# EVIDENCE-GUIDED COUNTERFACTUAL LEARNING FOR LONG-TAILED ROBOTIC POLICIES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

In the practical deployment of vision–language–action policies, handling long-tailed task distributions is a common challenge due to the highly imbalanced composition of robot demonstration datasets. To investigate this issue, we study the head-task action bias in shared robot policies, particularly focusing on its relationship with task-specific visual evidence. Our experimental analysis shows that, when the visual evidence required by a tail task is insufficient, the policy increasingly falls back on action patterns reinforced by data-rich head tasks, resulting in degraded performance on under-represented tasks. Motivated by this observation, we propose Evidence-Guided Counterfactual Learning, a training framework for long-tailed robot policies. Our method identifies decision-relevant visual evidence using instruction-related and action-related signals, and constructs counterfactual observations by removing the selected evidence. The resulting counterfactual predictions are used to strengthen the association among task instructions, visual evidence, and actions, with the supervision adaptively adjusted according to the number of demonstrations available for each task. The proposed framework requires neither additional training data nor region-level annotations and incurs no additional computation during deployment. Extensive experiments across simulated benchmarks, different VLA architectures, and real-world robot platforms demonstrate that our method robustly and consistently improves long-tailed robot policy learning, particularly for tail tasks, while maintaining reliable performance on head tasks.

## 1 INTRODUCTION

Vision–language–action models (VLAs) have emerged as a promising foundation for generalist robot policies, integrating visual perception, language understanding, and action generation within a unified framework (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Ma et al., 2026). Robot demonstration datasets, however, can be highly imbalanced across manipulation skills (Open X-Embodiment Collaboration et al., 2024). Similar to the long-tailed category distributions observed in datasets such as iNaturalist (Van Horn et al., 2018), robot task distributions can be skewed toward a few data-rich head tasks, leaving many tail tasks with limited demonstrations(Zhu et al., 2026).

Long-tailed learning addresses unequal supervision through approaches such as resampling, loss reweighting, and decoupled representation and classifier learning (Cui et al., 2019; Kang et al., 2019; Ma et al., 2022). However, most of these methods are developed for classification and are difficult to apply directly to VLAs due to differences in task formulation and model architecture.Zhu et al. (2026) propose Approaching-Phase Augmentation (APA) for long-tailed VLA learning, but it relies on additional tools, such as object detection and image inpainting, increasing the complexity of the approach.Moreover, APA primarily mitigates the problem through data augmentation, without directly constraining VLAs' reliance on head-task action bias.

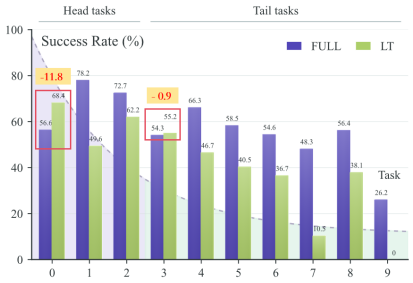
However, comparing the results of policies trained on the balanced FULL dataset and the long-tailed LT dataset (see Appendix A), we find that tail-task performance drops substantially under LT, as shown in Fig. 1(a). Interestingly, some head tasks perform worse under FULL than under LT: more complete and diverse training data do not yield uniform improvements across all tasks. This leads us to consider whether the long-tailed multi-task learning problem is related to how different tasks jointly influence a shared policy. We further visualize the model's attention during inference.



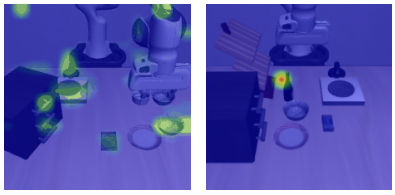
Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping 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.

Figures 1(b) and (c) show that head tasks attend reasonably well to visual evidence associated with the target object, but may exhibit more dispersed visual attention due to cross-task interference. Tail tasks face a more severe problem: they struggle to establish reliable task-specific visual grounding. Through the toy experiment in Section 3.1, we find that when such evidence is insufficient, the model must still produce an action, and the shared policy tends to fall back on frequent action patterns in the training data, giving rise to a **head-task action bias**. Thus, tail tasks do not merely suffer from insufficient data exposure; during decision-making, shared action priors associated with head tasks are more likely affect the visual evidence needed for the current task.

This finding raises a new question: ***how can we preserve reliable decisions on head tasks while preventing tail tasks from relying excessively on head-task action bias?***




**(a)** Task success rates



**(b)** Head task **(c)** Tail task

Figure 1: **Head vs.tail task performance and intermediate-level visual grounding.** (a)Success rates for policies trained on FULL and LT. (b) Visual attention for "Pick up the black bowl next to the plate and place it on the plate." (c) Visual attention for "Put the wine bottle on the rack."

Thus, we introduce **Evidence-Guided Counterfactual Learning (ECL)**, Without requiring additional annotations or training data, ECL directly uses the original data and a relation-aware counterfactual branch to identify visual evidence relevant to decisions during rollouts. It constructs counterfactual samples by removing key relational tokens and strengthens the associations among task instructions, visual relations, and actions through counterfactual training. Crucially, evidence guides both counterfactual construction and supervision in ECL, which adaptively assigns weaker regularization to head tasks and stronger supervision to tail tasks according to the number of demonstrations.

**Our contributions:** **1)**We reassess the conventional view that tail-task failures arise from data scarcity and show through a toy experiment that, when task-specific visual grounding is insufficient, tail-task predictions fall back on action bias shaped by head tasks. **2)** We propose ECL, which suppresses tail tasks' reliance on action priors via counterfactual supervised training to build their task-specific evidence, without requiring extra annotations or additional training data. **3)** We evaluate ECL on two VLA policies in LIBERO-Core-LT and on two real-world robot platforms, Our method achieves state-of-the-art performance, with macro-averaged gains and task-specific outcomes reported.

## 2 Related Work

**Long-Tailed Learning.** Long-tailed recognition studies class imbalance that biases models toward head classes (Zhang et al., 2023). Representative approaches include class-balanced re-weighting (Cui et al., 2019), decoupled representation and classifier learning (Kang et al., 2019), and prior-based logit adjustment (Menon et al., 2020). LDAM adjusts classification margins according to class frequencies (Cao et al., 2019), Balanced Softmax addresses label distribution shift (Ren et al., 2020), and DisAlign combines adaptive score calibration with prior-based re-weighting (Zhang et al., 2021). Feature augmentation methods enrich tail classes through transferable class-generic variations (Chu et al., 2020), learned semantic augmentation directions (Li et al., 2021), or head-class feature displacements (Wang et al., 2021). Contrastive methods improve representation geometry: PaCo introduces learnable class-wise centers (Cui et al., 2021), while BCL balances inter-class contributions (Zhu et al., 2022). These methods primarily address discrete class imbalance rather than long-tailed state and behavior distributions in sequential robotic manipulation.

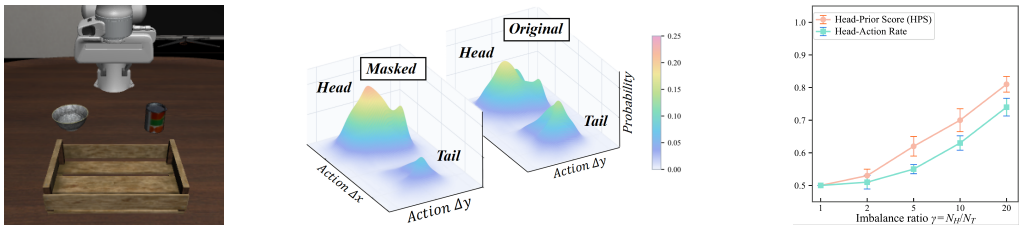
**Robotic Foundation Models.** Robotic foundation models build on large vision and language models (Touvron et al., 2023; Zhai et al., 2023; Oquab et al., 2023). PaLM-E integrates visual, state, and textual inputs for embodied reasoning and planning (Driess et al., 2023). Robot policies span autoregressive, diffusion-based, and continuous generative approaches. RT-1 (Brohan et al., 2022) scales

Transformer-based behavior cloning through action tokenization, while RT-2 (Brohan et al., 2023) transfers Internet-scale vision-language knowledge to robotic control. OpenVLA (Kim et al., 2024) combines a pretrained language model with SigLIP and DINOv2 features and trains on Open X-Embodiment demonstrations (Open X-Embodiment Collaboration et al., 2024). MiniVLA (Belkhale & Sadigh, 2024) adopts a lightweight Qwen2.5-0.5B backbone, while OpenVLA-OFT improves adaptation and inference through parallel decoding, action chunking, and continuous regression (Kim et al., 2025). Diffusion policies generate continuous actions through iterative denoising (Chi et al., 2025). Octo (Team et al., 2024) trains a Transformer-based diffusion policy on diverse robot trajectories for adaptation to new observation and action spaces. $\pi_0$ (Black et al., 2024) combines flow matching with a pretrained vision-language model for cross-embodiment control, while $\pi_{0.5}$ (Intelligence et al., 2025) broadens task and environment coverage. SmolVLA uses a compact architecture and asynchronous inference for efficient training and deployment (Shukor et al., 2025). Despite these advances, training data can remain imbalanced across tasks, states, and behaviors, motivating further investigation into how long-tailed distributions affect visuomotor representations and action prediction.

