EqC: Equilibrium Energy Control for Introspective Lifelong Robot Learning

Anonymous authors
Paper under double-blind review

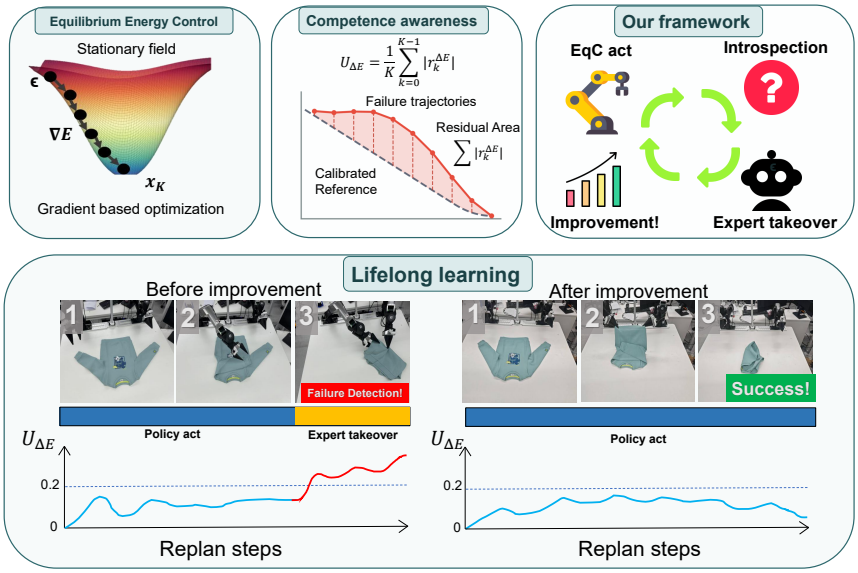


Figure 1: **Overview of Equilibrium Energy Control (EqC). Top:** EqC generates actions through energy-based refinement, while the geometry of the refinement path reveals when autonomous execution is weakly supported. Energy-Gated DAgger uses this signal to trigger expert takeover and subsequent policy improvement. **Bottom:** In real-world deployment, persistent $U_{\Delta E}$ alarms prompt timely assistance before failure, progressively converting assisted interventions into fully autonomous execution through lifelong learning.

Abstract

Long-term robot autonomy depends on more than effective action generation: a policy must recognize when its behavior is no longer sufficiently supported and use assistance to improve. We introduce **Equilibrium Energy Control (EqC)**, a framework for introspective lifelong robot learning built around an explicit conditional energy landscape. EqC casts generative action refinement as equilibrium-seeking optimization, while the geometry of the refinement process reveals whether the generated behavior remains consistent with the learned landscape. This competence signal enables Energy-Gated DAgger, where persistent energy-consistency violations trigger expert takeover and the resulting recovery trajectories are aggregated for subsequent policy updates. EqC integrates naturally into both Diffusion Transformer policies and flow-matching-based vision-language-action (VLA) policies. Across simulation and real-world bimanual manipulation, EqC matches or improves control performance and robustness while enabling timely intervention near the policy's competence boundary. A real-world study further shows how these interventions progressively expand the range of situations handled autonomously, providing a practical path from static generative controllers toward self-improving robot systems.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Robots deployed in open-ended environments must manage their autonomy as operating conditions change (Farid et al., 2024; Filos et al., 2020). A policy may perform reliably when the current situation is well represented by prior experience, yet become unreliable as deployment approaches the limits of that experience. This transition defines the policy's context-dependent competence boundary. An introspective robot recognizes its proximity to this boundary and regulates autonomous execution before uncertainty develops into failure. Such awareness is also central to lifelong robot learning: assistance requested near the competence boundary protects deployment and reveals where corrective experience is most valuable. Modern generative robot policies, however, are commonly deployed as static action generators that provide little indication of whether their generated behavior remains reliable in the current context (Chi et al., 2025; Black et al., 2024). Competence awareness can instead turn these boundary cases into targeted opportunities for learning rather than avoidable failures.

We introduce **Equilibrium Energy Control (EqC)**, an equilibrium energy formulation inspired by Equilibrium Matching (Wang & Du, 2025) that equips generative robot policies with an explicit conditional energy landscape. Given visual observations, language instructions, and robot state, EqC optimizes an action chunk on this landscape and executes a short prefix under receding-horizon control. The optimization trajectory also reveals whether refinement follows the energy-progress relation learned from expert demonstrations, providing an introspective view of the policy's competence during execution. Energy-Gated DAgger uses the resulting path-wise energy-consistency signal to trigger expert intervention and aggregate recovery trajectories near the policy's current competence boundary. This condition-dependent energy formulation therefore supports action generation, online competence monitoring, and targeted data acquisition, while integrating naturally with both Diffusion Transformer policies (Peebles & Xie, 2023) and flow-matching-based vision-language-action (VLA) policies (Black et al., 2024; Physical Intelligence et al., 2025).

We evaluate EqC on simulation benchmarks including LIBERO (Liu et al., 2023), RoboTwin 2.0 (Chen et al., 2025b), VLABench (Zhang et al., 2025), and LIBERO-Plus (Fei et al., 2025), together with real-world bimanual manipulation. EqC matches or improves control performance and robustness across visual and task variation, while its geometry-derived competence signal enables online detection of failure-prone execution. In real-world lifelong learning experiments, the resulting requests for assistance concentrate recovery data near the policy's competence boundary, and subsequent updates convert previously assisted situations into autonomous capabilities. This process progressively enlarges the range of situations the robot can handle independently, which we refer to as *autonomous competence expansion*. Code is available on the anonymous GitHub page. Video is provided in the supplementary material.

Our contributions are summarized as follows:

- **Equilibrium energy control.** EqC formulates action generation as equilibrium-seeking optimization under an explicit conditional energy function. The formulation integrates with both Diffusion Transformer and flow-matching-based VLA policies, providing a common energy-based control mechanism across distinct policy families.
- **Introspection from energy geometry.** EqC characterizes finite energy progress across the action-refinement trajectory and measures its agreement with the learned equilibrium relation. The resulting signal detects when autonomous execution approaches the policy's competence boundary and provides an online criterion for timely assistance.
- **Energy-Gated DAgger for lifelong robot learning.** Persistent energy-consistency violations trigger expert recovery, and the resulting trajectories are aggregated to update the policy around its current competence boundary. Real-world deployment shows how this closed loop converts assisted situations into autonomous capabilities and progressively expands robot competence.

2 RELATED WORK

Generative models for robotics. Generative modeling has driven substantial progress in robot learning through diffusion-based policies (Janner et al., 2022; Ajay et al., 2022; Chi et al., 2025; Huang et al., 2024; 2025), world models (Zhang & Du, 2026; Chen et al., 2025a; Yuan et al.,

2026), and vision-language-action policies (Wang et al., 2026). Within VLA policies, iterative action generation is commonly instantiated with diffusion, as in RDT-1B (Liu et al., 2025), or with flow-matching action experts, as in π_0 and $\pi_{0.5}$ (Lipman et al., 2022; Black et al., 2024; Physical Intelligence et al., 2025). Recent analysis relates the effectiveness of these generative policies to noise-covered supervision and iterative computation (Pan et al., 2026). EqC builds on this structure by realizing action refinement as optimization over an explicit conditional energy landscape.

Energy-based models for robotics. Energy-based models represent compatibility through a scalar function over candidate configurations, turning inference into optimization on a learned landscape (LeCun et al., 2006). In robot learning, observation–action energy functions have been learned for imitation and used to optimize actions at inference time (Florence et al., 2022), with subsequent work improving their contrastive objectives, interpolation behavior, and scaling to transformer policies (Singh et al., 2023; Davies et al., 2025). Recent energy-based VLA work uses terminal action energy as a confidence measure for visual anomaly detection, while leaving its use for intervention-driven real-world data aggregation unexplored (Naghiyev et al.). Equilibrium Matching (EqM) (Wang & Du, 2025) learns stationary generative dynamics with an implicit energy structure, and GeCO (Zhang et al., 2026) carries this implicit-energy formulation into robot control through iterative refinement and adaptive computation. EqC makes the conditional energy explicit in the deployed control field, allowing finite progress along the optimization trajectory to support both action generation and introspection.

Failure detection and competence-aware autonomy. Failure detection and uncertainty estimation provide useful signals for deciding when autonomous execution should yield to supervision (Farid et al., 2024; Filos et al., 2020). DAgger establishes the iterative aggregation of on-policy expert corrections (Ross et al., 2011), while robot-gated variants use predicted action error, ensemble disagreement, state novelty, and learned risk to decide when those corrections are needed (Zhang & Cho, 2016; Menda et al., 2019; Hoque et al., 2021). This principle also appears in competence-aware autonomy, where a robot adjusts its level of autonomy according to its estimated ability in the current situation (Basich et al., 2020). For generative manipulation policies, diffusion training loss has also served as a failure signal for interactive data collection (Lee et al., 2025). EqC instead reads introspective information from the consistency of energy progress across the full refinement path and uses persistent violations to gate recovery data collection.

3 EqC for Introspective Lifelong Robot Learning

3.1 Overview

EqC organizes robot control and lifelong improvement around the condition-dependent energy function $E_\theta(x, z)$. Its conditional energy landscape refines actions through energy gradients, while finite energy changes along the same refinement trajectory provide an introspective signal of policy competence. Energy-Gated DAgger uses this signal to request expert intervention and aggregate targeted recovery experience. We first formulate energy-based action refinement, then derive its introspective readout and use that signal for lifelong learning.

3.2 Equilibrium Energy Control

We consider a robot policy that predicts a continuous action chunk $a \in \mathbb{R}^D$ from a condition z, which may include images, a language instruction, robot proprioception, and recent observation history. Demonstrations are drawn from an unknown conditional distribution $p_{\text{data}}(a \mid z)$. We parameterize the bridge from noise to data by $t \in [0, 1]$, with $t = 0$ at noise and $t = 1$ at data, and sample t uniformly during training. For $\varepsilon \sim \mathcal{N}(0, I)$, the bridge displacement, intermediate state, and remaining schedule integral are

$$d := a - \varepsilon, \qquad x_t := \varepsilon + td, \qquad C(t) := \int_t^1 c(s)\,\mathrm{d}s. \tag{1}$$

A nonnegative schedule $c(t)$ controls the magnitude of the target field $c(t)d$.

EqC represents action quality with a differentiable energy $E_\theta : \mathbb{R}^D \times \mathcal{Z} \to \mathbb{R}$. Its gradient $g_\theta(x, z) = \nabla_x E_\theta(x, z)$ is a stationary control field: it depends on the current action and condition, but not on the bridge coordinate. At x_t, the demonstrated displacement supplies the desired direction,

and the energy is trained with the gradient-matching objective

$$\mathcal{L}_{\text{grad}} = \mathbb{E}_{(a,z),\varepsilon,t}\left[\frac{1}{D}\left\|\nabla_x E_\theta(x_t, z) - c(t)d\right\|_2^2\right]. \tag{2}$$

The bridge coordinate is used only to sample training targets; the deployed field depends on the current action and condition alone. At inference, EqC initializes $x^{(0)} \sim \mathcal{N}(0, I)$ and applies $x^{(k+1)} = x^{(k)} + \eta_k \nabla_x E_\theta(x^{(k)}, z)$ for $k = 0, \ldots, K-1$, yielding $\hat{x} = x^{(K)}$. The controller executes a prefix and replans from the next observation. Standard smoothness conditions give monotonic energy progress for sufficiently stable step sizes (Appendix C.4).

Architecture compatibility. The framework requires only a differentiable, per-sample scalar over an action chunk. It may be produced by a dedicated energy head or by a differentiable scalarization of action-conditioned features. In our implementations, a stationary backbone representation f_θ defines $E_\theta(x, z) = \langle x, f_\theta(x, z)\rangle$ over active action dimensions. The DiT model is conditioned on RGB observations, language, and robot state. The VLA model retains the π_0 multimodal prefix, action expert, action representation, and padding convention. Both expose the same interface $x \mapsto E_\theta(x, z) \mapsto \nabla_x E_\theta(x, z)$, so EqC changes the action objective and its optimization through an interface shared across observation encoders and policy scales.

3.3 Introspection from Energy Geometry

The gradient-matching target also determines finite energy progress along a demonstration bridge. For $x_s = \varepsilon + sd$, integrating $\frac{d}{ds}E_\theta(x_s, z) = \nabla_x E_\theta(x_s, z)^\top d = c(s)\|d\|_2^2$ gives

$$E_\theta(a, z) - E_\theta(x_t, z) = \|d\|_2^2 C(t). \tag{3}$$

Violation of this relation along a realized refinement trajectory reveals disagreement between local optimization and the energy geometry learned from demonstrations.

At deployment, we treat $\hat{x} = x^{(K)}$ as the approximate equilibrium and use it as a self-anchor. Let $v = \hat{x} - x^{(0)}$ and $D_v = \max(\|v\|_2^2, \delta)$. Projection onto the noise-to-equilibrium chord assigns each iterate a progress coordinate τ_k, after which we compare its normalized energy-to-go with the bridge reference:

$$\begin{aligned} \tau_k &:= \text{clip}\left(\frac{\langle x^{(k)} - x^{(0)}, v\rangle}{D_v}, 0, 1\right), \\ r_k^{\Delta E} &:= \frac{E_\theta(\hat{x}, z) - E_\theta(x^{(k)}, z)}{D_v} - C(\tau_k), \\ U_{\Delta E}(z) &:= \frac{1}{K}\sum_{k=0}^{K-1}\left|r_k^{\Delta E}(z)\right|. \end{aligned} \tag{4}$$

The score is small when realized energy progress follows the bridge relation and increases when that relation is violated. Absolute aggregation prevents cancellation across refinement states, while averaging makes the scale comparable across different values of K. It requires neither an expert action nor an auxiliary uncertainty model at deployment. Appendix C.5 relates the residual to field error and off-chord deviation.

3.4 Energy-Gated DAgger for Lifelong Robot Learning

We use EqC's energy-consistency score to gate a DAgger-style data aggregation loop (Ross et al., 2011), an instantiation we call Energy-Gated DAgger. Before each action prefix, the controller computes $U_{\Delta E}(z_j)$; m consecutive threshold crossings trigger expert takeover, and the recovery trajectory is retained for the next policy update. The gate therefore responds to persistent path-wise inconsistency rather than raw energy magnitude.

This deployment loop concentrates supervision on states where the current controller lacks support: the energy used to generate behavior also determines when autonomy should yield, focusing new experience near the policy's competence boundary.

Algorithm 1 Energy-Gated DAgger with EqC

Require: Initial demonstrations $\mathcal{D}_0$, threshold γ, persistence m, rounds R
$\theta_0 \leftarrow \textsc{TrainEqC}(\mathcal{D}_0)$
for $r = 0, \ldots, R-1$ **do**
  $\mathcal{Q}_r \leftarrow \varnothing, h \leftarrow 0$
  for each replan j in deployment round r **do**
    $z_j \leftarrow \textsc{Observe}$
    $(\hat{x}_j, \mathcal{X}_j) \leftarrow \textsc{EnergyRefine}(E_{\theta_r}, z_j)$
    $u_j \leftarrow \textsc{EnergyConsistency}(\mathcal{X}_j)$
    $h \leftarrow (h+1)\,\mathbf{1}[u_j > \gamma]$
    if $h = m$ **then**
      $\xi_j^\star \leftarrow \textsc{ExpertRecover}(z_j)$
      $\mathcal{Q}_r \leftarrow \mathcal{Q}_r \cup \{\xi_j^\star\}, h \leftarrow 0$
    else
      $\textsc{ExecutePrefix}(\hat{x}_j)$
    end if
  end for
  $\mathcal{D}_{r+1} \leftarrow \mathcal{D}_r \cup \mathcal{Q}_r$
  $\theta_{r+1} \leftarrow \textsc{TrainEqC}(\mathcal{D}_{r+1})$
end for
return E_{θ_R}

4 Why EqC Fits Generative Robot Policies

A favorable generation regime. Image synthesis must cover diverse pixel realizations left unspecified by a class or text prompt. In control, rich visual, language, and proprioceptive conditions narrow the action choice, success requires one competent action, and receding-horizon execution replans after a short prefix. Action outputs are also much smaller than images. These properties make conditional action generation a natural setting for a structured optimizer. This view agrees with the finding that noise-covered supervised iteration explains more of generative-control performance than distinct action modes on common benchmarks (Pan et al., 2026), and suggests why the explicit-energy cost observed for EqM image generation (Wang & Du, 2025) need not transfer directly to control.

Approximation and optimization. Appendix C.3 compares EqC with matched stationary and time-conditioned regressors and shows

$$R^{\Theta}_{\mathrm{EqC}} - R^{\star}_{\mathrm{tc}} \le \left(\sigma_{\mathrm{act}} + \sigma_{\mathrm{scale}} + \delta_{\Theta}\right)^2 . \tag{5}$$

The terms measure residual action ambiguity, uncertainty about bridge progress after removing time, and finite energy class approximation. When they are small, explicit energy has limited regression cost. It then exposes a property not provided by an unconstrained field: sufficiently small gradient-ascent steps provably increase the scalar optimized at inference. The same path calculus relates $U_{\Delta E}$ to mean absolute integrated field error and trajectory deviation. Full definitions and proofs are given in Appendix C.

5 Experiments

The experiments evaluate the three capabilities developed by EqC. We first study its effectiveness as a generative controller across DiT and VLA policies in simulation and real-world manipulation. We then test path-wise energy consistency as an introspective signal for online failure detection. Finally, we close the loop with Energy-Gated DAgger to evaluate lifelong improvement on a physical robot. Complete benchmark and implementation protocols, failure-detection analyses, real-robot details, and ablations are provided in Appendices A, B, D, and E, respectively.

5.1 Simulated and Real-World Benchmarks

5.1.1 Setup

We evaluate action generation and competence awareness on LIBERO (Liu et al., 2023), RoboTwin 2.0 (Chen et al., 2025b), VLABench (Zhang et al., 2025), and LIBERO-Plus (Fei et al., 2025). All methods use each benchmark's binary success criterion. The DiT experiments share one conditioning

Table 1: Success rates (%) on LIBERO with the DiT policy and VLABench with VLA policies.

(a) LIBERO–DiT

Method	Goal	Spatial	Object	Long	Avg.
Diffusion Policy	82.9	91.4	88.9	82.7	86.5
Rectified Flow	92.4	94.6	97.0	76.0	90.0
GeCO	95.2	95.8	**99.0**	83.8	93.5
EqC (ours)	**95.6**	**96.4**	98.1	**85.5**	**94.0**

(b) VLABench–VLA

Method	T1	T2	T3	T4	T6	Avg.
$\pi_{0.5}$	40.6	**22.6**	18.0	**16.1**	25.6	24.5
$\pi_{0.5}$ + **EqC (ours)**	**44.3**	21.3	**18.5**	15.4	**36.0**	**27.1**
π_0	47.0	21.2	29.1	17.3	32.2	29.4
π_0 + GeCO	61.0	22.0	**34.0**	**20.0**	45.0	36.0
π_0 + **EqC (ours)**	**64.4**	**29.0**	28.6	15.8	**45.6**	**36.7**

Table 2: VLA policy success rates (%) on RoboTwin 2.0 (Easy / Hard) and LIBERO-Plus.

