

When Plans Change, Remember: Replanning-Guided Visual Memory for Vision-Language-Action Models

Victoria Smelova¹, Daniil Kazachkov¹, Georgiy Kazantsev²
 Iuliia Tsvetkova³, Egor Cherepanov^{1,4,5}, Nikita Kachaev^{4,5}
 Artem Latyshev^{1,4,5}, Aleksandr I. Panov^{1,5}, Alexey K. Kovalev¹

¹MIRAI ²Independent Researcher ³Central University ⁴AXXX ⁵CogAI Lab

Abstract

Long-horizon, multi-stage Vision-Language-Action (VLA) policies often require visual information that is no longer present in the current observation. Retaining the complete observation history is computationally expensive and highly redundant, motivating selective visual memory. We introduce **RePlanMem-VLA** (*Replanning-Guided Keyframe Memory for VLA*), which turns a policy’s own plan revisions into an online memory-write signal. Under receding-horizon execution, consecutive action chunks predict some of the same future actions. RePlanMem-VLA treats unusually large disagreement between these temporally aligned predictions as evidence of a task-relevant event. We first compare 35 disagreement measures across single-task decoder-only and Flow-Matching Transformers trained from scratch, then validate the selected signals on pretrained OpenVLA-OFT and $\pi_{0.5}$ policies over MIKASA-Robo-VLA tasks. The resulting detectors achieve ROC-AUC of 0.981–0.997. We then use the selected signals in RePlanMem-VLA to trigger updates to a bounded visual memory, with a robot-masked write filter suppressing redundant frames. In the evaluated memory-retention experiments, RePlanMem-VLA preserves task-relevant observations until the later decision point in 100% of episodes. Real-robot replay data further show that event selection and redundancy suppression extend beyond simulation. These results establish plan revision as a policy-internal signal for controlling what a VLA remembers.

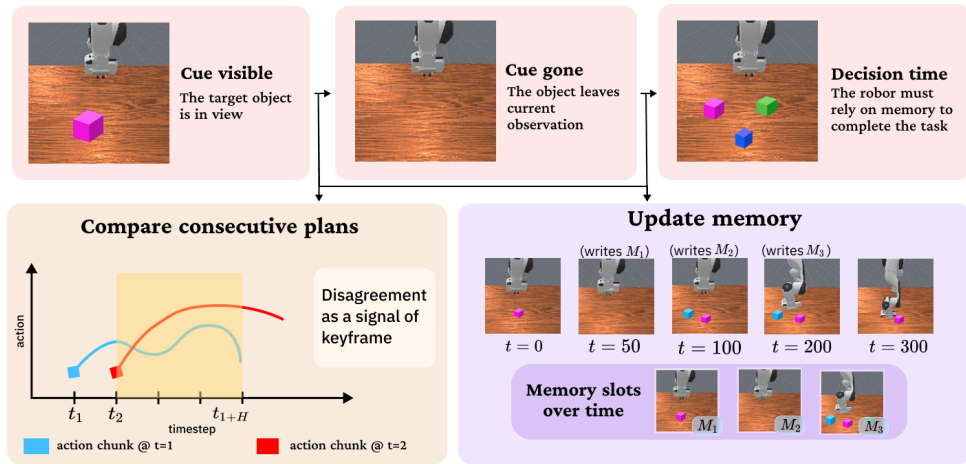


Figure 1: **RePlanMem-VLA**. Task-relevant cues may disappear from the current observation before they are needed. When consecutive replans disagree on their overlapping actions, the important observation is stored in memory and remains available after the cue is no longer visible.

1 Introduction

Long-horizon manipulation often requires VLA policies to use visual information that has left the current observation, including object properties (Chen et al., 2026a; Shi et al., 2026), earlier environment states (Chen et al., 2026a; Shi et al., 2026), and task progress (Buamane et al., 2026; Feng et al., 2026). Retaining the full history preserves these cues but increases context length and computation while adding visual redundancy (Jang et al., 2025; Qiu et al., 2026; Wang et al., 2026). We therefore seek a small set of informative observations.

We call these observations *keyframes*: they preserve cues needed later (Shi et al., 2026; Lei et al., 2026) or mark task-phase transitions (Buamane et al., 2026; Yi et al., 2026). Prior work selects them with vision-language models (Lei et al., 2026), waypoint heuristics (Shi et al., 2023), learned retrieval (Sridhar et al., 2026), progress estimates (Liu et al., 2026), or event detectors (Zeng et al., 2026b). We investigate whether the policy’s own plan revisions can provide an online signal for selecting these observations.

Under receding-horizon execution, consecutive action chunks predict some of the same future actions, while the new plan incorporates the latest observation (Fig. 1). Comparing these temporally aligned predictions reveals how the policy revises its intended behavior after receiving new visual information. **Our central hypothesis is that task-relevant observations induce unusually large revisions in predictions for the same future actions.** This motivates using plan disagreement as a policy-internal signal to identify candidate keyframes for visual memory.

We introduce **RePlanMem-VLA** (*Replanning-Guided Keyframe Memory for VLA*), which derives its selection signal from revisions between overlapping action predictions already produced during normal replanning. Selected keyframes become part of the policy’s visual context, keeping past observations available for subsequent action prediction, while a robot-masked write filter suppresses redundant frames, reducing repeated overwrites. At deployment, the trigger requires neither keyframe annotations nor privileged simulator state.

Our experiments are conducted on MIKASA-Robo-VLA (Cherepanov et al., 2026b), an extension of MIKASA-Robo (Cherepanov et al., 2026a) for evaluating memory-intensive manipulation with VLA policies. We first screen 35 disagreement metrics using decoder-only and Flow-Matching Transformers (Lipman et al., 2023) trained from scratch. We then evaluate detectors based on the selected signals with pretrained OpenVLA-OFT (Kim et al., 2025) and $\pi_{0.5}$ (Physical Intelligence et al., 2025) policies, obtaining ROC-AUC of 0.981–0.997. In the evaluated memory-retention experiments, RePlanMem-VLA preserves task-relevant observations until the later decision point in 100% of episodes, while robot-masked filtering reduces redundant overwrites. Real-robot replay data further support its event-selection and redundancy-suppression behavior beyond simulation (Fig. 2).

Our main contributions are as follows:

1. We introduce **RePlanMem-VLA**, which identifies keyframes from policy-internal revisions of predictions for the same future actions across consecutive replanning steps (Section 3.3).
2. We **identify disagreement measures that best align with keyframe events** through systematic analysis across decoder-only and Flow-Matching Transformer policies, and verify their transfer to pretrained OpenVLA-OFT and $\pi_{0.5}$ policies (Sections 4.1 and 4.2).
3. We develop a **bounded visual-memory mechanism** where plan disagreement determines when to write and redundancy-aware filtering, including robot-masked clustering, determines what to retain, substantially reducing redundant updates while preserving task-relevant observations (Sections 3 and 4.3).

2 Related Work

Memory for long-horizon, partially observable policies Long-horizon manipulation often requires access to information from earlier in the episode. Recent VLA work addresses this through recurrent or belief-state latents (Cherepanov et al., 2026c; Li et al., 2026a), explicit memory representations (Shi et al., 2026; Guo et al., 2026), long-context or compressed multi-frame conditioning (Jang et al., 2025), and retrieval- or structure-based memory (Sridhar et al., 2026; Haresh et al., 2026; Chung et al., 2026; Li et al., 2026b). Memory-dependent manipulation is also studied in dedicated benchmarks (Cherepanov et al., 2026b; Chen et al., 2026a; Lei et al., 2026). These works study how

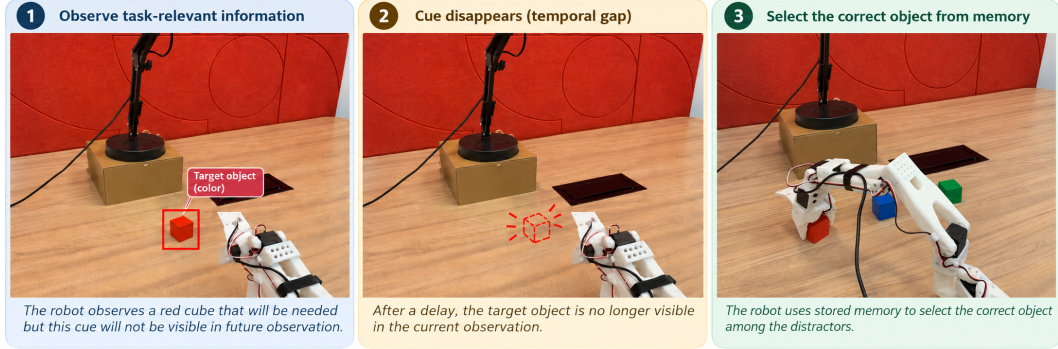


Figure 2: **Real-robot delayed-cue setup.** The robot observes the target object’s color. The cue then leaves the observation, and the robot must later select the matching object among distractors using stored visual information.

historical information is represented or retrieved, whereas our focus is on when an observation should be admitted to a bounded memory.