# 3 Diagnosing and Mitigating Head Task Action Bias

## 3.1 Toy Experiment on Head Task Action Bias

We construct a two-task diagnostic to examine how task imbalance and visual masking affect action preferences under tail-task instructions. Two objects are positioned symmetrically to the left and right of a shared target container (Fig. 2(a)). The tasks share the scene, robot initialization, and manipulation procedure, and differ in the instruction specifying the object to manipulate. We fix the number of head-task demonstrations and subsample the tail-task data to obtain frequency ratios $\gamma = N_H/N_T \in \{1, 2, 5, 10, 20\}$. This design measures the combined effect of increasing relative imbalance and reducing tail-data coverage.




**(a)** Task setup **(b)** Action distributions **(c)** Head-task bias

Figure 2: **Diagnosing Head Task Action Bias.** (a) Two-task environment. (b) Action distributions before and after relational masking at the same state under a tail-task instruction. (c) Head-Prior Score and Head-Action Rate versus task imbalance.

For each trained policy, we condition on the tail-task instruction at a common set of diagnostic states and inspect the predicted action-token distribution. We focus on horizontal displacement $\Delta x$, which distinguishes the two tasks, and exclude action dimensions shared by both. With the head task assigned to the left object and the tail task to the right, we partition directional tokens into a *head-consistent set* $A_H$, comprising displacements $\Delta x < -\tau$, and a *tail-consistent set* $A_T$, comprising displacements $\Delta x > \tau$. The threshold $\tau$ filters out negligible movements. Let $P_H(x)$ and $P_T(x)$ denote the probability mass assigned to $A_H$ and $A_T$, respectively. We define the *Head-Prior Score (HPS)* as:

$$\mathrm{HPS}(x) = \frac{P_H(x)}{P_H(x) + P_T(x)} \tag{1}$$

Here, $x$ indexes a diagnostic state. HPS is defined when $P_H(x) + P_T(x) > 0$ and measures the relative preference between the two directional sets under a tail-task instruction. A value of $0.5$ indicates equal probability, while larger values indicate a stronger head-task bias.

We compare predictions from the intact observation, $p(a \mid I, L_T)$, with predictions from the masked observation, $p(a \mid I_{\neg R}, L_T)$, at the same diagnostic states. Here, $L_T$ is the tail-task instruction and

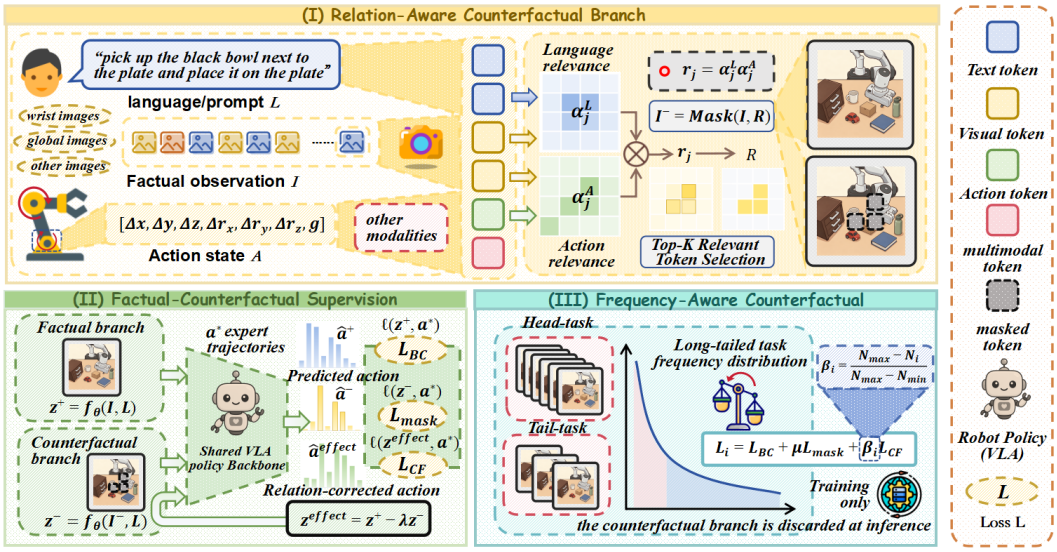



Figure 3: **Overview of ECL. (I) Joint language–action relevance selects top-$K$ visual tokens for masking. (II) A shared policy produces factual, masked, and contrastive training outputs, supervised by $\mathcal{L}_{\mathrm{BC}}$, $\mathcal{L}_{\mathrm{mask}}$, and $\mathcal{L}_{\mathrm{CF}}$, respectively. (III) Task frequency determines the per-sample weight of the counterfactual loss. At deployment, only the factual policy is used.**

$I_{\neg R}$ denotes the observation after masking selected task-relevant visual evidence. Masking shifts the action distribution further toward the head-task direction (Fig. 2(b)). We also measure the *Head-Action Rate*, the fraction of rollouts in which the initial movement is toward the head-task object despite a tail-task instruction. Both HPS and the Head-Action Rate increase with task imbalance rate (Fig. 2(c)). Exchanging the head–tail assignment reverses the directional preference, which argues against a fixed left–right bias in this setting.

### 3.2 Frequency-Aware Counterfactual Supervision

The above experiments show that bias induced by head tasks affects inference on tail tasks, reducing their performance. Based on this finding, We introduce **Evidence-Guided Counterfactual Learning (ECL)**, a counterfactual learning strategy that masks key visual regions for tail tasks. By constructing a counterfactual branch that disrupts the original visual information, FACS encourages the model to reduce its reliance on head tasks and re-establish connections to tail-task-specific evidence.Figure 3 summarizes the framework.

#### 3.2.1 Relation-Aware Counterfactual Branch

Let $I$ denote the visual observation and $L$ the instruction. For simplicity, we omit policy-specific conditioning variables and write the factual prediction from the intact observation as:

$$z^{+} = f_{\theta}(I, L) \tag{2}$$

Here, $z^{+}$ denotes the prediction output on which the supervised action loss is evaluated. The factual and counterfactual branches share the same prediction space, allowing their contrast to be supervised using the same action loss.

To identify task-relevant visual evidence, we combine instruction-related and action-related attention as complementary relevance signals, while recognizing that attention weights alone do not reliably explain model predictions (Jain & Wallace, 2019). The former reflects how instruction-related queries attend to visual tokens, providing cues for locating the target and associated regions described by the instruction. The latter reflects how action-prediction queries attend to visual tokens, indicating which visual information the model attends to when generating actions. Relying solely on instruction-related attention may select semantically relevant regions that are not sufficiently utilized for action prediction. Conversely, relying solely on action-related attention may select regions

shared across tasks that do not adequately distinguish the current task. We therefore combine both signals to prioritize visual tokens relevant to both task semantics and action prediction:

$$r_j = \alpha_j^L \alpha_j^A \tag{3}$$

Here, $\alpha_j^L$ and $\alpha_j^A$ denote the instruction-related and action-related attention scores for visual token $v_j$, respectively. The product suppresses tokens with a low score in either signal, emphasizing regions attended to by both. We select the top-$K$ tokens to form the set $R$, using a default selection ratio of 10%, with ablations on this ratio presented in Section 4.3.This joint score guides evidence selection without equating attention weights with the causal contributions of individual tokens.

