FRAME: Framework for Robotic Action and Motion Evaluation

Ameya Wagh [1] **Vishnu Rudrasamudram** [1]

Abstract

The rapid emergence of Vision-Language-Action (VLA) models has fundamentally shifted robotics toward end-to-end, generalist architectures capable of complex semantic reasoning. However, the lack of a unified evaluation standard remains a critical bottleneck, as research often relies on disparate metrics that fail to bridge the gap between offline action accuracy and online physical deployment. We introduce FRAME, a comprehensive, open-source evaluation framework built natively on TorchMetrics to facilitate reproducible and scalable robot policy research. FRAME provides a modular taxonomy of 15+ standardized metrics across four critical dimensions: Task Performance, Trajectory Quality, Safety, and Efficiency. Safety metrics (collision rate, obstacle proximity, risk factor) are implemented in the library; their deployment requires contact or proximity sensing signals, which we leave to future hardware-instrumented experiments. We demonstrate the utility of FRAME through an empirical study of four pre-trained policies (Diffusion Policy, VQ-BeT, and ACT) on the PushT manipulation task, and a physical robot evaluation of SmolVLA and pi_0 on the SO-101 manipulator. Our analysis reveals significant discrepancies between traditional success rates and trajectory-level quality, highlighting critical failure modes that binary metrics alone cannot capture. By providing interpretable process metrics and trace-level diagnostics, FRAME enables a more nuanced understanding of robot policy performance and establishes a common language for identifying and mitigating failure modes in embodied AI.

[1]IEEE Members. Correspondence to: Ameya Wagh <ameyawagh@ieee.org>, Vishnu Rudrasamudram <vishnurs@ieee.org>.

Accepted to the 1st Workshop on Combining Theory and Benchmarks, CTB@ICML 2026, Seoul, South Korea.

1. Introduction

1.1. Background and Motivation

Traditional robotics systems historically relied on modular architectures separating perception, planning, control, and task specific components. Initially, robotics was heavily based on meticulously hand-crafted mathematical models and heuristic algorithms. As the field of AI and machine learning progressed, the perception system and planning were replaced by end-to-end learning-based algorithms(Xiao et al., 2025; Khan & Waheed, 2025; Chen et al., 2024). As these systems were developed separately, they had individual evaluation criteria and metrics, and high level integration metrics.

Recent progress in foundation models enabled generalist Vision Language Action Models (VLAs) to achieve impressive results by fusing multi-modal information such as images, textual instructions and robot state to generate robot actions. They are highly capable of reasoning and show complex understanding of the world (Brohan et al., 2023; Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2026). Similarly, new paradigms have emerged in learning of behavior policy through imitation learning and reinforcement learning, such as ACT (Zhao et al., 2023), Diffusion-Policy (Chi et al., 2023) and VQ-Bet (Lee et al., 2024).

Evaluating end-to-end systems predominantly relies on binary criteria like success rates, which lack detailed performance insights into the failure modes of the policy. Two policies might share identical success rates, but trace-level diagnostics may reveal one is smooth yet slow, while the other is fast but jerky. Granular insights are crucial when trajectory quality matters, such as handling liquids. While recent papers propose ad-hoc metrics, existing frameworks remain fragmented, lacking unified trajectory analysis across diverse platforms. This motivates a standardized framework easily integrated into training pipelines to extract interpretable process metrics.

1.2. Contributions

The primary contribution of this work is FRAME, a lightweight extensible evaluation library built on TorchMetrics(Nicki Skafte Detlefsen et al., 2022) for trace-level diagnostics and fine-grained metrics. We provide the fol-

Table 1. Evaluation Metric Taxonomy

Category	Metric	Purpose
Task Performance	Success Rate, SR (Brohan et al., 2023)	% of tasks successfully finished.
	Task Completion Rate, TCR (Brohan et al., 2023)	Fraction of multi-step tasks fully executed.
	Mean Squared Error, MSE (Dobiš et al., 2022)	MSE between prediction and actions.
	Avg./Norm. MSE (Dobiš et al., 2022; Farag et al., 2021)	Action-error mean and scale-normalised.
Trajectory Quality	Path Length, PL (Fankhauser et al., 2015)	Total distance travelled in trajectory.
	Path Smoothness, PS (Guillén Ruiz et al., 2020)	Avg change in heading; detects oscillations.
	Curvature Change, CC (Hwang et al., 2004)	Variation in local curvature along path.
	Absolute Traj. Error, ATE (Sturm et al., 2012; Endres et al., 2012)	Global pose deviation from ref trajectory.
	Relative Traj. Error, RTE (Sturm et al., 2012; Endres et al., 2012)	Local drift over fixed temporal windows.
Safety & Robustness	Collision Rate, CR (Hoy et al., 2015)	Collisions per time-step; core safety indicator.
	Obstacle Proximity, OP (Blunder et al., 2022)	Closest distance to any obstacle encountered.
	Risk Factor, RF (Majumdar & Pavone, 2020)	Path-wide avg inverse distance to obstacles.
Efficiency	Inference Latency, IL (Brohan et al., 2023)	Wall-clock time for one model inference.
	Computation Time, CT (Hartmanis & Stearns, 1965)	Proc cycles consumed per decision step.
	Memory Usage, MU (Wang & Sun, 2014)	Peak RAM + VRAM used during execution.

lowing:

- Unified Python Framework: An extensible library that integrates directly into PyTorch loops for real-time monitoring of 15+ metrics across task performance, trajectory quality, safety, and efficiency. Code available at: `https://github.com/AmeyaWagh/robometric-frame`.
- Empirical Study: A holistic evaluation of four pre-trained policies (Diffusion Policy pixel,Diffusion Policy keypoint, VQ-BeT, ACT) in simulation and fine-tuned policy (SmolVLA) and pi_0 on a physical robot, revealing that trajectory-level metrics expose failure modes invisible to success rate alone.
- Discovery of the "Evaluation Gap": An empirical demonstration that standard Success Rates can mask critical differences in trajectory smoothness and efficiency, with policies at similar SR exhibiting substantially different motion quality profiles.