(a) RoboTwin 2.0 (Easy / Hard)

Method	Bottle	Hammer	Rank	Alarm	Bell	Avg.
π_0	90 / 56	43 / 21	**7** / **1**	63 / 11	44 / 3	49 / 18
π_0 + GeCO	88 / 66	52 / **36**	0 / 0	72 / 16	42 / 21	50 / 28
π_0 + **EqC (ours)**	**100** / **76**	**76** / 27	6 / 0	**83** / **32**	**49** / **30**	**63** / **33**

(b) LIBERO-Plus

Method	Orig	Cam	Rob	Lang	Light	Back	Noise	Lay	Avg.
OpenVLA	76.5	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
π_0	94.2	15.8	6.6	**61.0**	79.6	78.5	**79.4**	70.4	53.6
π_0 + **EqC (ours)**	**94.3**	**61.7**	**39.0**	54.8	**84.6**	**83.0**	74.3	**73.1**	**65.8**

stack, action interface, and demonstration set across objectives and are built on a modular diffusion-policy implementation (Dong et al., 2024). The VLA experiments use the same flow-matching VLA backbone within each comparison. EqC always refines a noise-initialized action chunk and executes a prefix under receding-horizon control. Appendix A gives the benchmark definitions, rollout budgets, and implementation details.

5.1.2 Diffusion Transformer

We adapt EqC to a DiT architecture and evaluate it on the four LIBERO suites. Table 1a follows the benchmark comparison reported by GeCO (Zhang et al., 2026) and places EqC's multi-seed mean alongside those published baselines. EqC achieves the highest aggregate result, with 94.0% average success across training seeds. Appendix A.4 reports the complete EqC statistics together with our independent multi-seed reproductions of the matched rectified-flow and GeCO baselines; Appendix A.2 describes the protocol.

5.1.3 Vision-Language-Action Policies

We integrate EqC into a VLA backbone by adapting the π_0 policy while preserving its multimodal architecture. EqC replaces flow matching with the energy objective $E_\theta(x, o) = \langle x, v_\theta(x, o, 0)\rangle$, evaluated at the fixed model time $t_{\text{model}} = 0$. Training matches energy gradients along interpolated action trajectories and masks inactive padded dimensions; inference refines actions iteratively with the learned energy field. We evaluate the resulting VLA policies on RoboTwin 2.0, VLABench, and LIBERO-Plus, with benchmark protocols in Appendix A.2; supplementary LIBERO results are provided in Appendix A.6.

RoboTwin 2.0. RoboTwin 2.0 evaluates bimanual manipulation under structured randomization of clutter, lighting, background, tabletop height, and language instructions. Under its clean Easy and domain-randomized Hard conditions, Table 2a shows that EqC raises the π_0 averages from $49/18$ to $63/33$, compared with $50/28$ for π_0+GeCO. The improvement is particularly pronounced on Adjust Bottle and Click Alarmclock, including under the more challenging Hard condition.

VLABench. We evaluate on VLABench across its five standardized tracks (T1–T4 and T6), covering object generalization, commonsense reasoning, semantics, and appearance shifts. As shown in Table 1b, EqC consistently enhances generalization for both flow-matching backbones, boosting π_0 from 29.4% to 36.7% and $\pi_{0.5}$ from 24.5% to 27.1%.

LIBERO-Plus. LIBERO-Plus tests robustness under seven controlled perturbation families. Table 2b shows that EqC raises overall success from 53.6% to 65.8%. The most prominent improvements occur under camera viewpoint ($15.8\% \rightarrow 61.7\%$) and robot initialization shifts ($6.6\% \rightarrow 39.0\%$), demonstrating superior out-of-distribution robustness.

Table 3: Control and competence-awareness evaluation. **(a)** Success and latency on three real-world bimanual tasks (20 trials each). **(b)** Failure detection on pooled LIBERO and LIBERO-Plus rollouts. "EqC+Diff-DAgger" denotes the Diff-DAgger flow-loss detector applied to the EqC backbone; EqC uses path-wise energy consistency.

(a) Real-world bimanual manipulation

Method	Success (%)				Latency (ms)			
	Flower	Towel	Sweep.	Avg.	Flower	Towel	Sweep.	Avg.
Galaxea G0.5	45.0	80.0	65.0	63.3	**421.9**	**435.6**	**439.6**	**432.4**
Galaxea G0.5 + **EqC (ours)**	**80.0**	**95.0**	**100.0**	**91.7**	526.4	556.5	548.7	543.9

(b) Online competence awareness

Method	AUROC	TPR/FPR	Acc. (%)	Saved (%)
Diff-DAgger	0.8430	61.3/10.2	82.2	33.3
EqC+Diff-DAgger	0.8888	72.8/8.8	85.5	42.1
EqC (ours)	**0.9244**	**77.9/8.5**	**87.5**	**45.5**

5.1.4 Real-World Experiments

We evaluate EqC on a bimanual workstation with two Galaxea A1Z manipulators using the Galaxea G0.5(Liu et al., 2026) flow-matching based VLA policy, with 20 independent trials per task under randomized object placements. Flower Insertion (Figure 2, top) requires the robot to grasp and reorient a flexible stem before inserting it through a narrow vase opening; Towel Folding (middle) requires coordinated deformable-object manipulation through lateral folding, regrasping, and a final vertical fold. Debris Sweeping (bottom) evaluates multi-stage bimanual tool coordination, requiring the robot to sweep scattered paper scraps into a handheld dustpan using a brush and deposit them cleanly into a waste bin without spillage. Appendix D provides task definitions, success criteria, and training settings.

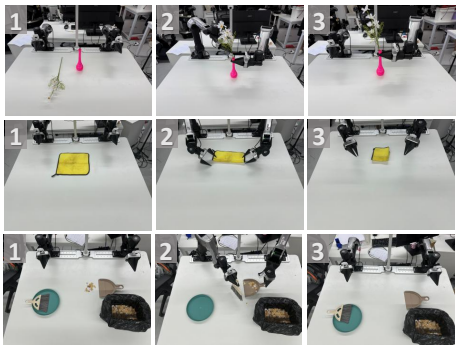


Figure 2: **Real-world bimanual execution.** *Top (1–3):* Flower Insertion. *Middle (1–3):* Towel Folding. *Bottom (1–3):* Debris Sweeping.

Table 3a shows that EqC raises average success from 63.3% to 91.7%. Success improves from 45.0% to 80.0% on *Flower Insertion*, from 80.0% to 95.0% on *Towel Folding*, and reaches 100.0% on *Debris Sweeping*, while introducing moderate inference latency overhead.

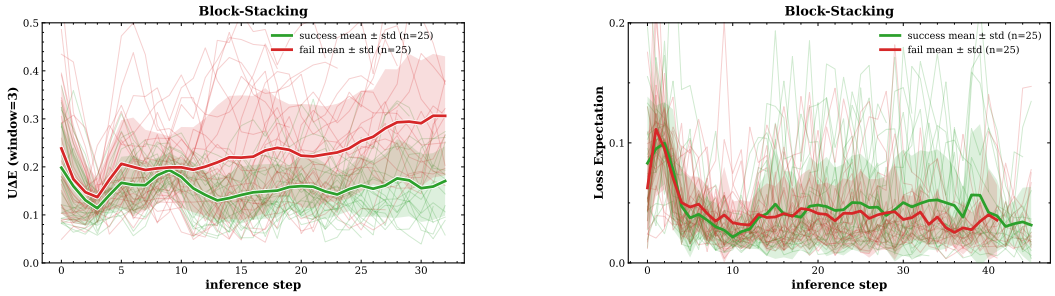
5.2 Competence Awareness

5.2.1 Setup

EqC's energy-consistency signal measures whether inferred energy progress remains aligned with the relation induced by expert demonstrations, enabling the policy to perform introspective competence assessment at inference time. We evaluate this property as online failure detection: successful rollouts are negative examples and failed rollouts are positive examples. An alarm requires the signal to exceed a fixed threshold for m consecutive replanning steps, and detection timing is measured by the fraction of the rollout remaining when an alarm is raised in a failed trial. Appendix B.1 formalizes the competence-aware task definition, while Appendix B.2 provides the simulation failure-detection protocol, including the rollout partition, Receiver Operating Characteristic (ROC) curve construction, and threshold calibration.

5.2.2 Simulated Rollout Analysis

In simulation, we evaluate online competence awareness on LIBERO and LIBERO-Plus using three configurations: Diff-DAgger's flow-loss detector on π_0 (Lee et al., 2025), the same detector on the EqC backbone, and the full energy-consistency signal $U_{\Delta E}$. Threshold-dependent metrics use a disjoint split: 10% of the rollouts select and freeze (m, τ^*), and the remaining 90% are used only for evaluation. Under this protocol, the calibration subset selects (m, τ^*) while constraining the FPR to approximately 10%; the pair remains frozen for held-out evaluation, while Area Under the



(a) EqC (ours)

(b) Diff-DAgger (baseline)

Figure 3: **Real-world failure detection** (mean ± 1 std, $n = 25$). (a) EqC cleanly separates success and failure early for timely intervention, whereas (b) Diff-DAgger exhibits heavy overlap.

Receiver Operating Characteristic (AUROC) summarizes the full score ranking. Table 3b shows that EqC's path-wise energy-consistency signal attains the highest AUROC, TPR, accuracy, and saved inference time, together with the lowest FPR, among Diff-DAgger and EqC + Diff-DAgger baseline; its AUROC reaches 0.9244, compared with 0.8430 for Diff-DAgger. Appendix B.5 details the adapted Diff-DAgger baseline, and Appendix B.6 reports calibration variants, window and Monte Carlo sensitivity, saved inference time, and inference latency.

5.2.3 Real-World Rollout Analysis

We next use *Hierarchical Block Stacking* as a real-world failure-detection testbed and collect 50 rollouts: 25 successful executions and 25 failed executions. Figure 3a shows that the signal remains low for successful executions and rises early for failures, whereas the Diff-DAgger trajectories in Figure 3b overlap through most of the horizon. The EqC signal therefore preserves a more useful distinction between supported and unsupported execution under physical sensing and control noise; Appendix D provides the task details.

5.3 Energy-Gated DAgger for Lifelong Learning

5.3.1 Experimental Setup and Protocol

Real-world robot data are costly, and deployment conditions cannot be fully anticipated. We therefore test whether EqC recognizes its competence boundary and uses intervention data to expand it across two bimanual tasks on the Galaxea platform: (1) *Hierarchical Block Stacking*, requiring the robot to construct a freestanding tower across $3! = 6$ color permutations under tight contact constraints; and (2) *Garment Folding*, requiring sequential multi-stage folding of deformable fabric. Both policies start from limited seed demonstrations and follow Algorithm 1 ($\gamma = 0.2$), where persistent $U_{\Delta E}$ violations prompt expert recovery. The aggregated trajectories are used for subsequent policy fine-tuning (Appendix D).

5.3.2 Progressive Performance

Figure 4 (top) tracks competence acquisition over Energy-Gated DAgger deployment rounds across *Hierarchical Block Stacking* and *Garment Folding*. We evaluate against an automated Galaxea G0.5 + Diff-DAgger baseline and a skilled human-in-the-loop teleoperator. Under identical data and training budgets, Energy-Gated DAgger substantially outperforms all baselines across three independent experimental runs. On *Hierarchical Block Stacking*, success rises from an initial 0.0% to an average of **96.7%** after incorporating 30 targeted recovery episodes, outperforming Galaxea G0.5 + Diff-DAgger (80.0%) and human-triggered intervention (58.3%). On *Garment Folding*, the policy progresses from 30.0% to an average of **98.3%** autonomous execution in just 2 rounds (12 targeted recovery episodes), surpassing Galaxea G0.5 + Diff-DAgger (81.7%) and skilled human intervention (55.0%). Full round-by-round statistics across all individual runs are reported in Table 17 (Appendix D.4). These results confirm that EqC's path-wise energy-consistency signal identifies weakly supported states more reliably than Diff-DAgger's signal or manual monitoring, yielding targeted recovery data that systematically expands autonomous competence. As qualitatively shown in Figure 4 (bottom),

the detection of execution anomalies trigger timely $U_{\Delta E}$ alarms to prompt expert recovery in earlier rounds, progressively enabling robust autonomous execution under perturbed setups. Furthermore, under sequential cross-task shift (*Hierarchical Cup Stacking*, Appendix D.5), EqC detects task boundary transitions, mitigates catastrophic forgetting, and restores 100.0% dual-task autonomy.

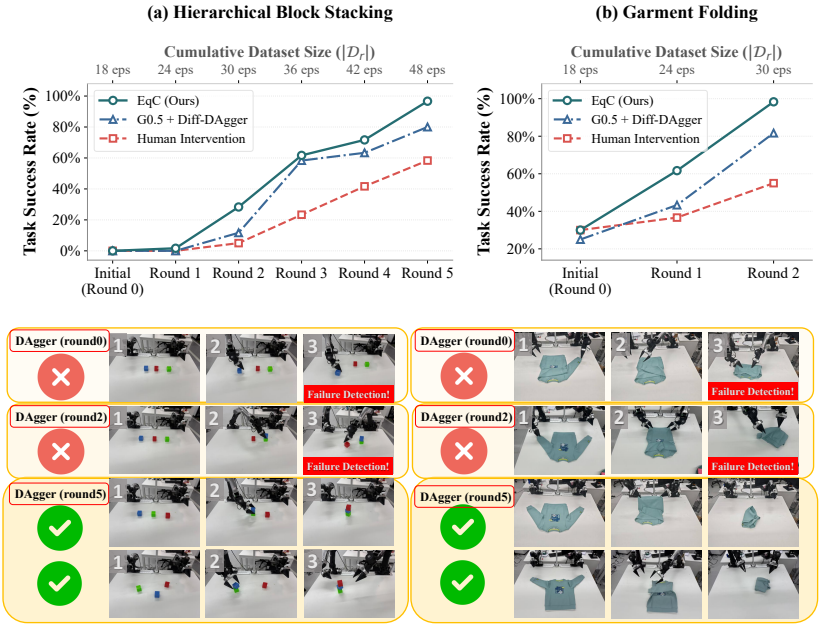


Figure 4: **Real-world lifelong learning evaluation with Energy-Gated DAgger. Top:** Quantitative success curves across deployment rounds (averaged over 3 independent runs, with 20 autonomous evaluation trials per checkpoint per run). **Bottom:** Qualitative rollouts: early drift triggers $U_{\Delta E}$ alarms (red banners) for expert recovery; fine-tuned policies achieve autonomous execution (Rows 3–4). Across both **(a)** *Hierarchical Block Stacking* and **(b)** *Garment Folding*, EqC achieves **96.7%** and **98.3%** mean success rates, respectively, decisively outperforming Diff-DAgger and human intervention.

6 Limitations and Scope

Our study focuses on continuous action-chunk policies with stationary refinement schedules. Future work should extend EqC to hybrid, memory-based, and hierarchical policies; deepen the analysis of energy geometry, calibration, and repeated data aggregation; and test how competence thresholds and intervention budgets transfer across embodiments and longer deployments.

7 Conclusion

We presented Equilibrium Energy Control (EqC), an energy-based framework connecting generative robot control with introspective competence awareness. Through an explicit conditional energy landscape, EqC refines actions toward equilibrium and uses path-wise energy consistency to assess whether behavior remains supported; Energy-Gated DAgger turns this introspective signal into targeted expert recovery and policy updates. Across simulation and physical robots, EqC matches or improves control performance and robustness. Real-world lifelong learning experiments further show that repeated recovery and updates expand autonomous competence. Its compatibility with diverse robot-manipulation architectures also offers a path toward energy-based control and introspection in locomotion, navigation, and autonomous driving. Embedding competence awareness into generative control may enable introspective, self-evolving embodied agents whose autonomy grows through lifelong learning while retaining a mechanism for safe intervention.

AI USE STATEMENT

Generative AI tools were used to improve the readability of the manuscript, create or modify scientific figures, and edit code. The authors carefully reviewed all AI-assisted text, figures, and code, checked them against the underlying research and experimental results, and take full responsibility for the final content of the paper, including any AI-assisted material.

REPRODUCIBILITY STATEMENT

We provide the EqC formulation and theoretical derivations in the main text and Appendix C. The appendices document model and training configurations, benchmark protocols, evaluation splits, failure-detection calibration, real-robot tasks, and baseline implementations (Appendices A, B, and D). The code for the EqC implementations and evaluation procedures is available at the anonymous repository linked in the Introduction and Appendix G.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Connor Basich, Justin Svegliato, Kyle Hollins Wray, Stefan Witwicki, Joydeep Biswas, and Shlomo Zilberstein. Learning to optimize autonomy in competence-aware systems. *arXiv preprint arXiv:2003.07745*, 2020.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

Boyuan Chen, Tianyuan Zhang, Haoran Geng, Caiyi Zhang, Peihao Li, Kiwhan Song, William T Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, et al. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025b.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Travis Davies, Yiqi Huang, Alexi Gladstone, Yunxin Liu, Xiang Chen, Heng Ji, Huxian Liu, and Luhui Hu. EBT-Policy: Energy unlocks emergent physical reasoning capabilities. *arXiv preprint arXiv:2510.27545*, 2025.

Zibin Dong, Yifu Yuan, Jianye Hao, Fei Ni, Yi Ma, Pengyi Li, and Yan Zheng. Cleandiffuser: An easy-to-use modularized library for diffusion models in decision making. *Advances in Neural Information Processing Systems*, 37:86899–86926, 2024.

Alec Farid, Sushant Veer, Divyanshu Pachisia, and Anirudha Majumdar. Task-driven detection of distribution shifts with statistical guarantees for robot learning. *IEEE Transactions on Robotics*, 41:926–945, 2024.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Angelos Filos, Panagiotis Tigas, Rowan McAllister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *International Conference on Machine Learning*, pp. 3145–3153. PMLR, 2020.

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

Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S Brown, and Ken Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *arXiv preprint arXiv:2109.08273*, 2021.

Runhan Huang, Haldun Balim, Heng Yang, and Yilun Du. Flexible locomotion learning with diffusion model predictive control. *arXiv preprint arXiv:2510.04234*, 2025.

Xiaoyu Huang, Yufeng Chi, Ruofeng Wang, Zhongyu Li, Xue Bin Peng, Sophia Shao, Borivoje Nikolic, and Koushil Sreenath. Diffuseloco: Real-time legged locomotion control with diffusion from offline datasets. *arXiv preprint arXiv:2404.19264*, 2024.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

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

Sung-Wook Lee, Xuhui Kang, and Yen-Ling Kuo. Diff-dagger: Uncertainty estimation with diffusion policy for robotic manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4845–4852. IEEE, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, et al. G0. 5: One autoregressive stream for robot reasoning and action. *arXiv preprint arXiv:2608.11739*, 2026.

Kunal Menda, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Ensembledagger: A bayesian approach to safe imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5041–5048. IEEE, 2019.

Jalal Naghiyev, Kirill Khvastow, Alexi Gladstone, Ralf Römer, Heng Ji, and Angela P Schoellig. Energy-based action heads know when they don't know. In *ICRA 2026 Workshop: From Data to Decisions: VLA Pipelines for Real Robots*.

