ConfAL-WM: Confidence-Guided Active Learning for Action-Conditioned World Models

Xiang Liu
xiang-liu25@mail.tsinghua.edu.cn

Sen Cui‡
cuis@mail.tsinghua.edu.cn

Changshui Zhang†
zcs@mail.tsinghua.edu.cn

Tsinghua University ‡ Project Leader & Corresponding Author † Corresponding Author

Abstract

Action-conditioned world models have become an important foundation for embodied prediction, planning, and synthetic data generation, but their errors under new task and scene distributions are often concentrated in localized spatiotemporal regions such as robot arms, manipulated objects, contact areas, and occluded objects. This paper presents **ConfAL-WM**, a confidence-guided active learning framework for post-training embodied world models. Built upon EVAC, we attach a lightweight confidence probe to UNet decoder features and predict dense confidence maps in the latent space. These maps are aggregated into task-, frame-, and patch-level scores, enabling both efficient data selection and localized training enhancement. Our pipeline first retrains the confidence probe and warms up EVAC with a small subset of target-domain data, then performs task-level prescreening to allocate sampling budgets, and finally applies selected-data retraining with optional frame or patch weighted data enhancement. Experiments on RoboTwin2.0 show that confidence-guided selection improves post-training efficiency, while dense frame and patch weighting further enhances prediction quality and embodied trajectory consistency compared with scalar reward, progress, and judge-based scoring baselines. A quick visual overview of this work is available at https://ConfAL-WM.github.io.

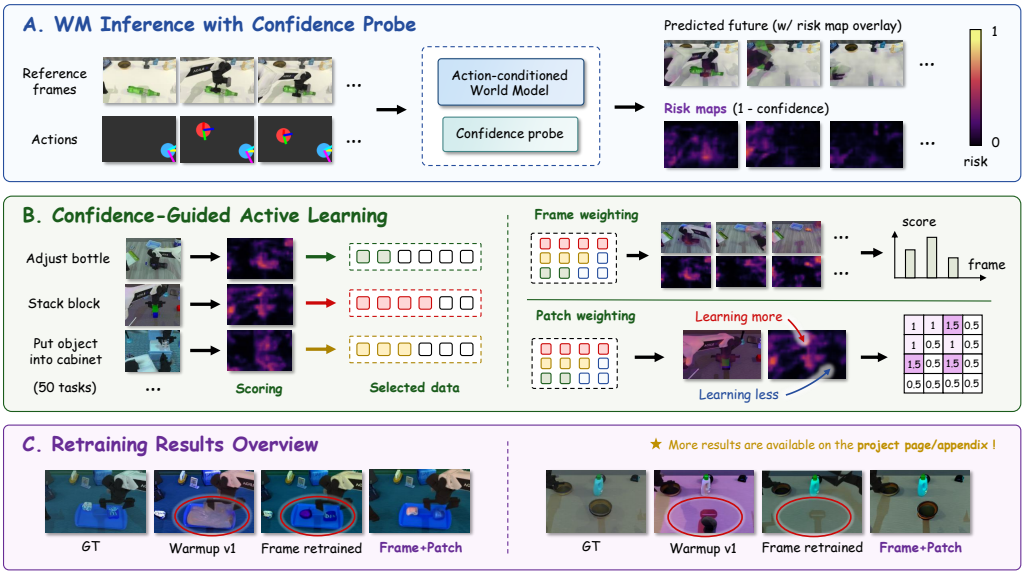


Figure 1: **An introduction of our work.** (A) Given reference frames and action conditions, we use a UNet-based latent diffusion world model as the backbone and train a confidence probe on decoder features to predict dense future risk maps. (B) Confidence guides active learning in two ways: selecting more informative post-training data and enhancing supervision on unreliable frames or patches. (C) Two representative episodes compare the same predicted frame from the ground truth, warmup model, frame-weighted retraining, and frame-and-patch-weighted retraining.

1 Introduction

Action-conditioned embodied world models aim to predict future visual observations under given robot actions, enabling offline policy evaluation, synthetic data generation, and planning without direct environment interaction. EnerVerse-AC (EVAC) is a representative action-conditioned world model that generates future multi-view observations conditioned on robot actions (Jiang et al., 2025). However, whether under zero-shot transfer to new datasets or during post-training on existing domains, prediction errors are rarely uniformly distributed over the whole video. They often concentrate around moving robot arms, manipulated objects, contact regions, occlusions, and long-horizon interaction errors. Recent studies on confidence-aware video generation and physics-reinforced world simulation also suggest that unreliable or physically inconsistent regions are usually spatially and temporally localized (Mei et al., 2025b; Zhang et al., 2026). This motivates a post-training pipeline that does not merely add more data globally, but selects and enhances data according to where the world model is likely to fail.

In this work, we study confidence-guided active learning for action-conditioned world models, and refer to the overall framework as **ConfAL-WM**. We build upon EVAC as the backbone world model and attach a lightweight confidence probe to its UNet decoder features. Unlike prior DiT-based dense confidence estimation, where each latent token naturally corresponds to a video patch (Mei et al., 2025b), EVAC requires an explicit feature-tapping design. We use decoder features rather than the bottleneck features, because they retain stronger spatial locality while still carrying global contextual information from the latent diffusion backbone. The probe predicts dense confidence maps in the latent space, which are further aggregated into patch-, frame-, and task-level scores for selection and retraining. To stabilize probe supervision, we also replace the random binary threshold range used in previous confidence training with an adaptive EMA-based thresholding strategy.

Our active learning pipeline uses confidence in a staged and budget-aware manner. EVAC is originally pretrained on AgiBot World, a large-scale real-world manipulation dataset (Bu et al., 2025), and we use RoboTwin2.0 as the post-training dataset to evaluate the effectiveness of active learning under a new task and scene distribution (Chen et al., 2025). First, a small subset of data is used to retrain the confidence probe and warm up EVAC, producing a domain-adapted EVAC-v1. Second, we perform fast task-level prescreening with the confidence probe: harder tasks are assigned larger sampling budgets, while easier tasks receive fewer selected scenes. Third, after scene selection, there are two possible retraining paths. The selected data can be directly used for EVAC-v2 retraining, or it can be further processed by EVAC-v1 and the confidence probe to obtain frame- and patch-level confidence scores. The latter path introduces additional inference and scoring cost, but enables confidence-guided data enhancement during retraining.

This design gives confidence two roles. At the selection level, it estimates which tasks and scenes deserve more retraining budget. At the training level, it localizes which frames and patches should receive stronger supervision. This differs from scalar reward, progress, or judge-based scoring models, which can rank trajectories or frames but usually cannot provide patch-level prediction risk for world-model retraining (Lee et al., 2026; Liang et al., 2026; Ma et al., 2024; Ji et al., 2026). In our comparison, these scoring models are used as alternative selection signals, and part of them are further equipped with frame-weighted retraining. Confidence is used as our main signal, and we additionally study frame-and-patch-weighted retraining as a stronger dense enhancement strategy. Following the EWMBench-style evaluation protocol (Yue et al., 2025), our results show that confidence-guided selection and confidence-guided frame/patch weighting consistently improve prediction quality and embodied trajectory consistency over scalar scoring baselines.

Our contributions are summarized as follows:

1. We propose ConfAL-WM, a staged confidence-guided active learning pipeline for efficient post-training of embodied action-conditioned world models.
2. We adapt dense confidence estimation to a UNet-based action-conditioned world model by designing a decoder-feature confidence probe and a more stable EMA-based training target.
3. We introduce confidence-guided data enhancement, especially frame-and-patch-weighted retraining, to focus supervision on unreliable spatiotemporal regions.
4. We show that confidence improves both data selection and weighted retraining compared with existing scalar reward, progress, and judge-based scoring methods.

2 Related Work

Action-conditioned world models and evaluation. Action-conditioned world models predict future observations under robot actions. EnerVerse-AC (EVAC), pretrained on AgiBot World, is the UNet-based backbone of our framework (Jiang et al., 2025; Bu et al., 2025); we post-train it on RoboTwin2.0 under new tasks, scenes, and robot embodiments (Chen et al., 2025). Recent alternatives improve action controllability or latent prediction, including IRASim and DINO-WM (Zhu et al., 2025; Zhou et al., 2025). Most closely related, Mei et al. (2025b) introduce C^3 for dense confidence estimation in DiT-based video models.

EWMBench evaluates embodied world models through reconstruction, scene, motion, and semantic metrics, and provides our main evaluation protocol (Yue et al., 2025). Other benchmarks connect generated futures with policy performance or assess instruction following and physical plausibility (Jang et al., 2025; Li et al., 2025b; NVIDIA, 2025).

Dense confidence, reward, progress, and judge signals. Dense uncertainty is the closest signal to our method. S^3 analyzes uncertainty in generative video models, while C^3 introduces dense calibrated confidence maps for controllable video generation (Mei et al., 2025a;b). PRM-as-a-Judge provides process-level robotic auditing with macro- and micro-level signals (Ji et al., 2026). Our confidence additionally serves as an acquisition score and a local training weight.

We compare against several scalar scoring methods. GVL estimates temporal progress through in-context value learning (Ma et al., 2024); RoboReward learns general-purpose vision-language rewards for robotics (Lee et al., 2026); Robometer combines frame-level progress with trajectory preferences (Liang et al., 2026); and LRMs generate process and completion rewards online (Wu et al., 2026). These methods score trajectories or frames, whereas we also localizes patch errors.

Localized supervision has recently been explored from complementary perspectives. CD-LAM reduces action-irrelevant latent bias through embodiment-focused reconstruction and action-aware objectives (Wei et al., 2026). PhysisForcing strengthens physical consistency by emphasizing physics-informative interaction regions (Zhang et al., 2026). Our method identifies such regions through world-model confidence and uses them for active selection and patch-weighted retraining.

Active learning, uncertainty-aware robot learning, and auxiliary world-model objectives. Classical active learning selects samples using uncertainty, disagreement, or coverage (Settles, 2009; Seung et al., 1992; Sener & Savarese, 2018). Robotics-specific methods extend these principles to view selection, rollout control, and demonstration curation (Eren & Oztop, 2024; Dasgupta et al., 2024; Li et al., 2025a; Dass et al., 2025). Most closely related, Römer et al. (2026) derive velocity-field disagreement (VFD) to quantify epistemic uncertainty in flow-based VLAs, and further introduce SAVE, which uses this uncertainty to prioritize tasks and initial scenes for active multitask fine-tuning. This shares our goal of using model uncertainty to reduce adaptation cost and focus supervision on informative data, but operates on action-policy uncertainty, whereas our method estimates dense video-prediction confidence and uses it for both data selection and localized world-model retraining. Future-aware auxiliary objectives can also improve embodied representations (Yuan et al., 2026; Lv et al., 2026; Luo et al., 2026).

3 Method

In this section, we first introduce a dense confidence probe for a UNet-based action-conditioned world model, which estimates patch-level prediction reliability from intermediate decoder features. We then describe how the resulting dense risk maps are aggregated into task-level acquisition scores and further converted into weights for confidence-guided EVAC retraining.

3.1 Dense Confidence Probe for UNet World Models

Confidence definition. Given an action-conditioned world model, we define confidence as the predicted probability that a local prediction error is below a threshold. In EVAC, the diffusion model predicts a denoising direction $\hat{v}^{(\tau)}$ for the ground-truth target $v^{(\tau)}$. For a predicted future frame t

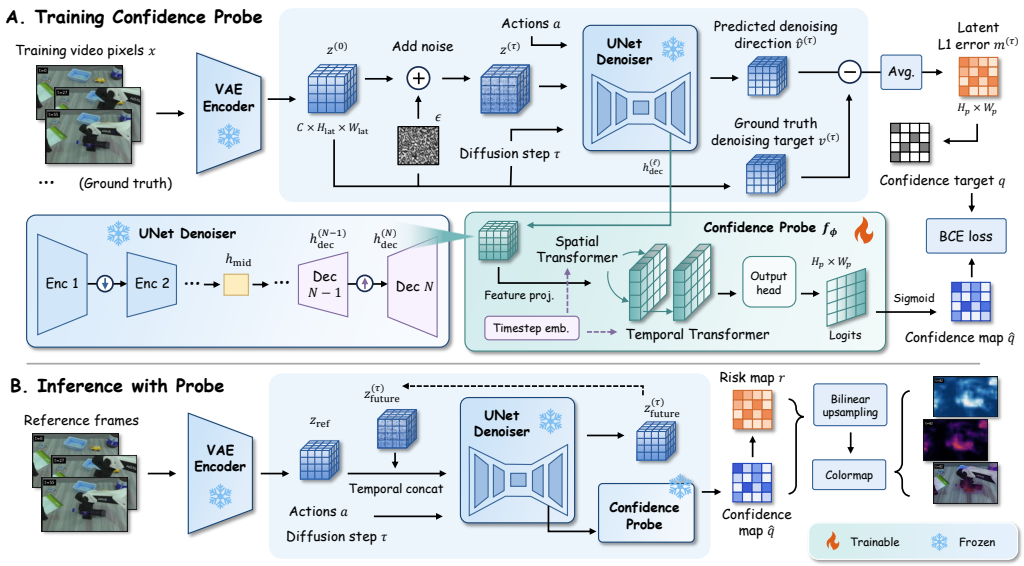


Figure 2: **Training and inference of the confidence probe in the UNet latent diffusion world model.** Video pixels are encoded into latent variables, diffused into $z^{(\tau)}$, processed by the UNet denoiser, and supervised by the denoising target $v^{(\tau)}$. The confidence probe is attached to selectable decoder features $h_{\text{dec}}^{(\ell)}$ and predicts dense confidence maps.

and latent patch (i, j), we compute a **local mean absolute error** as

$$m_{t,(i,j)}^{(\tau)} = \frac{\sqrt{1-\bar{\alpha}_\tau}}{|\mathcal{P}_{i,j}|} \sum_{p \in \mathcal{P}_{i,j}} \left| \hat{v}_{t,p}^{(\tau)} - v_{t,p}^{(\tau)} \right| = \frac{1}{|\mathcal{P}_{i,j}|} \sum_{p \in \mathcal{P}_{i,j}} \left| \hat{z}_{t,p}^{(\tau)} - z_{t,p}^{(0)} \right|. \tag{1}$$

Here, $p = (c, x, y)$ indexes one channel-spatial entry in the latent tensor: c is the latent feature channel and (x, y) is a spatial location in the latent grid. The set $\mathcal{P}_{i,j}$ denotes the latent region corresponding to probe patch (i, j). $z_{t,p}^{(0)}$ is the clean video latent encoded from the ground-truth video, and $\hat{z}_{t,p}^{(\tau)}$ is its closed-form prediction recovered from $\hat{v}_{t,p}^{(\tau)}$ at diffusion timestep τ. This equivalent form avoids executing a complete reverse diffusion trajectory when constructing the confidence target. $\bar{\alpha}_\tau$ is the cumulative noise-schedule coefficient at timestep τ.

The **binary confidence target** at frame t is defined as

$$q_t = \{q_{t,(i,j)}\}_{i,j} \in \{0,1\}^{H_p \times W_p}, \qquad q_{t,(i,j)} = \mathbb{I}\{m_{t,(i,j)}^{(\tau)} < \theta_t\}, \tag{2}$$

where θ_t is a stochastic threshold sampled from an adaptive interval. Further architectural details of the latent-space error construction are provided in Appendix A.1.

Decoder-feature confidence probe. Let $h_{\text{dec}}^{(\ell)}$ denote the feature map extracted from the ℓ-th UNet decoder block. For each predicted future frame, the **confidence probe prediction** is

$$\hat{q}_t = \sigma\Big(f_\phi\Big(h_{\text{dec}}^{(\ell)}, e_\tau, e_\theta\Big)\Big) \in [0,1]^{H_p \times W_p}. \tag{3}$$

where e_τ denotes the diffusion-timestep embedding, e_θ denotes the sampled error threshold embedding, f_ϕ is the trainable confidence probe parameterized by ϕ, and $\sigma(\cdot)$ is the element-wise sigmoid function. Internally, the probe applies channel projection, spatial Transformer layers, temporal Transformer layers, and a patch-wise output head. The decoder index ℓ allows the probe to trade coarse semantic context for finer spatial resolution, as shown in Figure 3.

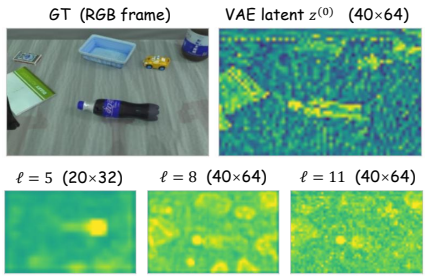


Figure 3: **Direct comparison for a single frame.** All operations are performed in the VAE latent space. The three below visualize the channel-averaged $\mathbb{E}_c\left[\left|h_{\text{dec}}^{(\ell)}\right|\right]$.