Keyframe and event selection for visual memory Existing approaches identify important observations using motion or bottleneck-pose heuristics (James et al., 2022), learned frame scoring (Mark et al., 2026; Chen et al., 2026c), progress, value, or critic signals (Liu et al., 2026; Yi et al., 2026; Buamane et al., 2026), trajectory annotation (Feng et al., 2026), or dedicated event detectors (Zeng et al., 2026b) — each relying on a separate selector or external signal. We instead derive the selection signal directly from changes in the acting policy’s own predicted action plans.

Action chunking, replanning, and plan consistency Action-chunking policies predict short action sequences and under receding-horizon execution successive chunks overlap on predictions for the same future actions (Zhao et al., 2023; Chi et al., 2023; Kim et al., 2025; Physical Intelligence et al., 2025). Prior work mainly uses this overlap to improve consistency (Zhao et al., 2023; Liu et al., 2025) or studies the trade-off between open-loop execution and replanning (Lazzati et al., 2026; Zeng et al., 2026a; Chen et al., 2026b; Jing et al., 2026). We instead use disagreement between overlapping predictions as a signal that the new observation has changed the policy’s intended behavior.

Memory representation, keyframe selection, and plan consistency are typically studied separately, whereas we connect them by using plan disagreement as a policy-internal signal for selective memory updates.

3 RePlanMem-VLA

3.1 Overview

In long-horizon manipulation, task-relevant visual information often leaves the cameras before it is needed. A policy that sees only the current frame cannot recover that cue, so we concatenate the live views with a bounded visual memory of B slots. Slots are empty at the start of an episode and are never carried across episode boundaries. The policy predicts a chunk of K actions from one observation and executes only the first $H < K$ before replanning, so consecutive chunks overlap on $K - H$ future steps: the same intended actions, predicted before and after the new frame. Prior work treats disagreement on that overlap as noise to suppress. We read a large revision as evidence that the new observation was task-relevant and use it to trigger a memory-update proposal. The direct update rule stores the immediately preceding frame, whereas the redundancy-aware variant retains only visual states that are not already represented in memory. Figures 1 and 3 and Algorithm 1 summarize the procedure. At every query the policy first predicts a pre-write chunk from the memory available before a potential write, then overlap disagreement is converted, without gradients, into a task-calibrated surprise score and an event triggers frame selection and a memory update only when a visually novel state is found. The final action prediction is conditioned on the resulting memory state, and only this prediction receives the original supervised fine-tuning (SFT) objective. Event detection never sees expert actions, so memory writes are driven only by the policy’s own plan revisions.

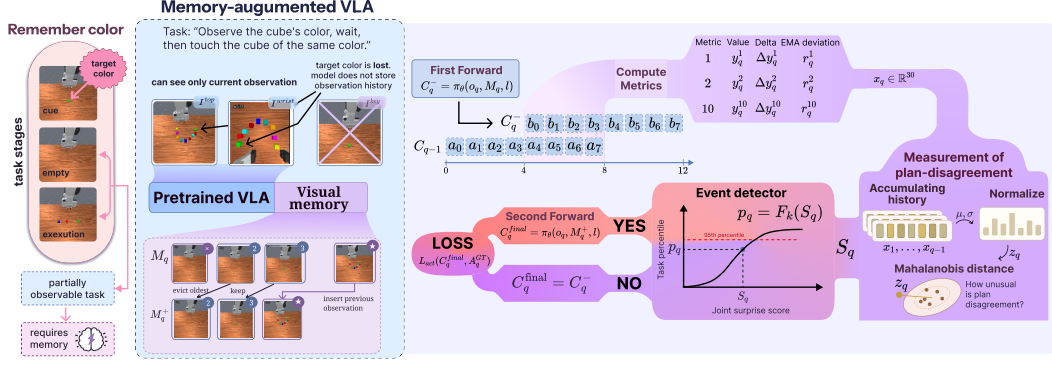


Figure 3: **RePlanMem-VLA overview.** At replanning step q , the VLA first predicts a chunk C_q^- from the current observation, instruction, and existing memory. The temporally aligned overlap with the preceding plan is converted into a task-calibrated surprise percentile p_q . If $p_q > \tau$, a selected observation is written to FIFO memory and the policy performs a second forward pass with the updated memory, yielding $C_q^{final} = C_q^+$, otherwise $C_q^{final} = C_q^-$.

3.2 Memory-augmented action policy

In partially observable manipulation, identical current observations can require different actions depending on earlier cues or task progress (Cherepanov et al., 2026a; Chen et al., 2026a), as illustrated by the delayed-cue setup in Fig. 2. A policy conditioned only on the current observation cannot distinguish such histories. Historical context can resolve this ambiguity, motivating multi-frame conditioning and memory-augmented policies (Jang et al., 2025; Shi et al., 2026). To bound the amount of historical visual input, we retain at most B observations. The resulting online selection problem is which observations to preserve for future control. We call these observations keyframes, focusing on transient cues needed later and observations marking changes in task phase.

Let q index policy queries, o_q denote the current multi-view observation, and ℓ denote the language instruction. Action chunking predicts multiple future controls from a single observation, supporting temporally coordinated behavior and efficient execution (Zhao et al., 2023; Chi et al., 2023; Kim et al., 2025). The policy predicts a chunk of K actions $C_q = (a_{q,0}, \dots, a_{q,K-1})$ but executes only the first $H < K$ of them before replanning. We refer to K and H as the *chunk horizon* and *execution stride*. Here, $a_{q,j} \in \mathbb{R}^d$ denotes the action predicted at query q for the j -th step of the chunk.

The visual memory before query q is an ordered collection of at most B keyframes $M_q = (M_q^1, \dots, M_q^{m_q})$, $m_q \leq B$, whose slots are appended to the policy’s visual context alongside the current camera views: with a top and a wrist camera the visual input is $X_q = [I_q^{top}, I_q^{wrist}, M_q^1, \dots, M_q^{m_q}]$.

All slots are empty at the beginning of an episode, and memory state is never carried across episode boundaries. At query q , a first forward pass uses the memory state that existed before the current event decision and yields the *pre-write plan* $C_q^- = \pi_\theta(o_q, M_q, \ell)$, which provides both a candidate control output and the policy-internal signal to decide whether memory should be updated. We denote the chunk used for execution after the memory-update decision by C_q^{final} .

3.3 Plan revision as an event signal

Two consecutive action chunks make predictions for some of the same future time steps. Because H actions of the preceding chunk have already been executed, the aligned overlap has length $K - H$. We therefore compare

$$U_q = C_{q-1}^{final}[H : K], \quad V_q = C_q^-[0 : K - H]. \quad (1)$$

Both sequences estimate the same future actions, but V_q is conditioned on the newly observed scene, so a large discrepancy indicates that new visual evidence has caused the policy to revise its intended behavior. We summarize this revision with D complementary detector signals $\mathbf{y}_q = [y_q^1, \dots, y_q^D]^\top$. These signals include aligned action error, directional disagreement, temporally localized deviations, translation, rotation, and gripper changes. The deployed detector also includes the distance between consecutive detached top-view VLA features as one of these signals. In our configuration $D = 10$.