2. Related Work

The rapid evolution of Vision-Language-Action (VLA) models has necessitated a shift from ad-hoc testing to rigorous, standardized benchmarking.

Simulation-Based Benchmarks: Foundational work in this domain focuses on task diversity and reproducibility. RLBench (James et al., 2020), a large-scale simulation benchmark, established the standard for multi-task learning by providing 100 uniquely designed manipulation tasks ranging from simple reaching to complex multi-stage sequences. Building on this foundation, The Colosseum (Pumacay et al., 2024) systematically evaluates generalization by stress-testing policies across 14 axes of environmental perturbations, such as lighting, texture, and physical property changes (mass, friction). While these benchmarks

excel at assessing task success under varying conditions, they primarily rely on binary outcomes in simulation.

Autonomous and Trustworthy Evaluation: Addressing the scalability bottleneck of physical deployment, recent works have focused on automating the evaluation pipeline. AutoEval (Zhou et al., 2025) introduces "round-the-clock" protocols using a secondary Vision-Language Model (VLM) to score real-world task progression, reducing manual monitoring effort by over 99%. Complementing this, Trustworthy Evaluation of Robotic Manipulation (Liu et al., 2026) tackles the "credibility crisis" in reported metrics. It proposes a framework to replace opaque success rates with transparent, fine-grained assessments, utilizing the Eval-Actions dataset to verify the authenticity of autonomous trajectories.

Multi-Task Benchmarks: Several large-scale benchmarks have established standard evaluation protocols for manipulation policies. CALVIN (Mees et al., 2022) evaluates long-horizon, language-conditioned manipulation across 34 tasks in four environments, introducing chained task-completion as a richer success metric. LIBERO (Liu et al., 2023) provides 130 manipulation tasks organized into suites targeting knowledge transfer and lifelong learning. Meta-World (Yu et al., 2020) offers 50 tabletop tasks with a multi-task protocol widely used to assess policy breadth. While these benchmarks differ in task scope, they share a reliance on binary success rates and do not measure trajectory quality. The Open X-Embodiment (Open X-Embodiment Collaboration, 2023) dataset and Octo (Octo Model Team et al., 2024) generalist policy further highlight the pressing need for evaluation frameworks that scale across robot morphologies, a gap that FRAME directly addresses.

Concurrent VLA Evaluation Frameworks: Several recent benchmarks propose complementary evaluations (Table 2). VLA-Arena (Zhang et al., 2025) tests models in interactive 3D environments with language goals, while SIMPLER (Li et al., 2024) evaluates sim-to-real correlation. REALM (Sedlacek et al., 2025) provides robust trajectory alignment protocols, and MolmoSpaces (Kim et al., 2026) focuses on large-scale spatial reasoning. RoboEval (Wang et al., 2025) explores trajectory metrics like jerk, but focuses on post-hoc evaluation. However, these benchmarks remain external to policy learning. Crucially, *none integrate natively into the PyTorch training loop* or expose fine-grained metrics (smoothness, curvature, safety) during development. While FRAME currently uses these metrics for real-time logging, its PyTorch-native implementation enables future use as auxiliary losses. FRAME is designed to augment, not replace, these simulators: its TorchMetrics layer can integrate *inside* pipelines like SIMPLER or VLA-Arena to provide continuous trajectory tracking alongside binary success.

While these frameworks address environmental robustness and autonomous monitoring, they often treat the policy's execution quality as a black box. There remains a lack of a unified, training-native ecosystem that evaluates how the motion is executed: specifically, quantifying smoothness, safety, and efficiency alongside success. FRAME bridges this gap by introducing a comprehensive, PyTorch-native taxonomy that integrates directly into training/eval loops. Unlike previous works that focus solely on "did it succeed?", FRAME provides comparable trace-level diagnostics and an early signal of performance decay, establishing a critical link between offline accuracy and online physical reality.

3. FRAME Methodology

3.1. Architecture Overview

FRAME provides a TorchMetrics-compatible modular library for a comprehensive Robot policy evaluation. Figure 1 illustrates the framework architecture, which supports both online computation during training/evaluation and offline analysis of recorded trajectories.

3.2. Metric Taxonomy

The FRAME suite is organized into a four-tier taxonomy, transitioning from high-level task goals to low-level hardware constraints described in Table 1.

3.2.1. Task Performance Metrics

These metrics provide a top-down assessment of a model's ability to interpret instructions and execute the correct sequence of actions. **Success Rate (SR)**: This metric provides a top-down assessment of a model's ability to interpret instructions and execute correct action sequences. However, SR alone often masks nuances in long-horizon tasks. We therefore implement the Task Completion Rate (TCR) to evaluate multi-step sequence execution:

$$SR = \frac{N_{\text{success}}}{N_{\text{total}}} \tag{1}$$

$$TCR = \frac{N_{\text{completed tasks}}}{N_{\text{task chains}}} \tag{2}$$

Action Accuracy: Measures the precision of predicted actions against ground truth trajectories, providing a direct assessment in offline scenarios where physical deployment is unfeasible. We evaluate using Mean Squared Error (MSE) and its variations, utilizing Normalized Average MSE (NAMSE) to account for action scale variance:

Table 2. Comparison of recent robotic evaluation frameworks. FRAME provides a training-native metrics taxonomy covering trajectory quality and efficiency; safety metrics (CR/OP/RF) are implemented in the library but require contact/proximity sensing not available in the current experiments.