Chaoyi Pan, Giridharan Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, et al. Much ado about noising: Dispelling the myths of generative robotic control. In *International Conference on Learning Representations*, volume 2026, pp. 90575–90614, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Sumeet Singh, Stephen Tu, and Vikas Sindhwani. Revisiting energy based models as policies: Ranking noise contrastive estimation and interpolating energy models. *arXiv preprint arXiv:2309.05803*, 2023.

Qiuyue Wang, Mingsheng Li, Jian Guan, Jinhui Ye, Sicheng Xie, Yitao Liu, Junhao Chen, Zhixuan Liang, Jie Zhang, Xintong Hu, et al. Qwen-vla: Unifying vision-language-action modeling across tasks, environments, and robot embodiments. *arXiv preprint arXiv:2605.30280*, 2026.

Runqian Wang and Yilun Du. Equilibrium matching: Generative modeling with implicit energy-based models. *arXiv preprint arXiv:2510.02300*, 2025.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv preprint arXiv:1605.06450*, 2016.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11142–11152. IEEE, 2025.

Xiangcheng Zhang and Yilun Du. World action planner: Generalizable decision-making with action-conditioned world models. *arXiv preprint arXiv:2607.27599*, 2026.

Zunzhe Zhang, Runhan Huang, Yicheng Liu, Shaoting Zhu, Linzhan Mou, and Hang Zhao. Generative control as optimization: Time unconditional flow matching for adaptive and robust robotic control. *arXiv preprint arXiv:2603.17834*, 2026.

A Evaluation Details

Table 4: Evaluation coverage and rollout budgets.

Benchmark	Coverage	Episodes
LIBERO–DiT	4 suites; 40 tasks	50/task
LIBERO–VLA	4 suites; 40 tasks	50/task
RoboTwin 2.0	5 tasks; Easy and Hard	100/task/setting
VLABench	Tracks 1, 2, 3, 4, and 6; 50 tasks	50/task
LIBERO-Plus	7 perturbation families	10030 total

A.1 Implementation Protocol

Both implementations use the convention $t = 0$ for noise and $t = 1$ for data, and query the stationary backbone at model time zero. The scalar energy is reduced independently for each sample over active action coordinates, $E_\theta(x, z) = \sum_{i \in \mathcal{A}} x_i f_{\theta,i}(x, z)$; the VLA mask excludes padded embodiment dimensions. Training differentiates this scalar with respect to the complete action chunk and retains the higher-order graph needed to update θ. Inference performs gradient ascent and detaches the action between refinement steps. The reported models use $c(t) = \min\{4, 20(1 - t)\}$, with $C(t)$ evaluated analytically; the analysis allows any schedule satisfying the stated integrability conditions.

A.2 BENCHMARKS

LIBERO. We evaluate the DiT policy on the LIBERO simulation benchmark, which contains four task suites—Goal, Spatial, Object, and Long—with ten tasks per suite. These suites evaluate different aspects of language-conditioned robot manipulation, including changes in task goals, spatial configurations, object categories, and long-horizon compositional tasks. We use the shared DiT architecture and data pipeline described in Section 5.

RoboTwin 2.0. We evaluate the VLA policy on RoboTwin 2.0, a benchmark for bimanual robot manipulation under both clean and domain-randomized conditions. We consider five representative tasks: Adjust Bottle, Beat Block Hammer, Blocks Ranking (Size), Click Alarmclock, and Click Bell. Each task is evaluated under the Easy and Hard settings. The Easy setting uses clean environments, whereas the Hard setting introduces additional domain randomization. We report task-wise success rates and average performance across the five tasks for both settings. All results are averaged over 100 evaluation episodes per task and setting using the benchmark's success criteria.

VLABench. We follow the current official VLA evaluation protocol and report the five tracks with released standardized episode configurations: in-distribution task learning (T1), cross-category and object generalization (T2), commonsense understanding (T3), semantic-instruction understanding (T4), and unseen background or table textures (T6). Each task is evaluated over 50 episodes, and we report track-wise success rates and their average. Cross-task skill transfer (T5) is part of the broader VLABench design, but the official protocol defines it as an open setting in which users choose their own training and evaluation tasks; it does not provide a fixed task split or standardized episode configuration. We therefore omit T5 and use the publicly reproducible five-track configuration rather than introducing a study-specific cross-task split.

LIBERO-Plus. We evaluate robustness under controlled distribution shifts using LIBERO-Plus, which extends the original LIBERO tasks with perturbations that preserve the underlying task semantics. We consider seven perturbation families: camera viewpoint, robot initialization, language, illumination, background, sensor noise, and object layout. We report success rates for each perturbation family and the corresponding robustness aggregate.

A.3 DIT DETAILS

For LIBERO experiments with the Diffusion Transformer (DiT), we encode visual observations using a pretrained DINOv2-base (`dinov2-base`) model and language instructions using T5-base (`t5-base`), with an observation horizon of $T_o = 1$. The policy predicts continuous action chunks with horizon $T_a = 16$ and $D_a = 7$ control dimensions ($D = 112$), executing the first 8 steps under receding-horizon control.

Training. We train all DiT models for 70k steps with a batch size of 256 in `bf16-mixed` precision. Training uses the AdamW optimizer with cosine learning-rate decay, a peak learning rate of 1×10^{-4}, 5,000 linear warmup steps, and weight decay of 1×10^{-4}.

A.4 MULTI-SEED DIT RESULTS

Table 5 reports the LIBERO results obtained across multiple EqC training seeds. The final row gives the mean and sample standard deviation for each suite. Unlike the published baseline values retained in the main comparison, Table 6 reports our independent multi-seed reproductions of the matched rectified-flow and GeCO objectives alongside EqC, without listing individual baseline runs.

A.5 VLA TRAINING DETAILS

We fine-tune π_0 and $\pi_{0.5}$ VLAs from OpenPI base checkpoints on LeRobot-format datasets using the AdamW optimizer with gradient clipping (`clip_gradient_norm=1.0`) and a cosine learning-rate schedule (10k warmup steps and a peak learning rate of 5×10^{-5}). Models are trained for 50k steps on LIBERO and 30k steps on RoboTwin 2.0 and VLABench.

Table 5: Detailed EqC success rates (%) across multiple training seeds on LIBERO.

Seed	Goal	Spatial	Object	Long	Avg.
0	95.6	97.2	98.6	85.4	94.3
1	96.0	97.0	98.4	87.0	94.7
2	95.0	96.0	97.6	85.4	93.5
3	95.8	95.2	97.8	84.2	93.3
Mean $\pm$ sample s.d.	95.60 ± 0.43	96.35 ± 0.93	98.10 ± 0.48	85.50 ± 1.15	93.95 ± 0.66

Table 6: Overall LIBERO success across multiple training seeds.

Method	Mean (%)	Sample s.d. (%)
Fixed RF, 20 steps	89.12	0.80
GeCO	92.30	1.02
EqC (ours)	**93.95**	**0.66**

RoboTwin 2.0. We fine-tune a task-specific π_0 model for each of the five bimanual tasks (batch size 32, 30k steps). Inputs comprise three RGB camera views (`cam_high`, `cam_left_wrist`, `cam_right_wrist`) and robot proprioception, predicting action chunks of horizon $H = 50$ with 14 active control dimensions ($D = 700$).

LIBERO and LIBERO-Plus. For LIBERO, we fine-tune π_0 on the `physical-intelligence/libero` dataset for 50k steps with batch size 32. For LIBERO-Plus, we directly evaluate the LIBERO-trained checkpoint in a zero-shot manner across all seven perturbation suites without any benchmark-specific adaptation.

VLABench. For VLABench, models are fine-tuned for 30k steps on `vlabench/vlabench_primitive_ft_lerobot` across five evaluation tracks, using a batch size of 256 for π_0 and 32 for $\pi_{0.5}$.

A.6 Supplementary LIBERO Results for VLA Policies

We additionally evaluate EqC with VLA policies (π_0) on LIBERO to examine whether the formulation transfers to large multimodal action models. Table 7 reports the full comparison across the four LIBERO task suites.

Table 7: LIBERO success rates (%) with the VLA backbone.

Method	Goal	Spatial	Object	Long	Avg.
π_0 + FAST (Pertsch et al., 2025)	88.6	96.4	96.8	60.2	85.5
π_0	95.8	96.8	**98.8**	85.2	94.2
π_0 + GeCO	**96.4**	96.8	97.0	85.4	93.9
π_0 + **EqC (ours)**	94.6	**97.6**	**98.8**	**86.2**	**94.3**

EqC achieves a 94.3% aggregate success rate with the VLA policy, together with strong results on Spatial, Object, and Long. Combined with the DiT experiments, these results demonstrate the applicability of the equilibrium-energy formulation across both policy families. Sections 5.2 and 5.3 further evaluate the introspective competence signal enabled by the VLA instantiation and its use for lifelong learning.

B COMPETENCE AWARENESS DETAILS

B.1 TASK DEFINITION: ONLINE FAILURE DETECTION

Our goal is online failure detection rather than detecting visual novelty or distribution shift. In robot manipulation, environmental shifts do not necessarily imply execution failure: an adaptable policy may handle unfamiliar scenes autonomously, where takeover is unnecessary and disruptive. Conversely, executions under nominal setups frequently fail due to contact slip, occlusions, or compounding control error. Intervention should therefore depend on whether the policy is actually failing the task, rather than whether the current observation looks unfamiliar.

Ground-truth labels are determined strictly by task outcomes: successful rollouts are labeled negative (autonomous completion, no intervention needed), while failed rollouts are labeled positive (intervention required). This formulation avoids penalizing successful generalization: standard novelty detectors flag unfamiliar yet competent executions as anomalies, prompting unnecessary human takeover. The reported metrics evaluate the ability to anticipate and catch execution failures rather than measuring observation density. Labels remain tied to final task success rather than the scene configuration.

B.2 SIMULATION FAILURE DETECTION PROTOCOL

We evaluate competence-aware failure detection using three methodological configurations:

1. **EqC (ours):** Our primary approach quantifies policy competence using the energy-consistency signal $U_{\Delta E}$ derived from the EqC-adapted π_0 policy. As introduced in Section 3.3, this score projects each post-update action onto the noise-to-equilibrium chord using the floored denominator $D_v = \max(\|x^{(K)} - x^{(0)}\|_2^2, 10^{-12})$ and averages the absolute energy-consistency residuals according to equation 4.
2. **Diff-DAgger (baseline):** The standard baseline using the original π_0 policy paired with Diff-DAgger's multi-pass flow-loss detector (Lee et al., 2025). Implementation details are provided in Section B.5.
3. **EqC + Diff-DAgger (hybrid):** A hybrid ablation variant that deploys our EqC-adapted π_0 policy for action generation while computing the failure-detection signal using Diff-DAgger's re-noising and denoising flow-loss scoring mechanism.

All three methods are evaluated on the LIBERO and LIBERO-Plus benchmark suites using 50k-step training checkpoints. A failure alarm is triggered when the detection signal exceeds a predefined threshold for m consecutive replanning steps.

The statistical unit of the analysis is a complete rollout rather than an individual replanning step. Before sampling either split, we stratify the evaluation pool across the 8 dataset variants (the original LIBERO and the 7 LIBERO-Plus perturbation families). Within every stratum, rollouts are assigned proportionally to a 10% calibration split and a 90% test split, ensuring that neither split is dominated by any particular task or perturbation type. A rollout is treated as an indivisible unit: its initial-state instance, outcome label, and complete sequence of replanning signals are assigned strictly to one split. No rollout or trajectory segment is shared across calibration and testing, and all compared detectors evaluate on the exact same split membership. Further details regarding the composition of the calibration split are provided in Section B.4.

Policy checkpoints are fixed before detector calibration and remain unchanged throughout the analysis. The calibration split is used only to select the temporal window m and operating threshold τ^*; it is not used for checkpoint selection or detector adaptation. Once selected, (m, τ^*) is frozen and applied once to the held-out test rollouts. Test outcomes and signal trajectories therefore cannot influence the alarm rule. By preserving the task, perturbation, and outcome composition across the two splits, the protocol evaluates rollout-level failure discrimination without confounding the comparison through a different benchmark mixture.

Each rollout produces a time-varying sequence of failure-detection signals. For a fixed window size m and threshold τ, a rollout raises an alarm if its signal exceeds τ for any sequence of m consecutive replanning steps; otherwise, no alarm is raised. We then compute TPR and FPR from

rollout-level success and failure labels and construct the ROC curve by sweeping τ. To examine whether performance remains stable across calibration procedures rather than being tied to a particular threshold-selection rule, we evaluate two calibration strategies on the calibration split:

1. **FPR-Constrained Calibration:** As reported in the main text (Section 5.2.2), all candidate methods are first evaluated on the 10% calibration split. By performing a fine-grained sweep over candidate thresholds logged from these rollouts, we fix a given window size m and select the optimal threshold τ^* that maximizes TPR while strictly constraining the FPR to approximately 10%. This calibrated parameter pair (m, τ^*) is then frozen and applied to evaluate the remaining 90% test split.
2. **Youden's J Index Calibration:** Alternatively, for a given sliding window size m, the optimal threshold τ^* is selected on the 10% calibration split by maximizing Youden's J statistic ($J = \text{TPR} - \text{FPR}$). The calibrated threshold pair (m, τ^*) is subsequently deployed on the 90% test split to trigger failure alarms.

Across both FPR-constrained calibration and Youden's J index calibration in Table 9, EqC consistently achieves the highest AUROC and delivers the most favorable overall trade-off among TPR, FPR, and Accuracy across the vast majority of window sizes m. This broader advantage indicates that our competence signal provides robust failure detection without depending on a specific threshold-selection rule.

AUROC is independent of the operating threshold τ, so we report a single global ranking statistic obtained by sweeping τ over all labeled rollouts rather than a separate AUROC for each calibration rule. This descriptive quantity is not used to select (m, τ^*) or to compute the threshold-dependent operating metrics, which retain the calibration/test separation above. Regarding the time-saved metric, when an alarm is raised at elapsed horizon h_{alarm} in an episode capped at $H_{\max}$, the percentage of saved inference time is calculated as $100(1 - h_{\text{alarm}}/H_{\max})\%$.

B.3 Real-World Failure Detection Protocol

The real-world testbed evaluates the intermediate Round 3 policy checkpoint fine-tuned on $\mathcal{D}_3$ (36 cumulative demonstrations) on *Hierarchical Block Stacking*. Across nominal block configurations , we collect 50 rollouts comprising 25 successful and 25 failed executions, with binary ground-truth labels assigned strictly from the final task outcome. To reduce short-timescale measurement noise, each detection signal is smoothed with a three-step moving average: starting from the third replanning step, the current value is averaged with the two preceding values. An alarm is raised when the smoothed signal remains above its threshold for m consecutive replanning steps. Appendix D provides the platform, task, and rollout-collection details.

B.4 Composition of the Calibration Split

The shared 10% calibration split comprises $1{,}203$ rollouts in total. To ensure balanced coverage across environmental conditions, we perform stratified sampling across the 8 dataset variants (the original LIBERO and 7 LIBERO-Plus perturbation types). Within each dataset variant, rollouts are selected purely at random and shared identically across all evaluated methods (though ground-truth outcome labels depend on each policy's execution). As detailed in Table 8, because rollouts are randomly sampled within each perturbation family, the distribution across the four suites naturally yields a nearly uniform breakdown ($\approx 25\%$ per suite) without requiring explicit suite-level stratification.

B.5 Diff-DAgger Baseline Implementation Adapted for π_0

To adapt Diff-DAgger as a failure-detection baseline for the π_0 backbone, we use π_0's conditional flow-matching formulation. The detector evaluates whether an action sequence generated by π_0 is consistent with the flow-matching vector field learned during training. Because this inference-only procedure does not modify the original π_0 training objective, it preserves the π_0 architecture, including the PaliGemma vision-language backbone, action expert, observation preprocessing, and action projection head. The detector is attached directly to the original Euler sampler.

Table 8: Detailed composition of the shared 10% calibration split (1,203 total rollouts).

(a) Breakdown by Perturbation Variant

Dataset	Count	Ratio (%)
Original LIBERO	200	16.63
Camera	160	13.30
Noise	160	13.30
Robot	155	12.88
Language	154	12.80
Layout	152	12.64
Light	114	9.48
Background	108	8.98

(b) Breakdown by Benchmark Suite

Suite	Count	Ratio (%)
LIBERO-Goal	307	25.52
LIBERO-Object	302	25.10
LIBERO-Spatial	299	24.85
LIBERO-10	295	24.52

Given an observation o, π_0 first generates a predicted action sequence $\hat{a}$ through its standard Euler sampler:

$$\hat{a} = S_\theta(o, \epsilon_0), \tag{6}$$

where $\epsilon_0 \sim \mathcal{N}(0, I)$ is an initial Gaussian noise sample. The generated action sequence $\hat{a}$ is then treated as a pseudo-label, so the detector requires no expert demonstrations or additional supervision at test time.

For each Monte Carlo sample $q \in \{1, \ldots, M\}$, we independently sample a flow-matching timestep t_q and Gaussian noise ϵ_q:

$$t_q \sim \mathrm{Beta}(1.5, 1), \qquad \epsilon_q \sim \mathcal{N}(0, I). \tag{7}$$

Following π_0's standard training formulation, we construct a perturbed version of the generated action x_{t_q}:

$$x_{t_q} = t_q \epsilon_q + (1 - t_q)\hat{a}, \tag{8}$$

with the corresponding target velocity defined as

$$u_{t_q} = \epsilon_q - \hat{a}. \tag{9}$$

The π_0 model is then queried using the original observation o, the perturbed action x_{t_q}, and the sampled timestep t_q, producing the predicted velocity vector:

$$v_\theta(x_{t_q}, t_q, o). \tag{10}$$

The Diff-DAgger score for the q-th Monte Carlo sample is the normalized flow-matching error

$$\ell_q = \frac{1}{H D_{\mathrm{act}}} \sum_{h=1}^{H} \sum_{d=1}^{D_{\mathrm{act}}} m_d \left[v_\theta(x_{t_q}, t_q, o)_{h,d} - u_{t_q,h,d} \right]^2, \tag{11}$$

where H represents the action horizon, D_{act} denotes the number of active action dimensions, and $m_d \in \{0, 1\}$ is an action-space mask. This mask is essential for π_0 as its action representation includes zero-padded dimensions, which are excluded from both the pseudo-label construction and loss evaluation.

The final per-query failure score is computed as the averaged flow loss over M Monte Carlo evaluations:

$$\mathcal{L}_{\mathrm{flow}} = \frac{1}{M} \sum_{q=1}^{M} \ell_q. \tag{12}$$

We reserve K for the number of EqC refinement iterates and use M for the number of Diff-DAgger Monte Carlo samples; the default EqC and Diff-DAgger configurations use $K = 10$ and $M = 10$, respectively.

B.6 Detailed Results

As presented in Table 9 and Table 10, we evaluate failure detection performance and early intervention capability across sliding window sizes $m \in [3, 13]$ under both calibration protocols. Note that

AUROC reflects the threshold-independent detection capacity evaluated globally across all dataset rollouts (and is thus independent of split partitions and identical across both subtables for any given m), whereas TPR, FPR, Accuracy, and Time Saved reflect operational performance on the 90% test split using operating thresholds calibrated from the 10% calibration split.