To distinguish persistent model uncertainty from abrupt changes, we augment these values with their temporal differences $\Delta \mathbf{y}_q = \mathbf{y}_q - \mathbf{y}_{q-1}$ and with causal deviations from an exponential moving average (EMA),

$$\widetilde{\mathbf{m}}_q = \beta \widetilde{\mathbf{m}}_{q-1} + (1 - \beta) \mathbf{y}_q, \quad \mathbf{d}_{q,\text{EMA}} = \mathbf{y}_q - \widetilde{\mathbf{m}}_{q-1}.$$

The complete detector input is

$$\mathbf{x}_q = [\mathbf{y}_q^\top, \Delta \mathbf{y}_q^\top, \mathbf{d}_{q,\text{EMA}}^\top]^\top \in \mathbb{R}^{3D}, \quad (2)$$

which is $\mathbb{R}^{30}$ in our configuration. All quantities entering Eqs. (1) and (2) are computed from stop-gradient copies of action chunks and visual features. Consequently, the event detector cannot shape the policy by differentiating through its decision rule.

3.4 Causal, task-conditioned event detection

The components of $\mathbf{x}_q$ have different units and dynamic ranges, so we standardize each of them using statistics available strictly before the current query, $z_q^i = (x_q^i - \mu_{q-1}^{i,(k)}) / (\sigma_{q-1}^{i,(k)} + \varepsilon)$. The current sample is scored before it is incorporated into the running statistics. All detector statistics — the moments, the matrix below, and the reference CDF of Eq. (4) — are maintained separately for each task k , so that no cross-task scale difference enters the score. Let $\Sigma_{q-1}^{(k)}$ be the causal second-moment matrix of the normalized features. Because the preceding normalization is component-wise, its diagonal is unity up to the lag of the running moments, so $\Sigma_{q-1}^{(k)}$ acts as a correlation matrix. The joint deviation of the features is measured by the regularized Mahalanobis surprise

$$S_q = \sqrt{\mathbf{z}_q^\top \left(\Sigma_{q-1}^{(k)} + \lambda \mathbf{I} \right)^{-1} \mathbf{z}_q}, \quad (3)$$

where $\lambda > 0$ stabilizes the inversion. We set it to a fixed fraction of the mean eigenvalue of $\Sigma_{q-1}^{(k)}$, which makes the regularizer scale-free across tasks. Unlike an unweighted Euclidean norm, Eq. (3) accounts for correlations among different revision signals.

Natural plan variability differs across tasks in multi-task SFT, which is why the calibration above is per-task while the decision threshold is shared. Writing $F_k(s) = \Pr(S \leq s \mid k)$ for the empirical cumulative distribution function (CDF) of past surprise scores for task k , the event percentile and binary decision are

$$p_q = F_k(S_q), \quad e_q = \mathbf{1}[p_q > \tau]. \quad (4)$$

Thus $\tau = 0.99$, for example, assigns the same interpretation — an event more surprising than 99% of the task-specific reference distribution — across heterogeneous tasks. At the beginning of SFT, memory writes are disabled for a short warm-up period (the first 5–10% of training), while the policy is optimized normally and detector statistics are collected. After warm-up, the CDFs and causal moments can be updated with rolling histories to track the changing policy. Once training is complete, these statistics are frozen and kept fixed at inference time to preserve a stable detector calibration. The first query in each episode has no preceding chunk and therefore cannot produce a plan-overlap event.

3.5 Redundancy-aware frame selection

An event determines *when* memory should be reconsidered, but not necessarily which frame should be stored. At an event, we jointly consider the current FIFO contents and a short causal window W_q of observations preceding the current query. We use the three observations preceding the event query ($W = 3$) and apply deterministic complete-link clustering with threshold $\delta = 0.01$. From each novel cluster, the most recent candidate is selected as its representative. The current observation is excluded so that selection cannot use information obtained after the detected plan revision.

We group visually similar frames using complete-link clustering and treat a candidate cluster as novel only if it contains no frame already represented in memory. From each novel cluster, we retain its most recent candidate and append the selected representatives to the bounded FIFO in chronological order. Thus, one detector event may result in zero, one, or multiple writes. Visual masks, when available, are used only to suppress irrelevant appearance changes during comparison, while the original RGB frames are stored and supplied to the policy. The direct update rule is the special case in which the selector returns only the immediately preceding frame without redundancy filtering.

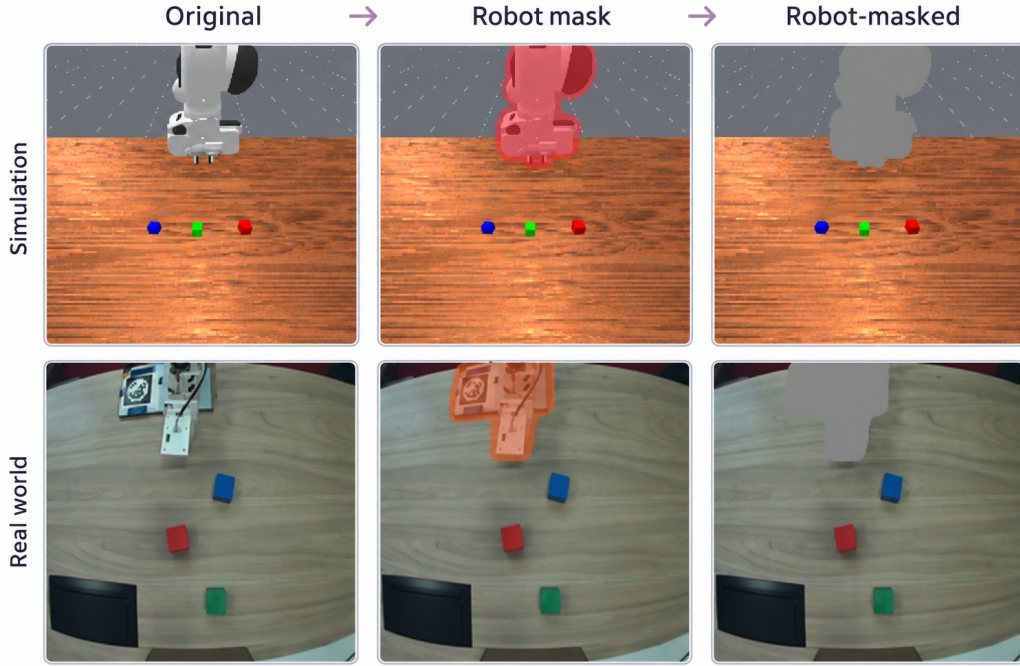


Figure 4: **Examples of robot masking in simulation and real-robot observations.** Each example shows the original observation, the estimated robot mask, and the masked image used for redundancy comparison. The policy receives the original RGB frame rather than the masked image.

3.6 Event-gated memory update and second forward pass

When an event is detected at query q , the frame-selection rule returns a possibly empty sequence R_q of candidate keyframes. The direct rule returns the immediately preceding top-view observation, whereas the redundancy-aware rule may return zero, one, or multiple frames. A fixed-capacity FIFO operator produces

$$M_q^+ = \text{FIFOAppend}(M_q, R_q),$$

inserting the selected frames in chronological order and discarding the oldest items when all B slots are occupied. If $R_q = \emptyset$, memory remains unchanged even though the detector emitted an event.

At inference time, an actual memory update is followed by a second evaluation of the same policy on the current observation and instruction,

$$C_q^{\text{final}} = \pi_\theta(o_q, M_q^+, \ell).$$

If no frame is written, no additional inference pass is required and $C_q^{\text{final}} = C_q^-$. During distributed SFT, the inference-only pre-write pass is followed by one differentiable final pass for the complete micro-batch, so rows without a memory update simply use the unchanged memory state. This keeps the distributed computation graph synchronized when event counts differ across ranks.

3.7 Supervised fine-tuning objective

Let A_q^{GT} be the expert action chunk associated with query q . We retain the backbone’s standard action objective and apply it only to the selected final chunk, $\mathcal{L}_{\text{SFT}}(\theta) = \sum_q \mathcal{L}_{\text{act}}(C_q^{\text{final}}, A_q^{\text{GT}})$. Gradients flow through the policy invocation that produced C_q^{final} , but not through the detector, its running statistics, or the discrete memory-write decision. Expert actions supervise control only: they are not used to identify events or select keyframes. The same causal detector and memory-update procedure can therefore be used at inference time without keyframe annotations or privileged simulator state.

Algorithm 1: Surprise keyframe memory at query q .