Framework	Primary Focus	Trajectory Quality	Safety Metrics	PyTorch-Native Integration	Continuous Monitoring
VLA-Arena (Zhang et al., 2025)	Interactive 3D Environments	×	×	×	×
SIMPLER (Li et al., 2024)	Sim-to-Real Fidelity	×	×	×	×
REALM (Sedlacek et al., 2025)	Trajectory Alignment	✓	×	×	×
MolmoSpaces (Kim et al., 2026)	Spatial Reasoning	×	×	×	×
RoboEval (Wang et al., 2025)	Post-hoc Trajectory Eval	✓	×	×	×
FRAME (Ours)	**Continuous Policy Benchmarking**	**✓(e.g., Smoothness, ATE)**	**✓(e.g., Collision, Risk)**	**✓(TorchMetrics)**	**✓(During Training)**

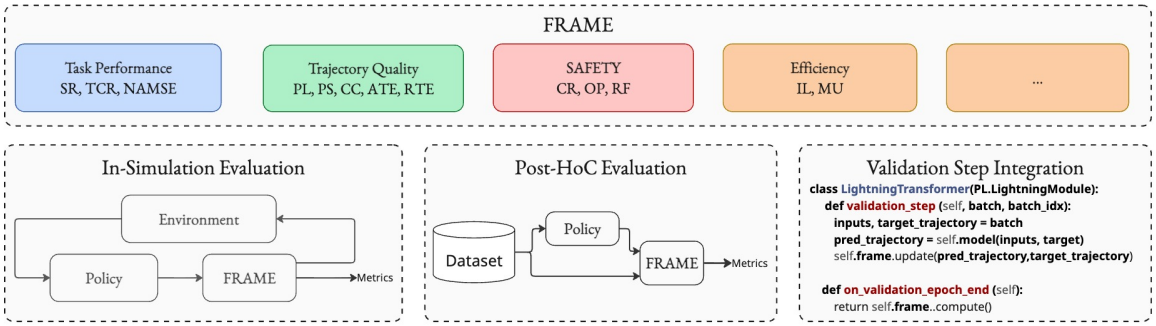


Figure 1. FRAME architecture. The framework integrates with PyTorch training loops via TorchMetrics, computing metrics from trajectory data produced by policy rollouts. Four metric categories (Task, Trajectory, Safety, Efficiency) enable comprehensive evaluation with native logging support.

$$MSE = \frac{1}{T}\sum_{t=1}^{T} \|\mathbf{a}_t - \hat{\mathbf{a}}_t\|_2^2 \tag{3}$$

$$AMSE = \frac{1}{K}\sum_{k=1}^{K} MSE_k \tag{4}$$

$$NAMSE = \frac{AMSE}{\sigma^2_{\text{action}}} \tag{5}$$

where $\mathbf{a}_t$ is the action vector, T is the sequence length and K is the number of trajectories

Although Task Performance measures what was achieved, it ignores the efficiency of physical movement.

3.2.2. Trajectory Quality Metrics

These metrics evaluate the "finesse" and mechanical viability of the robot's motion. **Path Length (PL)**: Quantifies the total distance traveled. Shorter paths indicate higher execution efficiency (Fankhauser et al., 2015) given by the equation 6 where p_i is the position and L is the number of samples in the trajectory.

$$PL = \sum_{i=1}^{L-1} \|\mathbf{p}_{i+1} - \mathbf{p}_i\|_2 \tag{6}$$

Path Smoothness (PS) (Dobiš et al., 2022) Given by equation 7 evaluates the rate of change in direction of the trajectory, detecting oscillations that may arise from velocity changes or directional adjustments (Guillén Ruiz et al., 2020). Smooth trajectories are essential for safe robot operation and reduced mechanical wear. **Curvature Change (CC)** (Hwang et al., 2004) provides a specialized evaluation for mobile robots, measuring trajectory smoothness while accounting for mobile robot heading or end-effector orientation given by equation 8 where θ is the orientation. This metric proves particularly valuable for car-like mobile robots where curvature directly relates to turning radius constraints. Unlike path smoothness, the curvature change incorporates the angular velocity, providing a more comprehensive trajectory assessment.

$$PS = \frac{1}{L-2}\sum_{i=1}^{L-2} \|(\mathbf{p}_{i+2} - \mathbf{p}_{i+1}) - (\mathbf{p}_{i+1} - \mathbf{p}_i)\|_2 \tag{7}$$

$$CC = \frac{1}{L-2}\sum_{i=1}^{L-2} |\kappa_{i+1} - \kappa_i|, \quad \kappa_i = \frac{\theta_{i+1} - \theta_i}{\|\mathbf{p}_{i+1} - \mathbf{p}_i\|_2} \tag{8}$$

Sampling note: Both PS and CC are sensitive to the trajectory sampling rate; we compute them on raw control-frequency data (50 Hz for PushT) without additional filtering. For deployments with noisy or low-frequency sensors, we recommend applying a low-pass filter prior to metric computation and reporting the sampling rate alongside results.