EMA-calibrated random thresholding. At training step s, we compute the lower and upper percentiles of the batch-wise local error distribution, and we update the threshold bounds as

$$p_l^{(s)} = \mathrm{Quantile}_{0.1}\left(m^{(s)}\right), \qquad p_h^{(s)} = \mathrm{Quantile}_{0.9}\left(m^{(s)}\right), \tag{4}$$

$$l^{(s)} = (1-\gamma_s)l^{(s-1)} + \gamma_s p_l^{(s)}, \qquad h^{(s)} = (1-\gamma_s)h^{(s-1)} + \gamma_s p_h^{(s)}. \tag{5}$$

The update rate γ_s is larger during learning-rate warmup and becomes substantially smaller afterwards, enabling rapid initialization followed by stable online calibration.

A separate threshold is sampled for each predicted future frame t as $\theta_t \sim \mathcal{U}\left(l^{(s)}, h^{(s)}\right)$. The sampled threshold θ_t is shared by all spatial patches (i, j) within the same frame. Further implementation details and the conditional interpretation of θ_t are provided in Appendix A.2.

Probe training objective. The probe is trained with binary cross-entropy over all supervised patches in the predicted future frames:

$$\mathcal{L}_{\mathrm{conf}} = -\mathbb{E}_{t,(i,j)}\Big[q_{t,(i,j)} \log \hat{q}_{t,(i,j)} + (1 - q_{t,(i,j)}) \log(1 - \hat{q}_{t,(i,j)})\Big], \tag{6}$$

During probe training, the EVAC backbone remains frozen and gradients are propagated only through f_ϕ. At inference time, the frame-wise dense risk map is defined as $r_{t,(i,j)} = 1 - \hat{q}_{t,(i,j)}$.

3.2 Confidence-Guided Active Learning

Full pipeline. Given a candidate data pool, our active learning pipeline uses confidence for both sample acquisition and localized retraining. First, we train the confidence probe and warm up EVAC on a small subset of the data pool. The resulting EVAC-v1 is used for task-level prescreening, where representative episodes estimate task difficulty and determine the per-task candidate quota, according to which episodes are randomly sampled within each task.

Second, the resulting candidate subset can then be directly used for selection-only EVAC-v2 retraining. Alternatively, we perform an additional inference and confidence-scoring stage on the selected episodes, and use the resulting confidence maps to weight the EVAC-v2 retraining objective.

Risk-based selection criteria. We consider three confidence-based selection criteria: **1. mean risk**, **2. tail risk**, and **3. persistent risk**. Mean risk measures the overall prediction difficulty by averaging the dense risk map over all spatial locations and predicted future frames, whereas tail risk emphasizes severe localized failures. For tail risk, we flatten all patch risks $\{r_{t,(i,j)}\}$ into a vector of length $K = TH_pW_p$ and sort its entries in ascending order as $r_{[1]} \le r_{[2]} \le \cdots \le r_{[K]}$. Let η denotes the selected high-risk tail ratio, the two scores are defined as

$$S_{\mathrm{mean}} = \mathbb{E}_{t,(i,j)}\left[r_{t,(i,j)}\right], \quad S_{\mathrm{tail}} = \frac{1}{\lceil \eta K \rceil} \sum_{n=K-\lceil \eta K \rceil+1}^{K} r_{[n]}. \tag{7}$$

Persistent risk captures high-risk local failures that recur across multiple frames. Let $K_p = H_pW_p$, for frame t, we sort its patch risks as $r_{t,[1]} \le \cdots \le r_{t,[K_p]}$ and average the largest fraction of them to obtain a frame-level score u_t. We then sort these frame scores as $u_{[1]} \le \cdots \le u_{[T]}$ and average the highest-scoring frames:

$$u_t = \frac{1}{\lceil \eta_p K_p \rceil} \sum_{n=K_p-\lceil \eta_p K_p \rceil+1}^{K_p} r_{t,[n]}, \quad S_{\mathrm{persistent}} = \frac{1}{\lceil \eta_t T \rceil} \sum_{n=T-\lceil \eta_t T \rceil+1}^{T} u_{[n]}. \tag{8}$$

Here, η_p denotes the fraction of high-risk patches retained within each frame, and η_t denotes the fraction of high-risk frames retained across the predicted video.

Confidence-guided frame and patch weighting. For weighted EVAC-v2 retraining, we perform an additional inference and confidence-scoring stage on the selected episodes. Frame weighting assigns the spatially averaged risk uniformly to all patches within frame t, whereas patch weighting retains the original dense risk $r_{t,(i,j)}$. We unify the two forms as

$$r_{t,(i,j)}^{(\alpha)} = \mathbb{E}_{(i',j')}\left[r_{t,(i',j')}\right] + \alpha\left(r_{t,(i,j)} - \mathbb{E}_{(i',j')}\left[r_{t,(i',j')}\right]\right), \qquad \alpha \in [0,1]. \tag{9}$$

When $\alpha = 0$, all patches within a frame receive the same frame-level risk; when $\alpha = 1$, the original patch-level risk map is recovered. Intermediate values preserve the frame-level baseline while introducing local residual modulation.

After quantile normalization and clipping, we denote the resulting risk by $\widetilde{r}^{(\alpha)}_{t,(i,j)} \in [0,1]$. It is converted into a local loss multiplier and applied to the world-model objective as

$$w_{t,(i,j)} = 1 + \lambda_{\text{eff}}(s)\widetilde{r}^{(\alpha)}_{t,(i,j)}, \qquad \mathcal{L}_{\text{WM}} = \frac{\mathbb{E}_{t,(i,j)}\left[w_{t,(i,j)}\ell_{t,(i,j)}\right]}{\mathbb{E}_{t,(i,j)}\left[w_{t,(i,j)}\right]}. \tag{10}$$

Here, $\ell_{t,(i,j)}$ is the local world-model training loss, and $\lambda_{\text{eff}}(s) = \lambda_{\text{conf}} \min(1, s/s_{\text{warm}})$ linearly increases the weighting strength during the first s_{warm} EVAC-v2 retraining steps. Thus, higher-risk frames and patches receive larger optimization weights. The confidence maps are detached during EVAC retraining, so gradients are propagated only through the world model.

4 Experiments

Our experiments are designed to answer two questions. **1. Why Confidence?** Is the confidence signal a meaningful indicator of world-model prediction errors? **2. Why Active Learning?** Can confidence-guided active learning improve post-training efficiency and final world-model quality?

4.1 Experimental Setup

Data setting. In our setting, EVAC is pretrained on AgiBot World (Bu et al., 2025), while active learning and post-training are conducted on a subset of RoboTwin2.0 (Chen et al., 2025). This creates a transfer setting involving new tasks, scenes, and robot embodiments. We use data from the Aloha-AgileX dual-arm robot, covering 50 manipulation tasks. Each task contains 500 randomized scenes, resulting in 24,992 videos in total. The video length ranges from 98 to 578 frames.

Implementation details. During confidence-probe training, the entire EVAC backbone is frozen and only the confidence probe f_ϕ is optimized. During EVAC-v1 warmup and EVAC-v2 retraining, the UNet denoiser and two projection modules are trainable, while the VAE, CLIP embedder, and action-conditioning resampler remain frozen. The full EVAC model has about 2.33B parameters, and the confidence probe has about 8.2M parameters.

We use 6,248 episodes, about 25% of the data, for confidence-probe training and EVAC-v1 warmup. The remaining 18,244 episodes form the candidate pool, from which 7,298 episodes are selected for EVAC-v2 retraining. We use AdamW with learning rate 5×10^{-5}, fp16 training, and 4,000 default optimization steps on two A800 GPUs. For confidence scoring, EVAC-v1 averages probe scores over three diffusion timesteps $\tau \in [50, 200]$. The main active-learning comparison is repeated with three random seeds, 42, 3407, and 123, and the main text reports the mean across these runs. Detailed per-seed results and bootstrap statistics are provided in Appendix B.2.

Baselines. For evaluating confidence quality, directly comparing with the original C^3 is not fully fair because it is designed for DiT-style video world models (Mei et al., 2025b). Our main comparisons focus on the scoring method of active learning: RoboReward[1], a general-purpose vision-language reward model for robotics (Lee et al., 2026); GVL[2], which uses VLMs as in-context value learners (Ma et al., 2024); Robometer-Prog and Robometer-Pref[3], trajectory progress and preference models (Liang et al., 2026); PRM-as-Judge[4], a dense process-level robotic auditing method (Ji et al., 2026); and LRMs[5], large reward models for online robot reward generation (Wu et al., 2026).

World-model evaluation. We follow the EWMBench evaluation pipeline (Yue et al., 2025). PSNR and SSIM measure low-level reconstruction quality. Scene consistency measures layout and object preservation. Logics evaluates higher-level physical and interaction plausibility. Sem.-CLIP and Sem.-BLEU measure visual-semantic and textual-semantic agreement. Traj-HSD, Traj-Dyn, and Traj-nDTW evaluate robot-trajectory, and higher values indicate better trajectory consistency.

Beyond quantitative evaluation, Appendix B.3 and Appendix B.4 provide qualitative examples of retraining evolution and confidence-map visualization, respectively.

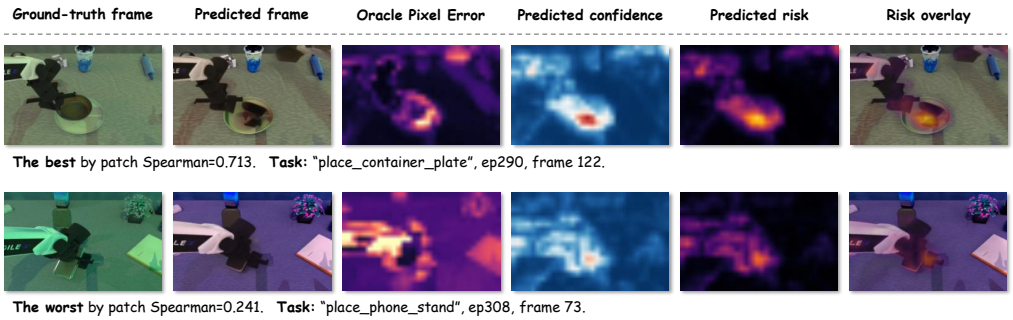


Figure 4: **Qualitative confidence visualization.** Two representative high-agreement episodes from different manipulation tasks are shown. High-risk regions generally coincide with errors around robot arms, manipulated objects, contacts, and occlusions. More results see Appendix B.4.

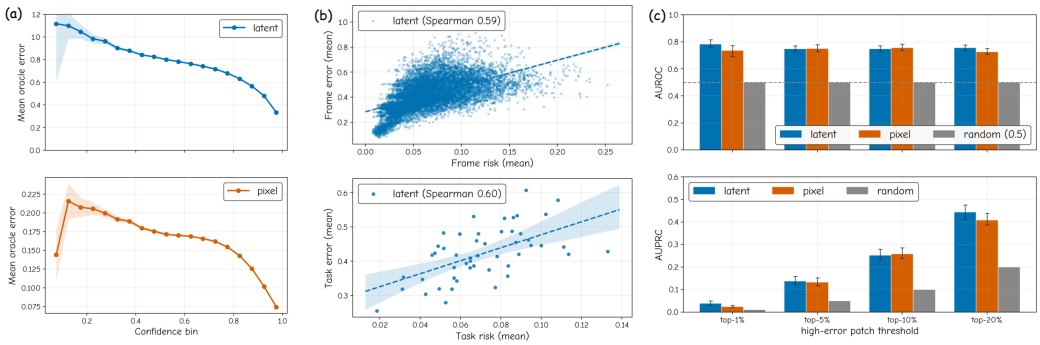


Figure 5: **Validity of confidence as a multi-scale risk signal.** (a) Mean oracle error across confidence bins in latent space (top) and pixel space (bottom). (b) Mean risk versus oracle error at the frame-level (top) and task-level (bottom) in latent space. (c) AUROC (top) and AUPRC (bottom) for detecting the top-$Q\%$ highest-error patches in latent and pixel spaces.

4.2 Why Confidence?

Before using confidence for active learning, we examine whether the predicted risk $r_{t,(i,j)}$ provides a meaningful estimate of world-model prediction error. We evaluate one episode from each of the 50 prescreened tasks using confidence maps and predictions generated by the EVAC-v1.

Qualitative localization. Figure 4 shows that the predicted risk maps respond to spatially localized failures rather than reflecting only global video quality. High-risk regions commonly appear around moving manipulators, object interactions, contacts, and temporarily occluded objects. Their agreement with latent prediction-error maps indicates that the probe captures local failures.

Confidence as a multi-scale ranking signal. We quantitatively compare risk with oracle error across patch-, frame-, and task-levels. As shown in Figure 5, oracle error generally decreases as confidence increases in both spaces, with a clearer monotonic trend in latent space. In latent space, risk achieves Spearman correlations of 0.540, 0.590, and 0.595 at the patch-, frame-, and task-levels, respectively. For detecting the top-5% highest-error patches, it obtains an AUROC of 0.761 and an AUPRC of 0.146, compared with random baselines of 0.5 and 0.05. These results show that dense risk can be reliably aggregated into frame- and task-level scores for active data selection.

Spatial and temporal behavior. Table 1 further shows that risk maps are temporally stable: adjacent frames obtain a top-region IoU of 0.740, while the flicker score is only 0.005. Risk and latent error are also temporally synchronized, with a peak correlation of 0.602 occurring near zero lag. Nevertheless, the top-5% spatial IoU is 0.130, suggesting that confidence reliably identifies error-prone regions but does not precisely reproduce their boundaries.

Overall, the probe provides a strong *ordinal* risk signal for ranking patches, frames, and tasks, although its output should not necessarily be interpreted as an absolutely calibrated probability. Additional pixel-space comparisons, calibration diagnostics, per-task results, failure cases, and parameter sensitivity analyses are provided in Appendix B.1.

Table 1: **Summary of latent-space confidence validity.** Frame- and task-level statistics use mean aggregation.

Property	Metric	Result
Multi-scale ranking	Patch / Frame / Episode Spearman (↑)	0.540/0.590/0.595
High-error detection	AUROC / AUPRC@top-5% (↑)	0.761/0.146
Spatial agreement	Top-5% IoU (↑)	0.130
Temporal stability	Adjacent-frame IoU (↑) / Flicker (↓)	0.740/0.005
Temporal alignment	Peak correlation (↑) / |lag| (↓)	0.602/ ≈ 0

4.3 Why Active Learning?

We evaluate whether confidence-guided selection and retraining improve the action-conditioned world model under a fixed data budget. We report selection-only retraining and selection with additional weighting, since the latter requires an extra inference and scoring stage after selection.

For compact comparison, all component metrics are normalized to $[0, 1]$ and aggregated into four dimensions: Reconstruction $=$ (PSNR $+$ SSIM)$/2$, Scene $=$ Scene Consistency, Semantics $=$ (Logics $+$ Sem.-CLIP $+$ Sem.-BLEU)$/3$, and Motion $=$ Traj-HSD $+$ Traj-Dyn $+$ Traj-nDTW.

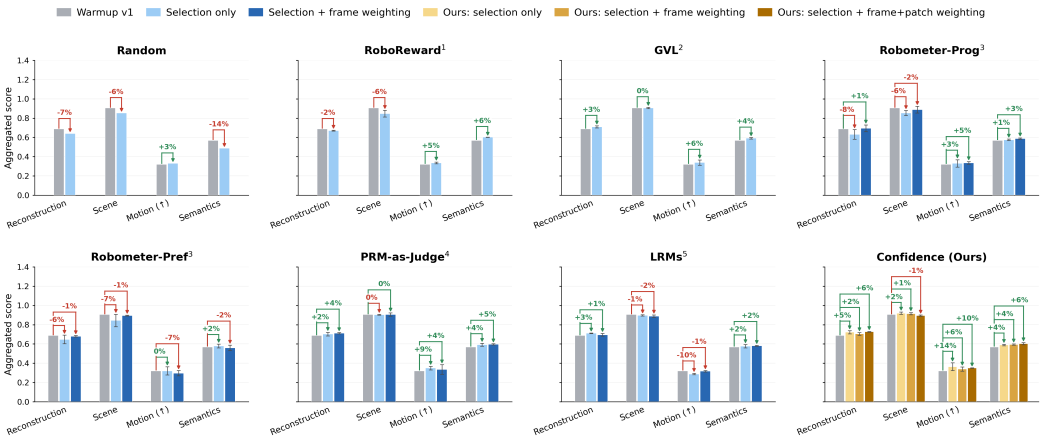


Figure 6: **Main comparison of active-learning scoring and retraining strategies.** Each panel compares EVAC-v1 with selection-only retraining and additional frame weighting. Confidence additionally enables frame+patch weighting. Bars report the mean over three seeds, and arrows report relative changes from EVAC-v1. Error bars indicate paired-bootstrap 95% confidence intervals.