Input : Observation o_q , instruction ℓ , task id k , memory M_q with B slots, previous chunk C_{q-1}^{final} , chunk horizon K , execution stride H , percentile threshold τ , causal statistics $(\mu, \sigma, \Sigma, F_k)_{q-1}$, candidate-window size $W = 3$, clustering threshold $\delta = 0.01$.

Output : Final action chunk C_q^{final} , updated memory M_{q+1} .

```
// 1. pre-write forward pass
 $C_q^- \leftarrow \pi_\theta(o_q, M_q, \ell)$ 
if  $q$  is the episode's first query then
     $M_q^+ \leftarrow M_q$ 
     $C_q^{\text{final}} \leftarrow C_q^-$ 
end
else
    // 2. plan-revision features (stop-gradient)
     $U_q \leftarrow C_{q-1}^{\text{final}}[H:K]$ ,
     $V_q \leftarrow C_q^-[0:K-H]$ 
     $x_q \leftarrow [y_q, \Delta y_q, d_{q,\text{EMA}}]$  by Eq. (2)
    // 3. causal scoring, then statistics update
     $z_q \leftarrow (x_q - \mu_{q-1}) \odot (\sigma_{q-1} + \varepsilon)$ 
     $S_q \leftarrow \text{Mahalanobis}(z_q)$  by Eq. (3)
     $p_q \leftarrow F_k(S_q)$ ;  $e_q \leftarrow \mathbf{1}[p_q > \tau]$ 
    update  $(\mu, \sigma, \Sigma, F_k)$  with  $x_q$ 
    // 4. event-gated frame selection and memory update
     $R_q \leftarrow \emptyset$ 
    if  $e_q = 1$  and warm-up has ended then
         $Q_q \leftarrow \{I_{q-W}^{\text{top}}, \dots, I_{q-1}^{\text{top}}\}$ 
         $Z_q \leftarrow M_q \cup Q_q$ 
        cluster  $Z_q$  using deterministic complete-link clustering with threshold  $\delta$ 
        foreach cluster containing no frame from  $M_q$  do
            append the most recent candidate to  $R_q$ 
        end
        sort  $R_q$  chronologically
    end
    if  $R_q \neq \emptyset$  then
         $M_q^+ \leftarrow \text{FIFOAppend}(M_q, R_q)$ 
         $C_q^{\text{final}} \leftarrow \pi_\theta(o_q, M_q^+, \ell)$ 
    else
         $M_q^+ \leftarrow M_q$ ;  $C_q^{\text{final}} \leftarrow C_q^-$ 
    end
end
// 5. execution and (training only) supervision
Execute  $C_q^{\text{final}}[0:H]$ ;  $M_{q+1} \leftarrow M_q^+$ 
Accumulate  $\mathcal{L}_{\text{act}}(C_q^{\text{final}}, A_q^{\text{GT}})$ ; no gradient flows through  $x_q$ , the statistics, or  $e_q$ 
```

4 Experimental evaluation

Our experiments address three research questions:

- RQ1.** Does revision of overlapping action chunks contain a task-relevant signal, and which disagreement measures extract it most reliably?
- RQ2.** Does this signal transfer to pretrained VLA policies across different backbones and replanning frequencies, and can it support accurate online event detection?
- RQ3.** Can the resulting surprise signal be used to drive selective visual-memory updates during VLA fine-tuning?

4.1 RQ1: Disagreement signals in single-task policies

We first examine whether revisions of overlapping action chunks contain task-relevant information in a controlled setting using lightweight single-task causal decoder-only and Flow-Matching Transformers trained from scratch. This controlled setting allows us to efficiently screen a large set of disagreement metrics at substantially lower computational cost before transferring the most informative signals to pretrained VLA policies. We evaluate 35 disagreement metrics (Appendix C) across eight diverse MIKASA-Robo-VLA tasks (Appendix A.1) and examine whether their behavior remains consistent across different task structures, amounts of available historical information, and successful versus failed rollouts. For each architecture, we compare two information regimes: full-context, where the policy has access to the accumulated observation-action history up to the current query, and current-only, where prediction is conditioned only on the current observation.

We use Frame ROC-AUC as the primary screening criterion, measuring whether a scalar disagreement signal consistently ranks transition frames above background frames independently of a decision threshold. Unlike PR-AUC, ROC-AUC is invariant to event prevalence, enabling comparison across tasks and replanning regimes with different positive rates. We report PR-AUC and thresholded F1 as complementary operating measures in RQ2. A more detailed analysis, including complete metric rankings, correlation analyses, and task-specific results, is provided in Appendix A.

Across decoder-only policies, causal temporal signals are particularly stable with respect to available context: Causal Z-score yields mean Frame ROC-AUC of 0.5985/0.6002 under full-context/current-only conditioning, with EMA deviation showing a similar pattern at 0.5904/0.5908. For Flow-Matching policies, distributional measures are stronger, with Wasserstein-1 reaching 0.7045/0.6678 and Energy distance 0.6831/0.6610. Nevertheless, Causal Z-score remains stable for Flow Matching as well (0.6373/0.6396), showing that the causal temporal signal persists across both action-generation paradigms.

Table 1: Representative high-performing disagreement signals from the single-task screening. Values report mean Frame ROC-AUC across eight MIKASA-Robo-VLA tasks under full-context and current-only conditioning. Top-10 rankings for each architecture and context regime are reported in Appendix A.3.

Policy	Metric	Full-context	Current-only
Decoder-only	Causal Z-score	0.5985	0.6002
Decoder-only	EMA deviation	0.5904	0.5908
Decoder-only	Energy distance	0.5597	0.6313
Decoder-only	Cumulative mean drift	0.5570	0.6210
Flow Matching	Wasserstein-1	0.7045	0.6678
Flow Matching	Energy distance	0.6831	0.6610
Flow Matching	Cumulative mean drift	0.6460	0.6456
Flow Matching	Causal Z-score	0.6373	0.6396

Thus, the result is not explained by a single universally optimal definition of disagreement. Instead, several different types of plan revision consistently provide useful information across tasks. Across decoder-only and Flow-Matching policies, disagreement between consecutive action chunks provides a stable signal of semantic keyframes. The analysis identifies two complementary classes of features: distributional and cumulative measures for frame-level separation, and causal temporal features for autonomous localization of plan changes. These patterns are preserved across action-generation architectures and context regimes and remain visible on failed trajectories (Appendix A.6). Correlation analysis further reveals substantial redundancy within individual metric families (Appendix A.5), indicating that several closely related variants often capture the same underlying plan-revision structure.

These findings motivate selecting a compact set of complementary disagreement features for the subsequent VLA experiments. We therefore retain 10 representative metrics spanning the informative distributional, cumulative, and causal temporal signals identified above, and in RQ2 evaluate whether their combination remains effective for online event detection in pretrained VLA policies across different backbones and replanning frequencies.

4.2 RQ2: Event detection with pretrained VLA policies

We next ask whether the overlap-disagreement features selected in RQ1 remain informative in pretrained VLAs, under frequent versus sparse replanning, and whether they support an online keyframe detector. Protocol details, per-task breakdowns, and additional ablations are reported in Appendix B.

Experiments use MIKASA-Robo-VLA (Cherepanov et al., 2026b), with simulator phase transitions as binary keyframe labels. We evaluate OpenVLA-OFT on all 22 tasks and $\pi 0.5$ on 19 shared tasks

(ShellGame excluded). For each task we collect 50 episodes and split at the episode level ($\approx 70/10/20$ train/val/test). Policies generate chunks of horizon $K = 8$ under receding horizon (RH1, $H = 1$) and partial execution (EXEC4, $H = 4$). At each query q , we compare temporally aligned overlap segments of length $K - H$ and compute the 10-dimensional surprise vector from RQ1. Detectors are frame-wise XGBoost/CatBoost and a causal LSTM. Thresholds maximize validation F1. We report PR-AUC, ROC-AUC, F1, positive rate, event_recall@1, and mean policy success (Mean SR).