Trajectory Errors (ATE/RTE): The trajectory error encompasses Absolute Trajectory Error (ATE) and Relative

Trajectory Error (RTE), measuring deviations between the predicted and reference trajectories (Sturm et al., 2012). ATE evaluates global consistency, while RTE evaluates local accuracy, both critical for navigation and manipulation tasks requiring precise positioning(Endres et al., 2012).

$$ATE = \frac{1}{L}\sum_{i=1}^{L} \|\mathbf{p}_i - \mathbf{p}_i^*\|_2 \tag{9}$$

$$RTE = \frac{1}{L-\Delta}\sum_{i=1}^{L-\Delta} \|(\mathbf{p}_{i+\Delta} - \mathbf{p}_i) - (\mathbf{p}_{i+\Delta}^* - \mathbf{p}_i^*)\|_2 \tag{10}$$

In our framework, reference trajectories are typically sourced from expert demonstrations or planner-generated paths; for tasks without a canonical ground truth reference, ATE and RTE can be computed post-hoc to evaluate deviations against successfully completed rollouts. For multimodal tasks where multiple valid paths exist to achieve the same goal, ATE and RTE should be computed against a distribution of expert demonstrations rather than a single trajectory, often by tracking the minimum distance to any valid reference path.

3.2.3. Safety and Robustness Metrics

Collision Rate (CR) quantifies the frequency of collisions during task execution, serving as a primary safety indicator (Hoy et al., 2015). This metric is particularly critical for mobile robots and humanoids operating in human environments where safety is paramount. **Obstacle Proximity (OP)** measures the minimum distance between the robot and environmental obstacles throughout task execution. This metric provides insights into safety margins and risk assessment capabilities of VLA models in cluttered environments(Blunder et al., 2022). The **Risk Factor (RF)** offers a comprehensive safety assessment (Majumdar & Pavone, 2020) by integrating proximity measurements throughout the route. Calculated as the average of the reciprocal distances from obstacles, this metric provides a holistic assessment of safety-conscious behavior.

$$CR = \frac{N_{\text{collisions}}}{T_{\text{steps}}} \tag{11}$$

$$OP = \min_t d_t^{\text{robot}\to\text{obstacle}} \tag{12}$$

$$RF = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{d_t} \tag{13}$$

Instrumentation note: CR, OP, and RF require real-time contact or proximity signals (e.g., force-torque sensors, depth cameras, or collision checkers in simulation). In the current experiments these signals were not available; accordingly, safety metrics are not reported in the results tables. Demonstrating these metrics on instrumented hardware is a primary direction for future work.

3.2.4. Efficiency Metrics

Inference Latency (IL): Inference Latency measures the time required to generate actions from inputs. This metric is crucial for real-time applications where responsive behavior is essential to understand behavior of policies such as action chunking. **Memory Usage (MU)**: Memory Usage assesses resource consumption during operation (Wang & Sun, 2014), particularly relevant for compact VLA models such as SmolVLA-450M (Shukor et al., 2025) designed for consumer hardware deployment.

$$IL = t_{\text{infer,end}} - t_{\text{infer,start}} \tag{14}$$

$$MU = \max_t(\text{RAM}_t + \text{VRAM}_t) \tag{15}$$

4. Experimentation And Results

4.1. Evaluating trained Pre-Policies

We conduct a systematic evaluation of pre-trained policies from the LeRobot model hub to demonstrate the value of multi-metric assessment. We chose PushT manipulation task from the LeRobot (Capuano et al., 2025) suite, as it has official releases of different vla and non-vla policies.

4.1.1. Evaluation setup

We evaluated and analyzed four different class of policies: **Diffusion Policy** (Chi et al., 2023) (both pixel-based and keypoint-based variants), **VQ-BeT** (Lee et al., 2024), and **ACT** (Zhao et al., 2023). Comparisons are conducted on an Nvidia RTX 4080 GPU. To ensure statistical robustness, each policy is evaluated over $N = 50$ episodes with different random seeds, collecting high-frequency trajectory data for post-hoc analysis.

4.1.2. Results

Table 3 presents the comprehensive evaluation results. Beyond the success rate, we report the PL, PS, CC, ATE, RTE, IL and MU. The results reveal several insights that success rate alone would obscure:

Trajectory Efficiency: Diffusion (keypoints) achieves the highest success rate (74.0%) with the most efficient path length (1647.3). In contrast, ACT exhibits the lowest success rate (34.0%) and the longest path length (2105.4), suggesting that failed episodes often involve inefficient or wandering trajectories before termination.

Motion Quality: Diffusion-based policies demonstrate superior motion quality, with Path Smoothness scores of

Table 3. Pre-trained Policy Evaluation on PushT Task. Values reported as mean ± std across seeds. Best values in **bold**.

Policy	SR ↑	TCR ↑	PL ↓	PS ↓	CC ↓	ATE ↓	RTE ↓	IL ↓	MU ↓
Diff. (kp)	**74.0 ± 3.6**	**74.0 ± 3.6**	**1647.3 ± 24.8**	2.384 ± 0.025	0.490 ± 0.023	175.6 ± 2.3	**12.4 ± 0.2**	7.0 ± 0.0	**1665.7 ± 79.2**
Diff. (px)	66.0 ± 2.9	66.0 ± 2.9	1833.9 ± 84.2	**2.338 ± 0.032**	**0.368 ± 0.027**	**172.2 ± 1.1**	12.7 ± 0.3	64.0 ± 0.0	1827.3 ± 15.3
VQ-BeT	63.7 ± 4.1	63.7 ± 4.1	1739.9 ± 101.3	2.903 ± 0.035	0.667 ± 0.110	175.4 ± 1.2	12.5 ± 0.1	**1.0 ± 0.0**	1952.0 ± 10.6
ACT	34.0 ± 1.4	34.0 ± 1.4	2105.4 ± 51.2	2.838 ± 0.044	0.679 ± 0.014	178.9 ± 2.6	12.6 ± 0.1	**1.0 ± 0.0**	2073.9 ± 0.2

SR: Success Rate (%), TCR: Task Completion Rate (%), PL: Path Length (px), PS: Path Smoothness, CC: Curvature Change, ATE: Absolute Trajectory Error, RTE: Relative Trajectory Error, IL: Inference Latency (ms), MU: Memory Usage (MB)

2.338–2.384, lower (better) than VQ-BeT (2.903) and ACT (2.838). This indicates a more consistent and less jerky motion, which is critical to reducing mechanical wear and ensuring safe operation in real-world deployment.