We then keep the instruction unchanged, mask the selected visual tokens, and process the intact and masked observations using the same policy:

$$I^- = \mathrm{Mask}(I, R), \tag{4}$$

$$z^- = f_\theta(I^-, L). \tag{5}$$

The masked branch addresses the following question: ***what action predictions does the policy still produce when key visual evidence relevant to the current task is removed?*** Our diagnostic experiments show that insufficient task-specific evidence makes tail-task predictions more susceptible to head-task action bias. We therefore use $z^-$ as a reference under incomplete visual evidence for comparison with the prediction from the intact observation. This reference still reflects the influence of unmasked information and should not be equated with head-task action bias itself.

Inspired by the use of factual–counterfactual prediction differences to guide learning (Mothilal et al., 2020; Chiappa, 2019; Goyal et al., 2019; Sauer & Geiger, 2021), we construct the following contrastive training output:

$$\boxed{z^{\mathrm{effect}} = z^+ - \lambda z^-}, \tag{6}$$

where $\lambda$ controls the contribution of the masked branch to the prediction contrast and is fixed at $0.3$ during training, with ablations presented in Section 4.3. By applying action supervision to $z^{\mathrm{effect}}$, we encourage the policy to learn action responses supported by task-specific visual evidence in the intact observation, reducing its reliance on head-task action bias. The selected visual regions thus serve not only to construct masked samples but also to inform supervision through the prediction contrast between the two branches, strengthening the associations among task instructions, visual evidence, and actions. The counterfactual branch is used only during training; at deployment, the policy generates actions directly from the intact visual observation and language instruction.

### 3.2.2 Frequency-Aware Counterfactual Supervision

Let $\ell(z, a^*)$ denote the base policy's supervised action-prediction loss, where $a^*$ denotes the expert action target in the corresponding prediction space. We supervise the factual prediction, the masked prediction, and their counterfactual action effect:

$$\mathcal{L}_{\mathrm{BC}} = \ell(z^+, a^*), \qquad \mathcal{L}_{\mathrm{mask}} = \ell(z^-, a^*), \qquad \mathcal{L}_{\mathrm{CF}} = \ell(z^{\mathrm{effect}}, a^*). \tag{7}$$

These objectives serve different roles. $\mathcal{L}_{\mathrm{BC}}$ retains the base policy's original supervision and directly optimizes the factual prediction used at deployment. $\mathcal{L}_{\mathrm{mask}}$ encourages the masked branch to predict actions using the remaining information, providing a meaningful reference under incomplete visual evidence. $\mathcal{L}_{\mathrm{CF}}$ supervises the counterfactual action effect derived from the two branches, encouraging the policy to associate task-specific visual evidence with actions and reduce its reliance on head-task action bias. The masked branch can still exploit task information that remains available and is not restricted to representing task-independent behavior.

Our diagnostic experiments show that tail tasks are more susceptible to head-task action bias when visual grounding is insufficient. We therefore allocate counterfactual supervision according to demonstration frequency, assigning stronger supervision to less frequent tasks while limiting additional constraints on frequent tasks. Let $N_i$ denote the number of training demonstrations for task $i$, and let $N_{\max}$ and $N_{\min}$ be the largest and smallest demonstration counts across tasks. For

$N_{\max} > N_{\min}$, we define the normalized rarity of task $i$ as:

$$r_i = \frac{N_{\max} - N_i}{N_{\max} - N_{\min}}, \qquad r_i \in [0, 1]. \tag{8}$$

We directly use $r_i$ as the the weight of the counterfactual supervision loss, yielding the training objective for a sample from task $i$:

$$\boxed{\mathcal{L}_i = \mathcal{L}_{\text{BC}} + \mu\mathcal{L}_{\text{mask}} + r_i\mathcal{L}_{\text{CF}}}. \tag{9}$$

Here, $\mu$ controls the auxiliary supervision strength of the masked branch. For the most frequent task, $r_i = 0$, retaining only factual and masked prediction supervision. For the least frequent task, $r_i = 1$, applying the full weight to counterfactual supervision. Intermediate tasks receive progressively larger weights as their demonstration counts decrease. This weighting operates per sample; each task's overall contribution to optimization also depends on its sampling frequency.

The frequency-based schedule uses demonstration counts as a proxy for the amount of supervision available to each task. The combined objective aims to balance action prediction and task-specific visual grounding, with its effectiveness assessed through factual-policy performance and loss-component ablations.

Table 1: Per-task success rates (%, mean ± binomial standard deviation of the rate) on the ten-task LIBERO-Core benchmark. The table reports four training configurations for MiniVLA and $\pi_{0.5}$.The Average standard deviation is computed across the three seed-wise mean success rates.

| | MiniVLA Success Rate (%) | | | | $\pi_{0.5}$ Success Rate (%, synthetic) | | | |
|---|---|---|---|---|---|---|---|---|
| Task | FULL | LT(Baseline) | APA | ECL(Ours) | FULL | LT(Baseline) | APA | ECL(Ours) |
| Task 0 | 56.6 ± 4.0 | 68.4 ± 2.3 | 72.6 ± 2.3 | 80.0 ± 2.2 | 56.7 ± 2.4 | 72.4 ± 3.4 | 69.5 ± 1.6 | 75.6 ± 2.5 |
| Task 1 | 78.2 ± 4.0 | 49.6 ± 3.6 | 67.3 ± 1.4 | 68.9 ± 1.3 | 61.5 ± 3.5 | 64.4 ± 1.6 | 65.8 ± 1.9 | 71.7 ± 1.8 |
| Task 2 | 72.7 ± 3.8 | 62.2 ± 3.2 | 66.1 ± 1.8 | 74.7 ± 1.8 | 72.6 ± 2.4 | 58.7 ± 2.4 | 57.8 ± 1.5 | 66.4 ± 0.7 |
| Task 3 | 50.3 ± 3.8 | 55.2 ± 3.5 | 70.0 ± 2.1 | 83.5 ± 1.8 | 50.2 ± 1.6 | 52.6 ± 2.6 | 55.4 ± 1.9 | 59.6 ± 0.4 |
| Task 4 | 66.3 ± 4.2 | 46.7 ± 3.6 | 56.5 ± 1.6 | 64.7 ± 1.4 | 68.3 ± 2.0 | 45.8 ± 1.5 | 43.3 ± 3.4 | 52.1 ± 3.4 |
| Task 5 | 58.5 ± 4.2 | 40.5 ± 3.4 | 44.5 ± 1.6 | 48.0 ± 1.3 | 58.8 ± 2.3 | 36.6 ± 2.7 | 40.3 ± 1.7 | 44.4 ± 2.4 |
| Task 6 | 54.6 ± 4.0 | 36.7 ± 3.4 | 37.2 ± 1.8 | 39.7 ± 1.1 | 53.9 ± 2.1 | 29.8 ± 3.5 | 27.9 ± 1.6 | 37.5 ± 3.0 |
| Task 7 | 48.3 ± 4.2 | 10.5 ± 4.0 | 11.8 ± 1.6 | 13.7 ± 0.9 | 46.5 ± 2.1 | 23.1 ± 0.8 | 27.6 ± 2.2 | 30.7 ± 1.2 |
| Task 8 | 56.4 ± 4.2 | 38.1 ± 3.4 | 20.9 ± 1.3 | 54.3 ± 1.1 | 58.2 ± 2.3 | 15.3 ± 3.2 | 19.4 ± 0.9 | 22.8 ± 2.4 |
| Task 9 | 26.2 ± 4.0 | 0.0 ± 0.0 | 1.3 ± 0.9 | 2.4 ± 0.7 | 26.2 ± 2.2 | 4.3 ± 0.7 | 10.1 ± 0.8 | 11.3 ± 3.7 |
| Average | 57.2 ± 4.0 | 40.8 ± 3.0 | 44.8 ± 1.6 | 53.0 ± 1.4 | 55.3 ± 2.0 | 40.3 ± 1.1 | 41.7 ± 1.6 | 47.2 ± 0.6 |

## 4 EXPERIMENTS

This section is organized around the following research questions:

*1) Does ECL improve long-tail policy performance in simulation and the real world?*
*2) How do different design choices affect the performance of ECL?*
*3) How robust is ECL to different forms of long-tailed distributions?*

We address the first question in Sections 4.1 and 4.2, the second in Section 4.3, and the third in Section 4.4. Task definitions and demonstration frequencies are provided in Appendix A.