Main comparison. Figure 6 and Table 2 show that the acquisition signal substantially affects post-training performance under the same data budget. Under selection-only retraining, confidence-guided mean-risk selection achieves the best result on eight of the nine component metrics, including PSNR, SSIM, Scene Consistency, Sem.-CLIP, Sem.-BLEU, and all three trajectory metrics.

Additional confidence-guided weighting further strengthens this result. Among methods with additional weighting, confidence with frame weighting achieves the best Scene Consistency, while confidence with frame-and-patch weighting achieves the best PSNR, SSIM, Logics, Sem.-CLIP, Sem.-BLEU, Traj-HSD, Traj-Dyn, and Traj-nDTW. These results support the two roles of confidence in our framework: **1. Efficient data selection:** the mean risk estimation provides an effective acquisition signal for selecting training data; **2. Effective data enhancement:** dense confidence maps further identify unreliable frames and local regions that benefit from stronger supervision.

A minor discrepancy remains in Scene Consistency: frame-and-patch weighting is slightly lower than frame-only weighting, despite improving Reconstruction, Semantics, and Motion. This discrepancy is consistent with the known tension between visual consistency and motion-oriented metrics, which may favor different characteristics of generated videos (Liao et al., 2024; Dou et al., 2026; Ye et al., 2026). It also highlights the limitations of using such proxy metrics to indirectly assess downstream embodied performance.

Table 2: **Detailed normalized results under the same active-learning setting.** The results are averaged over three seeds (42, 3407, and 123). Methods are separated into selection-only retraining and selection with additional weighting. Green, yellow, and red indicate the best, second-best, and third-best results within each block, respectively.

Scoring	Weighting	Reconstruction (↑)		Scene (↑)	Semantics (↑)			Motion (↑)		
		PSNR (↑)	SSIM (↑)	Scene Cons. (↑)	Logics (↑)	Sem.-CLIP (↑)	Sem.-BLEU (↑)	Traj-HSD (↑)	Traj-Dyn (↑)	Traj-nDTW (↑)
Base EVAC		0.5532	0.5778	0.8757	0.4298	0.8523	0.1799	0.0007	0.0001	0.0006
Base EVAC (Warmup v1)		0.6446	0.7309	0.9047	0.5537	0.8824	0.2708	0.1045	0.0721	0.1439
Selection-only retraining: no additional weighting after selection										
Random	None	0.5968	0.6849	0.8524	0.3554	0.8689	0.2372	0.1067	0.0725	0.1518
GVL[2]	None	0.6630	0.7546	0.9057	0.5992	0.8898	0.2849	0.1114	0.0688	0.1592
RoboReward[1]	None	0.6266	0.7158	0.8470	**0.6391**	0.8861	0.2815	0.1113	0.0693	0.1554
Robometer-Prog[3]	None	0.5809	0.6825	0.8532	0.5840	0.8810	0.2627	0.1090	0.0690	0.1526
Robometer-Pref[3]	None	0.5919	0.7029	0.8443	0.5813	0.8859	0.2695	0.1046	0.0638	0.1524
PRM-as-Judge[4]	None	0.6467	0.7580	0.9033	0.6033	0.8880	0.2852	0.1154	0.0746	0.1606
LRMs[5]	None	0.6617	0.7603	0.8966	0.5744	0.8827	0.2756	0.0946	0.0587	0.1352
Confidence (Ours)	**None**	**0.6746**	**0.7692**	**0.9196**	0.5923	**0.8903**	**0.2865**	**0.1181**	**0.0812**	**0.1650**
Selection + additional weighting: extra scoring after selection										
Robometer-Prog[3]	Frame	0.6439	0.7449	0.8882	0.5882	0.8893	0.2878	0.1097	0.0697	0.1563
Robometer-Pref[3]	Frame	0.6317	0.7289	0.8945	0.5289	0.8849	0.2665	0.0974	0.0654	0.1342
PRM-as-Judge[4]	Frame	0.6584	0.7661	0.9053	0.6226	0.8868	0.2795	0.1082	0.0728	0.1536
LRMs[5]	Frame	0.6463	0.7448	0.8882	0.5826	0.8869	0.2750	0.1053	0.0635	0.1483
Confidence (Ours)	**Frame**	0.6595	0.7500	**0.9143**	0.6171	0.8874	0.2789	0.1106	0.0684	0.1608
Confidence (Ours)	**Fr.+Patch**	**0.6772**	**0.7758**	0.8942	**0.6226**	**0.8925**	**0.2952**	**0.1118**	**0.0759**	**0.1633**

Table 3: **Ablation of selection criteria.** All experiments use the default seed 42 and no additional weighting. Green, yellow, and red indicate the best, second-best, and third-best results.

Variant	Reconstruction (↑)		Scene (↑)	Semantics (↑)			Motion (↑)		
	PSNR (↑)	SSIM (↑)	Scene Cons. (↑)	Logics (↑)	Sem.-CLIP (↑)	Sem.-BLEU (↑)	Traj-HSD (↑)	Traj-Dyn (↑)	Traj-nDTW (↑)
Random	0.5968	0.6849	0.8523	0.3554	0.8689	0.2372	0.1067	0.0725	0.1518
Mean Risk	**0.6838**	**0.7797**	**0.9322**	**0.6198**	0.8894	0.2754	**0.1313**	**0.0921**	**0.1780**
Tail Risk	0.6317	0.7197	0.9218	0.5579	0.8890	**0.3047**	0.1207	0.0711	0.1630
Persistent Risk	0.6529	0.7470	0.9230	0.5496	**0.8928**	0.2858	0.0926	0.0608	0.1342

Ablation studies. Table 3 isolates the effect of the confidence-risk aggregation used for data selection, without additional frame or patch weighting. Mean risk gives the strongest overall result, achieving the best performance on seven of the nine component metrics, including both reconstruction metrics, Scene Consistency, Logics, and all three trajectory metrics. Tail risk obtains the highest Sem.-BLEU, while persistent risk obtains the highest Sem.-CLIP. We therefore use mean risk as the default acquisition criterion in our main experiments.

5 Conclusion

We presented ConfAL-WM, a confidence-guided active learning framework for post-training action-conditioned world models. By attaching a lightweight confidence probe to UNet decoder features, our method supports task-level data selection as well as frame- and patch-level weighted retraining. Experiments on RoboTwin2.0 show that confidence provides an effective risk signal and improves reconstruction, scene consistency, and semantic quality over scalar scoring baselines.

Our current approach still has several limitations. First, the confidence probe is designed around the internal decoder features of a UNet diffusion backbone and is trained only with EVAC on RoboTwin2.0, making it difficult to transfer directly across world-model architectures and domains. Future work could train a general-purpose world confidence model using multiple backbone world models and diverse embodied datasets. Such a model would also require a more universal input interface. Second, confidence representations are difficult to evaluate directly: their usefulness is mainly inferred from error correlation, data ranking, and downstream retraining performance, rather than from an independent measure of whether the desired reliability representation has been learned. More explicit representation diagnostics and controlled causal evaluations are therefore needed. Third, current world-model evaluation remains incomplete, particularly because visual fidelity and motion accuracy may conflict. Future protocols could separately evaluate robot-arm motion, manipulated-object dynamics, interaction regions, and static backgrounds, providing a more precise account of both local reconstruction quality and action-conditioned physical evolution.

Impact Statement

This work aims to improve the data efficiency and reliability of embodied world-model post-training by focusing computation on informative data and unreliable regions. More accurate world models may benefit robotic simulation, planning, and synthetic-data generation, but their predictions should not be treated as guaranteed physical outcomes, especially under distribution shifts or safety-critical deployment.

References

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025.

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*, 2025.

Saptarshi Dasgupta, Akshat Gupta, Shreshth Tuli, and Rohan Paul. Uncertainty-aware active learning of nerf-based object models for robot manipulators using visual and re-orientation actions, 2024. URL https://arxiv.org/abs/2404.01812.

Shivin Dass, Alaa Khaddaj, Logan Engstrom, Aleksander Madry, Andrew Ilyas, and Roberto Martín-Martín. Datamil: Selecting data for robot imitation learning with datamodels. *arXiv preprint arXiv:2505.09603*, 2025.

Weijia Dou, Wenzhao Zheng, Weiliang Chen, Yu Zheng, Jie Zhou, and Jiwen Lu. Measuring 3d spatial geometric consistency in dynamic generated videos. *arXiv preprint arXiv:2603.19048*, 2026.

Mehmet Arda Eren and Erhan Oztop. Sample efficient robot learning in supervised effect prediction tasks, 2024. URL https://arxiv.org/abs/2412.02331.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. Dreamgen: Unlocking generalization in robot learning through video world models. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 5170–5194. PMLR, 2025. URL https://proceedings.mlr.press/v305/jang25a.html.

Yuheng Ji, Yuyang Liu, Huajie Tan, Xuchuan Huang, Fanding Huang, Yijie Xu, Cheng Chi, Yuting Zhao, Huaihai Lyu, Peterson Co, Mingyu Cao, Qiongyu Zhang, Zhe Li, Enshen Zhou, Pengwei Wang, Zhongyuan Wang, Shanghang Zhang, and Xiaolong Zheng. Prm-as-a-judge: A dense evaluation paradigm for fine-grained robotic auditing, 2026. URL https://arxiv.org/abs/2603.21669.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, and Guanghui Ren. Enerverse-ac: Envisioning embodied environments with action condition, 2025. URL https://arxiv.org/abs/2505.09723.

Tony Lee, Andrew Wagenmaker, Karl Pertsch, Percy Liang, Sergey Levine, and Chelsea Finn. Roboreward: General-purpose vision-language reward models for robotics, 2026. URL https://arxiv.org/abs/2601.00675.

Chenhao Li, Andreas Krause, and Marco Hutter. Uncertainty-aware robotic world model makes offline model-based reinforcement learning work on real robots, 2025a. URL https://arxiv.org/abs/2504.16680.

Daquan Li et al. Worldmodelbench: Judging video generation models as world models, 2025b. URL https://arxiv.org/abs/2502.20694.

Anthony Liang, Yigit Korkmaz, Jiahui Zhang, Minyoung Hwang, Abrar Anwar, Sidhant Kaushik, Aditya Shah, Alex S. Huang, Luke Zettlemoyer, Dieter Fox, Yu Xiang, Anqi Li, Andreea Bobu, Abhishek Gupta, Stephen Tu, Erdem Biyik, and Jesse Zhang. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons, 2026. URL https://arxiv.org/abs/2603.02115.

Mingxiang Liao, Hannan Lu, Xinyu Zhang, Fang Wan, Tianyu Wang, Yuzhong Zhao, Wangmeng Zuo, Qixiang Ye, and Jingdong Wang. Evaluation of text-to-video generation models: A dynamics perspective. *arXiv preprint arXiv:2407.01094*, 2024.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-h0.7: A latent world-action model from egocentric videos, 2026. URL https://arxiv.org/abs/2605.00078.

Jindi Lv, Hao Li, Jie Li, Yifei Nie, Fankun Kong, Yang Wang, Xiaofeng Wang, Zheng Zhu, Chaojun Ni, Qiuping Deng, Hengtao Li, Jiancheng Lv, and Guan Huang. Viva: A video-generative value model for robot reinforcement learning, 2026. URL https://arxiv.org/abs/2604.08168.

Yecheng Jason Ma, Joey Hejna, Ayzaan Wahid, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, Jonathan Tompson, Osbert Bastani, Dinesh Jayaraman, Wenhao Yu, Tingnan Zhang, Dorsa Sadigh, and Fei Xia. Vision language models are in-context value learners, 2024. URL https://arxiv.org/abs/2411.04549.

Zhiting Mei, Ola Shorinwa, and Anirudha Majumdar. How confident are video models? empowering video models to express their uncertainty, 2025a. URL https://arxiv.org/abs/2510.02571.

Zhiting Mei, Tenny Yin, Micah Baker, Ola Shorinwa, and Anirudha Majumdar. World models that know when they don't know: Controllable video generation with calibrated uncertainty, 2025b. URL https://arxiv.org/abs/2512.05927.

NVIDIA. Pbench: A physical ai benchmark for world models. https://research.nvidia.com/labs/cosmos-lab/pbench/, 2025.

Ralf Römer, Maximilian Seeliger, Saida Liu, Ben Sturgis, Marco Bagatella, Daniel Marta, Andreas Krause, and Angela P. Schoellig. Uncertainty quantification for flow-based vision-language-action models. *arXiv preprint arXiv:2606.18043*, 2026.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1708.00489.

Burr Settles. Active learning literature survey. *University of Wisconsin–Madison Computer Sciences Technical Report*, (1648), 2009. URL https://burrsettles.com/pub/settles.activelearning.pdf.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 287–294, 1992. doi: 10.1145/130385.130417.

Yufan Wei, Kun Zhou, Lingjun Mao, Zijun Zhang, Ziming Xu, Ziqiao Xi, Shuang Liang, Ruobing Han, Yuchen Yan, Xinyue Wang, Fan Feng, and Biwei Huang. Causally debiased latent action model for embodied action conditioned world models. *arXiv preprint arXiv:2607.09185*, 2026.

Yanru Wu, Weiduo Yuan, Ang Qi, Vitor Guizilini, Jiageng Mao, and Yue Wang. Large reward models: Generalizable online robot reward generation with vision-language models, 2026. URL https://arxiv.org/abs/2603.16065.

Xi Ye, Wenjia Yang, Yangyang Xu, Xiaoyang Liu, Duo Su, Mengfei Xia, and Jun Zhu. Shift: Motion alignment in video diffusion models with adversarial hybrid fine-tuning. *arXiv preprint arXiv:2603.17426*, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination?, 2026. URL https://arxiv.org/abs/2603.16666.

Hu Yue, Siyuan Huang, Yue Liao, Shengcong Chen, Pengfei Zhou, Liliang Chen, Maoqing Yao, and Guanghui Ren. Ewmbench: Evaluating scene, motion, and semantic quality in embodied world models, 2025. URL https://arxiv.org/abs/2505.09694.

Peiwen Zhang, Yufan Deng, Shangkun Sun, Juncheng Ma, Duomin Wang, Jonas Du, Zilin Pan, Ye Huang, Hao Liang, Songyan Huang, Ruihua Zhang, Enze Xie, Ming-Yu Liu, and Daquan Zhou. Physisforcing: Physics reinforced world simulator for robotic manipulation. *arXiv preprint arXiv:2606.28128*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9834–9844, 2025.

A More Details on Methods

A.1 Patch-level Error Construction

Figure 7 illustrates the overall architecture used to construct the patch-level confidence target.

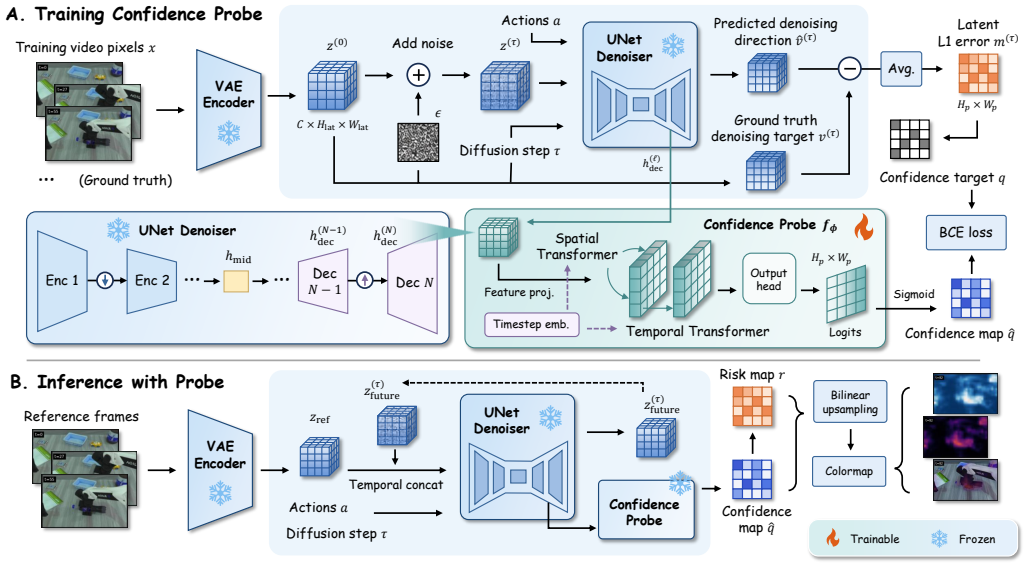


Figure 7: Training and inference of the confidence probe in the UNet latent diffusion world model. The figure follows the same training and inference pipeline as Figure 2.

Latent diffusion and velocity target. Given a ground-truth training video x, the frozen VAE encoder first maps it into a clean latent representation

$$z^{(0)} = \text{Enc}_{\text{VAE}}(x), \qquad z^{(0)} \in \mathbb{R}^{C \times H_{\text{lat}} \times W_{\text{lat}}}, \tag{11}$$

where C denotes the latent channel dimension and $(H_{\text{lat}}, W_{\text{lat}})$ denotes the spatial resolution of the latent grid. For simplicity, the batch and video-time dimensions are omitted in this subsection.