Latency-Performance Trade-offs: A distinct trade-off exists between model complexity and inference speed. Although Diffusion (pixels) achieves a high success rate (66.0%), it requires 64.0ms per inference step versus ≈1.0ms for ACT and VQ-BeT. However, these policies differ in input modality (pixels vs. keypoints), action chunk size, and control frequency, so latency figures should not be interpreted as a controlled comparison. They are reported here to illustrate FRAME's capacity to surface efficiency bottlenecks; normalizing by control frequency or chunking factor is recommended for apples-to-apples comparisons.

Trajectory Accuracy (ATE/RTE): The inclusion of Absolute and Relative Trajectory Errors provides granular insight into policy tracking behavior. Diffusion (pixels) achieves the best global tracking accuracy (lowest ATE: 172.2 ± 1.1), ensuring the end-effector remains close to the reference path. However, Diffusion (keypoints) maintains the best local motion consistency (lowest RTE: 12.4 ± 0.2), indicating that while its global path may deviate slightly more, its step-to-step relative movements are highly precise. These reference paths are sourced directly from the LeRobot expert dataset. For multi-modal tasks, the framework dynamically calculates the error against the nearest valid expert rollout.

Characterization of failure mode: The combination of metrics enables a richer failure analysis. For instance, ACT's high path length coupled with low success rates suggests that the policy attempts the task but struggles with precision or efficiency, rather than failing to initiate motion. While these results strongly motivate the existence of an "evaluation gap" between binary success and trajectory quality, we acknowledge that a formal statistical substantiation of this gap (e.g., via rank-correlation analysis across a wider variety of tasks) remains an important direction for future work.

4.2. Evaluating Metrics on Robot

To demonstrate FRAME's practical utility on physical hardware, we evaluated policies on the SO-101 manipulator. The hardware configuration, camera setup, and fine-tuning details are provided in Appendix A.

We evaluated the finetuned variant of smolVLA (Shukor et al., 2025) and π_0(Black et al., 2026) on the SO-101 physical robot, recording $N = 10$ evaluation episodes with manual success annotation. This limited sample size serves primarily as a demonstration of FRAME's feasibility in a real-world deployment scenario. Trajectory metrics (PL, PS, CC) were computed from recorded end-effector pose data using FRAME, and we report mean ± std across episodes in Table 4. Although online calculations of complex metrics are supported by our framework, we chose to perform post-hoc evaluation with forward kinematics to compute in task space on recordings for simplicity of experimentation.

Qualitatively, successful runs were characterized by smooth, purposeful motion towards the target, reflected in low PS values. As observed in Table 4, failed episodes often exhibited hesitation or high-frequency oscillations near the target object before missing the grasp. This can be observed in PL and PS being close to SmolVLA metrics despite the 0% vs. 50% success rate gap, suggesting π_0 initiates correct motion but fails at the grasp precision stage. Given $N = 10$, this is treated as a qualitative observation rather than a comparative benchmark claim.

By quantifying these behaviors, we can distinguish between "near-miss" failures (conceptually correct but imprecise) and catastrophic failures (out of distribution behavior), providing actionable insights for further fine-tuning.

4.3. Training-Time Metric Integration

Beyond post-hoc evaluation of pre-trained policies, FRAME can be integrated directly into a training loop to monitor trajectory quality as a policy learns. To demonstrate this, we integrated FRAME into the LeRobot training pipeline for ACT (Zhao et al., 2023) on the PushT task, computing trajectory quality metrics at each evalua-

Table 4. Fine-tuned policy evaluation on SO-101 physical robot. Values reported as mean ± std across $N = 10$ evaluation episodes. Best values in **bold**.

Policy	SR ↑	TCR ↑	PL (mm) ↓	PS ↓	CC ↓	IL (ms) ↓	MU (MB) ↓
smolVLA	**0.50**	**0.50**	**2517 ± 1954**	0.592 ± 0.045	0.440 ± 0.053	**4.1**	**887**
π_0 †	0.0	0.0	2903.2 ± 747	**0.58 ± 0.024**	**0.28 ± 0.0002**	28.5	5312

SR: Success Rate, TCR: Task Completion Rate, PL: Path Length (mm), PS: Path Smoothness (cart.), CC: Curvature Change (cart.), IL: Inference Latency (ms), MU: Memory Usage (MB). IL and MU for smolVLA measured on Nvidia RTX 4080.

tion checkpoint (every 2,000 gradient steps) alongside standard training losses. The goal of this experiment was to explore the training curve trends and to not achieve a best performing model.

Table 5 illustrates the progression of FRAME metrics at intermediate checkpoints over a single training run. Crucially, the standard **Success Rate (SR)** metric remains stagnant during certain training intervals, stuck at 0% from 2,000 to 4,000 steps, and flatlined at 2% from 6,000 to 10,000 steps. Under a traditional evaluation framework relying only on task success, the policy appears to show no significant progress during these periods. However, FRAME's trajectory metrics reveal a markedly different picture: looking at the progression from 6,000 to 10,000 steps, the trajectory quality actively improves, demonstrated by consistently smoother motion (PS decreases from 0.25 to 0.24) and falling training loss (0.09 to 0.073).

This illustrates FRAME's core value for continuous monitoring during training: even when SR plateaus and provides no useful learning signal, trajectory metrics continue to provide a differentiating gradient. This granular insight confirms whether a policy is steadily converging toward smooth, purposeful motion or stagnating.