### 4.1 SIMULATION EXPERIMENTS

We conduct simulation experiments in LIBERO (Liu et al., 2023), using the 10 manipulation tasks in the LIBERO-Core benchmark (Zhu et al., 2026). The tasks cover diverse manipulation skills, including object picking, transport, placement, pushing, and container interaction. All tasks are evaluated in closed loop with a 7-DoF Franka Panda manipulator.

**Experimental Setup.** Following Zhu et al. (2026), we use LIBERO-Core-FULL and LIBERO-Core-LT and evaluate MiniVLA (Belkhale & Sadigh, 2024) and $\pi_{0.5}$ (Intelligence et al., 2025).

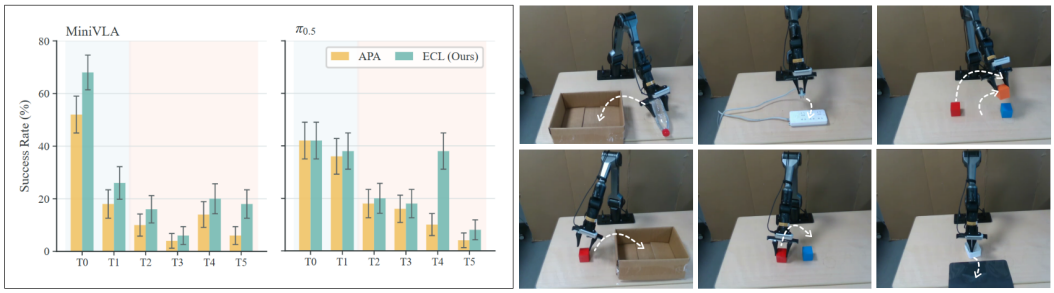



Figure 3: Real-world evaluation of ECL with MiniVLA and $\pi_{0.5}$ on AGILEX PiPER under long-tailed task distributions. Evaluation amd results on xArm6 are provided in Appendix B.

Table 2: Real-world per-task success rates (%) on AGILEX PIPER and xArm6. For each platform, APA and ECL are evaluated with MiniVLA and $\pi_{0.5}$ on six tasks ordered from head tasks (Task 0-1) to tail tasks (Task 2-5).

| | **AGILEX PIPER** | | | | **xArm6** | | | |
|---|---|---|---|---|---|---|---|---|
| | MiniVLA | | $\pi_{0.5}$ | | MiniVLA | | $\pi_{0.5}$ | |
| Task | APA | ECL(Ours) | APA | ECL(Ours) | APA | ECL(Ours) | APA | ECL(Ours) |
| Task 0 | $52 \pm 7.1$ | $68 \pm 6.6$ | $42 \pm 7.0$ | $42 \pm 7.0$ | $60 \pm 6.9$ | $62 \pm 6.9$ | $64 \pm 6.8$ | $68 \pm 7.0$ |
| Task 1 | $18 \pm 5.4$ | $26 \pm 6.2$ | $36 \pm 6.8$ | $38 \pm 6.9$ | $20 \pm 5.7$ | $38 \pm 6.9$ | $20 \pm 5.7$ | $40 \pm 6.9$ |
| Task 2 | $10 \pm 4.2$ | $16 \pm 5.2$ | $18 \pm 5.4$ | $20 \pm 5.7$ | $12 \pm 4.6$ | $20 \pm 5.7$ | $22 \pm 5.9$ | $24 \pm 6.0$ |
| Task 3 | $4 \pm 2.8$ | $6 \pm 3.4$ | $16 \pm 5.2$ | $18 \pm 5.4$ | $4 \pm 2.8$ | $6 \pm 2.8$ | $14 \pm 4.9$ | $20 \pm 5.7$ |
| Task 4 | $14 \pm 4.9$ | $20 \pm 5.7$ | $10 \pm 4.2$ | $38 \pm 6.9$ | $18 \pm 5.4$ | $18 \pm 5.4$ | $20 \pm 5.7$ | $20 \pm 5.7$ |
| Task 5 | $6 \pm 3.4$ | $18 \pm 5.4$ | $4 \pm 2.8$ | $8 \pm 3.8$ | $2 \pm 2.0$ | $12 \pm 4.6$ | $4 \pm 2.8$ | $12 \pm 4.6$ |
| Average | 17.3 | 25.7 | 21.0 | 27.3 | 19.3 | 26.0 | 24.0 | 30.7 |

MiniVLA is initialized from a checkpoint pretrained on LIBERO-90, and $\pi_{0.5}$ is initialized from $\pi_{0.5}$-Base. We compare four configurations: FULL, trained on LIBERO-Core-FULL; LT, trained directly on LIBERO-Core-LT; APA, using the augmentation strategy of Zhu et al. (2026); and ECL, trained on LIBERO-Core-LT with the proposed auxiliary objectives. All reported main results use factual-only inference: the masked branch, the contrastive output, and task-frequency weights are not used during action generation.

**Results and Analysis.** Table 1 reports success rates on the ten LIBERO-Core tasks. ECL increases macro-averaged success from 40.8% to 53.0% for MiniVLA and from 40.3% to 47.2% for $\pi_{0.5}$, corresponding to gains of 12.2 and 6.9 percentage points, or relative improvements of approximately 29.9% and 17.1%, respectively. These results indicate that ECL provides consistent overall performance gains across different VLA policies. Compared with APA Zhu et al. (2026), ECL improves macro-averaged success from 44.8% to 53.0% for MiniVLA and from 41.7% to 47.2% for $\pi_{0.5}$. On tail tasks (Tasks 7-9), average success increases from 11.3% to 23.5% for MiniVLA and from 19.0% to 21.6% for $\pi_{0.5}$.These gains are achieved using only the existing training data, without requiring additional data construction. Notably, ECL improves performance on tail tasks while also benefiting head tasks, avoiding the cross-task bias observed with FULL.

### 4.2 Real-World Experiments

We further evaluate ECL on two real robotic platforms, AGILEX PIPER and xArm6, having established ECL's improvements over the LT baseline in simulation, we focus the real-world evaluation on a direct comparison with APA Zhu et al. (2026).

**Experimental Setup.** We fine-tune MiniVLA and $\pi_{0.5}$ on a six-task real-world manipulation benchmark with imbalanced demonstration frequencies. Both policies are evaluated separately on two above manipulators, The sample counts and detailed descriptions of the six tasks are provided in Appendix A. The per-task outcomes for each platform–policy combination are reported in Table 2.




Figure 4: Ablation studies of ECL on **(a)** the training coefficient $\lambda$, **(b)** visual-token selection strategies, **(c)** the counterfactual masking ratio $K$, and **(d)** the masked-branch supervision weight $\mu$. Performance is measured by the average success rate (%) across 10 tasks in LIBERO-Core-LT using MiniVLA.

**Results and Analysis.** Table 2 reports higher macro-averaged success for ECL across all four platform-policy combinations. On AGILEX PIPER, success increases from 17.3% to 25.7% for MiniVLA and from 21.0% to 27.3% for $\pi_{0.5}$; on xArm6, it increases from 19.3% to 26.0% and from 24.0% to 30.7%, respectively. ECL improves tail tasks success rates over APA by an average of 9.0% on AGILEX PIPER and 4.3% on xArm6. APA Zhu et al. (2026) constructs augmented demonstrations through object replacement, for which its real-world pipeline proposes object localization using YOLOv8 Varghese & Sambath (2024), followed by image inpainting. ECL leverages existing training data without constructing object-replacement demonstrations. These results demonstrate ECL's effectiveness across policy models and robot platforms, highlighting its applicability to real-world robotic manipulation.

### 4.3 Ablation Studies

We conduct ablation studies to examine the contribution and sensitivity of the main design choices in ECL. Unless otherwise specified, the training and evaluation settings are the same as those in the simulation experiments.

**Training coefficient $\lambda$.** We examine $\lambda \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ for the head and tail task groups and measure performance in terms of average success rate. As shown in Fig. 4(a), success rate is highest at $\lambda = 0.3$ among the tested values, and decreases as $\lambda$ increases further. These results suggest that an appropriate weighting of the ECL objective benefits overall task performance, whereas an excessively large coefficient limits these gains.