At diffusion timestep τ, Gaussian noise $\varepsilon \sim \mathcal{N}(0, I)$ is sampled and added according to the forward diffusion schedule:

$$z^{(\tau)} = \sqrt{\bar{\alpha}_\tau}\, z^{(0)} + \sqrt{1 - \bar{\alpha}_\tau}\, \varepsilon, \tag{12}$$

where $\bar{\alpha}_\tau$ is the cumulative noise-schedule coefficient at timestep τ. The action-conditioned UNet predicts the corresponding velocity parameterization as

$$\hat{v}^{(\tau)} = \text{UNet}_\Theta\left(z^{(\tau)}, a, e_\tau\right), \qquad \hat{v}^{(\tau)} \in \mathbb{R}^{C \times H_{\text{lat}} \times W_{\text{lat}}}, \tag{13}$$

where a is the robot-action condition and e_τ is the diffusion-timestep embedding. Under the velocity-prediction formulation, the ground-truth denoising target is analytically constructed as

$$v^{(\tau)} = \sqrt{\bar{\alpha}_\tau}\, \varepsilon - \sqrt{1 - \bar{\alpha}_\tau}\, z^{(0)}, \qquad v^{(\tau)} \in \mathbb{R}^{C \times H_{\text{lat}} \times W_{\text{lat}}}. \tag{14}$$

Therefore, $\hat{v}^{(\tau)}$ and $v^{(\tau)}$ have the same tensor shape and lie in the same velocity-prediction latent space. The action a and timestep embedding e_τ condition the prediction $\hat{v}^{(\tau)}$, whereas the target $v^{(\tau)}$ is directly determined by $z^{(0)}$, ε, and the diffusion schedule.

Equivalent clean-latent reconstruction. Rather than executing the complete reverse diffusion trajectory, we can directly recover the predicted clean latent from the current noisy latent and the velocity prediction:

$$\hat{z}^{(\tau)} = \sqrt{\bar{\alpha}_\tau}\, z^{(\tau)} - \sqrt{1 - \bar{\alpha}_\tau}\, \hat{v}^{(\tau)}. \tag{15}$$

This provides an equivalent way to measure the local prediction error in the clean-latent space. Specifically, substituting equation 12 and equation 15 into $\left|\hat{z}^{(\tau)} - z^{(0)}\right|$ gives

$$\begin{aligned}\left|\hat{z}^{(\tau)} - z^{(0)}\right| &= \left|\sqrt{\bar{\alpha}_\tau}\, z^{(\tau)} - \sqrt{1-\bar{\alpha}_\tau}\, \hat{v}^{(\tau)} - z^{(0)}\right| \\ &= \left|(\bar{\alpha}_\tau - 1)\, z^{(0)} + \sqrt{\bar{\alpha}_\tau\,(1-\bar{\alpha}_\tau)}\,\varepsilon - \sqrt{1-\bar{\alpha}_\tau}\,\hat{v}^{(\tau)}\right| \\ &= \sqrt{1-\bar{\alpha}_\tau}\left|\sqrt{1-\bar{\alpha}_\tau}\, z^{(0)} - \sqrt{\bar{\alpha}_\tau}\,\varepsilon + \hat{v}^{(\tau)}\right|.\end{aligned} \tag{16}$$

Using the velocity target in equation 14, we therefore obtain

$$\left|\hat{z}^{(\tau)} - z^{(0)}\right| = \sqrt{1-\bar{\alpha}_\tau}\left|\hat{v}^{(\tau)} - v^{(\tau)}\right|. \tag{17}$$

Hence, at a fixed diffusion timestep τ, the clean-latent reconstruction error and the velocity-prediction error differ only by the scalar factor $\sqrt{1-\bar{\alpha}_\tau}$. This allows confidence supervision to use velocity-prediction error directly, without running the full reverse diffusion process.

Patch-level latent prediction error. We now restore the frame index t. For each latent spatial location (x, y), we first average the absolute clean-latent reconstruction error over the C latent channels:

$$\begin{aligned}E^{(\tau)}_{t,(x,y)} &= \frac{\sqrt{1-\bar{\alpha}_\tau}}{C}\sum_{c=1}^{C}\left|\hat{v}^{(\tau)}_{t,p} - v^{(\tau)}_{t,p}\right| \\ &= \frac{1}{C}\sum_{c=1}^{C}\left|\hat{z}^{(\tau)}_{t,p} - z^{(0)}_{t,p}\right|,\end{aligned} \tag{18}$$

Here $p = (c, x, y)$ denotes a latent entry, c indexes the latent channel and (x, y) denotes the spatial location on the latent grid. Thus, $E^{(\tau)}_t \in \mathbb{R}^{H_{\text{lat}} \times W_{\text{lat}}}$ forms a spatial error map for frame t.

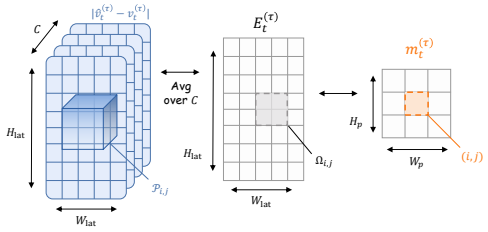


Figure 8: **Patch-level latent error construction.** Channel-wise latent prediction errors are first averaged into a spatial error map and then pooled over the region aligned with each confidence-probe patch.

As illustrated in Figure 8, let $\Omega_{i,j}$ denote the latent spatial region aligned with confidence-probe output location (i, j). The corresponding patch-level error is obtained by spatially averaging $E^{(\tau)}_t$ over this region. Equivalently, defining

$$\mathcal{P}_{i,j} = \{1, \ldots, C\} \times \Omega_{i,j}, \tag{19}$$

the same quantity can be written directly over all latent entries within the patch:

$$\begin{aligned}m^{(\tau)}_{t,(i,j)} &= \frac{1}{|\Omega_{i,j}|}\sum_{(x,y)\in\Omega_{i,j}} E^{(\tau)}_{t,(x,y)} \\ &= \frac{1}{C\cdot|\Omega_{i,j}|}\sum_{(x,y)\in\Omega_{i,j}}\sum_{c=1}^{C}\left|\hat{z}^{(\tau)}_{t,p} - z^{(0)}_{t,p}\right| \\ &= \frac{1}{|\mathcal{P}_{i,j}|}\sum_{p\in\mathcal{P}_{i,j}}\left|\hat{z}^{(\tau)}_{t,p} - z^{(0)}_{t,p}\right|.\end{aligned} \tag{20}$$

Hence, $m^{(\tau)}_t = \{m^{(\tau)}_{t,(i,j)}\}_{i,j} \in \mathbb{R}^{H_p \times W_p}$ has the same spatial indexing as the confidence output.

Binary confidence target. The binary supervision target is obtained by comparing each local prediction error with the adaptive threshold θ_t:

$$q_{t,(i,j)} = \mathbb{I}\left\{m^{(\tau)}_{t,(i,j)} < \theta_t\right\}, \qquad q_t = \left\{q_{t,(i,j)}\right\}_{i,j} \in \{0,1\}^{H_p \times W_p}. \tag{21}$$

A value $q_{t,(i,j)} = 1$ indicates that the corresponding latent patch is treated as reliable, while $q_{t,(i,j)} = 0$ denotes a high-error prediction. The adaptive construction of θ_t is described in the following subsection.

UNet feature tapping and confidence prediction. The denoising UNet contains multiple encoder blocks, a bottleneck, and multiple decoder blocks:

$$z^{(\tau)} \to h_{\text{enc}}^{(1)} \to \cdots \to h_{\text{enc}}^{(K)} \to h_{\text{mid}} \to h_{\text{dec}}^{(1)} \to \cdots \to h_{\text{dec}}^{(L)} \to \hat{v}^{(\tau)}. \tag{22}$$

Our confidence probe taps a selectable decoder feature $h_{\text{dec}}^{(\ell)}$, where the layer index ℓ controls the balance between global context and spatial locality. Earlier decoder layers provide coarser, more global representations, whereas later layers retain finer spatial information.

For future frame t, the dense confidence prediction is

$$\hat{q}_t = \sigma\left(f_\phi\left(h_{\text{dec}}^{(\ell)}, e_\tau, e_\theta\right)\right), \qquad \hat{q}_t \in [0,1]^{H_p \times W_p}, \tag{23}$$

where e_τ denotes the diffusion-timestep embedding, e_θ denotes the sampled error threshold embedding, f_ϕ is the trainable confidence probe and $\sigma(\cdot)$ denotes the element-wise sigmoid function. Each output location (i, j) is spatially aligned with $\Omega_{i,j}$, and therefore predicts the confidence associated with the local error $m_{t,(i,j)}^{(\tau)}$.

A.2 EMA-Calibrated Random Threshold Supervision

The numerical scale of the local prediction error can vary across datasets and training regimes, making a fixed confidence threshold difficult to transfer. We therefore estimate the threshold interval online from the batch-wise error distribution and stabilize it using exponential moving averages (EMA). For notational simplicity, the derivation below considers a fixed training sample and omits the sample index; the percentile statistics $p_l^{(s)}$ and $p_h^{(s)}$ are still computed from all local errors in the current training batch.

Batch-wise percentile estimation. At training step s, let $m^{(s)}$ denote the collection of patch-level latent errors in the current batch, where each element follows the definition of $m_{t,(i,j)}^{(\tau)}$ in Eq. equation 20. We estimate the lower and upper error statistics as

$$p_l^{(s)} = \text{Quantile}_{0.1}\left(m^{(s)}\right), \qquad p_h^{(s)} = \text{Quantile}_{0.9}\left(m^{(s)}\right). \tag{24}$$

These percentiles adapt the supervision range to the current error scale without manually specifying dataset-dependent MAE thresholds.

Two-stage EMA update. The lower and upper threshold bounds are updated as

$$l^{(s)} = (1-\gamma_s) l^{(s-1)} + \gamma_s p_l^{(s)}, \qquad h^{(s)} = (1-\gamma_s) h^{(s-1)} + \gamma_s p_h^{(s)}. \tag{25}$$

We use a two-stage update schedule,

$$\gamma_s = \begin{cases} 0.20, & s < S_{\text{warm}}, \\ 0.002, & s \geq S_{\text{warm}}, \end{cases} \qquad \beta_s = 1 - \gamma_s = \begin{cases} 0.80, & s < S_{\text{warm}}, \\ 0.998, & s \geq S_{\text{warm}}, \end{cases} \tag{26}$$

where γ_s is the injection rate of the current batch statistics and β_s is the equivalent EMA momentum. Thus, the conventional EMA form is $\mu^{(s)} = \beta_s \mu^{(s-1)} + (1-\beta_s) p^{(s)}$. The larger update rate during warmup rapidly adapts the threshold range to the target-domain error scale, while the smaller rate afterwards tracks slower distributional changes with stronger smoothing.

Random threshold sampling and interpretation. After updating the bounds, a separate threshold is sampled for each predicted future frame t:

$$\theta_t^{(s)} \sim \mathcal{U}\left(l^{(s)}, h^{(s)}\right), \qquad q_{t,(i,j)}^{(s)} = \mathbb{I}\left\{m_{t,(i,j)}^{(\tau)} < \theta_t^{(s)}\right\}. \tag{27}$$

The same $\theta_t^{(s)}$ is shared by all spatial patches (i, j) within frame t, while different future frames receive independently sampled thresholds.

For a fixed local error m, the binary target becomes random only through $\theta_t^{(s)}$. Since $\theta_t^{(s)} \sim \mathcal{U}(l^{(s)}, h^{(s)})$, its density is $1/(h^{(s)} - l^{(s)})$ over the current threshold interval. Marginalizing over the sampled threshold gives

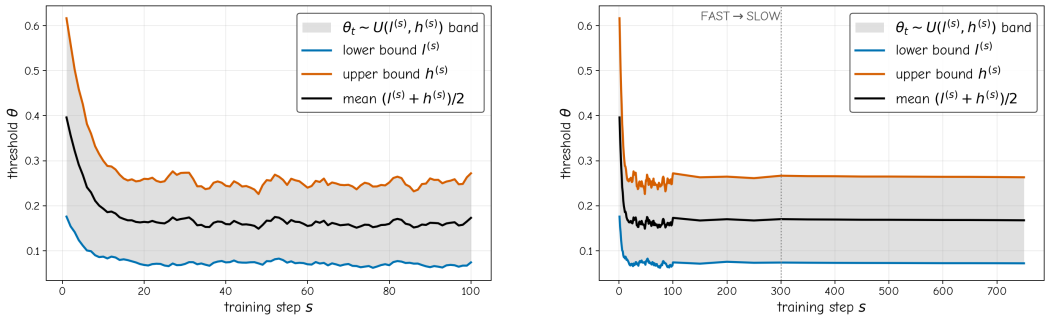
$$\begin{aligned}
\mathbb{E}_\theta[q = 1 \mid m] = \mathbb{P}_\theta(m < \theta) &= \int_{l^{(s)}}^{h^{(s)}} \mathbb{I}\{m < \theta\} \frac{1}{h^{(s)} - l^{(s)}} \, \mathrm{d}\theta \\
&= \frac{\left[h^{(s)} - \max\{m, l^{(s)}\}\right]_+}{h^{(s)} - l^{(s)}} = \mathrm{clamp}\left(\frac{h^{(s)} - m}{h^{(s)} - l^{(s)}}, 0, 1\right) \\
&= \begin{cases} 1, & m \le l^{(s)}, \\ \dfrac{h^{(s)} - m}{h^{(s)} - l^{(s)}}, & l^{(s)} < m < h^{(s)}, \\ 0, & m \ge h^{(s)}. \end{cases}
\end{aligned} \tag{28}$$

Hence, although each training target is binary, averaging over random thresholds induces a continuous target that decreases monotonically with prediction error.

In the threshold-marginalized case, the optimal prediction under binary cross-entropy is exactly this conditional probability:

$$\begin{aligned}
\hat{q}^*(m) &= \arg\min_{\hat{q} \in (0,1)} \mathbb{E}_{q|m}\left[-q \log \hat{q} - (1 - q) \log(1 - \hat{q})\right] \\
&= \mathbb{E}[q \mid m] = \mathrm{clamp}\left(\frac{h^{(s)} - m}{h^{(s)} - l^{(s)}}, 0, 1\right).
\end{aligned} \tag{29}$$

In our implementation, $\theta_t^{(s)}$ is additionally embedded into the confidence probe as described in the main method, so the probe learns threshold-conditioned reliability rather than only this marginalized form. Equation equation 28 nevertheless explains why randomized binary supervision naturally induces a continuous confidence ordering.



(a) Zoom-in of the first 100 training steps

(b) Extended training-step evolution

Figure 9: **Dynamics of the EMA-based adaptive threshold band.** The probe uses a stochastic threshold $\theta \sim \mathcal{U}(l^{(s)}, h^{(s)})$, where $l^{(s)}$ and $h^{(s)}$ are the EMA-tracked lower and upper bounds at training step s, and the black curve shows their midpoint $(l^{(s)} + h^{(s)})/2$. (a) During the first 100 steps, both bounds decrease rapidly and quickly approach a stable range, reflecting fast adaptation to the current error scale. (b) Over a longer set of sampled training steps, the threshold band shows a clear transition from a fast-changing warmup regime to a slower and more stable regime.

Implementation and EMA dynamics. For RoboTwin2.0, we initialize $l^{(0)} = 0.20$ and $h^{(0)} = 0.70$, use $S_{\text{warm}} = 300$, and continue updating the EMA bounds throughout the 6000-step probe training. The bounds eventually stabilize around $l \approx 0.0705$ and $h \approx 0.2610$, illustrating why online calibration is preferable to manually transferring fixed thresholds across datasets or error spaces.

Figure 9 visualizes the adaptation process. During the initial stage, the threshold band rapidly contracts toward the observed target-domain error scale. Over a longer range of sampled training steps, the evolution becomes substantially slower and more stable, showing the intended transition from fast initialization to long-term tracking.

B More Experimental Results

B.1 Additional Confidence Evaluation Details

This subsection supplements the confidence evaluation in Section 4.2. We report additional results for alternative risk aggregations, spatial localization, operational calibration, and parameter sensitivity. Latent-space errors remain our primary oracle, while pixel-space results are included as an external visual-domain reference.

Alternative task-level risk aggregations. Figure 10 compares mean, tail, and persistent risk aggregation in latent and pixel spaces. Mean risk achieves the strongest global Spearman correlation because averaging suppresses local noise and measures the overall prediction difficulty of an episode. Tail risk instead emphasizes sparse severe errors, while persistent risk targets failures that recur across multiple frames; consequently, they need not correlate as strongly with the episode-wide mean error. These correlations therefore verify that each score contains a meaningful error signal, but do not by themselves determine which acquisition rule yields the best post-training model. Their downstream differences are evaluated separately in Table 3.