Table 5. FRAME metrics logged during ACT training on PushT. While Success Rate plateaus over several epochs, trajectory metrics like Path Smoothness (PS) provide a continuous signal of policy improvement.

Steps	Loss ↓	SR ↑	PL (px) ↓	PS ↓
2,000	0.19	0%	1745	0.35
4,000	0.11	0%	2021.4	0.28
6,000	0.09	2%	2044.5	0.25
8,000	0.079	2%	2033.5	0.24
10,000	0.073	2%	2122.8	0.24

PL: Path Length (px), PS: Path Smoothness.

4.4. Holistic Policy Comparison

To holistically compare heterogeneous metrics with disparate scales (e.g., success rate in $[0, 1]$ vs. memory usage in MB), we employ a *ratio-to-best* normalization (Equation 16). This formulation preserves relative performance and enables unified evaluation by natively scaling higher-is-better metrics (SR, TCR) while inverting the ratio for lower-is-better metrics (PL, PS, CC, ATE, RTE, IL, MU).

$$\hat{m}_i^{high} = \frac{m_i}{\max_j m_j} \qquad \text{and} \qquad \hat{m}_i^{low} = \frac{\min_j m_j}{m_i} \quad (16)$$

When visualized as a radar chart in figure 2, this normalization yields an intuitive policy fingerprint that enables rapid qualitative comparison of multi-dimensional trade-off patterns. However, because the visual area of a radar polygon is strictly dependent on arbitrary axis ordering, it cannot serve as a mathematically valid scalar aggregate.

Instead, to rigorously compare policies, we calculate their 2D Pareto Hypervolume across the two most critical axes of VLA deployment: task capability (Success Rate) and execution efficiency (Inference Latency). This order-independent metric quantifies the exact objective space each policy dominates, revealing that Diffusion (keypoints) defines the capability frontier while VQ-BeT anchors the efficiency boundary. Crucially, this analysis mechanically filters out strictly inferior models; because ACT shares VQ-BeT's 1.0ms latency but yields a significantly lower success rate, it is entirely Pareto-dominated, contributing zero marginal hypervolume and mathematically proving it suboptimal for this specific trade-off.

5. Future Work

We aim to expand FRAME in several key directions. First, as an open-source initiative, we actively invite community contributions to broaden the metric taxonomy across diverse robotic embodiments. Second, we plan to develop **language-grounding metrics** that specifically evaluate semantic comprehension, quantifying how faithfully a policy executes natural language instructions. Third, while our current physical robot evaluation has a limited sample size, we intend to expand to a larger number of episodes across multiple tasks to establish statistically significant benchmarks for training-native metric monitoring. Finally, we intend to utilize FRAME's trace-level diagnostics to identify **reproducible triggers** for out-of-distribution failures. This will enable researchers to systematically evaluate mitigation and recovery strategies when a robot policy begins to drift, tying operational definitions of failure directly to measurable trajectory degradation.

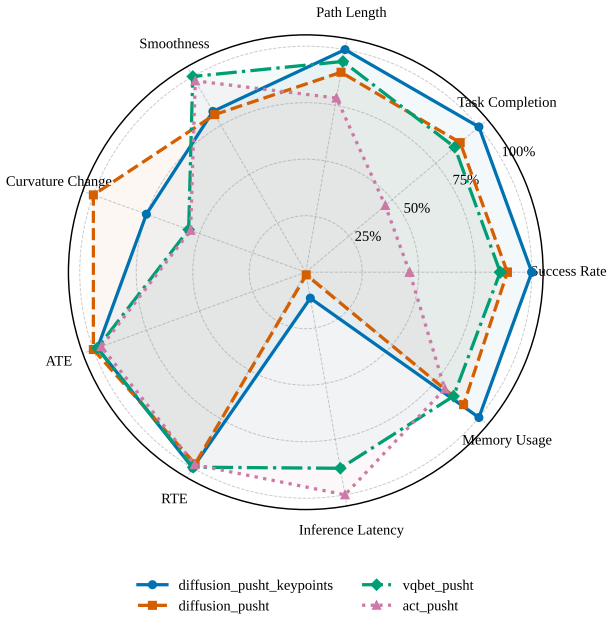


Figure 2. Qualitative policy fingerprints via ratio-to-best normalization. The chart illustrates multi-dimensional trade-offs, demonstrating how different architectures balance high-level capability with execution efficiency.

6. Conclusion

In this work, we introduced FRAME, a comprehensive evaluation framework library designed to address the critical benchmarking bottleneck in Vision-Language-Action (VLA) models. As the robotics community increasingly adopts end-to-end generalist policies, the standard reliance on binary success rates has proven insufficient, often masking critical failure modes in physical execution and safety.

By seamlessly integrating into distributed training and validation loops via TorchMetrics, FRAME empowers researchers to move beyond simply asking "did the robot succeed?" to rigorously diagnosing the exact mechanistic failures that caused it to fail. Ultimately, we release FRAME as an open-source tool to establish a standardized, reproducible language for extracting interpretable process metrics and improving the reliability of the next generation of embodied AI.

References

Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L. X., Tanner, J., Vuong, Q., Walling, A., Wang, H., and Zhilinsky, U. π_0: A vision-language-action flow model for general robot control. January 2026.

Blunder, N., Thiel, M., Schrick, M., Hinckeldeyn, J., and Kreutzfeldt, J. Integration and evaluation of a close proximity obstacle detection for mobile robots in public space. 2022.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, et al. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems XIX*, RSS2023. Robotics: Science and Systems Foundation, July 2023. doi: 10.15607/rss.2023.xix.025. URL http://dx.doi.org/10.15607/RSS.2023.XIX.025.