**Visual-token selection.** To assess how the masked tokens are selected, we compare **Object-only**, **Action-only**, **Random**, and the full **Object–Action Relation** strategy at the same masking ratio. Further details are provided in Appendices C.The first two strategies use only object-related or action-related attention, respectively; Random masks tokens without using either relevance score. The full strategy ranks tokens by the product of the two scores. In Fig. 4(b), this joint strategy achieves the highest reported success rate 53.0%.

**Counterfactual masking ratio.** We vary the proportion of visual tokens selected for masking by the top-$K$ strategy over $\{0.025, 0.05, 0.075, 0.10, 0.125\}$ while keeping the token-ranking criterion fixed. As shown in Fig. 4(c), the highest average success rate is achieved at a masking ratio of 0.10, which we adopt in the main experiments. A smaller masking ratio may leave much of the selected evidence available, whereas a larger ratio may also remove information needed for action prediction.

**Masked-branch supervision weight $\mu$.** We investigate the effect of $\mu$ by varying its value over $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ while keeping all other experimental settings fixed, and report the average success rate across tasks. When $\mu = 0$, performance remains close to that of the standard long-tailed training baseline. Without direct supervision on the masked branch, $z^-$ may become uninformative, weakening its role as a counterfactual reference. In particular, if $z^-$ approaches zero, $z^{\text{effect}}$ becomes nearly identical to $z^+$, making counterfactual supervision largely redundant with factual prediction supervision. This provides a possible explanation for the limited improvement at $\mu = 0$ and highlights the importance of supervising the masked branch to obtain a meaningful counterfactual reference.

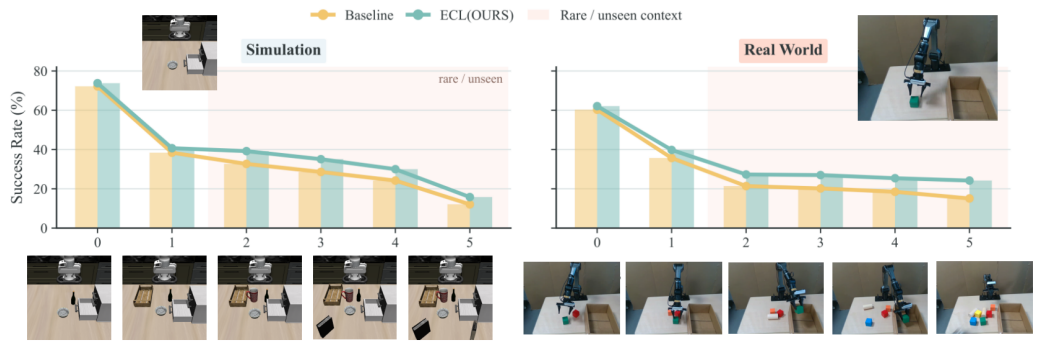



Figure 5: Robustness of ECL to context-level long-tailed distributions in simulation (left) and the real world (right). Contexts 1-5 correspond to the scenes below each plot. Yellow and teal lines and bars show success rates (%) for the LTC baseline and ECL respectively. Pink shading marks rare or unseen contexts (2-5).

### 4.4 Robustness to Context-Level Long-Tailed Distributions

The preceding experiments consider imbalance across different tasks. A related challenge is unequal coverage of scene configurations within a task. We therefore examine performance under context-level distribution shifts, where the task instruction and manipulation goal are held fixed while non-target objects vary.

To investigate this question, we construct a Long-Tailed Context (LTC) setting. We fix the task semantics, target object, and manipulation goal, while varying only the scene context. Specifically, we create scenes with different levels of complexity and control their numbers in the training set to induce a long-tailed distribution. This setting simulates the common situation in real-world robotic datasets where some contexts are well covered while low-frequency contexts have limited samples.We conduct experiments in both simulation and the real world using the MiniVLA policy, with the original LTC serving as the baseline. Detailed definitions of the context configurations and their corresponding training-sample counts are provided in Appendix A.

The results in Fig. 5 show that success rates generally decline as evaluation moves toward tail contexts, while ECL consistently outperforms the baseline in both simulation and real-world.ECL increases average success from 34.7% to 37.6% in simulation and from 28.5% to 34.3% in the real world.These results demonstrate that ECL effectively alleviates performance degradation caused by context distribution shifts, exhibiting strong generalization and decision robustness.

## 5 Conclusion, Limitations and Future Work

In this work, we investigate how long-tailed task distributions affect vision-language-action policies and identify weak task-specific visual grounding as a factor that leaves rare tasks susceptible to head action bias. Evidence-Guided Counterfactual Learning (ECL) identifies visual evidence jointly relevant to language instructions and action generation, uses this evidence to guide counterfactual observation construction through targeted token masking, and adapts the strength of counterfactual supervision to task frequency, encouraging task-specific grounding without requiring additional demonstrations or relation-level annotations.Therefore, we propose a lightweight, simple, and elegant solution to the long-tail problem in robotic policy learning. The source code and demonstrations are available in our project repository.

Despite its effectiveness, ECL introduces additional training overhead through counterfactual sample construction and extra forward passes. Under a fixed wall-clock training budget, this overhead creates an opportunity cost: computation devoted to improving tail-task performance could otherwise support further optimization of head tasks. Consequently, preserving head-task performance does not necessarily imply an optimal allocation of the available compute. Future work will explore more efficient counterfactual supervision and adaptive computation allocation to improve the trade-off between training cost and performance across the task distribution.

## 6 AI USE STATEMENT

During the preparation of this manuscript, an AI-based language tool was utilized to refine the prose and enhance overall clarity. The authors have rigorously examined all AI-generated suggestions and assume full accountability for the final submitted version, encompassing all textual content, scientific assertions, and research artifacts.

## REFERENCES


Suneel Belkhale and Dorsa Sadigh. Minivla: A better vla with a smaller footprint, 2024. URL https://github.com/Stanford-ILIAD/openvla-mini.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

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.

Silvia Chiappa. Path-specific counterfactual fairness. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 7801–7808, 2019.

Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *European conference on computer vision*, pp. 694–710, 2020.

Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 695–704. IEEE, 2021.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9260–9269, 2019.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2376–2384. PMLR, 09–15 Jun 2019.

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.

Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, 2019.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Shuang Li, Kailin Gong, Chenghao Liu, Yao Wang, Feng Qiao, and Xue Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5212–5221, 2021.

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.

Yanbiao Ma, Licheng Jiao, Fang Liu, Yuxin Li, Shuyuan Yang, and Xu Liu. Delving into semantic scale imbalance. *arXiv preprint arXiv:2212.14613*, 2022.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied AI. *IEEE Trans. Neural Networks Learn. Syst.*, 37:3031–3051, 2026.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pp. 607–617, 2020.

Open X-Embodiment Collaboration, Abbey O'Neill, Ahmad Rehman, Anish Maddukuri, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *IEEE International Conference on Robotics and Automation*, pp. 6892–6903, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.

Axel Sauer and Andreas Geiger. Counterfactual generative networks. In *International Conference on Learning Representations*, 2021.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 8769–8778. IEEE, 2018.

Rejin Varghese and M Sambath. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International conference on advances in data engineering and intelligent computing systems (ADICS)*, pp. 1–6. IEEE, 2024.

Jianfeng Wang, Thomas Lukasiewicz, Xiaolin Hu, Jianfei Cai, and Zhenghua Xu. Rsg: A simple but effective module for learning imbalanced datasets. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3783–3792, 2021.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952, 2023.

Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2361–2370. IEEE, 2021.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10795–10816, 2023.

Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6898–6907, 2022.

Junhong Zhu, Ji Zhang, Jingkuan Song, Lianli Gao, and Heng Tao Shen. Beyond the majority: Long-tail imitation learning for robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

# A Dataset Construction and Task Distribution

## A.1 Simulation Task-Level Long-Tailed Dataset

Following APA (Zhu et al., 2026), we use the same dataset construction protocol and Pareto-based task-frequency distribution to obtain LIBERO-Core-LT from the ten-task LIBERO-Core-FULL benchmark, built on LIBERO (Liu et al., 2023). Table 3 lists the task instructions and the demonstration counts used in our experiments. Tasks 0–2 form the head group, while all remaining tasks (3–9) form the tail group. The LT counts determine the frequency-aware correction strength in Eq. (8).