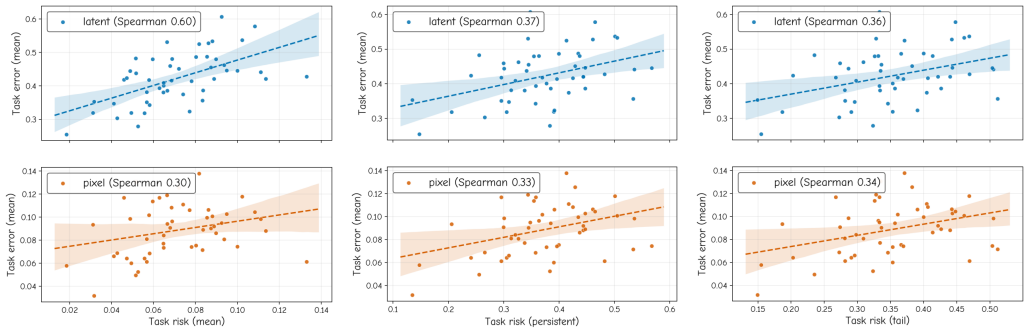
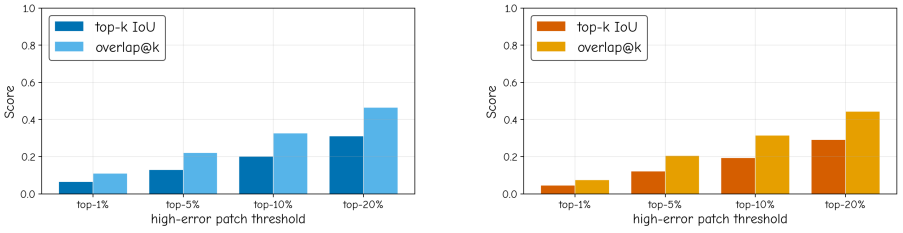


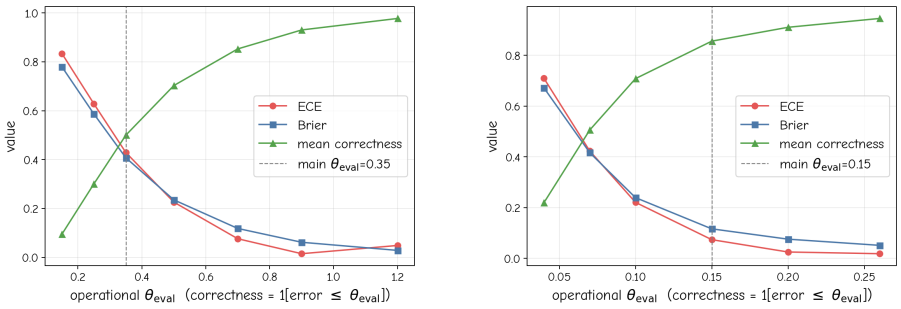
Figure 10: **Task-level risk aggregation in latent and pixel spaces.** Mean, tail, and persistent risk scores are compared with their corresponding oracle-error statistics over the 50 prescreened tasks. Dashed lines show linear fits, shaded regions denote confidence intervals, and each panel reports the Spearman correlation. Mean aggregation gives the strongest global correlation, whereas tail and persistent aggregation emphasize different failure patterns.

Spatial localization agreement. We further compare the spatial locations selected by risk and oracle error. For each ratio k, top-k IoU measures the intersection-over-union between the highest-risk and highest-error patch sets, while overlap@k measures the fraction of high-error patches recovered by the high-risk set. Figure 11 shows that both metrics increase as the selected region becomes larger. At top-5%, latent-space localization reaches approximately 0.13 IoU and 0.22 overlap, with slightly lower values in pixel space. The relatively high overlap but moderate IoU indicates that risk identifies many relevant error regions without exactly reproducing their boundaries. This supports soft patch weighting rather than hard spatial selection.



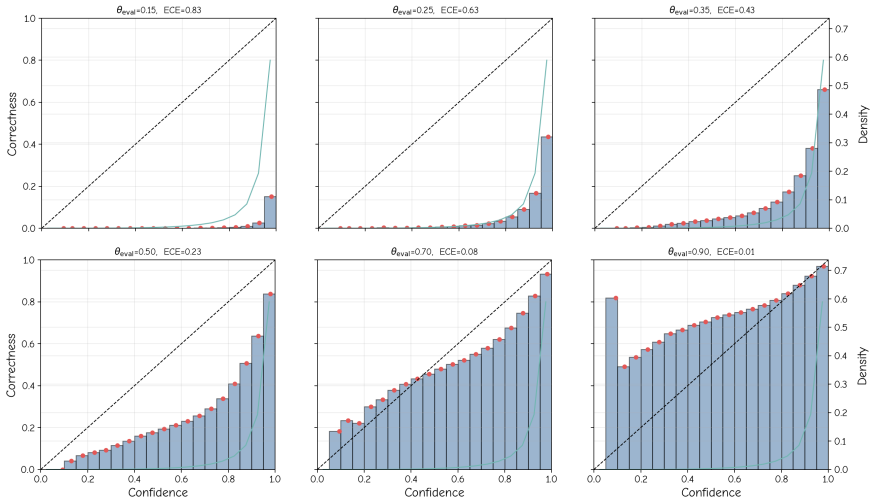
(a) Latent-space localization (b) Pixel-space localization

Figure 11: **Spatial agreement between high-risk and high-error patches.** Top-k IoU and overlap@k are reported for the top-1%, 5%, 10%, and 20% patches in (a) latent and (b) pixel spaces. The increasing scores at larger ratios reflect coarser spatial coverage and should not be interpreted as improved localization resolution.

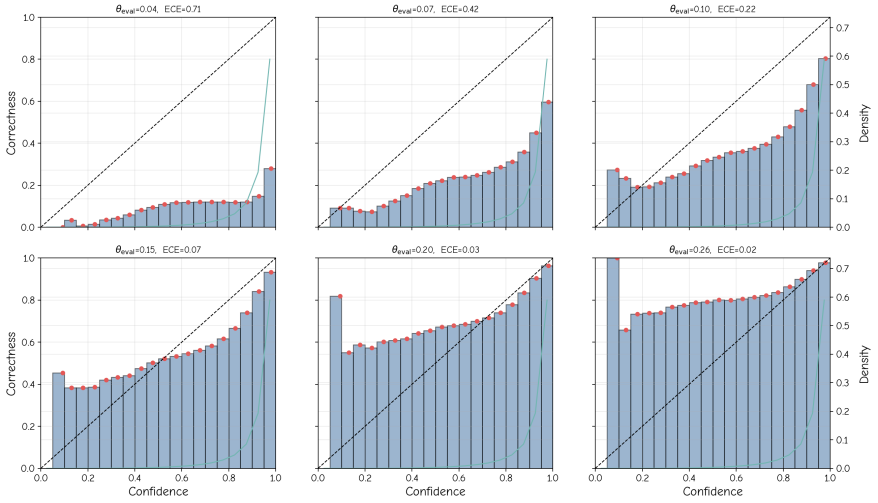


(a) Latent-space calibration sweep (b) Pixel-space calibration sweep

Figure 12: **ECE and Brier scores under different operational thresholds.** Calibration is evaluated by sweeping the correctness threshold θ_{eval} in (a) latent space and (b) pixel space. The confidence maps are fixed throughout; only the binary correctness definition changes.



(a) Latent-space reliability diagrams



(b) Pixel-space reliability diagrams

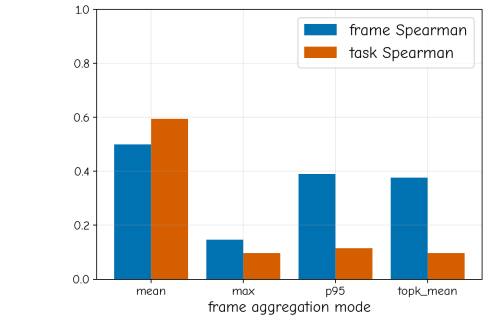
Figure 13: **Reliability diagrams across operational thresholds.** Each subfigure contains reliability diagrams under different choices of θ_{eval}: (a) latent space and (b) pixel space. As the threshold becomes looser, the observed correctness generally moves closer to the predicted confidence, although the latent-space probe remains more overconfident near the main operating point.

Operational calibration. Although the confidence probe is trained with stochastic binary targets, its calibration at test time depends on how prediction correctness is defined. To avoid confusion with the diffusion timestep τ, we denote the operational error threshold by θ_{eval} and define correctness as $\mathbb{I}\{m^{(\tau)}_{t,(i,j)} \leq \theta_{\text{eval}}\}$. Varying θ_{eval} does not change the predicted confidence maps; it only changes the post-hoc definition of whether a prediction is counted as correct.

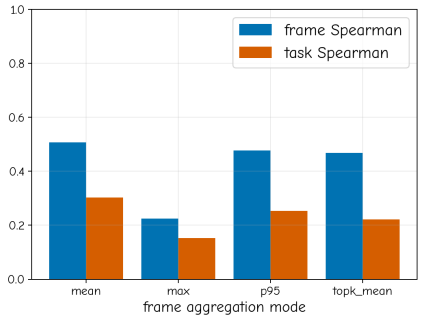
Figure 12 summarizes calibration quality under different choices of θ_{eval}. In latent space, ECE and Brier improve as the threshold becomes looser, but the probe remains noticeably overconfident around the main operating point. Pixel-space calibration appears better under its own selected threshold, although the latent and pixel settings are not directly comparable because they rely on different error scales. Figure 13 further visualizes the corresponding reliability diagrams. In both spaces, the curves become closer to the diagonal as θ_{eval} increases, confirming that the apparent calibration quality depends strongly on the operational correctness definition. Overall, these results support using confidence mainly as an ordinal ranking signal rather than as an absolutely calibrated probability.

Parameter sensitivity. We examine two implementation choices that are not reported in the main text. First, Figures 14(a–b) compare mean, maximum, percentile, and top-k frame aggregation. Mean aggregation provides the strongest frame- and task-level Spearman correlations in both latent and pixel spaces, whereas extreme-value aggregation amplifies isolated noisy patches. This motivates the mean aggregation adopted in our main evaluation.

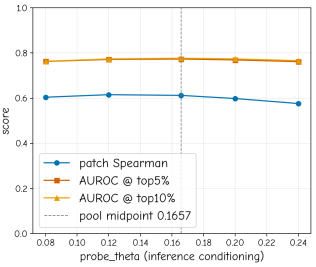
Second, Figures 14(c–d) vary the probe conditioning threshold θ_{cond} and regenerate the confidence maps. Low-to-moderate thresholds provide the strongest spatial discrimination, while large thresholds make most patches highly confident and reduce localization contrast. The default threshold lies near the best-performing range, and the overall variation remains moderate, indicating that the confidence-ranking conclusion is not sensitive to a narrowly tuned threshold.

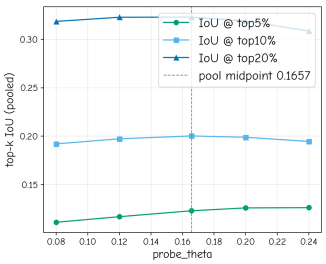


(a) Frame aggregation in latent space

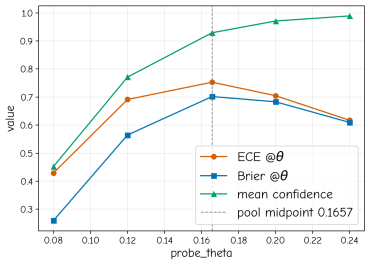


(b) Frame aggregation in pixel space





(c) Localization sensitivity



(d) Calibration sensitivity

Figure 14: **Sensitivity to confidence aggregation and probe conditioning.** (a–b) Frame- and task-level Spearman correlations under mean, maximum, percentile, and top-k aggregation in latent and pixel spaces. (c) High-error detection and spatial localization under different probe conditioning thresholds θ_{cond}. (d) ECE, Brier score, and mean confidence under the same thresholds. Unlike the operational threshold θ_{eval}, changing θ_{cond} regenerates the confidence maps.

B.2 Additional Numerical Results for Active Learning

This subsection provides the complete numerical results underlying the active-learning comparison in Section 4.2. We first report the four aggregated dimensions used in the main comparison, followed by paired-bootstrap improvements relative to EVAC-v1. We then provide the complete seed-wise component metrics for seeds 42, 3407, and 123.

Aggregated results and paired-bootstrap evidence. Table 4 reports the aggregated active-learning results. Reconstruction, Scene, Motion, and Semantics follow the same definitions as in Section 4.3. For learned scoring methods, the reported values are averaged over seeds 42, 3407, and 123. Table 5 further reports paired-bootstrap improvements over EVAC-v1. For each learned scoring method, we concatenate the episode-level paired differences from all three seeds and perform 10,000 percentile-bootstrap resamples at 95% confidence. The pairing with EVAC-v1 is preserved within each seed before pooling. Positive values indicate improvement for all four dimensions.

Table 4: Aggregated active-learning results. Each entry reports the aggregated mean followed by its relative change from EVAC-v1. Results for learned scoring methods are averaged over three seeds.

Scoring	Weighting	Reconstruction (↑)	Scene (↑)	Motion (↑)	Semantics (↑)
Base EVAC		0.5655 (-17.8%)	0.8757 (-3.2%)	0.0014 (-99.6%)	0.4873 (-14.3%)
Base EVAC (Warmup v1)		0.6878 (0.0%)	0.9047 (0.0%)	0.3205 (0.0%)	0.5690 (0.0%)
Selection-only retraining: no additional weighting after selection					
Random	None	0.6409 (-6.8%)	0.8524 (-5.8%)	0.3310 (+3.3%)	0.4872 (-14.4%)
RoboReward[1]	None	0.6712 (-2.4%)	0.8470 (-6.4%)	0.3360 (+4.8%)	**0.6022 (+5.8%)**
GVL[2]	None	0.7088 (+3.1%)	0.9057 (+0.1%)	0.3393 (+5.9%)	0.5913 (+3.9%)
Robometer-Prog[3]	None	0.6317 (-8.2%)	0.8532 (-5.7%)	0.3307 (+3.2%)	0.5759 (+1.2%)
Robometer-Pref[3]	None	0.6474 (-5.9%)	0.8443 (-6.7%)	0.3208 (+0.1%)	0.5789 (+1.7%)
PRM-as-Judge[4]	None	0.7024 (+2.1%)	0.9033 (-0.2%)	0.3505 (+9.4%)	0.5922 (+4.1%)
LRMs[5]	None	0.7110 (+3.4%)	0.8966 (-0.9%)	0.2885 (-10.0%)	0.5776 (+1.5%)
Confidence (Ours)	**None**	**0.7219 (+5.0%)**	**0.9196 (+1.6%)**	**0.3643 (+13.7%)**	0.5897 (+3.6%)
Selection + additional weighting: extra scoring after selection					
Robometer-Prog[3]	Frame	0.6944 (+1.0%)	0.8882 (-1.8%)	0.3357 (+4.7%)	0.5884 (+3.4%)
Robometer-Pref[3]	Frame	0.6803 (-1.1%)	0.8945 (-1.1%)	0.2969 (-7.4%)	0.5601 (-1.6%)
PRM-as-Judge[4]	Frame	0.7123 (+3.6%)	0.9053 (+0.1%)	0.3347 (+4.4%)	0.5963 (+4.8%)
LRMs[5]	Frame	0.6955 (+1.1%)	0.8882 (-1.8%)	0.3171 (-1.1%)	0.5815 (+2.2%)
Confidence (Ours)	**Frame**	0.7047 (+2.5%)	**0.9143 (+1.1%)**	0.3398 (+6.0%)	0.5944 (+4.5%)
Confidence (Ours)	**Fr.+Patch**	**0.7265 (+5.6%)**	0.8942 (-1.2%)	**0.3510 (+9.5%)**	**0.6034 (+6.1%)**

Table 5: Paired-bootstrap improvements over EVAC-v1. For each learned scoring method, episode-level paired differences from seeds 42, 3407, and 123 are pooled before bootstrap resampling. Each entry gives the 95% confidence interval followed by the paired mean in parentheses.