The signal transfers. For OpenVLA-OFT, Mean SR is essentially unchanged from RH1 to EXEC4 (17.7% vs. 18.7%), while detection remains strong: under RH1 all classifiers reach event_recall@1 = 1.0, with ROC-AUC 0.981–0.989. For $\pi 0.5$, control degrades without step-wise feedback (13.2% $\rightarrow$ 8.8%), but detection remains accurate (RH1 PR-AUC 0.924–0.933, EXEC4 0.981–0.983). On the 19 shared tasks, the success gap is large, yet EXEC4 detection is nearly identical across backbones (XGBoost PR-AUC 0.996 vs. 0.982). Chunk disagreement therefore marks semantic phase boundaries even when the policy fails at manipulation.

Table 2: Keyframe detection vs. policy success under RH1 and EXEC4. OpenVLA-OFT: 22 tasks, $\pi 0.5$: 19 shared tasks.

Policy	Regime	Mean SR	Classifier	PR-AUC	ROC-AUC	F1
OpenVLA-OFT	RH1	17.7%	XGBoost	0.932	0.982	0.841
OpenVLA-OFT	RH1	17.7%	CatBoost	0.929	0.981	0.831
OpenVLA-OFT	RH1	17.7%	LSTM	0.962	0.989	0.874
OpenVLA-OFT	EXEC4	18.7%	XGBoost	0.996	0.997	0.966
OpenVLA-OFT	EXEC4	18.7%	CatBoost	0.995	0.996	0.967
OpenVLA-OFT	EXEC4	18.7%	LSTM	0.984	0.990	0.964
$\pi 0.5$	RH1	13.2%	XGBoost	0.928	0.984	0.854
$\pi 0.5$	RH1	13.2%	CatBoost	0.924	0.984	0.854
$\pi 0.5$	RH1	13.2%	LSTM	0.933	0.985	0.859
$\pi 0.5$	EXEC4	8.8%	XGBoost	0.982	0.988	0.917
$\pi 0.5$	EXEC4	8.8%	CatBoost	0.981	0.987	0.921
$\pi 0.5$	EXEC4	8.8%	LSTM	0.983	0.988	0.942

The PR-AUC increase at $H = 4$ is partly a class-balance effect: fewer replans with the same number of phase events raise the positive rate (≈ 0.18 – 0.20 to ≈ 0.39 – 0.50). Stable ROC-AUC near 0.98 shows that the features remain informative after this densification is taken into account. The same pattern holds within a RememberColor5 episode: disagreement stays near baseline during cue observation, spikes at the transition to manipulation, and returns to baseline during later motion.

4.3 RQ3: Surprise-gated visual-memory updates

We finally ask whether the surprise signal identified in RQ1–RQ2 can serve as a practical write gate for bounded visual memory. In contrast to RQ2, which evaluates event detection itself, RQ3 focuses on the resulting memory behavior: whether detected events cause task-relevant observations to be retained while redundant updates are suppressed.

4.3.1 Direct surprise-gated writing

We first evaluate the simplest memory-update rule, in which each detected event directly writes the immediately preceding top-view observation to the FIFO. We evaluate this variant with the fine-tuned $\pi_{0.5}$ policy on REMEMBERCOLOR3 over 50 protocol seeds (Table 3). At the 95th-percentile threshold, the detector triggers in only 19/50 episodes, leaving the cue available at action start in 9/50. Lowering the threshold to the 80th percentile increases write frequency to 1.82 per episode and cue coverage to 24/50.

Quantity	$q_{0.95}$	$q_{0.80}$
Success rate	15/50 (0.30)	17/50 (0.34)
Write events per episode	1.00	1.82
Episodes with ≥ 1 write	19/50	48/50
Cue frame written	10/50	24/50
Cue frame in memory at action start	9/50	24/50
Memory empty at action start	37/50	4/50
SR cue in memory	4/9 (0.44)	7/24 (0.29)
SR non-empty, no cue	1/4 (0.25)	7/22 (0.32)
SR memory empty	10/37 (0.27)	3/4 (0.75)

Table 3: Direct surprise-gated memory updates for the fine-tuned $\pi_{0.5}$ policy on REMEMBERCOLOR3-Medium over 50 protocol seeds. The 19k-step run uses the 95th-percentile gate. The 5k-step run uses the 80th-percentile gate. Lowering the threshold increases write frequency and cue coverage.

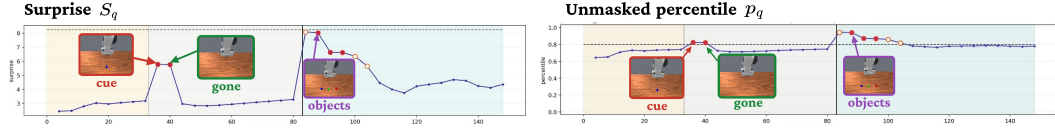


Figure 5: **Surprise-gated memory updates on the RememberColor task.** The surprise score S_q and task-calibrated percentile p_q identify candidate update times near transitions between the cue, empty, and action phases. The redundancy-aware selector then determines which observations, if any, are written to memory.

4.3.2 Redundancy-aware memory selection

We next replace direct writes with the redundancy-aware selector from Section 3.5. The detector still determines *when* memory should be reconsidered, but an event no longer implies an unconditional write: visually redundant candidates can now be rejected, whereas previously unseen scene states are admitted to the FIFO.

Thus, an event becomes a proposal for memory reconsideration rather than an unconditional write. Figure 5 illustrates this behavior on representative REMEMBERCOLOR trajectories. Across the evaluated REMEMBERCOLOR episodes, the relevant cue observation remained available until the later action phase, showing that redundancy suppression can reduce unnecessary updates without discarding the task-relevant state.

4.3.3 Robot-masked redundancy filtering

Visual comparison can still interpret changes in robot configuration as changes in the environment. We therefore augment the redundancy-aware selector with robot masking during pairwise frame comparison. Figure 4 shows examples of the masks used in simulation and on real-robot observations. Only the comparison image is masked, while the original RGB observation is retained in memory and supplied to the VLA.

For VLA-OFT, both the unmasked and SAM2-masked variants retained a cue-phase observation until the first action-phase query in all 50/50 episodes. Robot masking reduced FIFO writes from 336 to 169 (49.7%) and overwrites from 186 to 19 (89.8%).

Taken together, these results separate the two roles of the memory mechanism: surprise determines *when* memory should be reconsidered, while visual comparison determines *what* should be retained.

4.3.4 Transfer to real-robot replay

Finally, we evaluate the same event-detection and redundancy-filtering pipeline on recorded real-robot trajectories, testing the event detector, visual clustering, and robot-masked similarity outside simulation. This is a replay evaluation rather than closed-loop control.

Figure 6 shows representative replay examples. The detector continues to produce temporally localized event proposals, while the same clustering procedure can separate distinct scene states and reject visually redundant observations. Robot masking additionally removes changes caused primarily by the manipulator, using the same comparison procedure as in simulation.

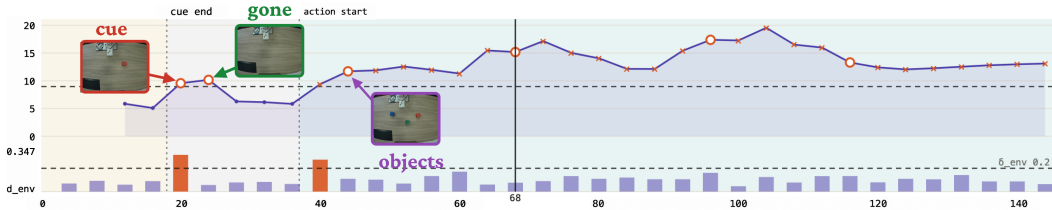


Figure 6: **Real-robot replay of the surprise-based event detector and redundancy-aware memory-selection pipeline.** The experiment evaluates event proposals, visual clustering, and robot-masked similarity on recorded real observations rather than closed-loop control.

5 Limitations and Future Work

Our experiments show that disagreement between overlapping action chunks can accurately identify task-relevant keyframes. However, we do not claim that such disagreement is specific to semantic events. Substantial plan revisions may also arise for other reasons, including policy prediction errors. The proposed signal should therefore be interpreted as an indicator of potentially important change rather than as an unambiguous semantic event detector.

A second limitation concerns the temporal structure of the evaluated tasks. The semantic phases in our experiments have relatively clear boundaries. With an execution stride of $H = 4$, sufficiently short phases may occur entirely between two replanning queries and therefore be missed, whereas gradual transitions may distribute the corresponding plan revision over several queries and produce no pronounced peak. How disagreement behaves under less distinct phase transitions remains to be investigated and is closely related to the question of when an observation should be committed to memory.