Capuano, F., Pascal, C., Zouitine, A., Wolf, T., and Aractingi, M. Robot learning: A tutorial. October 2025.

Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., and Li, H. End-to-end autonomous driving: Challenges and frontiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12): 10164–10183, December 2024.

Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B. C., and Song, S. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026.

Dobiš, M., Dekan, M., Beňo, P., Duchoň, F., and Babinec, A. Evaluation criteria for trajectories of robotic arms. *Robotics*, 11(1):29, February 2022.

Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., and Burgard, W. An evaluation of the RGB-D SLAM system. In *2012 IEEE International Conference on Robotics and Automation*. IEEE, May 2012.

Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M., and Siegwart, R. Kinect v2 for mobile robot navigation: Evaluation and modeling. In *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, July 2015.

Farag, K. K. A., Shehata, H. H., and El-Batsh, H. M. Mobile robot obstacle avoidance based on neural network with a standardization technique. *J. Robot.*, 2021:1–14, November 2021.

Guillén Ruiz, S., Calderita, L. V., Hidalgo-Paniagua, A., and Bandera Rubio, J. P. Measuring smoothness as a factor for efficient and socially accepted robot motion. *Sensors (Basel)*, 20(23):6822, November 2020.

Hartmanis, J. and Stearns, R. E. On the computational complexity of algorithms. *Trans. Am. Math. Soc.*, 117(0): 285–306, 1965.

Hoy, M., Matveev, A. S., and Savkin, A. V. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica*, 33(3):463–497, March 2015.

Hwang, J.-H., Arkin, R. C., and Kwon, D.-S. Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*. IEEE, 2004.

James, S., Ma, Z., Arrojo, D. R., and Davison, A. J. RLBench: The robot learning benchmark & learning environment. *IEEE Robot. Autom. Lett.*, 5(2):3019–3026, April 2020.

Khan, M. T. and Waheed, A. Foundation model driven robotics: A comprehensive review. *To appear*, 2025.

Kim, M. J., Pertsch, K., Karamcheti, S., et al. Openvla: An open-source vision-language-action model. In Agrawal, P., Kroemer, O., and Burgard, W. (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Kim, Y., Pumacay, W., et al. Molmospaces: A large-scale open ecosystem for robot navigation and manipulation, 2026. URL https://arxiv.org/abs/2602.11337.

Lee, S., Wang, Y., Etukuru, H., Kim, H. J., Shafiullah, N. M. M., and Pinto, L. Behavior generation with latent actions. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Li, X., Stengel-Eskin, E., Zhang, Y., et al. Evaluating real-world robot manipulation policies in simulation, 2024.

Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Stone, P., and Sun, Y. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Liu, M., Sheng, J., Li, P., Wang, Z., Xu, T., Xu, T., and Liu, H. Trustworthy evaluation of robotic manipulation: A new benchmark and AutoEval methods. January 2026.

Majumdar, A. and Pavone, M. How should a robot assess risk? towards an axiomatic theory of risk in robotics. In *Springer Proceedings in Advanced Robotics*, pp. 75–84. Springer International Publishing, Cham, 2020.

Mees, O., Hermann, L., Rosete-Beas, E., and Burgard, W. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. volume 7, pp. 7327–7334, 2022. doi: 10.1109/LRA.2022.3180108.

Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. TorchMetrics - Measuring Reproducibility in PyTorch, February 2022. URL https://github.com/Lightning-AI/torchmetrics.

Octo Model Team, Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., et al. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024.

Open X-Embodiment Collaboration. Open X-embodiment: Robotic learning datasets and RT-X models, 2023.

Pumacay, W., Singh, I., Duan, J., Krishna, R., Thomason, J., and Fox, D. THE COLOSSEUM: A benchmark for evaluating generalization for robotic manipulation. 2024.

Sedlacek, M., Yefanov, P., Ponimatkin, G., Bardhan, J., Pilc, S., Fourmy, M., Kazakos, E., Snoek, C. G. M., Sivic, J., and Petrik, V. Realm: A real-to-sim validated benchmark for generalization in robotic manipulation, 2025. URL https://arxiv.org/abs/2512.19562.

Shukor, M., Aubakirova, D., Capuano, F., Kooijmans, P., Palma, S., Zouitine, A., Aractingi, M., Pascal, C., Russi, M., Marafioti, A., Alibert, S., Cord, M., Wolf, T., and Cadene, R. Smolvla: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, October 2012.

Wang, D. and Sun, X.-H. APC: A novel memory metric and measurement methodology for modern memory systems. *IEEE Trans. Comput.*, 63(7):1626–1639, July 2014.

Wang, Y. R., Ung, C., Tannert, G., Duan, J., Li, J., Le, A., Oswal, R., Grotz, M., Pumacay, W., Deng, Y., Krishna, R., Fox, D., and Srinivasa, S. Roboeval: Where robotic manipulation meets structured and scalable evaluation, 2025. URL https://arxiv.org/abs/2507.00435.

Xiao, X., Liu, J., Wang, Z., Zhou, Y., Qi, Y., Jiang, S., He, B., and Cheng, Q. Robot learning in the era of foundation models: a survey. *Neurocomputing*, 638(129963): 129963, July 2025.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, pp. 1094–1100. PMLR, 2020.

Zhang, H. et al. VLA-arena: Evaluating vision-language-action models in the wild, 2025.

Zhao, T. Z., Kumar, V., Levine, S., and Finn, C. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Zhou, Z., Atreya, P., Tan, Y. L., Pertsch, K., and Levine, S. AutoEval: Autonomous evaluation of generalist robot manipulation policies in the real world. 2025.