A comprehensive comparative analysis highlights the clear advantages of EqC across window sizes, operating metrics, and thresholding strategies:

- **Dominance in Theoretical Detection Capacity (AUROC):** Across all evaluated window sizes $m \in [3, 13]$, EqC consistently achieves the highest AUROC values among all compared methods, ascending monotonically from 0.8877 ($m = 3$) to 0.9300 ($m = 13$). It maintains a decisive lead over both the baseline Diff-DAgger (0.8296–0.8576) and the hybrid variant EqC + Diff-DAgger (0.8778–0.8978), confirming that energy-derived competence signals provide an inherently stronger and more reliable failure detection capability.
- **Superiority under FPR-Constrained Calibration:** Under strict false-alarm control (Table 9a, target FPR $\approx 10\%$), EqC significantly outperforms Diff-DAgger in both TPR and overall Accuracy across almost all m. For example, at $m = 13$, EqC reaches 78.9% TPR and 87.4% Accuracy, compared to Diff-DAgger's 63.8% TPR and 83.1% Accuracy. Furthermore, EqC maintains a lead over the hybrid variant across the majority of window sizes, demonstrating higher TPR.
- **Superiority under Youden's J Calibration:** When optimizing the decision threshold via Youden's J index ($J = \text{TPR} - \text{FPR}$, Table 9b), EqC consistently achieves higher J statistic values, yielding a distinctly superior trade-off between TPR and FPR control. Under this objective, EqC attains higher peak TPR (reaching up to 87.5% at $m = 12$) while securing the highest overall classification Accuracy across the vast majority of window sizes (peaking at 86.4% at $m = 11$). This confirms that EqC's operational superiority stems from a fundamentally better signal trade-off profile rather than an artifact of a specific calibration rule.
- **Responsiveness in Early Intervention (Time Saved):** Beyond accurately classifying failures, a practical failure-detection system must intervene timely to allow trajectory recovery. Table 10 reports the execution time saved specifically on failed rollouts within the 90% test split. Across both calibration protocols, EqC achieves the highest overall time saved across almost all window sizes m, consistently outperforming the baseline Diff-DAgger and surpassing the hybrid variant (EqC + Diff-DAgger) in the vast majority of cases. While the hybrid model also improves upon pure Diff-DAgger—further confirming the value of incorporating energy-based information—EqC provides the most responsive signal overall, timely avoiding rollouts from reaching an irrecoverable failure state.

Overall, these results confirm that EqC provides a robust, scalable, and highly responsive failure-detection signal regardless of the choice of threshold calibration, window size, or performance metric.

To further establish the theoretical upper bound on detection performance independent of split partitioning, Table 11 presents an evaluation conducted directly on the entire dataset (without calibration/test splitting). In this setup, AUROC is calculated by sweeping the threshold globally, while TPR, FPR, and Accuracy are reported using the theoretical optimal threshold determined by maximizing Youden's J index over the full dataset. Across all evaluated sliding-window sizes $m \in [3, 13]$, EqC consistently achieves the highest AUROC and the best optimal TPR/FPR trade-off compared to the Diff-DAgger baseline ($M = 10$) and the hybrid variant. Furthermore, increasing Diff-DAgger's sample count from $M = 10$ to $M \in \{16, 32, 64, 128\}$ fails to close this performance gap, indicating that merely scaling Monte Carlo dropout samples does not yield a fundamentally better failure-detection signal.

Beyond detection accuracy, Table 12 provides a detailed breakdown of the theoretical maximum inference time saved across various environmental perturbation suites in LIBERO and LIBERO-Plus. Standalone EqC achieves the highest overall time saved, outperforming both the baseline Diff-DAgger and the hybrid EqC + Diff-DAgger variant across the majority of evaluation setups and representative window sizes $m \in \{4, 11, 13\}$, reaching up to 54.0% in weighted average time saved. Notably, even when scaling Diff-DAgger's sample budget to $M \in \{16, 32\}$, its performance in time saved

Table 9: Sensitivity analysis of detection window size m under two distinct threshold calibration strategies on the 90% test split.

(a) FPR-Constrained Calibration (Target FPR $\approx 10\%$ on calibration split)

	Diff-DAgger			EqC + Diff-DAgger			**EqC (ours)**		
m	AUROC	TPR / FPR (%)	Acc (%)	AUROC	TPR / FPR (%)	Acc (%)	AUROC	TPR / FPR (%)	Acc (%)
3	0.8296	58.1 / 9.1	82.1	0.8778	71.9 / 9.9	84.8	0.8877	69.5 / 9.1	84.6
4	0.8307	56.3 / 8.3	82.2	0.8774	70.4 / 8.8	85.1	0.8947	70.8 / 8.9	85.1
5	0.8310	57.1 / 8.6	82.2	0.8802	70.9 / 9.3	84.9	0.9005	70.7 / 7.9	85.8
6	0.8328	55.7 / 8.1	82.2	0.8810	71.2 / 9.2	85.1	0.9070	74.7 / 9.0	86.2
7	0.8343	58.3 / 9.2	82.1	0.8818	71.7 / 8.9	85.4	0.9117	74.8 / 9.1	86.1
8	0.8345	59.8 / 9.9	82.0	0.8829	72.4 / 9.5	85.2	0.9147	76.1 / 9.3	86.3
9	0.8352	61.5 / 11.0	81.6	0.8849	73.5 / 9.8	85.3	0.9183	76.7 / 8.9	86.9
10	0.8391	60.3 / 10.1	82.0	0.8862	72.5 / 9.1	85.5	0.9222	77.7 / 8.9	87.1
11	0.8430	61.3 / 10.2	82.2	0.8888	72.8 / 8.8	85.8	0.9244	77.9 / 8.5	87.5
12	0.8517	62.3 / 9.2	83.2	0.8936	74.6 / 10.4	85.2	0.9274	78.4 / 8.4	87.6
13	0.8576	63.8 / 9.9	83.1	0.8978	74.0 / 9.2	85.9	0.9300	78.9 / 9.0	87.4

(b) Youden's J Index Calibration (Maximizing $J = \text{TPR} - \text{FPR}$ on calibration split)

	Diff-DAgger			EqC + Diff-DAgger			**EqC (ours)**		
m	AUROC	TPR / FPR (%)	Acc (%)	AUROC	TPR / FPR (%)	Acc (%)	AUROC	TPR / FPR (%)	Acc (%)
3	0.8296	73.9 / 20.8	77.8	0.8778	78.3 / 16.9	81.7	0.8877	78.1 / 14.9	83.0
4	0.8307	65.2 / 13.3	80.9	0.8774	80.6 / 19.5	80.5	0.8947	82.4 / 20.0	80.7
5	0.8310	76.0 / 24.4	75.7	0.8802	79.2 / 16.8	82.0	0.9005	82.1 / 17.5	82.4
6	0.8328	73.0 / 19.9	78.2	0.8810	80.0 / 18.1	81.4	0.9070	79.4 / 13.0	84.7
7	0.8343	75.0 / 23.2	76.3	0.8818	78.6 / 15.9	82.5	0.9117	85.3 / 20.2	81.4
8	0.8345	70.4 / 18.9	78.2	0.8829	76.9 / 13.7	83.6	0.9147	81.9 / 14.5	84.4
9	0.8352	65.5 / 14.5	80.2	0.8849	73.8 / 10.1	85.2	0.9183	83.8 / 15.5	84.3
10	0.8391	71.3 / 17.6	79.4	0.8862	80.8 / 18.0	81.7	0.9222	84.0 / 14.7	85.0
11	0.8430	70.3 / 16.4	80.1	0.8888	78.1 / 14.2	83.6	0.9244	82.1 / 11.8	86.4
12	0.8517	65.1 / 11.0	82.6	0.8936	77.5 / 12.7	84.4	0.9274	87.5 / 18.3	83.4
13	0.8576	69.0 / 13.6	81.7	0.8978	74.6 / 9.6	85.8	0.9300	85.5 / 14.5	85.5

Table 10: Evaluation of early intervention capability (Time Saved %) on the 90% test split. The operational decision thresholds are calibrated using the 10% calibration split under FPR-Constrained and Youden's J index strategies.

	FPR-Constrained Calibration			Youden's J Index Calibration		
m	Diff-DAgger	EqC + Diff	**EqC (ours)**	Diff-DAgger	EqC + Diff	**EqC (ours)**
3	33.6	44.5	45.2	47.9	51.8	53.7
4	31.8	42.5	45.3	39.2	53.8	57.4
5	32.1	42.7	44.5	49.4	51.8	55.5
6	30.6	42.6	46.7	45.2	52.2	51.5
7	32.0	42.8	46.6	47.0	50.2	57.1
8	32.5	43.0	46.3	42.4	47.9	52.2
9	33.9	43.5	46.2	37.4	43.8	53.7
10	32.7	42.4	46.0	41.8	51.0	53.0
11	33.3	42.1	45.5	40.4	47.9	49.8
12	33.3	43.8	45.3	35.4	46.4	54.8
13	34.3	42.4	45.3	38.6	42.9	51.5

Table 11: Evaluation of the theoretical upper bound of online failure detection across window sizes $m \in \{3, \ldots, 13\}$ under Youden's optimal thresholding strategy (Best Youden). *Top:* Comparison among the baseline Diff-DAgger, the EqC + Diff-DAgger hybrid, and our proposed EqC; EqC and Diff-DAgger use their default $K = 10$ and $M = 10$ samples, respectively. *Bottom:* Performance of Diff-DAgger when scaling $M \in \{16, 32, 64, 128\}$. TPR and FPR are reported as percentages at Youden's optimal threshold.

(a) Primary Method Comparison

	Diff-DAgger (baseline)			EqC + Diff-DAgger			**EqC (ours)**		
m	AUROC	TPR / FPR (%)	Acc (%)	AUROC	TPR / FPR (%)	Acc (%)	AUROC	TPR / FPR (%)	Acc (%)
3	0.8296	72.9 / 19.4	78.5	0.8778	73.8 / 11.5	**84.2**	**0.8877**	78.0 / 15.0	82.9
4	0.8307	72.3 / 18.9	78.7	0.8774	76.7 / 14.8	82.7	**0.8947**	79.2 / 15.3	**83.0**
5	0.8310	67.8 / 15.1	80.3	0.8802	75.7 / 13.4	**83.4**	**0.9005**	81.1 / 15.9	83.2
6	0.8328	74.0 / 20.6	78.0	0.8810	75.4 / 12.6	83.9	**0.9070**	79.2 / 12.7	**84.9**
7	0.8343	69.5 / 16.0	80.1	0.8818	76.3 / 13.5	83.5	**0.9117**	78.2 / 11.7	**85.3**
8	0.8345	69.1 / 16.5	79.7	0.8829	76.6 / 13.3	83.7	**0.9147**	82.2 / 14.8	**84.3**
9	0.8352	72.1 / 19.5	78.2	0.8849	76.5 / 12.9	84.0	**0.9183**	83.0 / 14.4	**84.8**
10	0.8391	71.7 / 17.6	79.6	0.8862	78.2 / 14.1	83.6	**0.9222**	81.1 / 11.5	**86.3**
11	0.8430	71.8 / 16.9	80.1	0.8888	78.9 / 14.9	83.2	**0.9244**	82.7 / 12.5	**86.1**
12	0.8517	73.6 / 17.5	80.1	0.8936	79.1 / 14.3	83.8	**0.9274**	82.3 / 11.9	**86.4**
13	0.8576	75.9 / 19.2	79.5	0.8978	77.1 / 11.8	85.0	**0.9300**	86.6 / 15.4	**85.2**

(b) Diff-DAgger Scaling Across M

	Diff-DAgger ($M = 16$)		Diff-DAgger ($M = 32$)		Diff-DAgger ($M = 64$)		Diff-DAgger ($M = 128$)	
m	AUROC	TPR / FPR (%)	AUROC	TPR / FPR (%)	AUROC	TPR / FPR (%)	AUROC	TPR / FPR (%)
3	0.8266	68.0 / 14.6	0.8302	74.5 / 21.1	0.8342	71.6 / 16.2	0.8264	71.3 / 18.2
4	0.8255	69.3 / 16.8	0.8263	70.2 / 17.1	0.8315	72.4 / 18.1	0.8216	68.3 / 16.3
5	0.8280	71.6 / 18.5	0.8250	67.9 / 15.4	0.8294	71.8 / 17.9	0.8210	69.6 / 18.1
6	0.8270	67.2 / 14.6	0.8261	70.9 / 18.8	0.8295	68.9 / 15.4	0.8218	67.4 / 16.1
7	0.8289	68.9 / 15.8	0.8295	71.0 / 18.0	0.8289	68.8 / 15.7	0.8210	65.5 / 13.8
8	0.8298	69.8 / 16.7	0.8297	69.2 / 17.2	0.8311	69.8 / 16.7	0.8209	66.1 / 14.3
9	0.8304	69.6 / 16.2	0.8306	67.9 / 15.2	0.8281	69.9 / 17.0	0.8203	67.9 / 15.8
10	0.8316	70.9 / 17.1	0.8313	69.2 / 16.3	0.8292	69.2 / 16.4	0.8219	68.0 / 16.2
11	0.8338	68.8 / 14.7	0.8355	67.5 / 14.2	0.8334	70.3 / 17.1	0.8248	67.7 / 15.5
12	0.8389	72.1 / 18.1	0.8414	73.8 / 19.8	0.8406	70.1 / 16.0	0.8317	65.6 / 12.6
13	0.8437	70.8 / 15.8	0.8470	70.7 / 15.8	0.8481	70.6 / 15.9	0.8385	72.6 / 19.0

Table 12: Breakdown of theoretically optimal inference time saved (%) across LIBERO and LIBERO-Plus suites under representative evaluation setups. *Orig.* corresponds to the LIBERO tasks, while the remaining seven columns represent environmental perturbation variants in LIBERO-Plus. Diff-DAgger uses $M = 10$ unless otherwise specified.

Method	Orig.	Camera	Robot	Lang.	Light	Backgr.	Noise	Layout	W. Avg.
Diff-DAgger ($m = 4$)	27.7	50.5	50.1	46.4	30.0	34.0	44.1	47.9	46.1
Diff-DAgger ($m = 11$)	26.5	44.2	46.0	39.9	32.5	32.3	37.3	44.9	41.5
Diff-DAgger ($m = 13$)	28.9	47.2	48.6	44.1	33.8	34.5	43.1	46.1	44.6
Diff-DAgger ($M = 16, m = 4$)	27.0	45.0	47.5	44.1	33.8	28.0	39.0	44.0	42.7
Diff-DAgger ($M = 16, m = 11$)	23.5	41.3	42.4	37.4	28.5	30.2	35.1	40.6	38.4
Diff-DAgger ($M = 16, m = 13$)	25.9	42.8	43.5	39.0	30.7	30.5	37.1	41.8	39.8
Diff-DAgger ($M = 32, m = 4$)	28.1	48.5	48.3	42.8	34.9	34.4	38.5	45.6	43.9
Diff-DAgger ($M = 32, m = 11$)	28.3	40.4	42.9	34.8	31.1	28.7	36.1	38.9	38.0
Diff-DAgger ($M = 32, m = 13$)	28.6	42.9	44.1	37.4	32.3	31.3	38.6	40.6	40.0
EqC + Diff-DAgger ($m = 4$)	30.7	50.3	51.8	56.9	44.1	27.0	44.5	48.9	49.1
EqC + Diff-DAgger ($m = 11$)	30.3	51.4	50.8	53.7	42.6	29.1	46.3	48.3	48.5
EqC + Diff-DAgger ($m = 13$)	26.8	48.4	47.0	50.2	40.8	26.4	43.5	45.2	45.3
EqC (ours)($m = 4$)	26.7	54.8	57.3	70.7	41.7	29.0	40.1	55.4	**54.0**
EqC (ours)($m = 11$)	24.7	53.4	55.3	60.8	39.7	30.2	41.5	47.8	**50.5**
EqC (ours)($m = 13$)	26.7	56.4	58.2	61.0	40.3	32.2	46.1	49.5	**52.8**

does not improve, demonstrating that simply increasing trajectory sampling density cannot enhance its failure detection capability. While the hybrid model also demonstrates notable gains over pure Diff-DAgger—further underscoring the benefit of energy-derived features—standalone EqC provides the most effective and responsive signal overall for timely failure intervention.

Finally, we evaluate the computational efficiency of EqC by benchmarking per-replan inference latency, encompassing both action candidate generation and failure-score computation at each step. As shown in Table 13, standalone EqC exhibits an inference latency comparable to that of low-budget Diff-DAgger variants. Crucially, as demonstrated in our earlier evaluations, EqC achieves significantly stronger and more reliable failure-detection performance than Diff-DAgger. While increasing Diff-DAgger's sample budget M monotonically inflates its inference latency, it yields no significant improvement in failure detection capability, rendering larger sample budgets computationally inefficient for real-time introspection. Furthermore, EqC + Diff-DAgger incurs a substantial latency overhead compared to either standalone EqC or Diff-DAgger, proving impractical for physical deployment. Consequently, EqC delivers the optimal balance, offering superior failure detection while maintaining the computational responsiveness required for real-time robotic control.

Table 13: Comparison of per-replan inference latency (ms) across methods. The latency includes both model action generation and failure-detection signal computation (evaluated at batch size = 1 after JAX warm-up). The optimization step budget K for EqC is set to 10.

Method	Mean Latency (ms)
Diff-DAgger ($M = 10$)	79.44
Diff-DAgger ($M = 16$)	91.43
Diff-DAgger ($M = 32$)	122.11
EqC + Diff-DAgger ($M = 10$)	171.03
EqC (ours)	108.86

B.7 Extended Studies

To evaluate the fundamental advantage of tracking temporal trajectory dynamics over static point-wise metrics, we compare our path-wise energy-consistency EqC signal against two alternative single-point

energy-derived baselines: (1) **Terminal Energy**: the absolute energy value at the final step of the EqC action optimization; and (2) **Last-step Gradient Norm**: the gradient norm of the energy evaluated at the final optimization step.

To benchmark these signals, performance is evaluated across sliding-window sizes $m \in [3, 13]$, where AUROC is computed globally in a threshold-independent manner, while decision thresholds for TPR, FPR, and Accuracy are calibrated on a 10% split—using both FPR-constrained and Youden's J Index regimes—and evaluated on the remaining 90% test split. As shown in Table 14, comparing these signals reveals a clear progression: scalar **Terminal Energy** offers weak detection capability (AUROC 0.5237–0.6489) because uncalibrated absolute energy values vary significantly across different environmental states, whereas **Last-step Gradient Norm** improves performance substantially (AUROC ~ 0.87) by measuring local gradient stationarity at the final optimization step. Building on this, **EqC** achieves the highest threshold-independent discriminability across all m, reaching a peak AUROC of 0.9300. Under FPR-constrained calibration, EqC strictly dominates both baselines across all m, yielding higher TPR and Accuracy (up to 87.6%). Under Youden's J Index, EqC achieves the optimal trade-off between TPR and FPR across all sliding window sizes m, while recovering higher peak Accuracy as m grows. Overall, these results confirm that tracking path-wise temporal energy dynamics along the optimization trajectory provides a fundamentally richer and more reliable introspection signal than static, single-point evaluations.

Table 14: Ablation analysis comparing single-point energy signals against full path-wise EqC across window sizes m under two calibration strategies on the 90% test split.

