stacking

Place a cube onto a target block; stable stack required

Cup Placement

Position a cup precisely onto a coaster

Table Cleaning

Sweep a block into a dustpan with a brush

Toy Tidying

Place toys into a storage box

Evaluation Metrics

S@1	25% area Core (easiest)
S@2	50% area Medium
S@3	90% area Near-boundary (hardest)

20 trajectories per task per region.
Inspired by Recall@K in IR.

Main Results: Consistent Gains Across Tasks and Budgets

Table 1 — ACA vs. diversity-first π0.5 baseline under varying data budgets

Mean gain at N=100
+40.8%

ACA @N=50 ≈ Baseline @N=150
3×

S@3 improvement (Block Stacking, N=150)
0%→65%

Best mean success (N=150, ACA)
83.8%

Method	Block Stacking S@1 / S@2 / S@3	Cup Placement S@1 / S@2 / S@3	Table Cleaning S@1 / S@2 / S@3	Toy Tidying S@1 / S@2 / S@3	Mean (%)
N=50					
π0.5 (baseline)	5 / 2 / 0	6 / 4 / 0	2 / 0 / 0	8 / 4 / 2	13.8
π0.5 + ACA	10 / 12 / 9	14 / 10 / 10	7 / 6 / 5	12 / 10 / 6	46.3 (+32.5)
N=100					
π0.5 (baseline)	8 / 2 / 0	14 / 6 / 3	10 / 2 / 1	14 / 10 / 6	31.7
π0.5 + ACA	16 / 14 / 13	18 / 17 / 14	12 / 11 / 8	18 / 16 / 17	72.5 (+40.8)
N=150					
π0.5 (baseline)	14 / 8 / 4	18 / 13 / 8	9 / 7 / 4	18 / 14 / 10	52.9
π0.5 + ACA	17 / 18 / 16	20 / 20 / 18	14 / 11 / 9	20 / 19 / 19	83.8 (+30.9)

S@k = success rate within the k-th nested workspace region (20 trials each). Results shown as # successes / # successes / # successes out of 20.

Ablations: Anchor Count K and Spatial Distribution

Validating the inverted-U trend and centralization principle

Impact of Anchor Count K

① **Inverted-U confirmed:** Task success is non-monotonic in K.
② **Diversity trap at high K:** Beyond K*, success declines — density dilution destabilizes the policy.
③ **Budget-dependent optimum:** Larger N supports higher K* without sacrificing stability — consistent with Corollary 3.3.

Impact of Anchor Spatial Layout

→ All configurations beat baseline:

Even biased top-left layout: 4× S@3 improvement (51.0% vs 12.5%)

→ Geometry pattern matters less:

Rect vs. Circle difference is marginal — centralization dominates geometry

Conclusion

Anchor-Centric Adaptation (ACA)

Empirical Insight

Identified the Diversity Trap: maximizing coverage under tight budgets destabilizes VLA adaptation — evidenced by a characteristic inverted-U performance trend.

Theoretical Formalization

Coverage-Density Trade-off framework: worst-case error decomposes into estimation (density) and extrapolation (coverage) terms, with an interior optimal $K^* < N$.

Practical Framework

ACA: two-stage pipeline (anchor densification + error-driven boundary expansion + constrained residual) delivering up to +40.8% improvement on real-robot tasks.

Take-Home Message: Structured repetition is a more potent scaling lever than raw diversity for budget-constrained robot learning.
Repeat first, expand second — and let the data tell you where to expand.

Future Work: Hierarchical anchor decomposition for long-horizon tasks · Autonomous anchor discovery · Extension to broader data distributions