Scoring	Weighting	Δ Reconstruction (↑)	Δ Scene (↑)	Δ Motion (↑)	Δ Semantics (↑)
Base EVAC		[-0.133, -0.112] (-0.122)	[-0.038, -0.020] (-0.029)	–	[-0.113, -0.049] (-0.081)
Base EVAC (Warmup v1)		–	–	–	–
Selection-only retraining: no additional weighting after selection					
Random	None	[-0.055, -0.039] (-0.047)	[-0.060, -0.045] (-0.052)	[-0.035, +0.056] (+0.011)	[-0.112, -0.046] (-0.079)
RoboReward[1]	None	[-0.020, -0.013] (-0.017)	[-0.063, -0.052] (-0.058)	[-0.006, +0.038] (+0.015)	**[+0.016, +0.053] (+0.035)**
GVL[2]	None	[+0.018, +0.024] (+0.021)	[-0.001, +0.003] (+0.001)	[-0.008, +0.046] (+0.019)	[+0.003, +0.042] (+0.023)
Robometer-Prog[3]	None	[-0.061, -0.051] (-0.056)	[-0.056, -0.047] (-0.051)	[-0.017, +0.038] (+0.010)	[-0.014, +0.024] (+0.006)
Robometer-Pref[3]	None	[-0.046, -0.035] (-0.040)	[-0.067, -0.054] (-0.060)	[-0.024, +0.025] (+0.000)	[-0.012, +0.028] (+0.008)
PRM-as-Judge[4]	None	[+0.011, +0.018] (+0.015)	[-0.004, +0.001] (-0.001)	[+0.005, +0.056] (+0.030)	[+0.001, +0.040] (+0.021)
LRMs[5]	None	[+0.020, +0.027] (+0.023)	[-0.011, -0.006] (-0.008)	[-0.057, -0.008] (-0.032)	[-0.010, +0.029] (+0.010)
Confidence (Ours)	**None**	**[+0.031, +0.037] (+0.034)**	**[+0.012, +0.018] (+0.015)**	**[+0.018, +0.070] (+0.044)**	[-0.002, +0.038] (+0.018)
Selection + additional weighting: extra scoring after selection					
Robometer-Prog[3]	Frame	[+0.003, +0.011] (+0.007)	[-0.021, -0.012] (-0.017)	[-0.010, +0.041] (+0.015)	[-0.000, +0.038] (+0.019)
Robometer-Pref[3]	Frame	[-0.011, -0.004] (-0.007)	[-0.013, -0.007] (-0.010)	[-0.049, +0.002] (-0.024)	[-0.031, +0.008] (-0.011)
PRM-as-Judge[4]	Frame	[+0.021, +0.028] (+0.024)	[-0.003, +0.004] (+0.001)	[-0.010, +0.039] (+0.014)	[+0.008, +0.046] (+0.027)
LRMs[5]	Frame	[+0.004, +0.011] (+0.008)	[-0.020, -0.013] (-0.016)	[-0.027, +0.020] (-0.003)	[-0.005, +0.033] (+0.014)
Confidence (Ours)	**Frame**	[+0.014, +0.020] (+0.017)	**[+0.007, +0.012] (+0.010)**	[-0.007, +0.045] (+0.019)	[+0.007, +0.044] (+0.026)
Confidence (Ours)	**Fr.+Patch**	**[+0.036, +0.042] (+0.039)**	[-0.013, -0.008] (-0.010)	**[+0.008, +0.054] (+0.031)**	**[+0.011, +0.049] (+0.030)**

Seed-wise detailed results. Tables 6–8 report the complete normalized component metrics for seeds 42, 3407, and 123, respectively. Seed 42 is the default seed. Base EVAC, EVAC-v1, and Random are shared reference results and are repeated in each table for ease of comparison.

Table 6: Detailed normalized active-learning results for seed 42.

Scoring	Weighting	Reconstruction (↑)		Scene (↑)	Semantics (↑)			Motion (↑)		
		PSNR (↑)	SSIM (↑)	Scene Cons. (↑)	Logics (↑)	Sem.-CLIP (↑)	Sem.-BLEU (↑)	Traj-HSD (↑)	Traj-Dyn (↑)	Traj-nDTW (↑)
Base EVAC		0.5532	0.5778	0.8757	0.4298	0.8523	0.1799	0.0007	0.0001	0.0006
Base EVAC (Warmup v1)		0.6446	0.7309	0.9047	0.5537	0.8824	0.2708	0.1045	0.0721	0.1439
Selection-only retraining: no additional weighting after selection										
Random	None	0.5968	0.6849	0.8524	0.3554	0.8689	0.2372	0.1067	0.0725	0.1518
RoboReward[1]	None	0.6276	0.7067	0.8712	0.6322	0.8895	0.2832	0.1076	0.0662	0.1501
GVL[2]	None	0.6550	0.7449	0.9136	0.5826	0.8885	0.2733	0.1033	0.0607	0.1381
Robometer-Prog[3]	None	0.5547	0.6711	0.8208	0.5496	0.8800	0.2680	0.1166	0.0774	**0.1805**
Robometer-Pref[3]	None	0.6047	0.7005	0.9005	0.6157	0.8866	0.2729	0.0912	0.0492	0.1242
PRM-as-Judge[4]	None	0.6523	0.7641	0.9102	**0.6405**	**0.8926**	**0.3023**	0.1095	0.0696	0.1495
LRMs[5]	None	0.6577	0.7528	0.8869	0.5785	0.8803	0.2550	0.0960	0.0595	0.1422
Confidence (Ours)	**None**	**0.6838**	**0.7797**	**0.9322**	0.6198	0.8894	0.2754	**0.1313**	**0.0921**	0.1780
Selection + additional weighting: extra scoring after selection										
Robometer-Prog[3]	Frame	0.5932	0.6981	0.8399	0.6074	0.8883	**0.2976**	0.1151	0.0703	0.1655
Robometer-Pref[3]	Frame	0.6412	0.7324	0.8948	0.6033	0.8868	0.2805	0.0981	0.0678	0.1291
PRM-as-Judge[4]	Frame	0.6695	0.7733	0.9193	0.6281	0.8880	0.2937	0.1000	0.0592	0.1443
LRMs[5]	Frame	0.6716	0.7616	0.8958	0.5785	0.8896	0.2733	0.0985	0.0623	0.1497
Confidence (Ours)	**Frame**	0.6737	0.7650	**0.9295**	**0.6405**	0.8867	0.2843	**0.1179**	0.0728	**0.1706**
Confidence (Ours)	**Fr.+Patch**	**0.6795**	**0.7746**	0.9014	0.6240	**0.8945**	0.2859	0.1066	**0.0761**	0.1665

Table 7: Detailed normalized active-learning results for seed 3407.

Scoring	Weighting	Reconstruction (↑)		Scene (↑)	Semantics (↑)			Motion (↑)		
		PSNR (↑)	SSIM (↑)	Scene Cons. (↑)	Logics (↑)	Sem.-CLIP (↑)	Sem.-BLEU (↑)	Traj-HSD (↑)	Traj-Dyn (↑)	Traj-nDTW (↑)
Base EVAC		0.5532	0.5778	0.8757	0.4298	0.8523	0.1799	0.0007	0.0001	0.0006
Base EVAC (Warmup v1)		0.6446	0.7309	0.9047	0.5537	0.8824	0.2708	0.1045	0.0721	0.1439
Selection-only retraining: no additional weighting after selection										
Random	None	0.5968	0.6849	0.8524	0.3554	0.8689	0.2372	0.1067	0.0725	0.1518
RoboReward[1]	None	0.6262	0.7290	0.7985	**0.6488**	0.8863	0.2775	0.1089	0.0732	0.1554
GVL[2]	None	0.6546	0.7503	0.8959	0.6157	0.8892	0.2868	**0.1200**	0.0759	0.1681
Robometer-Prog[3]	None	0.6492	0.7490	0.8857	0.6033	0.8814	0.2580	0.1148	0.0715	0.1491
Robometer-Pref[3]	None	0.6442	0.7535	0.8779	0.6033	0.8895	0.2877	0.1066	0.0671	0.1565
PRM-as-Judge[4]	None	**0.6714**	**0.7728**	0.9020	0.5950	**0.8896**	0.2873	0.1197	**0.0784**	**0.1757**
LRMs[5]	None	0.6619	0.7658	0.8991	0.5331	0.8817	0.2672	0.0925	0.0564	0.1299
Confidence (Ours)	**None**	0.6521	0.7443	**0.9029**	0.5537	0.8891	**0.2953**	0.0998	0.0618	0.1469
Selection + additional weighting: extra scoring after selection										
Robometer-Prog[3]	Frame	0.6785	0.7739	**0.9148**	0.5661	0.8896	0.2925	0.1042	0.0683	0.1454
Robometer-Pref[3]	Frame	0.6329	0.7343	0.9000	0.5372	0.8875	0.2796	0.0867	0.0542	0.1241
PRM-as-Judge[4]	Frame	0.6544	0.7606	0.8816	0.6364	0.8861	0.2681	**0.1276**	**0.0949**	**0.1827**
LRMs[5]	Frame	0.6391	0.7323	0.8984	0.5868	0.8872	0.2884	0.1047	0.0582	0.1508
Confidence (Ours)	**Frame**	0.6644	0.7518	0.9127	0.6074	0.8887	0.2734	0.1121	0.0704	0.1663
Confidence (Ours)	**Fr.+Patch**	**0.6798**	**0.7822**	0.8948	**0.6612**	**0.8926**	**0.2979**	0.1161	0.0679	0.1665

Table 8: Detailed normalized active-learning results for seed 123.

Scoring	Weighting	Reconstruction (↑)		Scene (↑)	Semantics (↑)			Motion (↑)		
		PSNR (↑)	SSIM (↑)	Scene Cons. (↑)	Logics (↑)	Sem.-CLIP (↑)	Sem.-BLEU (↑)	Traj-HSD (↑)	Traj-Dyn (↑)	Traj-nDTW (↑)
Base EVAC		0.5532	0.5778	0.8757	0.4298	0.8523	0.1799	0.0007	0.0001	0.0006
Base EVAC (Warmup v1)		0.6446	0.7309	0.9047	0.5537	0.8824	0.2708	0.1045	0.0721	0.1439
Selection-only retraining: no additional weighting after selection										
Random	None	0.5968	0.6849	0.8524	0.3554	0.8689	0.2372	0.1067	0.0725	0.1518
RoboReward[1]	None	0.6260	0.7117	0.8712	**0.6364**	0.8825	0.2838	0.1174	0.0685	0.1607
GVL[2]	None	0.6795	0.7686	0.9075	0.5992	0.8918	0.2946	0.1109	0.0698	0.1713
Robometer-Prog[3]	None	0.5388	0.6272	0.8532	0.5992	0.8815	0.2620	0.0955	0.0582	0.1282
Robometer-Pref[3]	None	0.5267	0.6546	0.7544	0.5248	0.8816	0.2479	0.1160	0.0752	**0.1764**
PRM-as-Judge[4]	None	0.6164	0.7372	0.8977	0.5744	0.8817	0.2659	0.1168	0.0758	0.1565
LRMs[5]	None	0.6655	0.7624	0.9037	0.6116	0.8862	**0.3046**	0.0952	0.0603	0.1335
Confidence (Ours)	**None**	**0.6878**	**0.7836**	**0.9237**	0.6033	**0.8924**	0.2889	**0.1232**	**0.0896**	0.1702
Selection + additional weighting: extra scoring after selection										
Robometer-Prog[3]	Frame	0.6600	0.7626	0.9099	0.5909	0.8899	0.2734	0.1097	0.0706	**0.1582**
Robometer-Pref[3]	Frame	0.6210	0.7199	0.8887	0.4463	0.8804	0.2395	0.1073	0.0741	0.1493
PRM-as-Judge[4]	Frame	0.6514	0.7645	**0.9150**	**0.6033**	0.8862	0.2766	0.0971	0.0643	0.1339
LRMs[5]	Frame	0.6282	0.7404	0.8706	0.5826	0.8839	0.2634	**0.1128**	0.0700	0.1443
Confidence (Ours)	**Frame**	0.6403	0.7332	0.9008	**0.6033**	0.8867	0.2790	0.1019	0.0620	0.1455
Confidence (Ours)	**Fr.+Patch**	**0.6723**	**0.7705**	0.8864	0.5826	**0.8903**	**0.3017**	0.1126	**0.0837**	0.1570

B.3 Qualitative Evolution from Base EVAC to EVAC-v2

We further visualize how world-model predictions evolve throughout the post-training pipeline. We select six representative RoboTwin2.0 episodes covering object placement, cabinet interaction, block stacking, bowl stacking, and switch manipulation. These examples expose a consistent progression from severe zero-shot cross-embodiment failure to target-domain adaptation and, finally, confidence-guided refinement.

Per-episode progression. Table 9 summarizes the four aggregated evaluation dimensions for the six selected episodes. All relative changes are computed with respect to EVAC-v1, which serves as the common post-warmup baseline. Across the selected examples, Base EVAC performs substantially worse under zero-shot cross-embodiment transfer, while EVAC-v1 already recovers much of the target-domain prediction quality. Confidence-guided EVAC-v2 further improves most dimensions, with frame-and-patch weighting achieving the strongest Reconstruction in all six episodes.

For this per-episode analysis, Semantics is aggregated from normalized Sem.-CLIP and Sem.-BLEU only; the binary Logics score is omitted because a single $0/1$ decision would disproportionately affect an individual episode.

Table 9: **Aggregated evolution on six representative RoboTwin2.0 episodes.** Each entry reports the normalized aggregate followed by its relative change from EVAC-v1. Green and red percentages indicate improvement and degradation, respectively. For Motion, relative changes are omitted when the EVAC-v1 score is zero. Green, yellow, and red cells indicate the best, second-best, and third-best values within each episode.

Episode	Variant	Reconstruction (↑)	Scene (↑)	Semantics (↑)	Motion (↑)
place burger fries (ep130)	Base EVAC	0.4592 (-27.3%)	0.7890 (-10.0%)	0.3591 (-28.9%)	0.0000 (-100.0%)
	EVAC-v1	0.6318 (0.0%)	0.8770 (0.0%)	0.5050 (0.0%)	0.4570 (0.0%)
	EVAC-v2 Frame	0.6618 (+4.7%)	**0.9430** (+7.5%)	0.5221 (+3.4%)	1.0830 (+137.0%)
	EVAC-v2 Fr.+Patch	**0.7020** (+11.1%)	0.9370 (+6.8%)	**0.5486** (+8.6%)	**1.3080** (+186.2%)
place empty cup (ep294)	Base EVAC	0.4605 (-38.6%)	0.7120 (-27.0%)	0.5454 (+2.0%)	0.0000 (-100.0%)
	EVAC-v1	0.7503 (0.0%)	**0.9750** (0.0%)	0.5349 (0.0%)	0.5600 (0.0%)
	EVAC-v2 Frame	0.7353 (-2.0%)	0.9700 (-0.5%)	0.2604 (-51.3%)	0.4150 (-25.9%)
	EVAC-v2 Fr.+Patch	**0.7558** (+0.7%)	0.9610 (-1.4%)	**0.6015** (+12.5%)	**0.5900** (+5.4%)
put object cabinet (ep282)	Base EVAC	0.4372 (-23.2%)	0.7770 (-10.4%)	0.4856 (-0.5%)	0.0000 (–)
	EVAC-v1	0.5693 (0.0%)	0.8670 (0.0%)	0.4879 (0.0%)	0.0000 (0.0%)
	EVAC-v2 Frame	0.6178 (+8.5%)	**0.9260** (+6.8%)	0.5264 (+7.9%)	0.0000 (–)
	EVAC-v2 Fr.+Patch	**0.6207** (+9.0%)	0.8950 (+3.2%)	**0.5651** (+15.8%)	0.0000 (–)
stack blocks two (ep325)	Base EVAC	0.3793 (-38.4%)	0.7970 (-10.4%)	0.4391 (-4.6%)	0.0000 (-100.0%)
	EVAC-v1	0.6160 (0.0%)	0.8900 (0.0%)	0.4603 (0.0%)	1.1470 (0.0%)
	EVAC-v2 Frame	0.6937 (+12.6%)	0.9220 (+3.6%)	0.4606 (+0.1%)	1.1770 (+2.6%)
	EVAC-v2 Fr.+Patch	**0.7053** (+14.5%)	**0.9240** (+3.8%)	**0.4622** (+0.4%)	**1.2470** (+8.7%)
stack bowls three (ep087)	Base EVAC	0.5532 (-15.5%)	0.8000 (+12.5%)	0.4340 (+4.8%)	0.0000 (-100.0%)
	EVAC-v1	0.6545 (0.0%)	0.7110 (0.0%)	0.4143 (0.0%)	0.3390 (0.0%)
	EVAC-v2 Frame	0.7407 (+13.2%)	**0.9060** (+27.4%)	0.4403 (+6.3%)	0.0000 (-100.0%)
	EVAC-v2 Fr.+Patch	**0.7555** (+15.4%)	0.8880 (+24.9%)	**0.4687** (+13.1%)	**0.4550** (+34.2%)
turn switch (ep310)	Base EVAC	0.5578 (-19.4%)	0.8260 (-11.2%)	0.4431 (-31.5%)	0.0000 (–)
	EVAC-v1	0.6923 (0.0%)	0.9300 (0.0%)	0.6464 (0.0%)	0.0000 (0.0%)
	EVAC-v2 Frame	0.7028 (+1.5%)	**0.9560** (+2.8%)	0.6869 (+6.3%)	0.0000 (–)
	EVAC-v2 Fr.+Patch	**0.7350** (+6.2%)	0.9410 (+1.2%)	**0.8269** (+27.9%)	0.0000 (–)

Representative visual evolution. Figures 15–20 provide matched temporal comparisons for the six representative episodes. Base EVAC is strongly out-of-distribution under the RoboTwin2.0 embodiment and often fails to preserve a recognizable robot arm or coherent interaction trajectory. After warmup, EVAC-v1 recovers the robot embodiment and basic scene structure, but noticeable color drift, object corruption, temporal flickering, and interaction errors remain. EVAC-v2 with frame weighting substantially improves the overall rollout, although local object geometry and gripper–object interactions can still fail. Frame-and-patch weighting further concentrates learning on these difficult local regions and most clearly improves manipulated objects, gripper geometry, and fine interaction details in the selected cases.