(a) FPR-Constrained Calibration (Target FPR $\approx 10\%$ on calibration split)

	Terminal Energy			Last-step Gradient Norm			**EqC (ours)**		
m	AUROC	TPR / FPR (%)	Acc (%)	AUROC	TPR / FPR (%)	Acc (%)	AUROC	TPR / FPR (%)	Acc (%)
3	0.5237	17.1 / 10.0	68.4	0.8734	68.0 / 9.3	84.0	0.8877	69.5 / 9.1	84.6
4	0.5404	19.3 / 10.1	68.9	0.8707	67.1 / 9.1	83.9	0.8947	70.8 / 8.9	85.1
5	0.5604	23.1 / 10.3	69.9	0.8713	65.5 / 8.7	83.6	0.9005	70.7 / 7.9	85.8
6	0.5754	26.0 / 10.8	70.5	0.8724	65.7 / 9.4	83.2	0.9070	74.7 / 9.0	86.2
7	0.5861	26.8 / 10.2	71.1	0.8737	66.0 / 9.3	83.3	0.9117	74.8 / 9.1	86.1
8	0.5974	28.7 / 9.8	71.9	0.8737	66.2 / 9.5	83.3	0.9147	76.1 / 9.3	86.3
9	0.6082	29.9 / 9.4	72.6	0.8744	64.7 / 9.0	83.2	0.9183	76.7 / 8.9	86.9
10	0.6214	32.5 / 10.0	72.9	0.8758	64.2 / 8.5	83.4	0.9222	77.7 / 8.9	87.1
11	0.6309	33.1 / 9.6	73.4	0.8769	65.4 / 9.3	83.1	0.9244	77.9 / 8.5	87.5
12	0.6410	33.8 / 9.6	73.6	0.8766	64.1 / 8.8	83.1	0.9274	78.4 / 8.4	87.6
13	0.6489	34.9 / 9.7	73.8	0.8766	64.6 / 8.8	83.3	0.9300	78.9 / 9.0	87.4

(b) Youden's J Index Calibration (Maximizing $J = \text{TPR} - \text{FPR}$ on calibration split)

	Terminal Energy			Last-step Gradient Norm			**EqC (ours)**		
m	AUROC	TPR / FPR (%)	Acc (%)	AUROC	TPR / FPR (%)	Acc (%)	AUROC	TPR / FPR (%)	Acc (%)
3	0.5237	21.6 / 13.2	67.5	0.8734	67.6 / 9.0	84.0	0.8877	78.1 / 14.9	83.0
4	0.5404	22.5 / 12.4	68.3	0.8707	67.7 / 9.5	83.8	0.8947	82.4 / 20.0	80.7
5	0.5604	42.5 / 32.8	59.9	0.8713	64.3 / 7.8	83.9	0.9005	82.1 / 17.5	82.4
6	0.5754	24.0 / 9.0	71.1	0.8724	72.1 / 14.1	81.8	0.9070	79.4 / 13.0	84.7
7	0.5861	28.3 / 11.5	70.6	0.8737	68.6 / 11.0	83.0	0.9117	85.3 / 20.2	81.4
8	0.5974	34.5 / 17.4	68.4	0.8737	65.3 / 9.1	83.3	0.9147	81.9 / 14.5	84.4
9	0.6082	36.7 / 16.1	69.9	0.8744	76.0 / 16.4	81.4	0.9183	83.8 / 15.5	84.3
10	0.6214	33.6 / 11.0	72.5	0.8758	75.8 / 16.3	81.4	0.9222	84.0 / 14.7	85.0
11	0.6309	48.1 / 25.6	66.6	0.8769	72.5 / 13.4	82.4	0.9244	82.1 / 11.8	86.4
12	0.6410	49.7 / 24.0	68.2	0.8766	80.7 / 21.6	79.1	0.9274	87.5 / 18.3	83.4
13	0.6489	44.4 / 18.1	70.8	0.8766	80.4 / 20.9	79.5	0.9300	85.5 / 14.5	85.5

C THEORY: ASSUMPTIONS AND GUARANTEES

This appendix establishes the theoretical properties underlying Equilibrium Energy Control (EqC). Section C.1 introduces the bridge construction, the energy-based control field, and the gradient-matching objective. Section C.2 relates local gradient matching to finite-path energy consistency. Section C.3 characterizes when the time-dependent bridge target can be represented by a stationary energy field. Section C.4 establishes monotone energy ascent during inference. Section C.5 shows that the deployment-time consistency score measures integrated field mismatch along the realized refinement trajectory.

C.1 PROBLEM SETUP

Let $\mathcal{Z}$ denote the condition space and let

$$(A, Z) \sim p_{\mathrm{data}}$$

denote an action–condition pair, where $A \in \mathbb{R}^D$ is an expert action chunk and $Z \in \mathcal{Z}$ contains the corresponding observations, language instruction, robot state, or other policy inputs. Let

$$\varepsilon \sim \mathcal{N}(0, I_D)$$

be independent of (A, Z). For a realization (a, z, ε), define

$$d := a - \varepsilon, \qquad x_t := \varepsilon + td, \qquad t \in [0, 1]. \tag{13}$$

Thus, $x_0 = \varepsilon$ and $x_1 = a$.

Let

$$c : [0, 1] \to \mathbb{R}_{\geq 0}$$

be a continuous schedule satisfying $c(1) = 0$, and define the remaining schedule integral

$$C(t) := \int_t^1 c(s)\, ds. \tag{14}$$

EqC represents action quality by a differentiable scalar energy

$$E_\theta : \mathbb{R}^D \times \mathcal{Z} \to \mathbb{R}.$$

The induced stationary control field is

$$g_\theta(x, z) := \nabla_x E_\theta(x, z). \tag{15}$$

The field is stationary in the sense that it depends on the current action state x and condition z, but not explicitly on the bridge coordinate t.

At the bridge point x_t, the target field is

$$c(t)d.$$

The gradient-matching objective is

$$\mathcal{L}_{\mathrm{grad}}(\theta) := \mathbb{E}_{(A,Z),\varepsilon,T}\left[\frac{1}{D}\left\|\nabla_x E_\theta(X_T, Z) - c(T)(A - \varepsilon)\right\|_2^2\right], \tag{16}$$

where

$$T \sim \mathrm{Uniform}[0, 1], \qquad X_T := \varepsilon + T(A - \varepsilon).$$

The corresponding finite-path target follows from integrating the target field along the bridge:

$$\int_t^1 (c(s)d)^\top d\, ds = \|d\|_2^2 C(t).$$

This relation is induced by the gradient target and will provide the reference for the deployment-time consistency score.

At inference time, EqC initializes

$$x^{(0)} \sim \mathcal{N}(0, I_D)$$

and applies gradient ascent:

$$x^{(k+1)} = x^{(k)} + \eta_k \nabla_x E_\theta(x^{(k)}, z), \qquad k = 0, \ldots, K - 1. \tag{17}$$

The final action prediction is

$$\hat{x} := x^{(K)}.$$

C.2 From Local Gradient Matching to Path Consistency

Fix a bridge $(x_s)_{s\in[0,1]}$ defined by equation 13. Define the local gradient-matching error

$$e_\theta(s) := \nabla_x E_\theta(x_s, z) - c(s)d. \tag{18}$$

The following result relates finite energy inconsistency to local gradient-matching error accumulated along the bridge.

Lemma 1 (Path-consistency identity)**.** *Suppose that $E_\theta(\cdot, z)$ is continuously differentiable on the bridge*

$$\{x_s : s \in [t, 1]\}.$$

Then, for every $t \in [0, 1]$,

$$E_\theta(a, z) - E_\theta(x_t, z) - \|d\|_2^2 C(t) = \int_t^1 e_\theta(s)^\top d\, ds. \tag{19}$$

Proof. Since

$$x_s = \varepsilon + sd,$$

we have

$$\frac{dx_s}{ds} = d.$$

Applying the chain rule,

$$\frac{d}{ds} E_\theta(x_s, z) = \nabla_x E_\theta(x_s, z)^\top d. \tag{20}$$

By equation 18,

$$\nabla_x E_\theta(x_s, z) = c(s)d + e_\theta(s).$$

Substituting this identity into equation 20 gives

$$\frac{d}{ds} E_\theta(x_s, z) = c(s)\|d\|_2^2 + e_\theta(s)^\top d.$$

Integrating from t to 1 yields

$$\begin{aligned} E_\theta(x_1, z) - E_\theta(x_t, z) &= \int_t^1 c(s)\|d\|_2^2\, ds + \int_t^1 e_\theta(s)^\top d\, ds \\ &= \|d\|_2^2 C(t) + \int_t^1 e_\theta(s)^\top d\, ds. \end{aligned}$$

Since $x_1 = a$, rearranging proves equation 19. □

Corollary 1 (Accumulated path-consistency error)**.** *Under the assumptions of Lemma 1,*

$$\left|E_\theta(a, z) - E_\theta(x_t, z) - \|d\|_2^2 C(t)\right| \le \|d\|_2 \sqrt{1-t} \left(\int_t^1 \|e_\theta(s)\|_2^2\, ds\right)^{1/2}. \tag{21}$$

Proof. By Lemma 1,

$$\left|E_\theta(a, z) - E_\theta(x_t, z) - \|d\|_2^2 C(t)\right| = \left|\int_t^1 e_\theta(s)^\top d\, ds\right|.$$

Cauchy–Schwarz first gives

$$|e_\theta(s)^\top d| \le \|e_\theta(s)\|_2 \|d\|_2.$$

Therefore,

$$\begin{aligned} \left|\int_t^1 e_\theta(s)^\top d\, ds\right| &\le \|d\|_2 \int_t^1 \|e_\theta(s)\|_2\, ds \\ &\le \|d\|_2 \left(\int_t^1 1\, ds\right)^{1/2} \left(\int_t^1 \|e_\theta(s)\|_2^2\, ds\right)^{1/2} \\ &= \|d\|_2 \sqrt{1-t} \left(\int_t^1 \|e_\theta(s)\|_2^2\, ds\right)^{1/2}. \end{aligned}$$

□

Lemma 1 shows that the consistency residual is the integrated component of the local field error in the bridge direction. Consequently, reducing the gradient-matching error also controls the finite energy relation underlying the deployment-time consistency test.

C.3 Stationary Representation of the Bridge Target

The bridge target $c(T)(A-\varepsilon)$ depends explicitly on T, whereas the deployed EqC field $\nabla_x E_\theta(X, Z)$ is stationary. We now characterize the approximation cost associated with this restriction.

Let

$$X := \varepsilon + T(A-\varepsilon), \qquad Y := c(T)(A-\varepsilon).$$

For $T < 1$, define

$$\alpha(T) := \frac{c(T)}{1-T}. \tag{22}$$

The value of $\alpha(1)$ may be assigned arbitrarily because $\mathbb{P}(T=1) = 0$.

Since

$$A - X = A - \varepsilon - T(A-\varepsilon) = (1-T)(A-\varepsilon),$$

the target satisfies

$$Y = \alpha(T)(A-X) \qquad \text{almost surely.} \tag{23}$$

Assume

$$\mathbb{E}\|A\|_2^2 < \infty,$$

and define the conditional mean action

$$a_\star(Z) := \mathbb{E}[A \mid Z]. \tag{24}$$

Define the radius of the current state relative to this conditional reference by

$$R_X := \|X - a_\star(Z)\|_2. \tag{25}$$

Definition 1 (Condition-dependent radial scale)**.** *A condition-dependent radial scale is a measurable function*

$$\lambda : \mathcal{Z} \times [0,\infty) \to [0,\infty),$$

written

$$\lambda_z(r) := \lambda(z, r),$$

such that $r \mapsto \lambda_z(r)$ is continuous for p_Z-almost every z.

The random variable $\lambda_Z(R_X)$ represents a stationary approximation of the bridge coefficient $\alpha(T)$ using only the current state X and condition Z.

For a radial scale λ, define the per-dimension action-dispersion and scale-recovery errors by

$$\sigma_{\text{act}}^2 := \frac{1}{D}\mathbb{E}\left[\alpha(T)^2\|A - a_\star(Z)\|_2^2\right], \tag{26}$$

$$\sigma_{\text{scale}}^2(\lambda) := \frac{1}{D}\mathbb{E}\left[\big(\alpha(T) - \lambda_Z(R_X)\big)^2 R_X^2\right]. \tag{27}$$

The first quantity measures the residual variation of the expert action after conditioning on Z. Since T is independent of (A, Z),

$$D\sigma_{\text{act}}^2 = \mathbb{E}[\alpha(T)^2]\,\mathbb{E}\|A - a_\star(Z)\|_2^2 \tag{28}$$

whenever both factors are finite. The second quantity measures the error incurred when $\alpha(T)$ is represented as a function of (R_X, Z).

Lemma 2 (Optimal measurable radial scale)**.** *Among all measurable functions of (R_X, Z), the weighted error*

$$\mathbb{E}\left[\big(\alpha(T) - \lambda_Z(R_X)\big)^2 R_X^2\right]$$

is minimized, for $R_X > 0$, by any version of

$$\lambda_z^\star(r) := \mathbb{E}\left[\alpha(T) \mid R_X = r, Z = z\right]. \tag{29}$$

At $r = 0$, the value of $\lambda_z^\star(0)$ does not affect the weighted objective.

Proof. Condition on $(R_X, Z) = (r, z)$. For any scalar ℓ,

$$\mathbb{E}\left[\big(\alpha(T) - \ell\big)^2 R_X^2 \mid R_X = r, Z = z\right] = r^2 \mathbb{E}\left[\big(\alpha(T) - \ell\big)^2 \mid R_X = r, Z = z\right].$$

For $r > 0$, multiplication by r^2 does not change the minimizer. The conditional squared error is minimized by the conditional mean

$$\ell = \mathbb{E}\left[\alpha(T) \mid R_X = r, Z = z\right].$$

For $r = 0$, the weighted conditional objective is identically zero for every ℓ. □

If the conditional expectation in equation 29 has a continuous version, it satisfies Definition 1. Otherwise, any continuous approximation may be used in the bound below.

Given a radial scale λ, define

$$E_\lambda(x, z) := -\int_0^{\|x - a_\star(z)\|_2} s\lambda_z(s)\, ds. \tag{30}$$

Lemma 3 (Gradient of the radial witness)**.** *If $r \mapsto \lambda_z(r)$ is continuous, then $E_\lambda(\cdot, z)$ is continuously differentiable and*

$$\nabla_x E_\lambda(x, z) = \lambda_z(\|x - a_\star(z)\|_2)\big(a_\star(z) - x\big). \tag{31}$$

Proof. Fix z and write

$$r(x) := \|x - a_\star(z)\|_2.$$

For $r(x) > 0$, the chain rule gives

$$\begin{aligned}\nabla_x E_\lambda(x, z) &= -r(x)\lambda_z(r(x))\nabla_x r(x) \\ &= -r(x)\lambda_z(r(x))\frac{x - a_\star(z)}{r(x)} \\ &= \lambda_z(r(x))(a_\star(z) - x).\end{aligned}$$

It remains to consider $x = a_\star(z)$. By continuity, λ_z is bounded on a neighborhood of zero. Hence, for sufficiently small h, there exists $M < \infty$ such that

$$\begin{aligned}|E_\lambda(a_\star(z) + h, z) - E_\lambda(a_\star(z), z)| &= \left|\int_0^{\|h\|_2} s\lambda_z(s)\, ds\right| \\ &\le \frac{M}{2}\|h\|_2^2 \\ &= o(\|h\|_2).\end{aligned}$$

Therefore,

$$\nabla_x E_\lambda(a_\star(z), z) = 0,$$

which agrees with equation 31. Continuity of the gradient follows from continuity of λ_z. □

The field

$$g_\lambda(x, z) := \nabla_x E_\lambda(x, z)$$

is a stationary conservative field. We use it as an explicit witness for the approximation capacity of the EqC energy class.

For a measurable predictor

$$h : \mathbb{R}^D \times [0, 1] \times \mathcal{Z} \to \mathbb{R}^D,$$

define its normalized squared risk by

$$R(h) := \frac{1}{D}\mathbb{E}\left[\|Y - h(X, T, Z)\|_2^2\right]. \tag{32}$$

The optimal time-conditioned regressor is

$$m_{\mathrm{tc}}(X,T,Z) := \mathbb{E}[Y \mid X,T,Z], \tag{33}$$

with risk

$$R^{\star}_{\mathrm{tc}} := R(m_{\mathrm{tc}}). \tag{34}$$

A stationary predictor depends only on (X,Z). Define the optimal stationary risk by

$$R^{\star}_{\mathrm{st}} := \inf_{h:\, h=h(X,Z)} \frac{1}{D}\mathbb{E}\left[\|Y-h(X,Z)\|_2^2\right]. \tag{35}$$

For the EqC energy class

$$\mathcal{E}_{\Theta} := \{E_\theta : \theta \in \Theta\},$$

define

$$R^{\Theta}_{\mathrm{EqC}} := \inf_{\theta\in\Theta} \frac{1}{D}\mathbb{E}\left[\|Y-\nabla_x E_\theta(X,Z)\|_2^2\right]. \tag{36}$$

Finally, define the approximation error of the EqC class relative to the radial witness:

$$\delta_{\Theta}(\lambda) := \inf_{\theta\in\Theta}\left(\frac{1}{D}\mathbb{E}\left[\|\nabla_x E_\theta(X,Z) - g_\lambda(X,Z)\|_2^2\right]\right)^{1/2}. \tag{37}$$

Proposition 1 (Stationary energy approximation)**.** *Suppose that*

$$\sigma_{\mathrm{act}} < \infty, \qquad \sigma_{\mathrm{scale}}(\lambda) < \infty, \qquad \delta_{\Theta}(\lambda) < \infty.$$

Then

$$0 \le R^{\Theta}_{\mathrm{EqC}} - R^{\star}_{\mathrm{st}} \le R^{\Theta}_{\mathrm{EqC}} - R^{\star}_{\mathrm{tc}} \le \left(\sigma_{\mathrm{act}} + \sigma_{\mathrm{scale}}(\lambda) + \delta_{\Theta}(\lambda)\right)^2. \tag{38}$$

Proof. For a square-integrable random vector W, define

$$\|W\|_{L^2_D} := \left(\frac{1}{D}\mathbb{E}\|W\|_2^2\right)^{1/2}.$$

The orthogonality principle for conditional expectations gives, for every measurable $h(X,T,Z)$,

$$R(h) - R^{\star}_{\mathrm{tc}} = \|h - m_{\mathrm{tc}}\|^2_{L^2_D}. \tag{39}$$

Indeed,

$$\begin{aligned}\|Y-h\|^2_{L^2_D} &= \|Y - m_{\mathrm{tc}} + m_{\mathrm{tc}} - h\|^2_{L^2_D}\\ &= \|Y-m_{\mathrm{tc}}\|^2_{L^2_D} + \|m_{\mathrm{tc}}-h\|^2_{L^2_D},\end{aligned}$$

because

$$\mathbb{E}\left[(Y-m_{\mathrm{tc}})^{\top}(m_{\mathrm{tc}}-h)\right] = 0.$$

Since $g_\lambda(X,Z)$ is measurable with respect to (X,T,Z),

$$\begin{aligned}\|m_{\mathrm{tc}} - g_\lambda\|_{L^2_D} &= \|\mathbb{E}\left[Y - g_\lambda(X,Z) \mid X,T,Z\right]\|_{L^2_D}\\ &\le \|Y - g_\lambda(X,Z)\|_{L^2_D},\end{aligned} \tag{40}$$

where the inequality follows from conditional Jensen.