Table 3: LIBERO-Core task definitions and demonstration counts. "FULL" and "LT" denote the full and long-tailed training configurations, respectively. Task IDs follow the ordering in the project benchmark map.

| Task | Language instruction | FULL | LT |
|---|---|---|---|
| 0 | Pick up the black bowl next to the plate and place it on the plate | 46 | 46 |
| 1 | Pick up the black bowl next to the cookie box and place it on the plate | 47 | 28 |
| 2 | Pick up the black bowl on the cookie box and place it on the plate | 45 | 19 |
| 3 | Pick up the ketchup and place it in the basket | 42 | 15 |
| 4 | Pick up the alphabet soup and place it in the basket | 47 | 11 |
| 5 | Push the plate to the front of the stove | 39 | 9 |
| 6 | Put the bowl on top of the cabinet | 47 | 8 |
| 7 | Put the cream cheese in the bowl | 39 | 7 |
| 8 | Put the wine bottle on top of the cabinet | 45 | 6 |
| 9 | Put the wine bottle on the rack | 38 | 5 |
| | Total demonstrations | 435 | 154 |

## A.2 Real-World Task-Level Long-Tailed Dataset

We define a six-task manipulation benchmark for real-world evaluation on AGILEX PIPER and xArm6. The tasks involve grasping and placing objects onto supports or into receptacles. The AGILEX PIPER demonstrations contain synchronized wrist-camera and global-camera views, whereas the xArm6 demonstrations contain global-camera views. For each robot, the FULL and LT demonstration counts follow the per-task allocations in Table 4. Tasks are ordered by decreasing demonstration count in the long-tailed configuration (LT). The reference configuration (FULL) allocates 45–50 demonstrations per task, whereas LT uses 50, 46, 29, 13, 10, and 9 demonstrations, respectively. The resulting allocation contains 285 demonstrations for FULL and 157 for LT, with an LT maximum-to-minimum frequency ratio of approximately $5.56{:}1$.

Table 4: Real-world task definitions and demonstration allocation. FULL denotes the approximately balanced reference configuration, and LT denotes the long-tailed configuration. Task IDs follow the order used in the real-world success-rate table.

| Task | Language instruction | FULL | LT |
|---|---|---|---|
| 0 | Pick up the block and place it in the box | 50 | 50 |
| 1 | Stack the red block on top of the blue block | 48 | 46 |
| 2 | Stack the two blocks on the left on top of the block on the right | 47 | 29 |
| 3 | Insert the three-prong plug into the socket | 49 | 13 |
| 4 | Push the box to the designated location | 46 | 10 |
| 5 | Pick up the water bottle and place it in the box | 45 | 9 |
| | Total demonstrations | 285 | 157 |

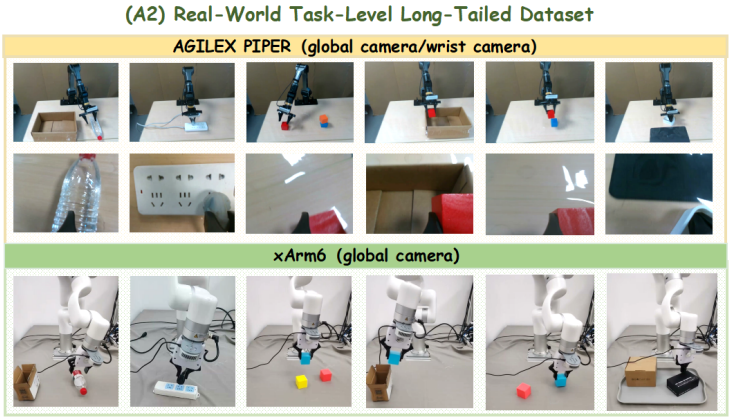



Figure 6: Examples from the real-world task-level long-tailed dataset. AGILEX PIPER provides global-camera and wrist-camera observations, while xArm6 provides global-camera observations. The columns show the six manipulation tasks listed in Table 4.




Figure 7: Examples from the Long-Tailed Context (LTC) dataset. Columns correspond to Scenes 0–5 in Table 5; the top and bottom rows show the global-camera and wrist-camera observations, respectively.

### A.3 Long-Tailed Context (LTC) Dataset

The Long-Tailed Context (LTC) dataset used in Section 4.4 keeps the language instruction and manipulation goal fixed while varying the scene configuration. Scene IDs 0–5 denote increasingly rare context configurations. This dataset is collected on AGILEX PIPER and contains synchronized wrist-camera and global-camera views. Figure 7 visualizes the six scene configurations and both camera views. Table 5 reports the balanced FULL allocation and the long-tailed LT allocation.

Table 5: Context-level task definition and demonstration counts used in Section 4.4. All scenes share the same language instruction. FULL contains 50 demonstrations per scene, while LT progressively reduces the number of demonstrations for rarer scene configurations.

| Language instruction | Scene | FULL | LT |
|---|---|---|---|
| Put the green block into the box | 0 | 50 | 50 |
| | 1 | 50 | 25 |
| | 2 | 50 | 20 |
| | 3 | 50 | 15 |
| | 4 | 50 | 10 |
| | 5 | 50 | 5 |
| | Total demonstrations | 300 | 125 |

# B EXPERIMENTAL ROLLOUT RESULTS

This appendix complements the aggregate success rates with representative closed-loop trajectories. Section B.1 contains one rollout for each of two representative tasks and each policy backbone (four comparisons in total). Section B.2 reports real-world results on AGILEX PIPER and xArm6. The simulation tables show frames sampled at fixed evaluation steps. The real-world tables use the selected PIPER global-camera and wrist-camera frames and the selected xArm6 global-camera frames.

## B.1 SIMULATION ROLLOUTS

**Evaluation configuration.** Table 6 fixes the protocol used for both backbones. Each task is evaluated for 675 episodes (225 episodes for each of three seeds), and every episode terminates at success, failure, or the fixed evaluation horizon. For $\pi_{0.5}$, we use the official OpenPI LIBERO configuration: a 10-step action horizon, batch size 256, learning rate $5 \times 10^{-5}$, 10k warm-up steps, 30k optimization steps, and EMA decay 0.999. The evaluation budget, seeds, and stopping rule are identical for MiniVLA and $\pi_{0.5}$ so that the rollout comparisons use the same protocol.

Table 6: Configuration for the simulation rollout evaluation.

| Backbone | Rollout limit | Episodes/task | Seeds | Episodes/seed | Action horizon |
|---|---|---|---|---|---|
| MiniVLA | 350 steps | 225 | 7, 14, 21 | 75 | — |
| $\pi_{0.5}$ | 350 steps | 225 | 7, 14, 21 | 75 | 10 actions |

Table 7: MiniVLA rollout for Task 3, "Pick up the ketchup and place it in the basket."

| Method | Simulation step | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 50 | 100 | 150 | 200 | 250 | 300 | 350 |
| FULL |  | | | | | | | |
| LT | | | | | | | | |
| APA | | | | | | | | |
| ECL | | | | | | | | |

*Source:* Official OpenPI $\pi_{0.5}$ LIBERO training configuration.

Table 8: MiniVLA rollout for Task 6, “Push the plate to the front of the stove.”

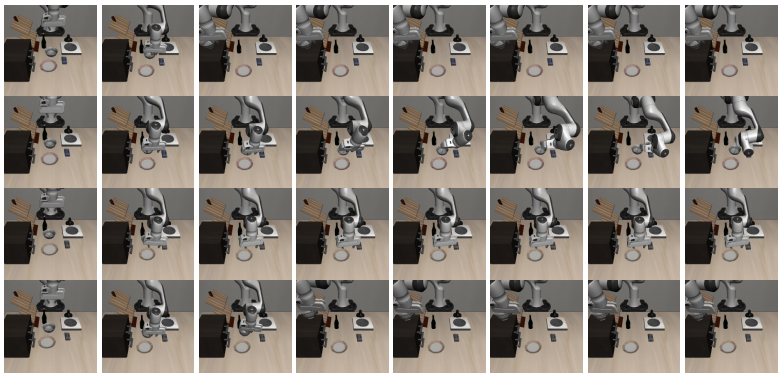
| Method | Simulation step | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 50 | 100 | 150 | 200 | 250 | 300 | 350 |
| FULL |  | | | | | | | |
| LT | | | | | | | | |
| APA | | | | | | | | |
| ECL | | | | | | | | |

Table 9: $\pi_{0.5}$ rollout for Task 8, “Put the wine bottle on top of the cabinet.”