Our current implementation stores selected observations as full images in a small visual memory and uses visual clustering to suppress redundant entries. A promising direction is therefore to develop more compact representations of stored observations, together with comparison and aggregation mechanisms that preserve task-relevant information while making historical context easier for the policy to use.

Finally, the current surprise estimate is defined relative to task-conditioned statistics accumulated from previous replanning queries, and the write threshold is selected empirically. Consequently, the present formulation requires calibration data for each task and has not yet been shown to transfer directly to previously unseen tasks without accumulating new reference statistics and selecting an appropriate threshold.

6 Conclusion

We investigated whether revisions of overlapping action chunks can serve as an internal signal of observation importance in long-horizon robot control. Across single-task decoder-only and Flow-Matching policies, we find that disagreement between consecutive plans contains a consistent signal of semantic task transitions, with complementary distributional, cumulative, and causal temporal measures capturing different aspects of plan revision. We further show that this signal transfers to pretrained OpenVLA-OFT and $\pi_{0.5}$ policies across different replanning frequencies and remains predictive even when control performance varies substantially. Building on these observations, we use disagreement-derived surprise to drive selective visual-memory updates in a memory-augmented VLA policy. The resulting framework connects three stages that are usually treated separately: replanning, event detection, and memory formation. Rather than introducing an external keyframe selector, the policy reuses a signal that naturally arises from its own action predictions to decide when the observation history should be updated. Overall, our results suggest that changes in a policy’s intended future behavior provide a useful source of temporal proposals for sparse memory updates in partially observable, multi-stage manipulation.

References

- Thanpimon Buamane, Masato Kobayashi, and Yuki Uranishi. Bi-HIL: Bilateral control-based multimodal hierarchical imitation learning via subtask-level progress rate and keyframe memory for long-horizon contact-rich robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026. URL <https://arxiv.org/abs/2603.13315>.
- Tianxing Chen, Yuran Wang, Mingleyang Li, Yan Qin, Hao Shi, Zixuan Li, Yifan Hu, Yingsheng Zhang, Kaixuan Wang, Yue Chen, Hongcheng Wang, Junjie Wang, Tianhang Yang, Renjing Xu, Ruihai Wu, Yao Mu, Yaodong Yang, Hao Dong, and Ping Luo. RMBench: Memory-dependent robotic manipulation benchmark with insights into policy design. *arXiv preprint arXiv:2603.01229*, 2026a. URL <https://arxiv.org/abs/2603.01229>.
- Wenxi Chen, Kaidi Zhang, Chi Lin, Zhiyuan Zhang, Yu She, Yuejiang Liu, Raymond A. Yeh, Shaoshuai Mou, and Yan Gu. DREAM-Chunk: Reactive action chunking with latent world model. *arXiv preprint arXiv:2606.18589*, 2026b. URL <https://arxiv.org/abs/2606.18589>.

- Yipeng Chen, Wentao Tan, Lei Zhu, Fengling Li, Jingjing Li, Guoli Yang, and Heng Tao Shen. Non-markovian long-horizon robot manipulation via keyframe chaining. *arXiv preprint arXiv:2603.01465*, 2026c. URL <https://arxiv.org/abs/2603.01465>.
- Egor Cherepanov, Nikita Kachaev, Alexey Kovalev, and Aleksandr I. Panov. Memory, benchmark & robots: A benchmark for solving complex tasks with reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL <https://openreview.net/forum?id=9cLPurIZMj>.
- Egor Cherepanov, Nikita Kachaev, Aleksandr I. Panov, and Alexey K. Kovalev. Mikasa-roboto-vla: Benchmarking memory in vla models for long-horizon manipulation, 2026b. URL <https://arxiv.org/abs/2610.00604>.
- Egor Cherepanov, Nikita Kachaev, Daniil Zelezetsky, Aydar Bulatov, Artem Pshenitsyn, Yuri Kuratov, Alexey Skrynnik, Aleksandr I. Panov, and Alexey K. Kovalev. μ vla: On recurrent memory for partially observable manipulation in VLA models. *arXiv preprint arXiv:2606.12497*, 2026c. URL <https://arxiv.org/abs/2606.12497>.
- Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.
- Nhat Chung, Taisei Hanyu, Toan Nguyen, Huy Le, Frederick Bumgarner, Duy Minh Ho Nguyen, Khoa Vo, Kashu Yamazaki, Chase Rainwater, Tung Kieu, Anh Nguyen, and Ngan Le. Rethinking progression of memory state in robotic manipulation: An object-centric perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 3407–3415, 2026. doi: 10.1609/aaai.v40i5.37337.
- Youhe Feng, Hansen Shi, Haoyang Li, Xinlei Guo, Yang Wang, Chengyang Zhang, Jinkai Zhang, Xiaohan Zhang, Jie Tang, and Jing Zhang. ProcVLM: Learning procedure-grounded progress rewards for robotic manipulation. *arXiv preprint arXiv:2605.08774*, 2026. URL <https://arxiv.org/abs/2605.08774>.
- Xinying Guo, Chenxi Jiang, Hyun Bin Kim, Yuhang Han, Ying Sun, Yang Xiao, and Jianfei Yang. Chameleon: Control-indexed prospective memory for visuomotor manipulation. *arXiv preprint arXiv:2603.24576*, 2026. URL <https://arxiv.org/abs/2603.24576>.
- Sanjay Haresh, Daniel Dijkman, Apratim Bhattacharyya, and Roland Memisevic. Notes-to-self: Scratchpad augmented vlas for memory dependent manipulation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL <https://arxiv.org/abs/2602.21013>.
- Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J. Davison. Coarse-to-fine Q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13739–13748, 2022.
- Huiwon Jang, Sihyun Yu, Heeseung Kwon, Hoin Jeon, Younggyo Seo, and Jinwoo Shin. ContextVLA: Vision-language-action model with amortized multi-frame context. *arXiv preprint arXiv:2510.04246*, 2025. URL <https://arxiv.org/abs/2510.04246>.
- Dong Jing, Gang Wang, Jiaqi Liu, Weiliang Tang, Zelong Sun, Yunchao Yao, Zhenyu Wei, Yunhui Liu, Zhiwu Lu, and Mingyu Ding. Mixture of horizons in action chunking. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026.
- Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025. URL <https://arxiv.org/abs/2502.19645>.
- Filippo Lazzati, Kyle Stachowicz, William Chen, Alberto Maria Metelli, Andrew Wagenmaker, and Sergey Levine. Why does action chunking improve behavioral cloning performance in robotic control? *arXiv preprint arXiv:2608.02547*, 2026. URL <https://arxiv.org/abs/2608.02547>.
- Huashuo Lei, Wenxuan Song, Huarui Zhang, Jieyuan Pei, Jiayi Chen, Haodong Yan, Han Zhao, Pengxiang Ding, Zhipeng Zhang, Lida Huang, Donglin Wang, Yan Wang, and Haoang Li. RoboMemArena: A comprehensive and challenging robotic memory benchmark. *arXiv preprint arXiv:2605.10921*, 2026. URL <https://arxiv.org/abs/2605.10921>.
- Hang Li, Fengyi Shen, Dong Chen, Liudi Yang, Xudong Wang, Jinkui Shi, Zhenshan Bing, Ziyuan Liu, and Alois Knoll. Remem-vla: Empowering vision-language-action model with memory via dual-level recurrent queries. *arXiv preprint arXiv:2603.12942*, 2026a. URL <https://arxiv.org/abs/2603.12942>.
- Pengteng Li, Weiyu Guo, He Zhang, Tiefu Cai, Xiao He, Yandong Guo, and Hui Xiong. Spatial memory for out-of-vision manipulation in vision-language-action. *arXiv preprint arXiv:2605.22283*, 2026b. URL <https://arxiv.org/abs/2605.22283>.

- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.
- Yuanzhe Liu, Jingyuan Zhu, Yuchen Mo, Gen Li, Xu Cao, Jin Jin, Yifan Shen, Zhengyuan Li, Tianjiao Yu, Wenzhen Yuan, Fangqiang Ding, and Ismini Lourentzou. PALM: Progress-aware policy learning via affordance reasoning for long-horizon robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28096–28110, 2026.
- Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Maximilian Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations*, 2025.
- Max Sobol Mark, Maria Attarian, Chuyuan Fu, Debidatta Dwibedi, Jacky Liang, Dhruv Shah, and Aviral Kumar. Long-context robot imitation learning by focusing on key history frames. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.201.
- Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL <https://arxiv.org/abs/2504.16054>.
- Weikang Qiu, Huashuo Lei, Tinglin Huang, and Rex Ying. Efficient long-horizon vision-language-action models via static-dynamic disentanglement. *arXiv preprint arXiv:2602.03983*, 2026. URL <https://arxiv.org/abs/2602.03983>.
- Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=54U3XHf7qq>.
- Lucy Xiaoyang Shi, Archit Sharma, Tony Z. Zhao, and Chelsea Finn. Waypoint-based imitation learning for robotic manipulation. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2195–2209. PMLR, 2023. URL <https://proceedings.mlr.press/v229/shi23b.html>.
- Ajay Sridhar, Jennifer Pan, Satvik Sharma, and Chelsea Finn. MemER: Scaling up memory for robot control via experience retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://arxiv.org/abs/2510.20328>.
- Ziye Wang, Modi Shi, Chaojun Ni, Jiazhi Yang, Mengdi Li, Zhizhong Su, Tianwei Lin, and Hongyang Li. NativeMEM: Native memory compression for long-horizon robotic manipulation. *arXiv preprint arXiv:2607.06678*, 2026.
- Pengfei Yi, Yingjie Ma, Wenjiang Xu, Yanan Hao, Shuai Gan, Wanting Li, and Shanlin Zhong. Critic in the loop: A tri-system vla framework for robust long-horizon manipulation. *arXiv preprint arXiv:2603.05185*, 2026. URL <https://arxiv.org/abs/2603.05185>.
- Michael Zeng, Abhinav Agarwal, Ajay Bati, Brian Lee, Siddharth Ancha, and Russ Tedrake. Revisiting open-loop execution in robotics: Toward reactive, higher-performing policies. *arXiv preprint arXiv:2608.15938*, 2026a. URL <https://arxiv.org/abs/2608.15938>.
- Yihan Zeng, Minghao Ye, Yiyuan Chen, Yide Shentu, Philipp Wu, Zike Yan, and Zhongyu Li. KEMO: Event-driven keyframe memory for long-horizon robot manipulation with vla policies. *arXiv preprint arXiv:2606.23589*, 2026b. URL <https://arxiv.org/abs/2606.23589>.
- Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023.

Table of Contents

A Additional Analysis for RQ1	14
A.1 Experimental setup and reference events	14
A.2 Candidate disagreement measures	15
A.3 Cross-task screening of scalar disagreement measures	15
A.4 Complementary evaluation criteria	16
A.5 Correlation structure and metric redundancy	17
A.6 Robustness to trajectory outcome	18
A.7 Detector feature representation used in RQ2	18
B Additional Analysis for RQ2	19
B.1 Experimental setup and evaluation	19
B.2 Features and detector architectures	20
B.3 OpenVLA-OFT under RH1 vs. EXEC4	21
B.4 Keyframe detection for $\pi_{0.5}$	21
B.5 Class balance and PR-AUC across execution strides	22
B.6 Cross-model comparison on shared tasks	22
B.7 Qualitative localization	22
B.8 Summary	22
C Complete Registry of Disagreement Metrics	22

A Additional Analysis for RQ1

This appendix provides the detailed experimental protocol and supporting analyses for RQ1. Section 4.1 reports the representative cross-task results used in the main text. Here we provide the task setup and reference-event definition, organize the candidate disagreement measures, report the top-ranked metrics for each architecture and context regime, evaluate complementary frame- and event-level criteria, analyze metric redundancy and robustness to trajectory outcome, and finally specify the compact detector representation used in RQ2.

A.1 Experimental setup and reference events

Tasks and training. The single-task screening uses eight MIKASA-Robo-VLA tasks (Cherepanov et al., 2026b): RememberColor3-VLA-v0, ShellGameColorLampTouch-VLA-v0, ShellGameShuffleColorLampTouch-VLA-v0, ShellGameShuffleTouch-VLA-v0, FindImposterShape3-VLA-v0, InterceptSlow-VLA-v0, RotateLenientPos-VLA-v0, and TakeItBack-VLA-v0. Each policy is trained from scratch on 250 demonstrations of a single task.

Policies and replanning. We evaluate two single-task action-generation architectures: a causal decoder-only Transformer and a Flow-Matching Transformer. Both policies predict action chunks of length $K = 8$ and execute one action before replanning, i.e., $H = 1$. Using the notation introduced in Section 3, the temporally aligned plan segments compared at query q are

$$U_q = C_{q-1}[H : K], \quad V_q = C_q[0 : K - H].$$

The two segments predict the same future execution times but are generated before and after the newly received observation.

The decoder-only policy predicts the action chunk directly. The Flow-Matching policy generates the same-sized chunk using conditional flow matching. To prevent stochastic resampling from introducing artificial disagreement, a fixed Gaussian noise tensor is reused within each Flow-Matching rollout.

Context regimes. For each architecture, we evaluate two information regimes. In the **full-context** setting, the policy has access to the accumulated observation-action history available up to the current query. In the **current-only** setting, prediction is conditioned only on the current observation. This comparison tests whether the informativeness and internal structure of the disagreement signal depend on access to historical context.

Table 4: Main families of candidate disagreement measures used in RQ1.

Family	Property	Examples
Point-wise	Aligned action error	overlap_l2_mean, overlap_nl1, overlap_huber
Scale-aware	Disagreement relative to action scale or covariance	relative_frobenius, diag_mahalanobis_rms
Action components	Translation, rotation, and gripper disagreement	a14_trans, a15_rot, a16_grip
Plan structure	Shape and temporal variation of the action chunk	pearson_shape_distance, first_difference_nmse
Cumulative	Accumulated disagreement along the prediction horizon	cumulative_mean_drift, cumulative_endpoint_drift
Distributional	Difference between distributions of predicted actions	Energy distance, Wasserstein-1, RBF MMD
Causal temporal	Current revision relative to preceding revisions	causal_z_exec_l2, cusum_exec_l2, ema_dev_exec_l2

Reference transitions. The tasks contain semantically distinct phases, whose transitions define the reference events used throughout RQ1. For a semantic phase transition occurring between frames $b - 1$ and b , we define the reference event window as

$$\mathcal{I}_e = [b - 1, b].$$

Thus, both the final observation of the preceding phase and the first observation of the following phase are treated as valid transition frames.

Primary evaluation criterion. For each task and candidate metric, we compute **Frame ROC-AUC**. This measures whether the scalar disagreement signal ranks transition-associated frames above background observations independently of a decision threshold. Cross-task summaries report the mean and median of the eight task-level ROC-AUC values.

A.2 Candidate disagreement measures

We screen 35 candidate disagreement measures computed from consecutive aligned plans. The candidates capture different aspects of plan revision, ranging from direct action-space deviations to cumulative, distributional, and causal temporal changes. Their main families are summarized in Table 4, and complete mathematical definitions are provided in Appendix C.

A.3 Cross-task screening of scalar disagreement measures

We first evaluate each candidate metric as a fixed scalar signal across all eight tasks. For each metric, task-level Frame ROC-AUC values are aggregated across tasks, and metrics are ranked separately for each architecture and context regime. Tables 5–8 report the ten highest-ranked metrics in the four configurations.

A.3.1 Decoder-only Transformer

Full-context. Causal temporal measures are particularly competitive under full context. Causal Z-score achieves the highest cross-task mean ROC-AUC (0.5985), while EMA deviation, cumulative mean drift, and Wasserstein-1 each appear among the task-wise top ten on 6/8 tasks. The result therefore already suggests that useful information is distributed across several forms of plan revision rather than captured by a single scalar definition.

Current-only. Under current-only conditioning, distributional and cumulative measures become stronger. Energy distance reaches mean ROC-AUC 0.6313, followed by cumulative mean drift (0.6210) and Wasserstein-1 (0.6202). Causal Z-score and EMA deviation remain competitive,

Table 5: Top-10 decoder-only disagreement metrics under full-context conditioning.