Using equation 23 and equation 31,

$$\begin{aligned}Y - g_\lambda(X,Z) &= \alpha(T)(A-X) - \lambda_Z(R_X)(a_\star(Z)-X)\\ &= \alpha(T)(A-a_\star(Z)) + \big(\alpha(T)-\lambda_Z(R_X)\big)(a_\star(Z)-X).\end{aligned} \tag{41}$$

Applying Minkowski's inequality,

$$\begin{aligned}\|Y-g_\lambda(X,Z)\|_{L^2_D} &\le \|\alpha(T)(A-a_\star(Z))\|_{L^2_D}\\ &\quad + \big\|\big(\alpha(T)-\lambda_Z(R_X)\big)(a_\star(Z)-X)\big\|_{L^2_D}\\ &= \sigma_{\mathrm{act}} + \sigma_{\mathrm{scale}}(\lambda),\end{aligned}$$

where

$$\|a_\star(Z) - X\|_2 = R_X.$$

Together with equation 40, this gives

$$\|m_{\mathrm{tc}} - g_\lambda\|_{L^2_D} \le \sigma_{\mathrm{act}} + \sigma_{\mathrm{scale}}(\lambda). \tag{42}$$

For any $\theta \in \Theta$, Minkowski's inequality and equation 42 imply

$$\begin{aligned}\|m_{\mathrm{tc}} - \nabla_x E_\theta\|_{L^2_D} &\le \|m_{\mathrm{tc}} - g_\lambda\|_{L^2_D} + \|g_\lambda - \nabla_x E_\theta\|_{L^2_D} \\ &\le \sigma_{\mathrm{act}} + \sigma_{\mathrm{scale}}(\lambda) + \|g_\lambda - \nabla_x E_\theta\|_{L^2_D}.\end{aligned}$$

Taking the infimum over θ gives

$$\inf_{\theta\in\Theta} \|m_{\mathrm{tc}} - \nabla_x E_\theta\|_{L^2_D} \le \sigma_{\mathrm{act}} + \sigma_{\mathrm{scale}}(\lambda) + \delta_\Theta(\lambda).$$

By equation 39,

$$R^\Theta_{\mathrm{EqC}} - R^\star_{\mathrm{tc}} = \inf_{\theta\in\Theta} \|m_{\mathrm{tc}} - \nabla_x E_\theta\|^2_{L^2_D}.$$

Therefore,

$$R^\Theta_{\mathrm{EqC}} - R^\star_{\mathrm{tc}} \le \left(\sigma_{\mathrm{act}} + \sigma_{\mathrm{scale}}(\lambda) + \delta_\Theta(\lambda)\right)^2.$$

Every EqC gradient $\nabla_x E_\theta(X, Z)$ is a stationary predictor. Hence,

$$R^\star_{\mathrm{st}} \le R^\Theta_{\mathrm{EqC}}.$$

Moreover, every stationary predictor is contained in the class of time-conditioned predictors, so

$$R^\star_{\mathrm{tc}} \le R^\star_{\mathrm{st}}.$$

Combining these inequalities completes the proof. □

Proposition 1 separates the stationary approximation error into three terms: conditional action dispersion, recovery of the bridge scale from the current state, and approximation by the chosen neural energy class.

Corollary 2 (Exact stationary representation)**.** *Suppose that the following conditions hold:*

$$A = a_\star(Z) \qquad \textit{almost surely}, \tag{43}$$

$$\mathrm{Var}(\alpha(T) \mid R_X, Z) = 0 \qquad \textit{almost surely}. \tag{44}$$

Let

$$\lambda_z(r) := \mathbb{E}\left[\alpha(T) \mid R_X = r, Z = z\right]$$

be a continuous version of the conditional expectation. If there exists $\theta_\star \in \Theta$ such that

$$\nabla_x E_{\theta_\star}(x, z) = \lambda_z(\|x - a_\star(z)\|_2)\big(a_\star(z) - x\big) \tag{45}$$

on the support of (X, Z), then

$$\nabla_x E_{\theta_\star}(X, Z) = Y \qquad \textit{almost surely}. \tag{46}$$

Consequently,

$$R^\Theta_{\mathrm{EqC}} = R^\star_{\mathrm{st}} = R^\star_{\mathrm{tc}} = 0. \tag{47}$$

Proof. Equation equation 44 implies that $\alpha(T)$ is measurable with respect to (R_X, Z). Therefore,

$$\alpha(T) = \mathbb{E}\left[\alpha(T) \mid R_X, Z\right] = \lambda_Z(R_X) \qquad \text{almost surely.}$$

Using equation 23 and equation 43,

$$\begin{aligned}Y &= \alpha(T)(A - X) \\ &= \lambda_Z(R_X)(a_\star(Z) - X).\end{aligned}$$

By equation 45,

$$\lambda_Z(R_X)(a_\star(Z) - X) = \nabla_x E_{\theta_\star}(X, Z).$$

Hence,

$$Y = \nabla_x E_{\theta_\star}(X, Z)$$

almost surely. The corresponding squared risk is zero, which proves equation 47. □

In the concentrated case $A = a_\star(Z)$,

$$R_X = (1 - T)\|\varepsilon - a_\star(Z)\|_2. \tag{48}$$

Thus, whenever (R_X, Z) determines $\alpha(T)$, a stationary energy can represent a nonconstant bridge schedule without receiving T explicitly.

C.4 MONOTONE ENERGY ASCENT

We next establish the optimization property of the inference update equation 17.

Assumption 1 (Local smoothness)**.** *Fix z. There exists $L > 0$ such that, for every $k \in \{0, \dots, K-1\}$,*

$$\|\nabla_x E_\theta(x, z) - \nabla_x E_\theta(y, z)\|_2 \le L\|x - y\|_2 \tag{49}$$

for all x and y on the segment

$$[x^{(k)}, x^{(k+1)}] := \left\{ x^{(k)} + s(x^{(k+1)} - x^{(k)}) : s \in [0, 1] \right\}.$$

Theorem 1 (Monotone energy ascent)**.** *Suppose Assumption 1 holds and EqC uses the exact gradient-ascent update*

$$x^{(k+1)} = x^{(k)} + \eta_k \nabla_x E_\theta(x^{(k)}, z)$$

with

$$0 < \eta_k < \frac{2}{L}. \tag{50}$$

Then

$$E_\theta(x^{(k+1)}, z) - E_\theta(x^{(k)}, z) \ge w_k \left\| \nabla_x E_\theta(x^{(k)}, z) \right\|_2^2, \tag{51}$$

where

$$w_k := \eta_k \left(1 - \frac{L\eta_k}{2} \right) > 0. \tag{52}$$

Therefore,

$$E_\theta(x^{(k+1)}, z) \ge E_\theta(x^{(k)}, z),$$

with strict inequality whenever

$$\nabla_x E_\theta(x^{(k)}, z) \neq 0.$$

Proof. Let

$$g_k := \nabla_x E_\theta(x^{(k)}, z).$$

The update is

$$x^{(k+1)} = x^{(k)} + \eta_k g_k.$$

Define the segment

$$q_k(s) := x^{(k)} + s\eta_k g_k, \qquad s \in [0, 1].$$

By the fundamental theorem of calculus,

$$\begin{aligned} E_\theta(x^{(k+1)}, z) - E_\theta(x^{(k)}, z) &= \int_0^1 \nabla_x E_\theta(q_k(s), z)^\top (\eta_k g_k)\, ds \\ &= \eta_k \|g_k\|_2^2 + \eta_k \int_0^1 \left(\nabla_x E_\theta(q_k(s), z) - g_k \right)^\top g_k \, ds. \end{aligned} \tag{53}$$

By Cauchy–Schwarz and Assumption 1,

$$\begin{aligned} \left(\nabla_x E_\theta(q_k(s), z) - g_k \right)^\top g_k &\ge - \|\nabla_x E_\theta(q_k(s), z) - g_k\|_2 \, \|g_k\|_2 \\ &\ge -L\|q_k(s) - x^{(k)}\|_2 \|g_k\|_2 \\ &= -Ls\eta_k \|g_k\|_2^2. \end{aligned}$$

Substituting this bound into equation 53,

$$\begin{aligned} E_\theta(x^{(k+1)}, z) - E_\theta(x^{(k)}, z) &\ge \eta_k \|g_k\|_2^2 - L\eta_k^2 \|g_k\|_2^2 \int_0^1 s \, ds \\ &= \eta_k \left(1 - \frac{L\eta_k}{2} \right) \|g_k\|_2^2 \\ &= w_k \|g_k\|_2^2. \end{aligned}$$

The step-size condition equation 50 implies $w_k > 0$, proving the result. □

Corollary 3 (Finite-step stationarity)**.** *Under the assumptions of Theorem 1,*

$$\min_{0\le k<K} \left\|\nabla_x E_\theta(x^{(k)}, z)\right\|_2^2 \le \frac{E_\theta(x^{(K)}, z) - E_\theta(x^{(0)}, z)}{\sum_{k=0}^{K-1} w_k}. \tag{54}$$

If

$$E_\theta(x^{(k)}, z) \le E_{\sup} \qquad \textit{for } k = 0, \dots, K,$$

then

$$\min_{0\le k<K} \left\|\nabla_x E_\theta(x^{(k)}, z)\right\|_2^2 \le \frac{E_{\sup} - E_\theta(x^{(0)}, z)}{\sum_{k=0}^{K-1} w_k}. \tag{55}$$

Proof. Summing equation 51 over $k = 0, \dots, K-1$ gives

$$\begin{aligned} E_\theta(x^{(K)}, z) - E_\theta(x^{(0)}, z) &= \sum_{k=0}^{K-1} \Big(E_\theta(x^{(k+1)}, z) - E_\theta(x^{(k)}, z) \Big) \\ &\ge \sum_{k=0}^{K-1} w_k \left\|\nabla_x E_\theta(x^{(k)}, z)\right\|_2^2 \\ &\ge \left(\min_{0\le k<K} \left\|\nabla_x E_\theta(x^{(k)}, z)\right\|_2^2 \right) \sum_{k=0}^{K-1} w_k. \end{aligned}$$

Since each $w_k > 0$, division by $\sum_{k=0}^{K-1} w_k$ proves equation 54. The bound equation 55 follows from

$$E_\theta(x^{(K)}, z) \le E_{\sup}.$$

□

Theorem 1 establishes that every EqC refinement step increases the same scalar energy under the stated smoothness and step-size conditions. Corollary 3 quantifies the resulting approach to a first-order stationary point.

C.5 Energy Consistency as a Path-Wise Mismatch Certificate

Let

$$\hat{x} := x^{(K)}, \qquad v := \hat{x} - x^{(0)}.$$

For a fixed normalization constant $\delta > 0$, define

$$D_v := \max(\|v\|_2^2, \delta), \qquad \gamma_v := \frac{\|v\|_2^2}{D_v}. \tag{56}$$

For $u \in \mathbb{R}$, define

$$\mathrm{clip}(u, 0, 1) := \min\{1, \max\{0, u\}\}.$$

The projected progress coordinate of iterate $x^{(k)}$ is

$$\tau_k := \mathrm{clip}\left(\frac{\langle x^{(k)} - x^{(0)}, v\rangle}{D_v}, 0, 1 \right). \tag{57}$$

The corresponding point on the chord from $x^{(0)}$ to $\hat{x}$ is

$$\bar{x}^{(k)} := x^{(0)} + \tau_k v, \tag{58}$$

and the off-chord displacement is

$$h_k := x^{(k)} - \bar{x}^{(k)}. \tag{59}$$

Define the field mismatch along the endpoint chord by

$$e_v(s) := \nabla_x E_\theta(x^{(0)} + sv, z) - c(s)v, \qquad s \in [0, 1]. \tag{60}$$

The deployment-time energy residual is

$$r_k^{\Delta E} := \frac{E_\theta(\hat{x}, z) - E_\theta(x^{(k)}, z)}{D_v} - C(\tau_k), \tag{61}$$

and the energy-consistency score is

$$U_{\Delta E} := \frac{1}{K} \sum_{k=1}^{K} |r_k^{\Delta E}|. \tag{62}$$

Theorem 2 (Energy-consistency decomposition)**.** *Suppose that $E_\theta(\cdot, z)$ is continuously differentiable on the chord*

$$\{x^{(0)} + sv : s \in [0, 1]\}.$$

Then, for each $k \in \{1, \dots, K\}$,

$$r_k^{\Delta E} = \frac{1}{D_v} \int_{\tau_k}^{1} e_v(s)^\top v \, ds + \frac{E_\theta(\bar{x}^{(k)}, z) - E_\theta(x^{(k)}, z)}{D_v} + (\gamma_v - 1) C(\tau_k). \tag{63}$$

Proof. Define the chord

$$q(s) := x^{(0)} + sv.$$

Since

$$q(\tau_k) = \bar{x}^{(k)}, \qquad q(1) = \hat{x}, \qquad \frac{dq(s)}{ds} = v,$$

the fundamental theorem of calculus gives

$$\begin{aligned} E_\theta(\hat{x}, z) - E_\theta(\bar{x}^{(k)}, z) &= \int_{\tau_k}^{1} \nabla_x E_\theta(q(s), z)^\top v \, ds \\ &= \int_{\tau_k}^{1} \big(c(s)v + e_v(s)\big)^\top v \, ds \\ &= \|v\|_2^2 \int_{\tau_k}^{1} c(s) \, ds + \int_{\tau_k}^{1} e_v(s)^\top v \, ds \\ &= \|v\|_2^2 C(\tau_k) + \int_{\tau_k}^{1} e_v(s)^\top v \, ds. \end{aligned}$$

Adding

$$E_\theta(\bar{x}^{(k)}, z) - E_\theta(x^{(k)}, z)$$

to both sides gives

$$\begin{aligned} E_\theta(\hat{x}, z) - E_\theta(x^{(k)}, z) = \|v\|_2^2 C(\tau_k) + \int_{\tau_k}^{1} e_v(s)^\top v \, ds \\ + E_\theta(\bar{x}^{(k)}, z) - E_\theta(x^{(k)}, z). \end{aligned}$$

Dividing by D_v and subtracting $C(\tau_k)$ yields

$$\begin{aligned} r_k^{\Delta E} = \frac{1}{D_v} \int_{\tau_k}^{1} e_v(s)^\top v \, ds \\ + \frac{E_\theta(\bar{x}^{(k)}, z) - E_\theta(x^{(k)}, z)}{D_v} \\ + \left(\frac{\|v\|_2^2}{D_v} - 1 \right) C(\tau_k). \end{aligned}$$

Using

$$\gamma_v = \frac{\|v\|_2^2}{D_v}$$

proves equation 63. □

The three terms in equation 63 have distinct origins. The first is the integrated field mismatch along the endpoint chord. The second measures the energy discrepancy caused by deviation of the realized iterate from that chord. The third is the correction introduced by the normalization floor.

Define the mean integrated chord mismatch

$$A_K := \frac{1}{KD_v} \sum_{k=1}^{K} \left| \int_{\tau_k}^{1} e_v(s)^\top v \, ds \right|. \tag{64}$$

Assumption 2 (Bounded field on off-chord segments)**.** *There exists $G < \infty$ such that*

$$\|\nabla_x E_\theta(y, z)\|_2 \le G$$

for every y on every segment

$$[x^{(k)}, \bar{x}^{(k)}], \qquad k = 1, \dots, K.$$

Under Assumption 2, define

$$B_K := \frac{G}{KD_v} \sum_{k=1}^{K} \|h_k\|_2 + \frac{|\gamma_v - 1|}{K} \sum_{k=1}^{K} |C(\tau_k)|. \tag{65}$$

Proposition 2 (Bounds on the energy-consistency score)**.** *Under Assumption 2,*

$$[A_K - B_K]_+ \le U_{\Delta E} \le A_K + B_K, \tag{66}$$

where

$$[q]_+ := \max(q, 0).$$

Moreover,

$$A_K \le \frac{\|v\|_2}{KD_v} \int_0^1 N_K(s) \|e_v(s)\|_2 \, ds, \tag{67}$$

where

$$N_K(s) := \sum_{k=1}^{K} \mathbf{1}\{\tau_k \le s\}. \tag{68}$$

Proof. For each k, write

$$\begin{aligned} a_k &:= \frac{1}{D_v} \int_{\tau_k}^{1} e_v(s)^\top v \, ds, \\ b_k &:= \frac{E_\theta(\bar{x}^{(k)}, z) - E_\theta(x^{(k)}, z)}{D_v}, \\ q_k &:= (\gamma_v - 1) C(\tau_k). \end{aligned}$$

By Theorem 2,

$$r_k^{\Delta E} = a_k + b_k + q_k.$$

The triangle and reverse-triangle inequalities give

$$|a_k| - |b_k| - |q_k| \le |r_k^{\Delta E}| \le |a_k| + |b_k| + |q_k|. \tag{69}$$

By the fundamental theorem of calculus on the segment from $x^{(k)}$ to $\bar{x}^{(k)}$,

$$\begin{aligned} &\left| E_\theta(\bar{x}^{(k)}, z) - E_\theta(x^{(k)}, z) \right| \\ &\quad \le \sup_{y \in [x^{(k)}, \bar{x}^{(k)}]} \|\nabla_x E_\theta(y, z)\|_2 \, \|\bar{x}^{(k)} - x^{(k)}\|_2 \\ &\quad \le G \|h_k\|_2. \end{aligned}$$

Therefore,

$$|b_k| \le \frac{G}{D_v} \|h_k\|_2, \qquad |q_k| = |\gamma_v - 1| \, |C(\tau_k)|.$$

Averaging equation 69 over $k = 1, \dots, K$ gives

$$A_K - B_K \le U_{\Delta E} \le A_K + B_K.$$

Since $U_{\Delta E} \ge 0$, the lower bound can be written as $[A_K - B_K]_+$.

For the second result, Cauchy–Schwarz implies

$$|e_v(s)^\top v| \le \|e_v(s)\|_2 \|v\|_2.$$

Consequently,

$$\begin{aligned} A_K &\le \frac{\|v\|_2}{KD_v} \sum_{k=1}^{K} \int_{\tau_k}^{1} \|e_v(s)\|_2 \, ds \\ &= \frac{\|v\|_2}{KD_v} \int_0^1 \left(\sum_{k=1}^{K} \mathbf{1}\{\tau_k \le s\} \right) \|e_v(s)\|_2 \, ds \\ &= \frac{\|v\|_2}{KD_v} \int_0^1 N_K(s) \|e_v(s)\|_2 \, ds. \end{aligned}$$

□

Proposition 2 shows that the observed score $U_{\Delta E}$ is controlled by the integrated chord-wise field error, up to explicit corrections for off-chord motion and normalization.