Zitkovich, B., Yu, T., Xu, S., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Tan, J., Toussaint, M., and Darvish, K. (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A. Experimental Details

A.1. Pre-trained Policy Evaluation (PushT Simulation)

A.1.1. Environment and Policies

Experiments were conducted on the PushT-v0 environment from LeRobot (Capuano et al., 2025) using four pre-trained policies from the official LeRobot model hub: **Diffusion Policy** (pixel-based and keypoint-based variants), **VQ-BeT** (Lee et al., 2024), and **ACT** (Zhao et al., 2023). Each policy was evaluated over $N = 50$ episodes with different random seeds. All evaluations were run on a single **Nvidia RTX 4080 GPU**. Experiment source code can be found in https://github.com/vvrs/vla-eval

A.1.2. FRAME Training-Time Integration: ACT Hyperparameters

To demonstrate training-time metric monitoring (Section 4.3), we trained ACT from scratch using the configuration in Table 6.

Table 6. ACT training hyperparameters on PushT (from WandB run `pusht_act`).

Hyperparameter	Value
Dataset	`lerobot/pusht`
Policy	ACT (ResNet18, IMAGENET1K_V1)
Batch size	64
Optimizer	AdamW
Learning rate	1×10^{-4}
Backbone LR	1×10^{-5}
Weight decay	1×10^{-4}
(β_1, β_2)	$(0.9,\ 0.999)$
ε	1×10^{-8}
Grad clip norm	10
Total steps	10,000
Eval frequency	every 1,000 steps
Chunk size	100
Model dim	512
Feedforward dim	3,200
Latent dim	32
KL weight	10
Dropout	0.1
Encoder layers	4
Decoder layers	1
Attention heads	8
Seed	1000
Hardware	Nvidia RTX 4090

Table 7. SmolVLA fine-tuning hyperparameters on SO-101 dataset.

Hyperparameter	Value
Base model	SmolVLA-450M (`lerobot/smolvla`)
Training demos	36 (80% of 45)
Test demos	9 (20% of 45)
Batch size	16
Optimizer	AdamW
Learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Training steps	10,000
Chunk size	40
Image resolution	384×384
Cameras	Overhead + wrist (2 views)
Hardware	Nvidia RTX 4080

A.2. Evaluation on SO-101 Physical Robot

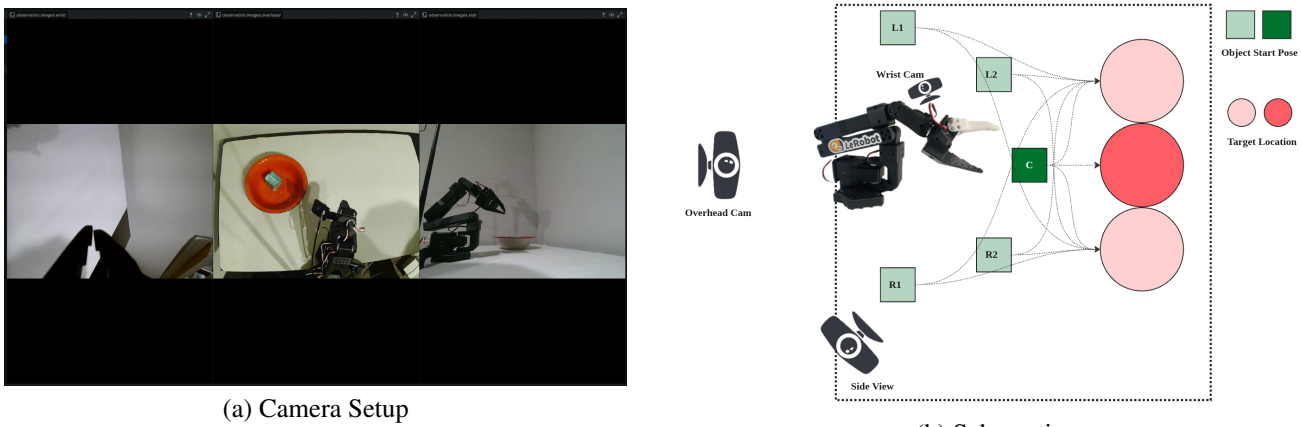
A.2.1. Experimental Setup and Data Collection

To evaluate FRAME on a physical platform, we deployed the SO-101 low-cost robotic manipulator (a 5-DOF open-source arm). We collected an expert demonstration dataset of 45 kinesthetic pick-and-place demonstrations using the setup shown in Figure 3. The dataset was split into **36 training** and **9 held-out test** demonstrations (80/20 split).

The task consists of grasping a target object and placing it into a designated bin. The robotic workspace (Figure 3-(b)) is equipped with a fixed overhead camera and a wrist-mounted camera, providing the visual observations required by the policy.

A.2.2. Fine-tuning Hyperparameters

We fine-tuned pre-trained **SmolVLA-450M** (Shukor et al., 2025) and π_0 (Black et al., 2026) on the collected demonstration data using the LeRobot training pipeline. Key hyperparameters for SmolVLA fine-tuning are listed in Table 7.



(a) Camera Setup

(b) Schematic

Figure 3. Physical robot evaluation setup. (a) External view showing the SO-101 robot and workspace. (b) Schematic of the camera and environment configuration.

A.2.3. Trajectory Metric Computation

End-effector Cartesian poses were reconstructed from joint encoder readings using SO-101's forward kinematics model ($L_1 = 117$ mm, $L_2 = 136$ mm, base height $= 120$ mm). FRAME's Cartesian metric suite (PL, PS, CC) was then applied to the reconstructed trajectories offline. Inference latency and memory usage were measured on the same Nvidia RTX 4080 used for training.