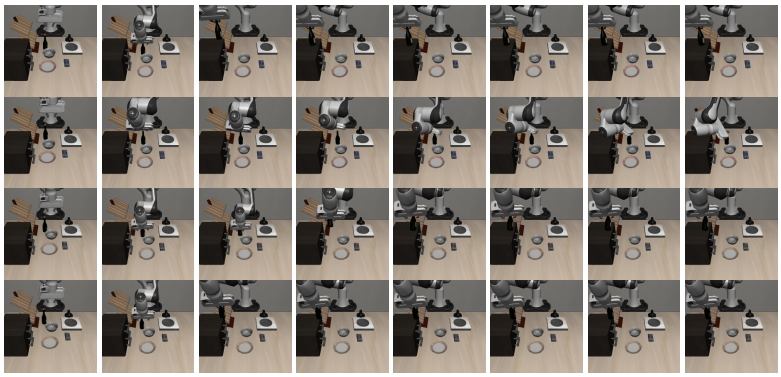
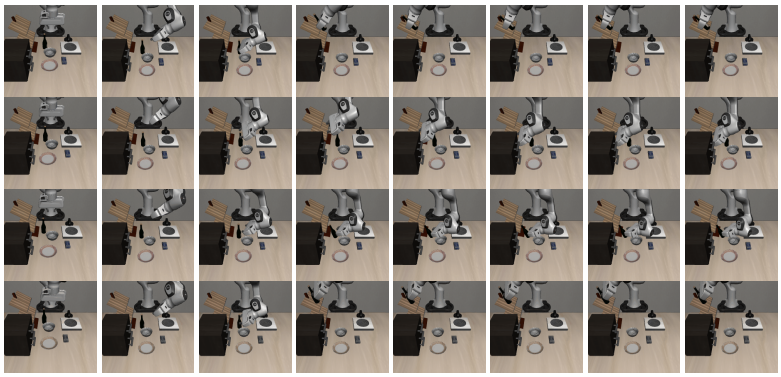
| Method | Simulation step | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 50 | 100 | 150 | 200 | 250 | 300 | 350 |
| FULL |  | | | | | | | |
| LT | | | | | | | | |
| APA | | | | | | | | |
| ECL | | | | | | | | |

Table 10: $\pi_{0.5}$ rollout for Task 9, “Put the wine bottle on the rack.”

| Method | Simulation step | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 50 | 100 | 150 | 200 | 250 | 300 | 350 |
| FULL |  | | | | | | | |
| LT | | | | | | | | |
| APA | | | | | | | | |
| ECL | | | | | | | | |

### B.2 Real-World Rollouts

**Evaluation configuration.** We evaluate MiniVLA and $\pi_{0.5}$ separately on AGILEX PIPER and xArm6. Each policy–robot pair is tested for 50 trials per task with randomized initial object poses. PIPER uses synchronized global-camera and wrist-camera inputs, whereas xArm6 uses a global camera. Consistent with the real-world study in Section 4.2, this part compares only APA and ECL. Each real-world rollout uses eight frames sampled uniformly over its normalized duration, from 0% (start) to 100% (final observation). This axis supports comparison across robots even when their control rates differ.

Table 11: Configuration for the real-world rollout evaluation.

| Robot | Visual observations | Backbones | Comparison / trials |
|---|---|---|---|
| AGILEX PIPER | Global camera + wrist | MiniVLA, $\pi_{0.5}$ | APA vs. ECL / 50 per task |
| xArm6 | Global camera | MiniVLA, $\pi_{0.5}$ | APA vs. ECL / 50 per task |

#### B.2.1 AGILEX PIPER

Table 12: AGILEX PIPER rollout with MiniVLA (global-camera and wrist-camera views).

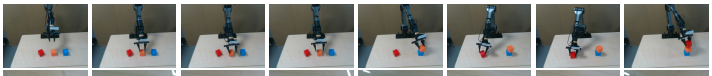
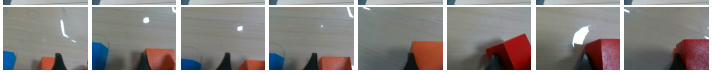
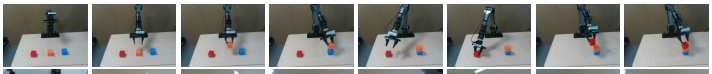
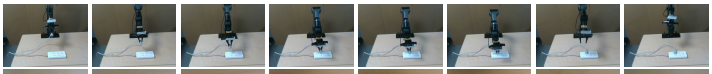
| Method | View | Normalized rollout progress | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 0% | 14% | 29% | 43% | 57% | 71% | 86% | 100% |
| **APA** | Global camera |  | | | | | | | |
| | Wrist |  | | | | | | | |
| **ECL** | Global camera |  | | | | | | | |
| | Wrist |  | | | | | | | |

Table 13: AGILEX PIPER rollout with $\pi_{0.5}$ (global-camera and wrist-camera views).

| Method | View | Normalized rollout progress | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 0% | 14% | 29% | 43% | 57% | 71% | 86% | 100% |
| **APA** | Global camera |  | | | | | | | |
| | Wrist |  | | | | | | | |
| **ECL** | Global camera |  | | | | | | | |
| | Wrist |  | | | | | | | |

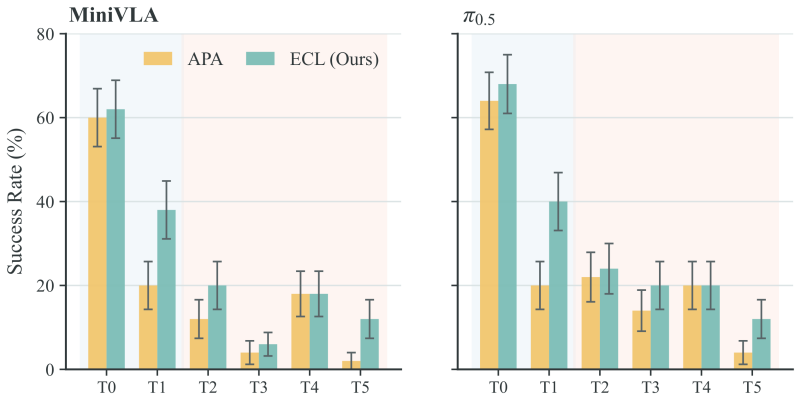



Figure 8: xArm6 per-task success rates for MiniVLA and $\pi_{0.5}$, comparing APA with ECL.

### B.2.2 xArm6

Table 14: xArm6 rollout comparison with MiniVLA.

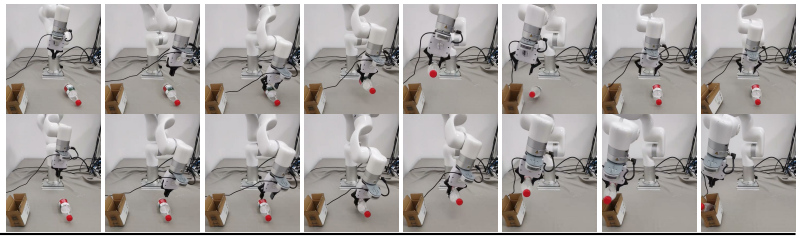
| Method | Normalized rollout progress | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0% | 14% | 29% | 43% | 57% | 71% | 86% | 100% |
| **APA** |  | | | | | | | |
| **ECL** |  | | | | | | | |

Table 15: xArm6 rollout comparison with $\pi_{0.5}$.