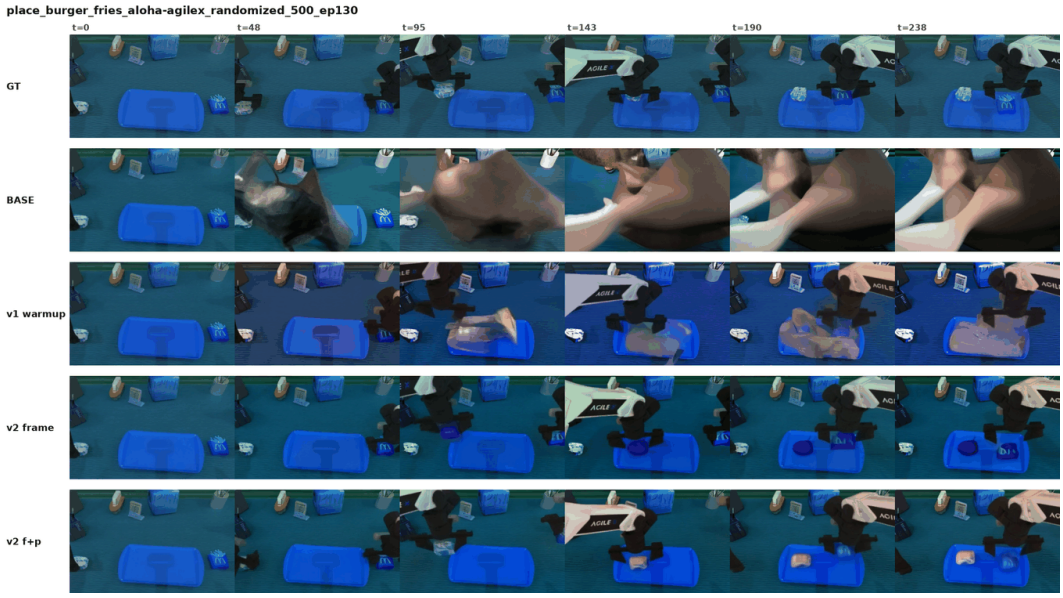


Figure 15: **Training evolution on `place_burger_fries` (ep130).** EVAC-v1 develops a strong blue color shift and the tray contents collapse from the middle of the rollout. EVAC-v2 Frame removes most tray corruption, but the left arm hallucinates an object instead of grasping the burger and the object held by the right arm is visibly deformed. EVAC-v2 Fr.+Patch correctly grasps and places the burger while preserving the fries most faithfully.

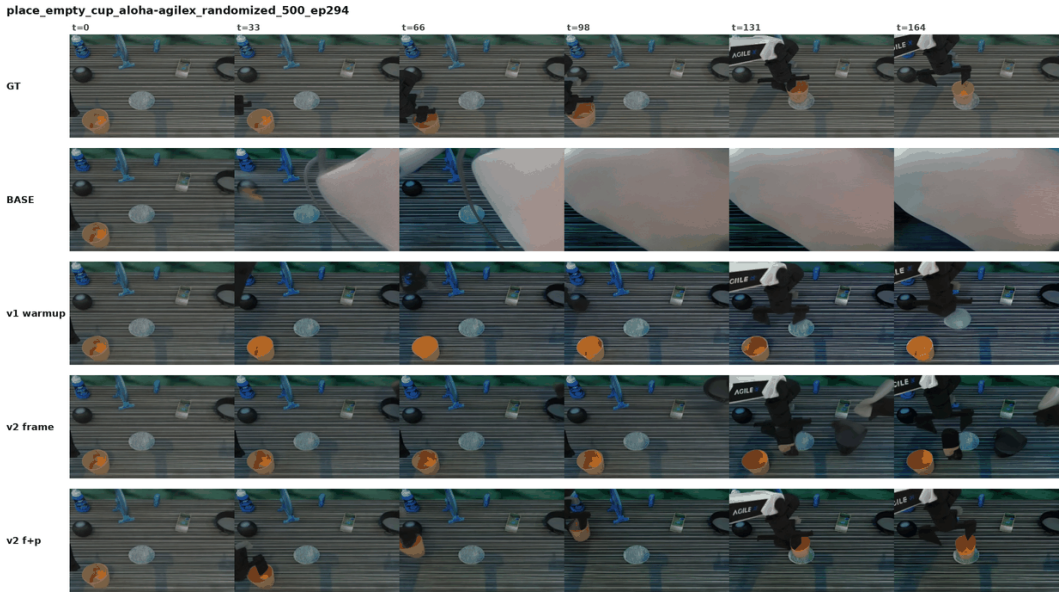


Figure 16: **Training evolution on `place_empty_cup` (ep294).** In both EVAC-v1 and EVAC-v2 Frame, the arm only partially appears near the image boundary and fails to grasp the cup at the beginning; EVAC-v2 Frame additionally deteriorates toward the end of the rollout. EVAC-v2 Fr.+Patch correctly grasps the cup from the boundary, places it on the plate, and maintains a coherent scene throughout the prediction.

Figure 17: **Training evolution on `put_object_cabinet` (ep282).** EVAC-v1 shows moderate color bias and eventually produces a floating arm together with a corrupted cabinet. EVAC-v2 Frame does not resolve these failures: the blue box in the upper-left region disappears and the arm is generated at an incorrect location. EVAC-v2 Fr.+Patch preserves the objects, cabinet structure, and robot geometry most consistently.

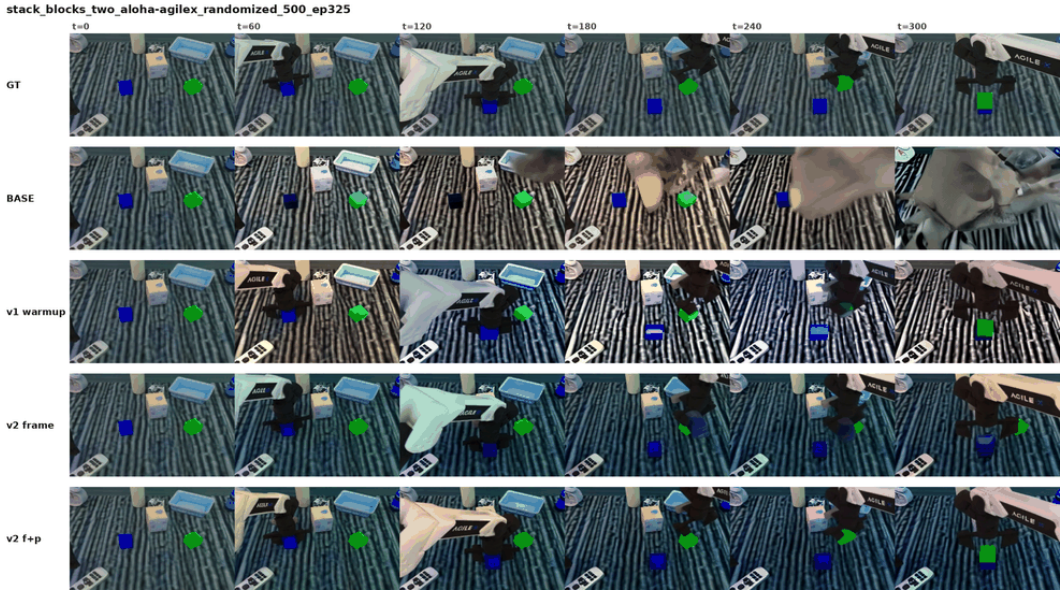


Figure 18: **Training evolution on `stack_blocks_two` (ep325).** EVAC-v1 exhibits strong color flickering, and the lower blue block disappears when the green block is lifted and stacked. EVAC-v2 Frame largely removes the color shift but fails to keep the green block attached to the gripper and consequently misses the stacking interaction. EVAC-v2 Fr.+Patch resolves both the object-persistence and grasping failures.

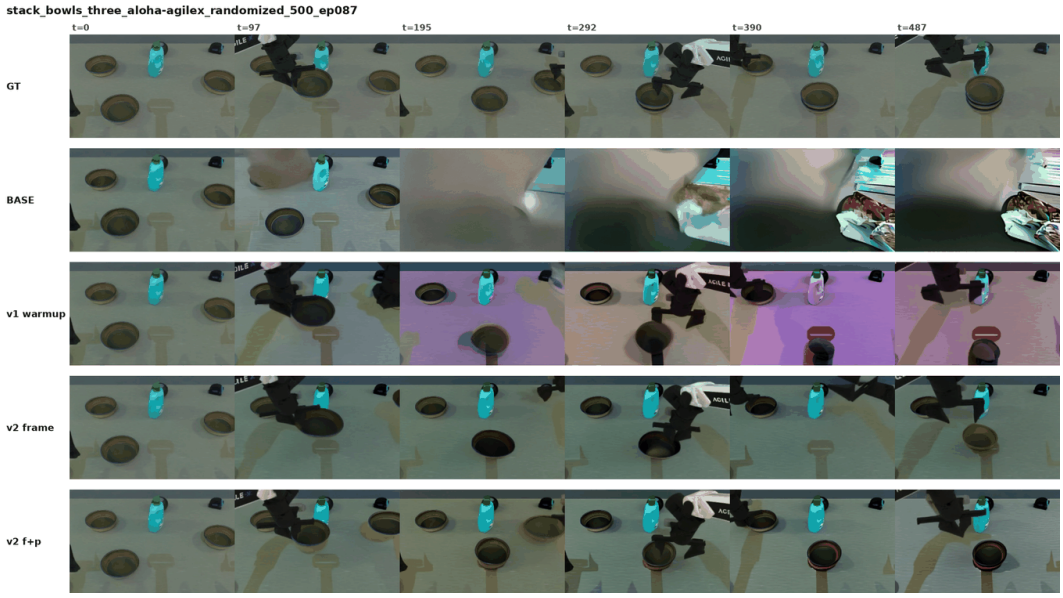


Figure 19: **Training evolution on `stack_bowls_three` (ep087).** EVAC-v1 suffers from pronounced color drift and repeated red flickering, followed by severe object corruption late in the rollout. EVAC-v2 Frame removes the color shift but still produces substantial bowl deformation, disappearance, and reappearance. EVAC-v2 Fr.+Patch better preserves the bowl geometry and maintains object persistence through the later frames.

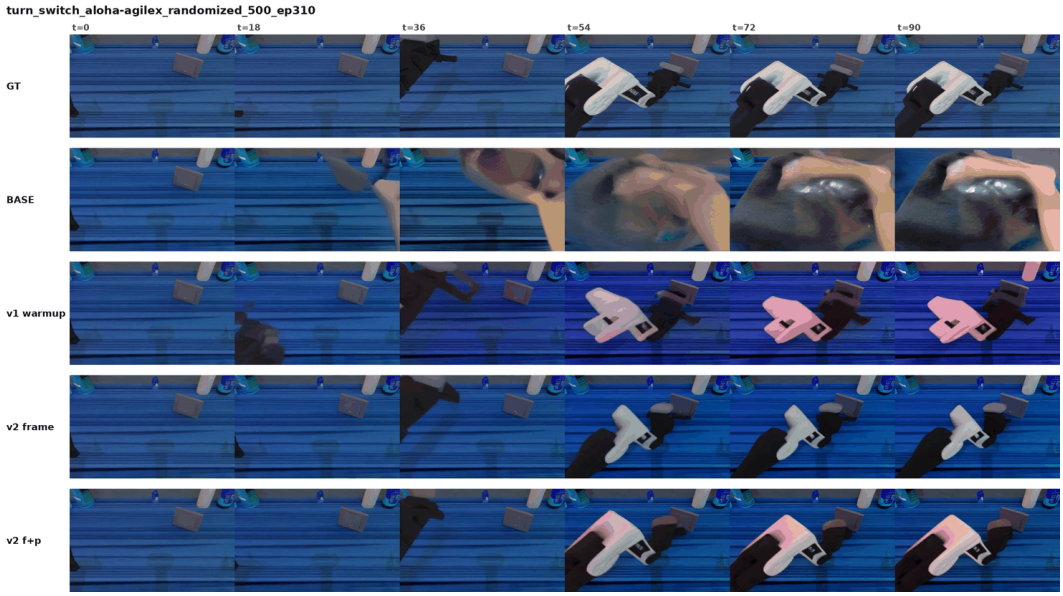


Figure 20: **Training evolution on `turn_switch` (ep310).** EVAC-v1 produces a reddish robot arm, a bluish background, and implausible floating-arm geometry. EVAC-v2 Frame substantially reduces the color bias, but the arm shape remains distorted. EVAC-v2 Fr.+Patch further improves the local arm geometry and produces an embodiment substantially closer to the ground truth.

Detailed EWMBench measurements. Table 10 reports the normalized component metrics underlying these representative examples. Overall, the transition from Base EVAC to EVAC-v1 substantially improves target-domain prediction quality, while confidence-guided EVAC-v2 provides further gains across reconstruction, scene, semantic, and trajectory dimensions. Among the selected episodes, frame-and-patch weighting gives the strongest overall progression, particularly for semantic quality and Motion when valid trajectory signals are available. These quantitative trends are consistent with the qualitative improvements shown above.

Table 10: **Detailed normalized EWMBench results for the qualitative evolution study.** All metrics are higher-is-better. Green, yellow, and red indicate the best, second-best, and third-best distinct values within each episode, respectively.

Episode	Variant	Reconstruction (↑)		Scene (↑)	Semantics (↑)		Motion (↑)		
		PSNR (↑)	SSIM (↑)	Scene Cons. (↑)	Sem.-CLIP (↑)	Sem.-BLEU (↑)	Traj-HSD (↑)	Traj-Dyn (↑)	Traj-nDTW (↑)
place burger fries (ep130)	Base EVAC	0.4413	0.477	0.789	0.7183	0.000	0.000	0.000	0.000
	EVAC-v1	0.5957	0.668	0.877	0.7879	0.222	0.095	0.001	0.361
	EVAC-v2 Frame	0.6777	0.646	**0.943**	0.8121	0.232	0.545	**0.024**	0.514
	EVAC-v2 Fr.+Patch	**0.6980**	**0.706**	0.937	**0.8182**	**0.279**	**0.761**	0.002	**0.545**
place empty cup (ep294)	Base EVAC	0.5100	0.411	0.712	0.8448	0.246	0.000	0.000	0.000
	EVAC-v1	**0.7217**	0.779	**0.975**	0.8038	0.266	0.108	0.365	0.087
	EVAC-v2 Frame	0.7117	0.759	0.970	0.3568	0.164	0.097	0.238	0.080
	EVAC-v2 Fr.+Patch	0.7197	**0.792**	0.961	**0.8760**	**0.327**	**0.117**	**0.379**	**0.094**
put object cabinet (ep282)	Base EVAC	0.4593	0.415	0.777	0.8081	0.163	0.000	0.000	0.000
	EVAC-v1	0.5377	0.601	0.867	0.7927	0.183	0.000	0.000	0.000
	EVAC-v2 Frame	0.5727	**0.663**	**0.926**	0.8708	0.182	0.000	0.000	0.000
	EVAC-v2 Fr.+Patch	**0.5903**	0.651	0.895	**0.9262**	**0.204**	0.000	0.000	0.000
stack blocks two (ep325)	Base EVAC	0.4537	0.305	0.797	0.7823	0.096	0.000	0.000	0.000
	EVAC-v1	0.5650	0.667	0.890	**0.8246**	0.096	0.303	0.205	0.639
	EVAC-v2 Frame	0.6313	0.756	0.922	0.8052	**0.116**	0.304	0.218	0.655
	EVAC-v2 Fr.+Patch	**0.6447**	**0.766**	**0.924**	0.8244	0.100	**0.319**	**0.259**	**0.669**
stack bowls three (ep087)	Base EVAC	0.4953	0.611	0.800	0.7860	0.082	0.000	0.000	0.000
	EVAC-v1	0.5630	0.746	0.711	**0.8287**	0.000	**0.063**	0.161	0.115
	EVAC-v2 Frame	0.6803	**0.801**	**0.906**	0.8116	0.069	0.000	0.000	0.000
	EVAC-v2 Fr.+Patch	**0.7160**	0.795	0.888	0.8204	**0.117**	0.062	**0.225**	**0.168**
turn switch (ep310)	Base EVAC	0.5507	0.565	0.826	0.7632	0.123	0.000	0.000	0.000
	EVAC-v1	0.6587	0.726	0.930	0.9108	0.382	0.000	0.000	0.000
	EVAC-v2 Frame	0.6807	0.725	**0.956**	0.9308	0.443	0.000	0.000	0.000
	EVAC-v2 Fr.+Patch	**0.7170**	**0.753**	0.941	**0.9318**	**0.722**	0.000	0.000	0.000

B.4 More Results of Qualitative Confidence Visualization

We provide additional qualitative examples to complement the confidence analysis in the main text. Figures 21 and 22 show the two episodes with the highest patch-level Spearman correlations, reaching 0.713 and 0.695, respectively. Across different future frames, the predicted risk maps consistently concentrate around manipulators, operated objects, and interaction regions, and largely agree with both pixel- and latent-space oracle errors. The corresponding risk overlays further illustrate that confidence provides spatially localized signals rather than merely reflecting global video quality.