Corollary 4 (Exact on-chord characterization)**.** *Suppose that the normalization floor is inactive,*

$$D_v = \|v\|_2^2, \tag{70}$$

and every refinement iterate lies on the endpoint chord:

$$x^{(k)} = \bar{x}^{(k)}, \qquad k = 1, \dots, K. \tag{71}$$

Then

$$U_{\Delta E} = \frac{1}{K\|v\|_2^2} \sum_{k=1}^{K} \left| \int_{\tau_k}^{1} e_v(s)^\top v \, ds \right|. \tag{72}$$

Furthermore,

$$\frac{1}{K\|v\|_2^2} \left| \int_0^1 N_K(s) e_v(s)^\top v \, ds \right| \le U_{\Delta E} \le \frac{1}{K\|v\|_2} \int_0^1 N_K(s) \|e_v(s)\|_2 \, ds. \tag{73}$$

Proof. If $D_v = \|v\|_2^2$, then

$$\gamma_v = 1.$$

If $x^{(k)} = \bar{x}^{(k)}$, then

$$E_\theta(\bar{x}^{(k)}, z) - E_\theta(x^{(k)}, z) = 0.$$

The decomposition in equation 63 therefore reduces to

$$r_k^{\Delta E} = \frac{1}{\|v\|_2^2} \int_{\tau_k}^{1} e_v(s)^\top v \, ds.$$

Taking absolute values and averaging proves equation 72.

For the lower bound, observe that

$$\begin{aligned} \int_0^1 N_K(s) e_v(s)^\top v \, ds &= \int_0^1 \sum_{k=1}^{K} \mathbf{1}\{\tau_k \le s\} e_v(s)^\top v \, ds \\ &= \sum_{k=1}^{K} \int_{\tau_k}^{1} e_v(s)^\top v \, ds. \end{aligned}$$

Hence, by the triangle inequality,

$$\left|\int_0^1 N_K(s) e_v(s)^\top v \, ds\right| \le \sum_{k=1}^{K} \left|\int_{\tau_k}^1 e_v(s)^\top v \, ds\right|.$$

Dividing by $K\|v\|_2^2$ proves the lower bound.

For the upper bound,

$$\begin{aligned} U_{\Delta E} &= \frac{1}{K\|v\|_2^2} \sum_{k=1}^{K} \left|\int_{\tau_k}^1 e_v(s)^\top v \, ds\right| \\ &\le \frac{1}{K\|v\|_2} \sum_{k=1}^{K} \int_{\tau_k}^1 \|e_v(s)\|_2 \, ds \\ &= \frac{1}{K\|v\|_2} \int_0^1 N_K(s) \|e_v(s)\|_2 \, ds. \end{aligned}$$

□

Corollary 4 provides the exact interpretation of the EqC score in the ideal geometric regime: $U_{\Delta E}$ is the average normalized magnitude of the remaining integrated longitudinal field error along the refinement path.

C.6 Scope of the auxiliary analysis

- The empirical findings of Pan et al. (2026) motivate the conditional-concentration regime; they do not prove that σ_{act} or σ_{scale} is small for EqC. These quantities require direct diagnostics on the selected datasets and trained policies.
- Proposition 1 is an existence upper bound based on a single-basin witness. Low conditional dispersion is sufficient, not necessary: a nonlinear energy may represent several action basins. Its weighted moment conditions also cover schedules for which $c(T)/(1-T)$ grows near the endpoint.
- Theorem 1 concerns optimization of the learned scalar only. Neither stationarity nor increasing energy is a correctness guarantee, and a generic stationary field can have a separately established Lyapunov function.
- Theorem 2 decomposes the mean absolute self-consistency residual. Absolute averaging prevents cancellation across refinement iterates, but opposite field errors can still cancel within an individual remaining-work integral. The normalization floor matters for short chords, and a wrong but internally coherent trajectory can have a small score.

D Real-Robot Experimental Setup and Evaluation Protocol

D.1 Platform and Hardware Architecture

We conduct all physical experiments on a bimanual manipulation workstation comprising two lightweight, force-controlled, 6-DOF **Galaxea A1Z** robotic arms. The policies are fine-tuned from the **Galaxea G0.5** VLA model. Each A1Z manipulator weighs 5 kg and has an operating reach radius of 626.5 mm, an operating height of 722 mm, and a rated continuous payload of 3 kg (with a peak payload of 5 kg at full extension).

Both manipulators are equipped with parallel-jaw grippers, yielding 14 active control degrees of freedom (twelve arm joints and two gripper channels). Visual inputs are captured simultaneously by three synchronized RGB cameras: an overhead global camera and two wrist-mounted cameras (left and right). Policy inputs also incorporate 14-dimensional bimanual proprioception (joint angles and gripper states). The physical control frequency is set to 15 Hz, and the receding-horizon action chunk size is set to $T_a = 32$.

D.2 Task Specifications and Success Criteria

Benchmark 1: Flower Insertion. The *Flower Insertion* task (Section 5.1.4) evaluates bimanual coordination, 3D spatial alignment, and contact-rich insertion. To successfully complete the task, the dual-arm system must:

1. Approach and grasp the flexible stem from the workspace surface;
2. Lift and reorient the stem in 3D space to align its longitudinal axis with the vertical entry axis of the vase opening;
3. Smoothly feed the stem through the aperture until fully seated, dynamically correcting minor contact misalignments at the rim without tipping the vase over.

A trial is scored as a success if the stem is fully inserted and the gripper safely releases without disturbing the upright posture of the vase.

Benchmark 2: Towel Folding. The *Towel Folding* task evaluates bimanual spatial coordination, deformable object manipulation, and sequential multi-stage execution. The robot must perform a two-stage quarter-fold (folding both laterally and longitudinally):

1. **First Fold (Horizontal):** Simultaneously approach and pinch two adjacent corners/edges of the unfolded towel using both grippers, lift and fold the towel in half laterally, and lay it down smoothly;
2. **Reorientation:** Rotate the laterally folded towel using both arms to reorient it for the longitudinal fold;
3. **Second Fold (Vertical) and Release:** Reposition both arms to regrasp the newly formed folded edges, lift the towel, and fold it in half longitudinally to form a compact four-layer square; then smooth the top layer down and cleanly release both grippers without disturbing the folded creases or dragging the fabric.

A trial is scored as a success if and only if: (1) both the lateral and longitudinal folds are successfully executed, (2) the final folded fabric is reduced to approximately a quarter of its original footprint with aligned edges, and (3) the towel remains stably and neatly folded without unfolding or flipping after gripper release.

Benchmark 3: Debris Sweeping. The *Debris Sweeping* task evaluates multi-stage bimanual tool coordination and granular object collection. The dual-arm robot must coordinate a hand brush (or mini broom) and a dustpan to gather scattered paper scraps from the tabletop and dump them into a designated waste bin:

1. **Dual-Tool Grasping and Positioning:** The left manipulator picks up the brush from the holder tray, while the right manipulator grasps the dustpan and places its front edge flat against the table surface near the perimeter of the debris cluster;
2. **Contact-Rich Sweeping:** The left arm coordinates with the right arm to sweep scattered paper scraps cleanly into the cavity of the dustpan using multiple sweeping strokes, maintaining proper surface contact without crushing or scattering the debris beyond the reachable workspace;
3. **Transport and Dumping:** The right arm lifts the loaded dustpan, transfers it over the rim of the desktop waste bin, tilts it to dump the collected scraps into the container, and returns to a neutral position.

A trial is scored as a success if and only if: (1) all of the scattered paper scraps are cleanly collected and transferred into the waste bin, (2) neither the brush nor the dustpan is dropped or detached from the grippers, and (3) the waste bin remains standing upright without being tipped over during the dumping motion.

Benchmark 4: Hierarchical Block Stacking. The *Hierarchical Block Stacking* task serves as the testbed for both online competence awareness (Section 5.2.3) and closed-loop lifelong learning

(Section 5.3), evaluating multi-stage sequential execution, precise contact and balance, and tolerance to compounding errors. The task involves three color-coded blocks (green, blue, and red). In the nominal collinear configuration, the three blocks are placed collinearly on the tabletop in each of the $3! = 6$ possible color permutations (e.g., green-blue-red, blue-green-red, and red-blue-green). The green block serves as a stationary base, remaining at its initial position while the dual-arm system identifies the scene layout and manipulates the other blocks. In the disordered configuration, the blocks are initialized at arbitrary, non-collinear 2D positions with unaligned orientations. The robot must perform the following stages:

1. **Base Identification and Blue Block Placement:** The dual-arm system identifies the scene layout, grasps the blue block, and balances it on the green base;
2. **Red Block Placement:** The robot then grasps the red block and balances it on the blue block to construct a freestanding three-tier tower;
3. **Release and Stabilization:** Both grippers release the assembly and clear the scene, leaving the tower upright.

A trial is scored as a success if and only if: (1) the blue block is successfully balanced on the green base, (2) the red block is successfully balanced on the blue block, and (3) the final three-tier assembly (green $\rightarrow$ blue $\rightarrow$ red) remains upright and stable for at least 3 seconds after both grippers release and clear the scene.

Benchmark 5: Garment Folding. The *Garment Folding* task evaluates long-horizon bimanual deformable-object manipulation. Starting from an unfolded garment laid flat on the table, the dual-arm system must:

1. **Sleeve Folding:** Sequentially grasp, lift, and fold the left and right sleeves inward across the torso;
2. **Longitudinal Fold:** Reposition both grippers to pinch the bottom hem and fold the lower half of the garment upward along the vertical centerline;
3. **Lateral Fold:** Grasp the outer folded edge and fold the garment horizontally to form a compact rectangular profile;
4. **Smoothing and Alignment:** Dynamically smooth out wrinkles and creases across the top layer with downward contact wiping.

A trial is scored as a success if and only if: (1) all four folding and smoothing stages are sequentially completed, (2) the final footprint is reduced cleanly without unfolding or severe edge misalignments, and (3) both grippers safely disengage without dragging or wrinkling the finished garment.

Benchmark 6: Hierarchical Cup Stacking (Cross-Task Shift). To evaluate the policy's lifelong adaptation and catastrophic forgetting mitigation under explicit cross-task shift, we introduce the *Hierarchical Cup Stacking* task. The setup employs three color-coded tapered cups (green, blue, and orange). Identical to the collinear setup in *Hierarchical Block Stacking*, the three cups are initialized collinearly in front of the robot across all $3! = 6$ spatial permutations. The green cup serves as the fixed base. The dual-arm system must:

1. Identify the target cup layout, approach, and securely grasp the blue cup, carefully placing it upside down on top of the green base cup with its rim facing downward, without knocking the base cup over;
2. Reposition the manipulators to grasp the orange cup, align its orientation, and stack it neatly atop the blue cup to form a compact three-tier nested cup assembly.

The success criterion strictly mirrors block stacking: a trial is deemed successful if and only if both stacking stages are completed without tipping, and the resulting three-cup tower remains stable and upright for at least 3 seconds after the grippers fully release and clear the scene.

D.3 Training Configurations and Computational Overhead

All models are fine-tuned from base Galaxea G0.5 checkpoints on an 8× NVIDIA H20 GPU cluster using the AdamW optimizer with a cosine learning-rate decay schedule. Table 15 lists the hyperparameters shared across the real-world experiments.

During physical deployment, EqC performs $K = 3$ energy gradient ascent steps starting from Gaussian noise $\mathcal{N}(0, I)$ to iteratively refine the action chunk, employing a decaying step-size schedule $[\eta_0, \eta_1, \eta_2] = [0.12, 0.10, 0.05]$. This decaying schedule facilitates rapid initial progress toward the high-compatibility energy basin followed by fine-grained local stabilization, while keeping the overall inference latency within practical limits.

Table 15: Unified training and inference hyperparameters for real-robot experiments.

Hyperparameter	Value
Base VLA Backbone	Galaxea G0.5
Compute Cluster	8× NVIDIA H20 GPUs
Optimizer	AdamW
Batch Size	16
Peak Learning Rate	8×10^{-5}
Warmup Schedule	Linear warmup (500 steps)
Decay Schedule	Cosine decay (`warmup_constant_cosine`)
Action Chunk Horizon (T_a)	32
Control Frequency	15 Hz
EqC Refinement Steps (K)	3
EqC Step Sizes (η_k)	$[0.12, 0.10, 0.05]$
Training Epochs (Flower Insertion)	6 epochs (82 demonstrations)
Training Epochs (Towel Folding)	4 epochs (36 demonstrations)
Training Epochs (Debris Sweeping)	4 epochs (142 demonstrations)
Training Epochs (Lifelong Learning)	2 epochs per update session

Wall-Clock Fine-Tuning Overhead. To assess the computational cost of learning an explicit energy function, Table 16 reports wall-clock fine-tuning times for *Flower Insertion* (6 epochs, 82 demonstrations), *Towel Folding* (4 epochs, 36 demonstrations), and *Debris Sweeping* (4 epochs, 142 demonstrations). Computing explicit energy gradients during backpropagation introduces only a modest overhead, increasing training time by 25.9% on Flower Insertion, 22.1% on Towel Folding, and 17.5% on Debris Sweeping.

Table 16: Wall-clock fine-tuning time across real-robot tasks on an 8× NVIDIA H20 cluster.

Method	Time (min)	Overhead
Flower Insertion		
Galaxea G0.5	**11.58**	—
Galaxea G0.5 + **EqC (ours)**	14.58	+25.9%
Towel Folding		
Galaxea G0.5	**40.48**	—
Galaxea G0.5 + **EqC (ours)**	49.42	+22.1%
Debris Sweeping		
Galaxea G0.5	**115.30**	—
Galaxea G0.5 + **EqC (ours)**	135.47	+17.5%

D.4 ENERGY-GATED DAGGER PROTOCOL

Initial Demonstration Dataset ($\mathcal{D}_0$). For *Hierarchical Block Stacking*, the base policy E_{θ_0} is initialized by fine-tuning on a small seed demonstration set $\mathcal{D}_0$ comprising $N_0 = 18$ teleoperated episodes collected exclusively under nominal, collinear block initializations (spanning all $3! = 6$ color permutations with 3 demonstrations per configuration), yielding a 0% initial success rate on perturbed setups. For *Garment Folding*, the base policy is initialized on $N_0 = 18$ demonstrations on standard unfolded garments, yielding an initial autonomous success rate of 30%.

Human Intervention Baseline. To isolate the benefit of automated competence-aware intervention, we additionally evaluate a Human Intervention DAgger baseline under the same data and training budget as Energy-Gated DAgger. The human baseline is initialized from exactly the same seed demonstration dataset $\mathcal{D}_0$ containing $N_0 = 18$ teleoperated episodes. Intervention decisions are made by skilled teleoperators with more than three months of experience collecting VLA training data. At each round, the current checkpoint is deployed on the same nominal collinear configurations used for autonomous evaluation, spanning all $3! = 6$ possible color permutations for Block Stacking, while the operator continuously monitors whether autonomous execution is likely to fail or has moved beyond the policy's apparent competence. When the operator identifies such a failure-prone state, autonomous execution is suspended and the operator intervenes to provide a recovery trajectory. For Block Stacking, we collect exactly 6 targeted recovery episodes per round over $R = 5$ rounds, yielding a total of 30 additional recovery demonstrations and expanding the dataset from 18 to 48 episodes. For Garment Folding, we follow the same procedure over $R = 2$ rounds, collecting exactly 6 targeted recovery episodes per round, yielding 12 additional recovery demonstrations and expanding the dataset from 18 to 30 episodes. The human-in-the-loop baseline therefore follows the same task-specific schedule and collection budget as Energy-Gated DAgger: $R = 5$ rounds and 30 recovery demonstrations for Block Stacking, and $R = 2$ rounds and 12 recovery demonstrations for Garment Folding. The two methods share the same initial dataset, collection budget, deployment schedule, and policy-update procedure, and differ in how intervention decisions are triggered: external human judgment for the baseline and the policy-internal energy-consistency signal $U_{\Delta E}$ for our method.

Galaxea G0.5 + Diff-DAgger Baseline. To provide a rigorous automated baseline for lifelong learning, we adapt Diff-DAgger (Lee et al., 2025) to the physical Galaxea G0.5 policy. Its online failure-detection scoring is identical to the flow-matching formulation detailed in Appendix B.5, evaluating $M = 10$ Monte Carlo samples per replan across active embodiment dimensions. Analogous to EqC, persistent threshold exceedances over m consecutive replanning steps trigger expert recovery intervention. Crucially, G0.5 + Diff-DAgger operates under the exact same experimental conditions: it starts from the identical seed dataset, collects the exact same budget of targeted recovery demonstrations per round, and follows the identical iterative fine-tuning hyperparameters as Energy-Gated DAgger and the human baseline.

Active Deployment and Intervention Mechanism. During autonomous deployment in round r, the policy executes action prefixes under receding-horizon control while continually monitoring the energy-consistency signal $U_{\Delta E}$. If $U_{\Delta E}$ exceeds the fixed alarm threshold γ for m consecutive replanning steps, autonomous execution is suspended, and an expert intervenes to provide a recovery trajectory $\xi_j^\star$ that guides the robot to successful task completion. Each round collects $|\mathcal{Q}_r| = 6$ targeted recovery episodes.

Iterative Fine-Tuning. Following Energy-Gated DAgger, each round aggregates newly collected recovery trajectories into the historical dataset ($\mathcal{D}_{r+1} \leftarrow \mathcal{D}_r \cup \mathcal{Q}_r$). The policy is re-initialized from the same initial checkpoint and independently trained for 2 epochs on $\mathcal{D}_{r+1}$ using the hyperparameters in Table 15. For Block Stacking, across $R = 5$ rounds, the robot incorporates 30 active recovery demonstrations, expanding the dataset from $|\mathcal{D}_0| = 18$ to $|\mathcal{D}_5| = 48$ episodes. For Garment Folding, the policy completes lifelong adaptation in $R = 2$ rounds, incorporating 12 targeted recoveries and expanding the dataset from $|\mathcal{D}_0| = 18$ to $|\mathcal{D}_2| = 30$ episodes. To ensure statistical consistency, all lifelong learning experiments across both tasks and all compared methods are repeated over 3 independent end-to-end runs. After each fine-tuning round, the resulting policy checkpoint is evaluated over 20 independent autonomous trials, and we report the mean success rate and standard deviation across the 3 runs in Table 17 and Figure 4.

Table 17: Detailed progressive success rates (%) across 3 independent lifelong learning runs on the physical Galaxea bimanual workstation (20 autonomous trials per checkpoint per run). Both individual run outcomes and summary statistics (Mean ± Std) are reported across deployment rounds.