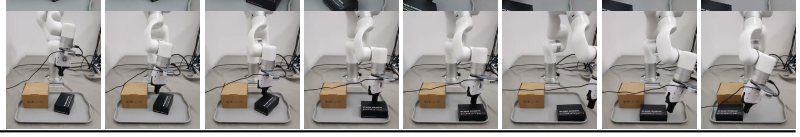
| Method | Normalized rollout progress | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0% | 14% | 29% | 43% | 57% | 71% | 86% | 100% |
| **APA** |  | | | | | | | |
| **ECL** |  | | | | | | | |

Figure 8 reports the per-task success rates on xArm6. Both xArm6 rollout tables use the same time axis as the PIPER tables. The final version will pair each rollout with its task instruction and outcome. These examples illustrate controller behavior; the quantitative claims remain based on the complete success-rate evaluation.

## C Comparison of Mask Selection Strategies

Figure 9 compares four patch-selection strategies on the same observation from the task "put the wine bottle on the rack." Each panel selects 26 of the $16 \times 16 = 256$ visual patches, corresponding to the nominal 10% masking budget. Keeping the observation and patch budget fixed makes the spatial differences between selection rules visible.

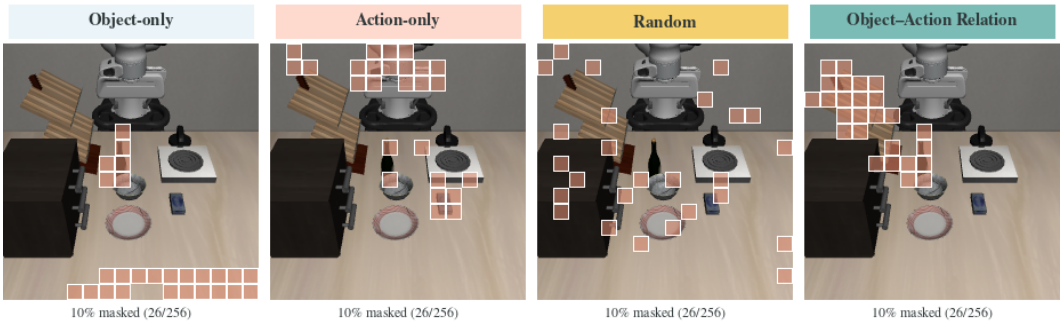



Figure 9: **Illustrative comparison of mask selection strategies.** From left to right: object-only, action-only, random, and object–action relation selection. Peach overlays with white borders mark selected patches; all panels select 26/256 patches. The supplied visualization uses semantic-mask and saved-attention proxies, as described below, rather than a direct export of the complete ECL selector.

**Reading the comparison.** Object-only selection uses target-object coverage; action-only selection uses a saved patch-attention map; random selection samples patches without task information. The relational illustration combines object coverage and attention, with an additional goal-region contribution. Consequently, random patches are dispersed across the image, while the relational panel includes regions around the bottle and rack. These panels illustrate why the spatial location of a fixed masking budget matters when constructing an evidence-limited reference.

## D Effectiveness of Visual Evidence in ECL

The purpose of this analysis is to verify whether ECL can learn the correct visual evidence for under-represented tail tasks. As discussed in the Introduction, tail-task failures are closely associated with weak task-specific visual grounding: without reliable evidence for the current task, a shared policy is more likely to rely on action patterns learned from frequent head tasks. We therefore visualize two LT tail tasks—"put the bowl on top of the cabinet" and "put the wine bottle on the rack"—and compare the LT baseline with ECL.

Figure 10 shows that the LT baseline often assigns attention to the workspace background or regions unrelated to the instruction. In contrast, ECL consistently focuses on the visual evidence required to complete the task, including the instructed object, the end-effector–object interaction, and the target location. These results show that ECL does more than improve the final success rate: it enables the policy to learn task-specific visual evidence from scarce tail-task demonstrations and use that evidence to support action prediction.

This is the central motivation for *Evidence-Guided Counterfactual Learning*. Instruction- and action-related signals identify the visual evidence that guides the counterfactual intervention. Training on the resulting intact–counterfactual difference then strengthens the association among the instruction, the correct visual evidence, and the predicted action. In this way, ECL explicitly encourages tail tasks to recover reliable visual grounding, which also reduces their dependence on head-task action bias.

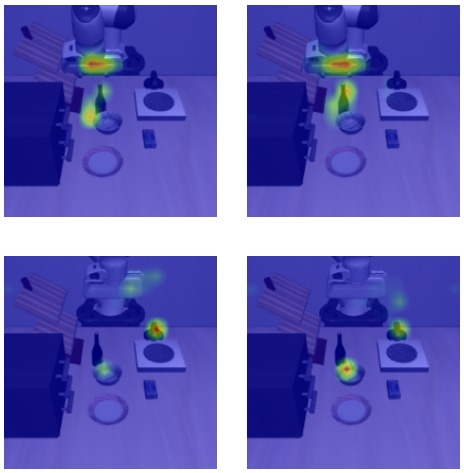


Figure 10: **Visual evidence on representative LT tail tasks.** Each square shows a paired comparison with the LT baseline on the left and ECL on the right. The top row shows "put the bowl on top of the cabinet," and the bottom row shows "put the wine bottle on the rack." Relative to the LT baseline, ECL more consistently grounds its decisions in instruction-relevant objects, interaction regions, and goal locations, supporting our finding that ECL effectively learns the correct task-specific visual evidence for tail-task decision making.

## E Training Cost Details

**Recorded training timings.** Table 16 summarizes the supplied training logs. For a log containing $M$ timing records $t_j$, we report the arithmetic mean $\bar{t} = M^{-1} \sum_{j=1}^{M} t_j$, the median, and the accumulated recorded time $T_{\log} = \sum_{j=1}^{M} t_j/3600$ in hours. All valid records are retained, including initialization and unusually slow steps. The timer fields are `VLA Train/Step Time` for FULL, LT, and APA, and `seconds` for ECL. Their measurement boundaries cannot be verified from the logs alone, so $T_{\log}$ is not interpreted as end-to-end training time: unrecorded data preparation, checkpointing, evaluation, and restart intervals are not added.

Table 16: Recorded MiniVLA training cost. GPU counts and batch sizes are taken from saved configurations. Batch denotes per-device/global batch size. Logged hours cover only the supplied timing records, which span different training intervals; these rows are not a controlled overhead comparison.

| Method | Data size | GPUs | Batch | Logged steps | Mean (s) | Median (s) | Logged (h) |
|---|---|---|---|---|---|---|---|
| FULL | 55,966 | 1 | 20/20 | 100,738 | 0.692 | 0.660 | 18.41 |
| LT (Baseline) | 19,234 | 1 | 20/20 | 34,622 | 0.608 | 0.570 | 5.89 |
| APA | 21,049 | 1 | 20/20 | 37,888 | 0.673 | 0.651 | 7.21 |
| ECL (Ours) | 19,234 | 1 | 20/20 | 34,622 | 1.068 | 1.090 | 10.27 |

**Cost of the counterfactual branch.** Within the ECL run, the first 1,923 steps have no active counterfactual samples, consistent with the configured two-epoch warm-up. Their mean recorded time is 0.519 s/step. The subsequent 32,698 steps activate the counterfactual branch and average 1.100 s/step, with a median of 1.090 s/step. The ratio of these phase means is approximately $2.12\times$. This is a within-run observation across different training phases, not a matched measurement against the LT baseline. These operations accompany the original and masked predictions and a separate no-gradient selector pass in the training procedure; their individual timing contributions are not separately logged.

**Data volume, training time, and performance trade-off.** The LT baseline uses 19,234 samples and has the lowest mean step time (0.608 s). APA expands the dataset to 21,049 samples, an increase of 1,815 samples (9.4%), while also increasing the logged steps from 34,622 to 37,888 and the mean step time to 0.673 s. Consequently, its recorded training time rises from 5.89 to 7.21 hours. In addition, producing the augmented APA samples requires object detection, image inpainting, and data validation; this time-consuming data construction is not included in the logged hours reported in Table 16. ECL uses exactly the same 19,234 samples and 34,622 logged steps as LT, but its mean step time increases to 1.068 s and its recorded training time to 10.27 hours. This increase is explained by the additional intact and counterfactual predictions and the no-gradient evidence selection performed during training. Although ECL therefore incurs a higher per-step training cost, it requires no additional demonstrations or augmented data, and the additional computation is confined to training. Considering the success-rate improvements in Table 1, the substantial time required to construct APA's extra data, and the absence of additional deployment-time computation, ECL provides the preferable overall balance between policy performance and training cost.