Figures 23 and 24 show two challenging cases with lower patch-level Spearman correlations of 0.241 and 0.265. In these examples, oracle errors are more spatially diffuse and the predicted risk maps do not precisely reproduce their local boundaries, although they still respond to many of the dominant high-error regions. Together, the best and worst cases illustrate both the localization capability and the remaining limitations of the confidence probe, consistent with our quantitative finding that confidence reliably captures the relative severity and spatial concentration of prediction errors, even when the predicted risk map does not exactly reproduce the oracle error boundaries.

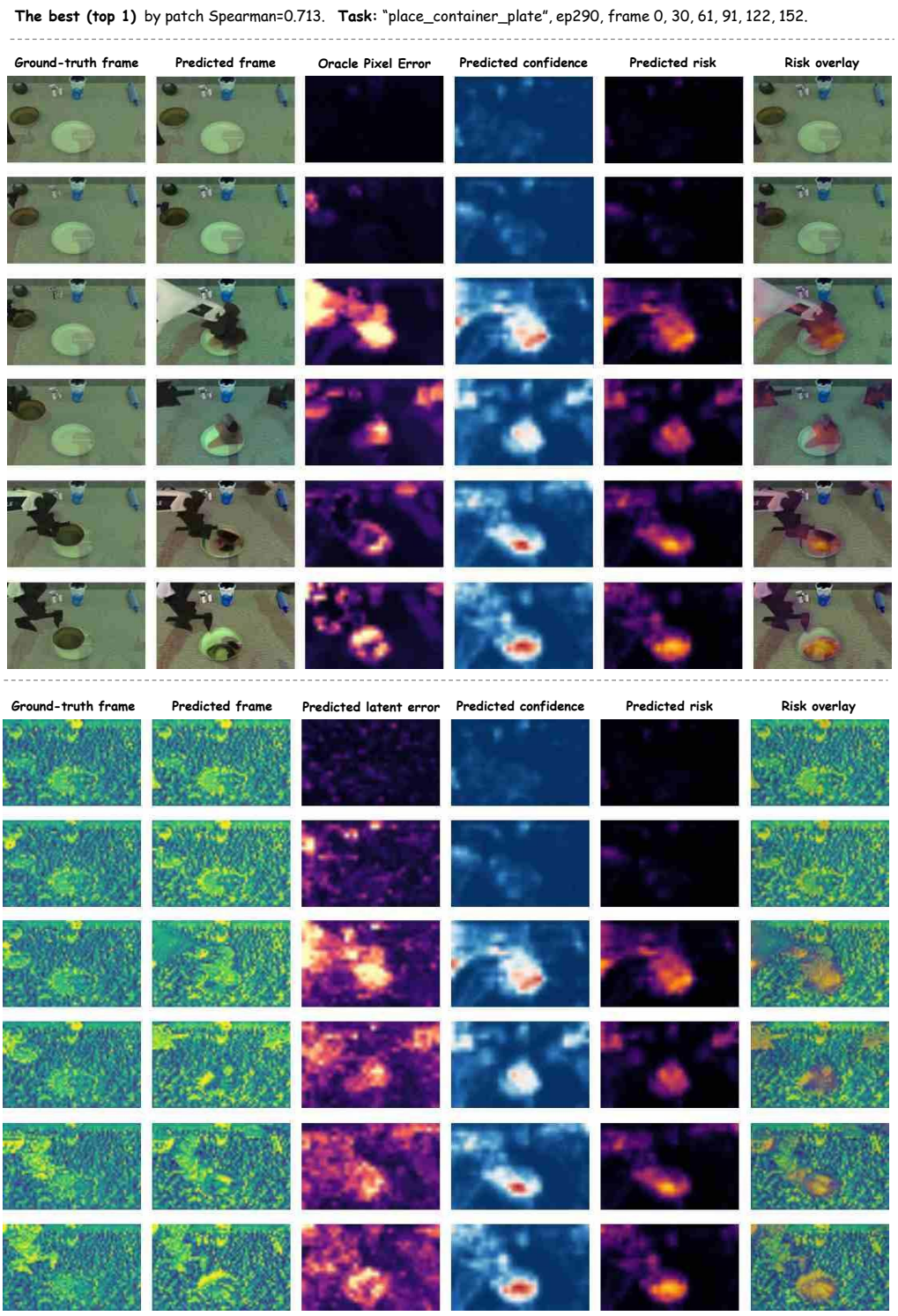


Figure 21: **More results of qualitative confidence visualization.** Best-ranked example (top 1) by patch-level Spearman correlation.

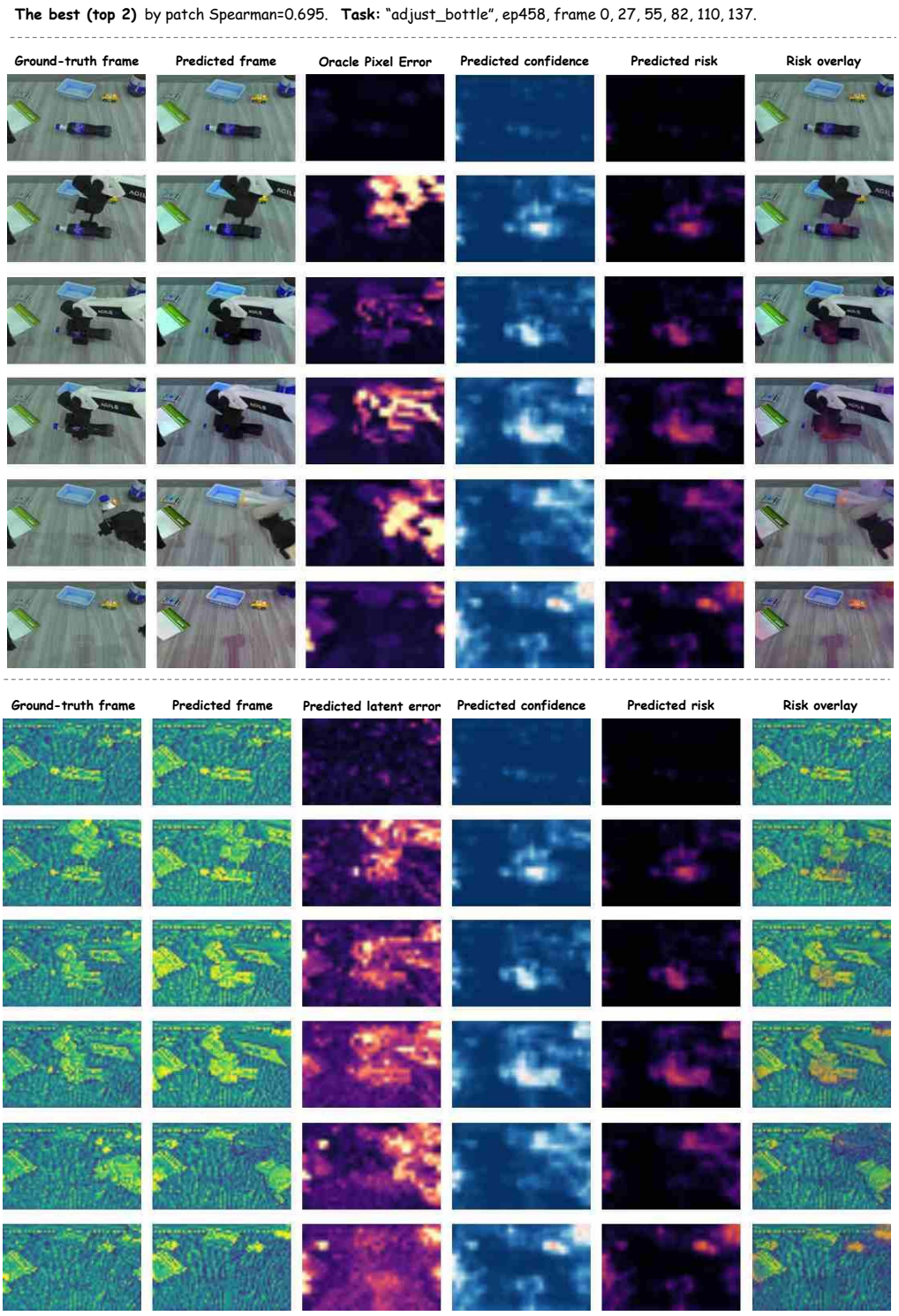


Figure 22: **More results of qualitative confidence visualization.** Best-ranked example (top 2) by patch-level Spearman correlation.

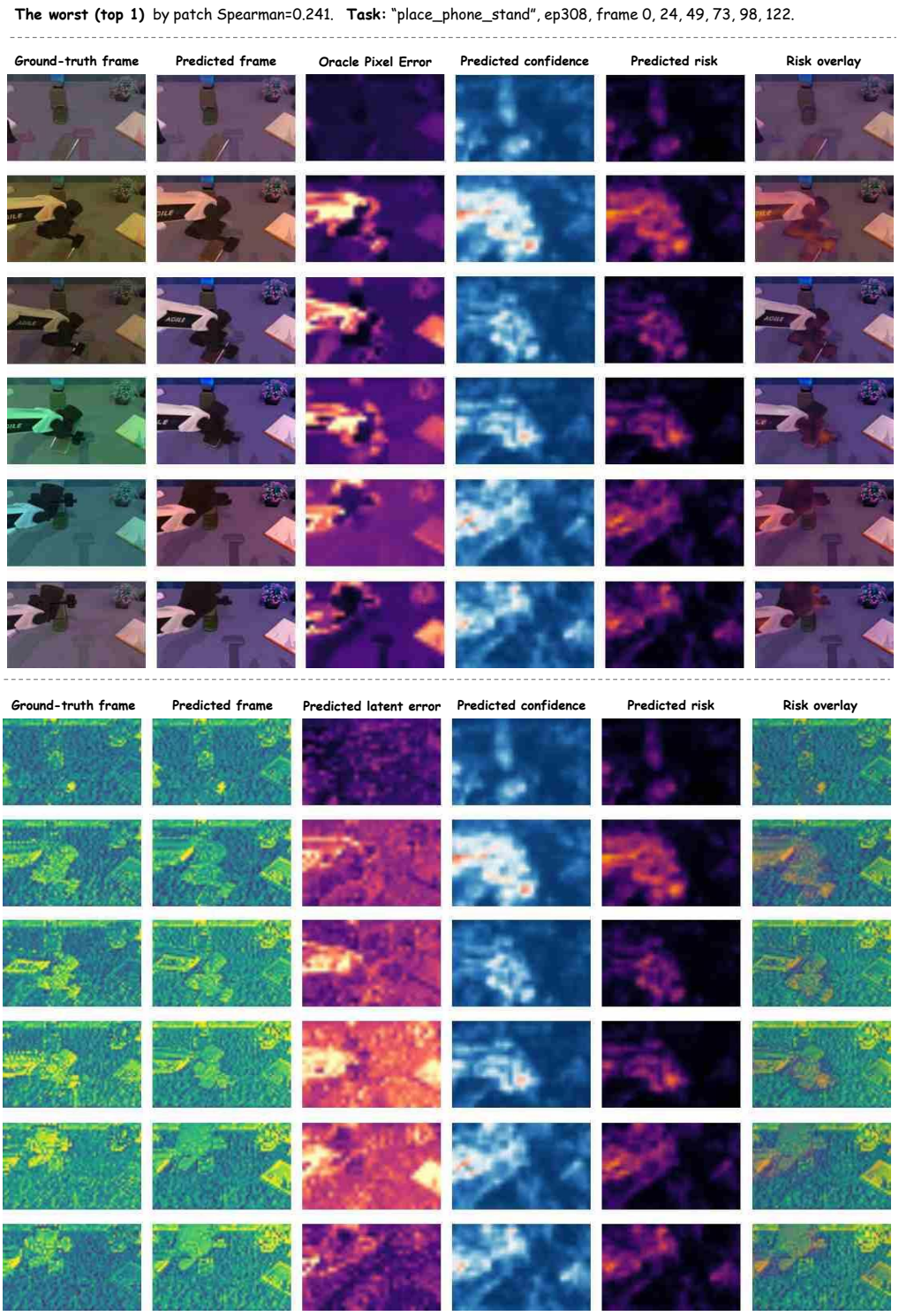


Figure 23: **More results of qualitative confidence visualization.** Lowest-ranked example (top 1) by patch-level Spearman correlation.

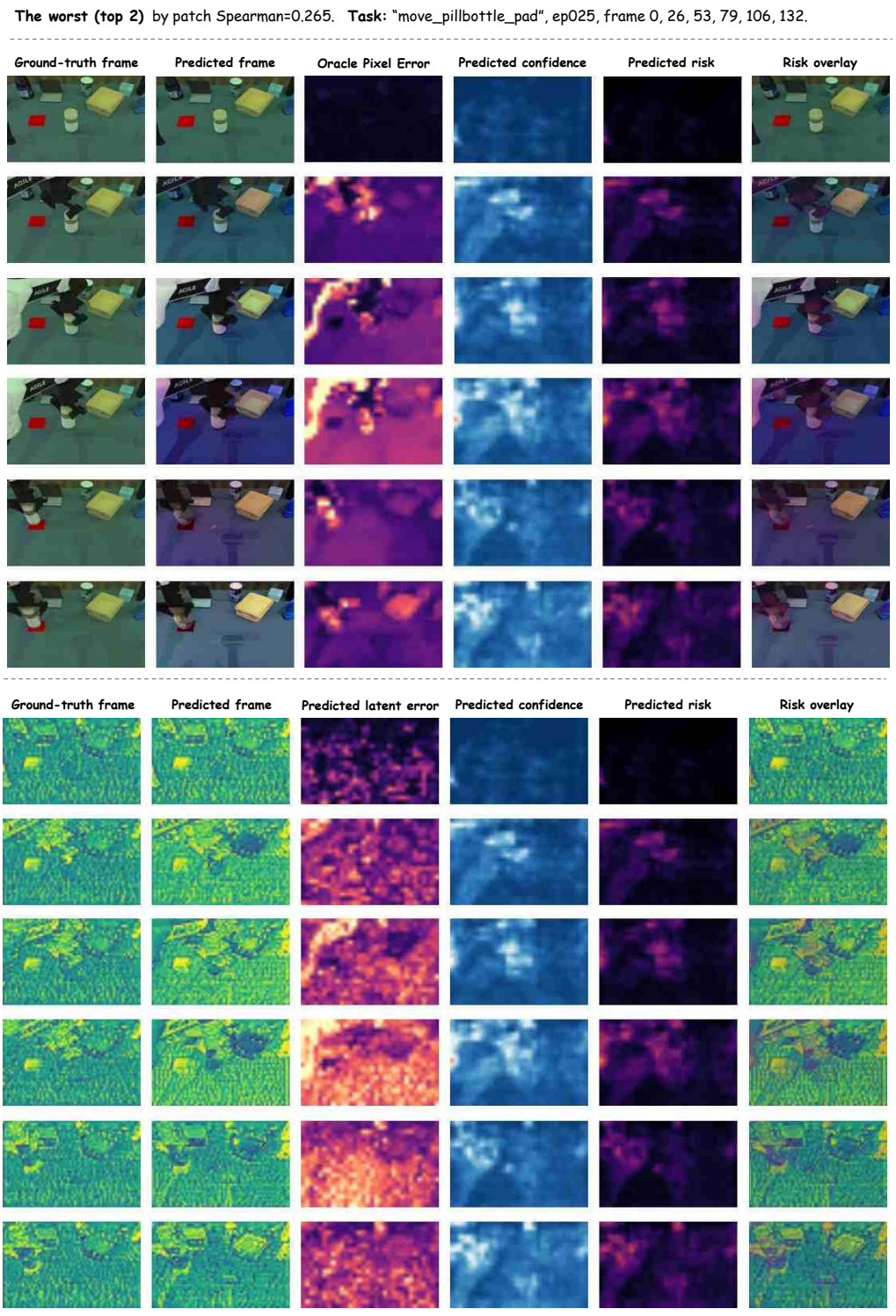


Figure 24: **More results of qualitative confidence visualization.** Lowest-ranked example (top 2) by patch-level Spearman correlation.