(a) Hierarchical Block Stacking ($|\mathcal{D}_0| = 18 \rightarrow |\mathcal{D}_5| = 48$)

Method	Run	R0 (18)	R1 (24)	R2 (30)	R3 (36)	R4 (42)	R5 (48)
Human Intervention	Run 1	0.0	0.0	5.0	15.0	20.0	50.0
	Run 2	0.0	0.0	10.0	35.0	60.0	70.0
	Run 3	0.0	0.0	0.0	20.0	45.0	55.0
	Mean±Std	0.0 ± 0.0	0.0 ± 0.0	5.0 ± 5.0	23.3 ± 10.4	41.7 ± 20.2	58.3 ± 10.4
Galaxea G0.5 + Diff-DAgger	Run 1	0.0	0.0	20.0	60.0	70.0	90.0
	Run 2	0.0	0.0	10.0	80.0	70.0	70.0
	Run 3	0.0	0.0	5.0	35.0	50.0	80.0
	Mean±Std	0.0 ± 0.0	0.0 ± 0.0	11.7 ± 7.6	58.3 ± 22.5	63.3 ± 11.5	80.0 ± 10.0
EqC (ours)	Run 1	0.0	0.0	25.0	60.0	50.0	100.0
	Run 2	0.0	5.0	30.0	55.0	85.0	95.0
	Run 3	0.0	0.0	30.0	70.0	80.0	95.0
	Mean±Std	$\mathbf{0.0 \pm 0.0}$	$\mathbf{1.7 \pm 2.9}$	$\mathbf{28.3 \pm 2.9}$	$\mathbf{61.7 \pm 7.6}$	$\mathbf{71.7 \pm 18.9}$	$\mathbf{96.7 \pm 2.9}$

(b) Garment Folding ($|\mathcal{D}_0| = 18 \rightarrow |\mathcal{D}_2| = 30$)

Method	Run	R0 (18)	R1 (24)	R2 (30)
Human Intervention	Run 1	30.0	35.0	40.0
	Run 2	30.0	40.0	65.0
	Run 3	30.0	35.0	60.0
	Mean±Std	30.0 ± 0.0	36.7 ± 2.9	55.0 ± 13.2
Galaxea G0.5 + Diff-DAgger	Run 1	25.0	35.0	90.0
	Run 2	25.0	50.0	80.0
	Run 3	25.0	45.0	75.0
	Mean±Std	25.0 ± 0.0	43.3 ± 7.6	81.7 ± 7.6
EqC (ours)	Run 1	30.0	45.0	100.0
	Run 2	30.0	80.0	100.0
	Run 3	30.0	60.0	95.0
	Mean±Std	$\mathbf{30.0 \pm 0.0}$	$\mathbf{61.7 \pm 17.6}$	$\mathbf{98.3 \pm 2.9}$

D.5 Cross-Task Continual Learning and Catastrophic Forgetting

Protocol. Real-world lifelong deployment often requires agents to acquire novel manipulation skills sequentially while retaining competence on previously mastered behaviors. To stress-test this capability, we select the fully converged Block Stacking policy checkpoint from Round 5 that achieved 100.0% autonomous success (corresponding to Run 1 in Table 17, trained on $|\mathcal{D}_5| = 48$ demonstrations) and deploy it directly onto the novel *Hierarchical Cup Stacking* benchmark without any prior cup interactions.

- **Round 0:** Evaluated directly on the cup task, this base checkpoint achieves 0.0% success due to geometric discrepancy and altered contact dynamics, while maintaining its initial 100.0% competence on block stacking. As visualized in Figure 5 (Top row), trajectory drift and grasp slippage immediately trigger persistent $U_{\Delta E}$ violations, prompting expert assistance before hardware collisions occur.
- **Round 1:** Using EqC's Energy-Gated DAgger with the frozen threshold γ, persistent $U_{\Delta E}$ violations detect out-of-competence execution and prompt human recovery, collecting $|\mathcal{Q}_1^{\text{cup}}| = 6$ cup stacking demonstrations. After fine-tuning on $\mathcal{D}_5 \cup \mathcal{Q}_1^{\text{cup}}$, cup stacking performance jumps to 50.0%. However, objective conflicts between tasks induced representation interference and parameter drift, causing the original block stacking success to degrade to 60.0% and manifesting classic catastrophic forgetting in continual learning.
- **Round 2:** During Round 2 deployment, EqC monitors both tasks simultaneously. It detects residual competence gaps in cup stacking as well as regression in block stacking, triggering targeted interventions that collect an additional 6 cup stacking episodes and 6 block stacking episodes ($|\mathcal{Q}_2| = 12$). Fine-tuning on the cumulative dataset successfully resolves task competition: both *Hierarchical Block Stacking* and *Hierarchical Cup Stacking* achieve

100.0% autonomous success (with autonomous cup stacking illustrated in Figure 5, Bottom row; Table 18).

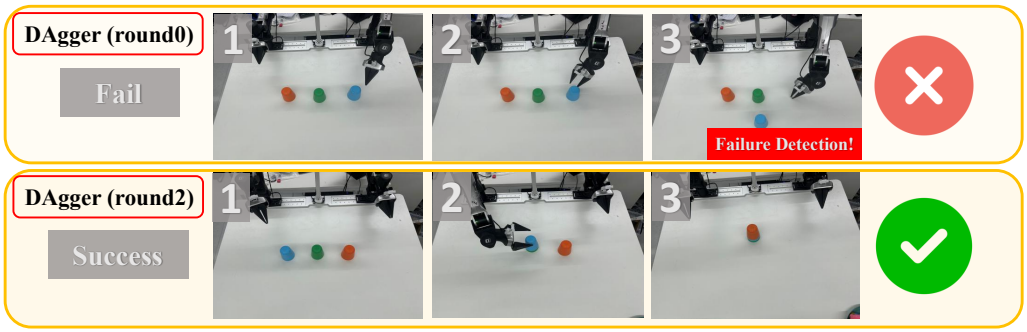


Figure 5: **Cross-task adaptation on *Hierarchical Cup Stacking*. Top (Round 0):** Zero-shot deployment causes grasp slippage and drift, promptly caught by a persistent $U_{\Delta E}$ failure alarm. **Bottom (Round 2):** After two rounds of Energy-Gated DAgger, the policy achieves fully autonomous execution, cleanly assembling the three-tier cup tower.

Table 18: Progressive performance (%) across continual learning rounds under task shift.

Metric / Task	Round 0 (Base)	Round 1 (+6 Cup)	Round 2 (+6 Cup, +6 Block)
Cumulative Demos	48 (Block only)	54 (48 Block, 6 Cup)	66 (54 Block, 12 Cup)
Block Stacking Success	**100.0**	60.0	**100.0**
Cup Stacking Success	0.0	50.0	**100.0**
Dual-Task Average	50.0	55.0	**100.0**

This result demonstrates that actively gating data collection for degraded and novel skills provides a practical mechanism for recovering from task-interference-induced forgetting.

E Ablation Study

E.1 Robustness to Optimization Step Schedules

To further evaluate the sensitivity of EqC to the underlying optimization step-size schedule, we conduct an ablation study across the LIBERO benchmark. Specifically, we compare our default Cosine decay schedule against uniform step schedules (with 15 and 20 optimization steps). Table 19 presents the performance breakdown across the four standard LIBERO task suites (Goal, Spatial, Object, and Long). The results demonstrate that EqC is highly robust to variations in the optimization scheduling strategy. While the Cosine decay schedule yields the best average performance (achieving 94.3), the uniform schedules maintain competitive results, with the 20-step uniform schedule reaching 94.2 and the 15-step schedule maintaining 93.1. This consistent performance indicates that our energy-based failure detection mechanism relies on the fundamental energy landscape rather than highly tuned optimization step sizes, ensuring reliable and stable operation across different scheduling choices.

E.2 Adaptive Inference

To investigate whether our action optimization converges to an equilibrium during inference, we adapt EqC to DiT backbones and analyze the optimization dynamics across the LIBERO benchmark. During inference procedure, we observe that candidate actions rapidly reach a stationary regime, as evidenced by two main aspects in Table 20:

Table 19: Success rate (%) of EqC under different optimization step schedules across the LIBERO suites. The highest values in each column are highlighted in bold.

Methods	Goal	Spatial	Object	Long	Average
EqC (uniform 15 steps)	**95.8**	95.2	98.4	83.0	93.1
EqC (uniform 20 steps)	95.4	96.2	98.0	**87.2**	94.2
EqC (Cosine decay)	95.6	**97.2**	**98.6**	85.4	**94.3**

- **Rapid Gradient Decay**: the mean gradient norm $\|\nabla E\|$ decreases almost monotonically from an initial noise scale (49.3) at step 0 to around 1.0 by step 14, ultimately flattening to a floor of 0.3 at the final step.
- **Early Energy Saturation**: the normalized energy convergence ratio—defined as $(E_k - E_0)/(E_{\text{final}} - E_0)$—saturates remarkably early, reaching 99.0% by step 7 and exceeding 99.9% by step 9.

These empirical observations indicate that our method rapidly reaches an equilibrium, as action candidate refinement enters a stable energy state within the first 10–15 steps.

Motivated by this steady-state behavior, we introduce an adaptive early-stopping mechanism that terminates action chunk optimization once the gradient norm falls below a threshold $g_{\min} \in \{1.0, 3.0\}$. As summarized in Table 21, adaptive early stopping significantly enhances inference efficiency without incurring any notable performance degradation. Specifically, setting $g_{\min} = 1.0$ reduces the average optimization steps to 16.8 (a 47.5% reduction in computation compared to the full budget) while maintaining an equivalent average success rate (93.7 vs. 94.0 baseline). Setting the threshold to $g_{\min} = 3.0$ further truncates the average optimization steps down to 11.1 (saving 65.3% of inference computation) while achieving a success rate of 94.2. These results confirm that our proposed method—Equilibrium Energy Control—can effectively eliminate redundant computation while preserving policy execution quality through an adaptive gradient-based stopping strategy.

Table 20: Convergence profile of mean gradient norm $\|\nabla E\|$ and normalized energy saturation ratio across optimization steps k on LIBERO.

Step k	0	5	6	7	8	9	14	19	32
Mean Gradient Norm $\|\nabla E\|$	49.3	44.4	29.9	12.9	6.0	3.6	1.2	0.7	0.3
E Saturation Ratio (%)	0.00	82.51	95.18	99.22	99.79	99.91	99.99	–	–

Table 21: Performance and inference efficiency under adaptive early stopping across LIBERO task suites. Metrics are reported as Success Rate (%) / Average Optimization Steps per action chunk.

Task Suite	Baseline ($K = 32$)	$g_{\min} = 1.0$	$g_{\min} = 3.0$
libero_goal	95.6 / 32.0	97.0 / 16.9	97.0 / 11.1
libero_spatial	96.4 / 32.0	95.2 / 17.3	98.0 / 11.3
libero_object	98.1 / 32.0	98.0 / 16.2	97.6 / 10.9
libero_long	85.5 / 32.0	84.6 / 16.8	84.2 / 10.9
Average	**94.0 / 32.0**	**93.7 / 16.8**	**94.2 / 11.1**

F Case Study: Detection Timeliness and Failure Modes

A practical introspective robot system must trigger assistance *at the competence boundary*: alarming neither prematurely on benign execution, nor belatedly after irreversible physical damage has occurred. In this section, we present real-world qualitative case studies on the physical Galaxea bimanual manipulation platform to evaluate detection timeliness across two domains: rigid-body multi-stage contact (*Hierarchical Block Stacking*) and deformable manipulation (*Garment Folding*). We first highlight EqC's proactive detection capability, followed by a detailed diagnostic of the baseline's characteristic pathology modes.

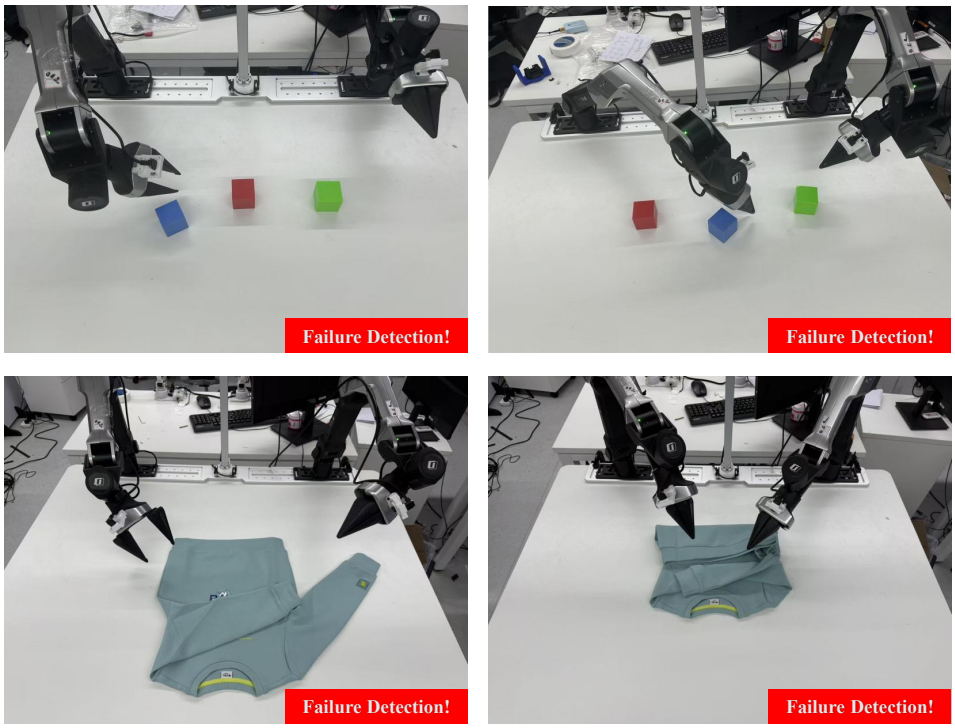


Figure 6: **Proactive competence-boundary detection of EqC across rigid and deformable manipulation.** Persistent $U_{\Delta E}$ violations trigger timely expert recovery at the earliest onset of execution drift, preserving an ample window for teleoperation before irreversible physical compounding occurs. **Top row (Hierarchical Block Stacking):** *Left:* Alarm raised when the block is accidentally pushed away, preventing it from being pushed further; *Right:* Alarm triggered by angular misalignment. **Bottom row (Garment Folding):** *Left:* Alarm triggered when the robot attempts to grasp the right sleeve but fails to secure it; *Right:* Alarm triggered upon failure to fold the garment to the designated position.

F.1 Proactive Detection by EqC at the Competence Boundary

As illustrated in Figure 6, EqC's path-wise energy consistency residual $U_{\Delta E}$ demonstrates sharp sensitivity to the earliest emergence of off-manifold execution drift:

- **Rigid-Body Manipulation (Top row):** During *Hierarchical Block Stacking*, when the block is accidentally pushed away (Top-left) or angular misalignment occurs during contact placement (Top-right), $U_{\Delta E}$ persistently exceeds the threshold γ, arresting autonomous motion *before* blocks are knocked over or the tower collapses.
- **Deformable Manipulation (Bottom row):** In *Garment Folding*, EqC reliably flags failure to secure the right sleeve during grasping (Bottom-left) and failure to reach the target folding position (Bottom-right), requesting expert takeover while the fabric remains completely recoverable.

F.2 Pathology Diagnostics of Flow-Loss Detection (Diff-DAgger)

In contrast to EqC's proactive warnings, point-wise velocity error in flow-matching policies displays significant variance and poor temporal sensitivity to compounding geometric error. As illustrated in Figure 7, evaluating **Galaxea G0.5 + Diff-DAgger** across both rigid stacking (Rows 1–2) and deformable garment folding (Row 3) reveals three distinct, pervasive pathology modes across multiple independent trials:

1. **Delayed Detection during Cumulative Drift (Figure 7, Left column):** When the manipulators suffer from progressive spatial drift, Diff-DAgger persistently fails to raise an alarm

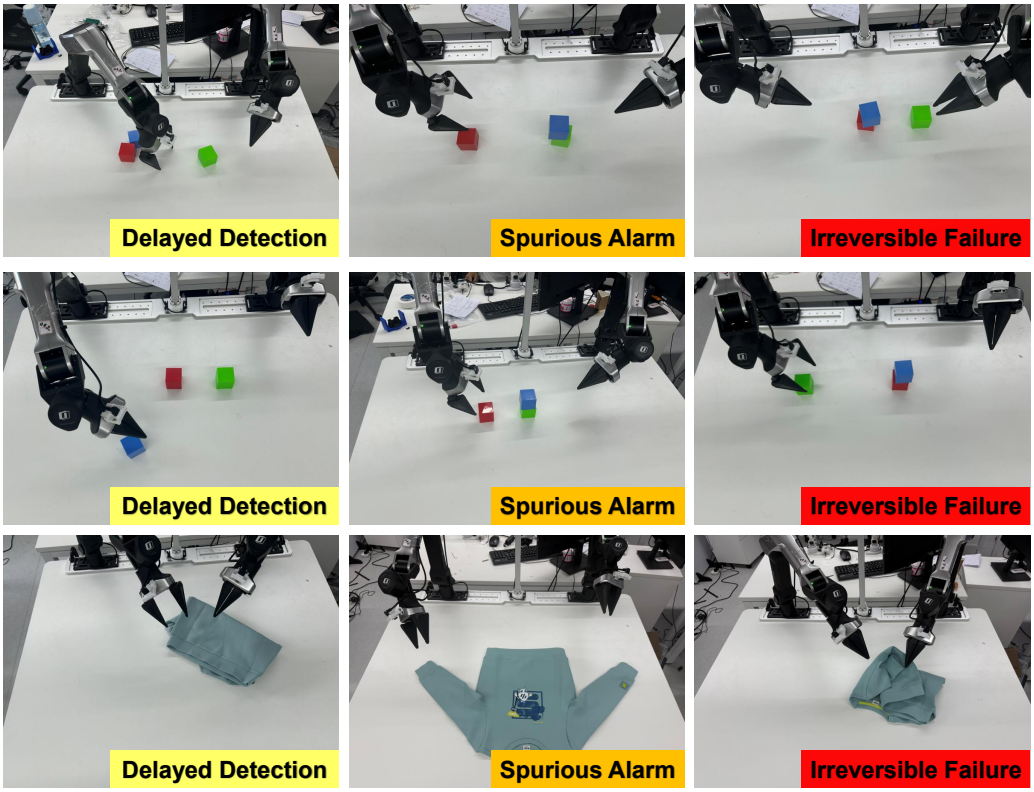


Figure 7: **Qualitative failure modes of Galaxea G0.5 + Diff-DAgger across rigid and deformable manipulation.** Rows 1–2 evaluate *Hierarchical Block Stacking* under nominal setups (two independent instances per category), while Row 3 evaluates *Garment Folding*. **Left column (Delayed Detection):** The detector remains silent throughout compounding drift, alarming only belatedly after the block arrangement is disrupted or a severe angular tilt has already occurred, missing the teleoperation recovery window. **Middle column (Spurious Alarm):** During nominal execution—including pristine, fully unfolded garment setups (Row 3)—noisy flow-loss expectation triggers false-positive alarms, needlessly aborting valid rollouts. **Right column (Irreversible Failure):** The detector triggers only after irrecoverable failures have occurred (e.g., mis-sequenced block placement or fabric bunched into an entangled knot), leaving no viable recovery path without a complete manual scene reset.

throughout the approach trajectory. Across both block stacking (Rows 1–2) and garment folding (Row 3), it alarms only at a belated stage—after the block cluster is disrupted or the fabric has already suffered a severe angular tilt—severely compressing or completely missing the reaction window required for human teleoperator intervention.

2. **Spurious False Alarm on Supported Execution (Figure 7, Middle column):** During nominal, error-free execution, Diff-DAgger's noisy loss expectation frequently induces spurious spikes. This occurs not only during mid-trajectory block transit, but even on completely flat, undisturbed garments (Row 3, Middle), mistakenly triggering false-positive alarms that prematurely halt valid execution.
3. **Irreversible Failure Triggering (Figure 7, Right column):** When execution diverges critically (e.g., stacking onto the wrong block or severely bunching fabric), Diff-DAgger fails to catch the deviation during transit. It alerts only post-contact or after the garment is balled into an entangled knot (Row 3, Right). At this terminal stage, in-place teleoperation is impossible, requiring a full manual reset of the environment.

G CODE AVAILABILITY

We have released the code at `https://anonymous.4open.science/r/eqc-iclr-anonymous-21AB`.