Rank	Metric	Mean ROC-AUC	Median ROC-AUC	Top-10 tasks
1	causal_z_exec_l2	0.5985	0.6158	5/8
2	ema_dev_exec_l2	0.5904	0.5813	6/8
3	overlap_energy_distance	0.5597	0.5095	5/8
4	cumulative_mean_drift	0.5570	0.5176	6/8
5	mean_dimension_wasserstein1	0.5518	0.5030	6/8
6	cumulative_endpoint_drift	0.5452	0.5070	5/8
7	overlap_rbf_mmd2_multiscale	0.5445	0.5120	5/8
8	abs_log_norm_ratio	0.5361	0.5572	4/8
9	first_nmse	0.5273	0.4996	3/8
10	delta_action_acc_ema09	0.5142	0.5098	3/8

Table 6: Top-10 decoder-only disagreement metrics under current-only conditioning.

Rank	Metric	Mean ROC-AUC	Median ROC-AUC	Top-10 tasks
1	overlap_energy_distance	0.6313	0.5934	5/8
2	cumulative_mean_drift	0.6210	0.5899	4/8
3	mean_dimension_wasserstein1	0.6202	0.5861	4/8
4	overlap_rbf_mmd2_multiscale	0.6188	0.5718	4/8
5	bid_lb	0.6094	0.5959	3/8
6	cumulative_endpoint_drift	0.6030	0.5779	5/8
7	causal_z_exec_l2	0.6002	0.5589	3/8
8	abs_log_norm_ratio	0.5973	0.5758	5/8
9	ema_dev_exec_l2	0.5908	0.5850	5/8
10	first_nmse	0.5863	0.5534	4/8

showing that causal temporal information remains present even when the policy does not receive accumulated observation-action history.

A.3.2 Flow-Matching Transformer

Full-context. For Flow Matching, distributional disagreement is especially consistent across tasks. Wasserstein-1 and Energy distance appear among the task-wise top ten on all eight tasks and achieve mean ROC-AUC 0.7045 and 0.6831, respectively. Wasserstein-1 also remains above chance on every task, with minimum task-level ROC-AUC 0.5446.

Current-only. The same broad pattern persists under current-only conditioning. Wasserstein-1 and Energy distance reach mean ROC-AUC 0.6678 and 0.6610, while Causal Z-score remains competitive and achieves the highest median ROC-AUC among the top-ranked metrics (0.6740).

Across all four architecture-context configurations, individual rankings change but the same broad signal families repeatedly appear among the strongest measures: distributional, cumulative, and causal temporal disagreement.

A.4 Complementary evaluation criteria

Frame ROC-AUC is the primary cross-task screening measure, but it does not capture all properties relevant to online event detection. We therefore perform two complementary analyses: prevalence-aware frame-level ranking and autonomous temporal peak localization.

A.4.1 Prevalence-normalized Frame AP

We normalize Average Precision relative to the random baseline implied by positive-frame prevalence,

$$\text{AP}_{\text{norm}} = \frac{\text{AP} - \pi}{1 - \pi},$$

where π denotes the positive-frame prevalence.

Table 7: Top-10 Flow-Matching disagreement metrics under full-context conditioning.

Rank	Metric	Mean ROC-AUC	Median ROC-AUC	Top-10 tasks
1	mean_dimension_wasserstein1	0.7045	0.6683	8/8
2	overlap_energy_distance	0.6831	0.6536	8/8
3	overlap_rbf_mmd2_multiscale	0.6542	0.6201	6/8
4	cumulative_endpoint_drift	0.6468	0.5969	5/8
5	cumulative_mean_drift	0.6460	0.5824	5/8
6	overlap_max_step_l2	0.6380	0.6123	4/8
7	causal_z_exec_l2	0.6373	0.6259	3/8
8	overlap_huber	0.6321	0.5971	4/8
9	overlap_l2_mean	0.6315	0.5823	3/8
10	overlap_max_abs	0.6305	0.5928	2/8

Table 8: Top-10 Flow-Matching disagreement metrics under current-only conditioning.

Rank	Metric	Mean ROC-AUC	Median ROC-AUC	Top-10 tasks
1	mean_dimension_wasserstein1	0.6678	0.6680	6/8
2	overlap_energy_distance	0.6610	0.6242	5/8
3	cumulative_endpoint_drift	0.6521	0.6289	3/8
4	cumulative_mean_drift	0.6456	0.6275	4/8
5	overlap_rbf_mmd2_multiscale	0.6450	0.6020	3/8
6	a14_trans	0.6406	0.6471	4/8
7	causal_z_exec_l2	0.6396	0.6740	4/8
8	relative_frobenius	0.6391	0.6116	3/8
9	ema_dev_exec_l2	0.6364	0.6676	4/8
10	overlap_l2_mean	0.6320	0.6466	4/8

The task-level analysis reveals substantial heterogeneity. The clearest separation is observed on RememberColor3, FindImposterShape3, and ShellGameColorLampTouch, whereas InterceptSlow and RotateLenientPos yield much weaker scalar disagreement signals.

A.4.2 Autonomous temporal localization

Frame-level separability does not imply that a disagreement measure forms an isolated temporal response around a semantic event. We therefore additionally evaluate Autonomous Peak AP, where local maxima are matched one-to-one to annotated phase transitions.

This analysis complements the Frame ROC-AUC screening. Causal temporal signals produce localized responses around semantic transitions even in settings where distributional disagreement provides stronger overall frame-level separation.

A.5 Correlation structure and metric redundancy

Many disagreement measures quantify related aspects of the same underlying plan revision. We therefore analyze their redundancy using pairwise within-episode Spearman correlations, aggregated equally across the eight tasks.

The correlation matrices contain clear blocks of strongly related signals. Point-wise and action-component measures are often strongly correlated with one another, and cumulative and distributional measures also exhibit substantial redundancy. In contrast, causal temporal measures occupy a more distinct region of the correlation structure, consistent with their complementary temporal behavior in the peak-localization analysis.

The global correlation organization is highly stable across context regimes. Correlating the upper triangles of the aggregated full-context and current-only matrices gives

$$\rho_{\text{context}}^{\text{Dec}} = 0.9647, \quad \rho_{\text{context}}^{\text{Flow}} = 0.9742.$$

Thus, although the absolute informativeness of individual metrics changes with the available context, their relationships to one another remain largely preserved. This redundancy motivates carrying a

Table 9: Task-wise prevalence-normalized Frame AP for decoder-only and Flow-Matching policies. Higher values indicate a larger improvement over the prevalence-defined baseline.

Task	Decoder-only		Flow Matching	
	Full	Current-only	Full	Current-only
Remember Color 3	0.6311	0.6978	0.7559	0.6924
Find Imposter Shape	0.4680	0.6651	0.5043	0.6283
ShellGame Color Lamp Touch	0.5211	0.4525	0.4894	0.5671
ShellGame Shuffle Touch	0.3331	0.1375	0.4062	0.2614
ShellGame Shuffle Color Lamp Touch	0.2845	0.1304	0.0591	0.0900
Take It Back	0.2030	0.1147	0.0579	0.2661
Intercept Slow	0.0359	0.0855	0.0480	0.0315
Rotate Lenient Pos	0.0259	0.0119	0.0987	0.2131

Table 10: Autonomous peak localization for representative causal temporal signals. Normalized Peak AP is measured relative to the corresponding null baseline.

Policy	Context	Metric	Median Peak AP	Median norm. Peak AP
Decoder-only	Full	Causal Z-score	0.5607	0.5186
Decoder-only	Full	EMA deviation	0.5708	0.5190
Decoder-only	Current-only	Causal Z-score	0.3696	0.3136
Decoder-only	Current-only	EMA deviation	0.2852	0.2038
Flow Matching	Full	Causal Z-score	0.5641	0.5268
Flow Matching	Full	EMA deviation	0.4827	0.4429
Flow Matching	Current-only	Causal Z-score	0.4600	0.4221
Flow Matching	Current-only	EMA deviation	0.3707	0.3023

compact representation rather than the full set of 35 candidate metrics into the more computationally expensive VLA experiments.

A.6 Robustness to trajectory outcome

We next test whether disagreement remains informative when the policy ultimately fails to complete the task. This distinguishes a signal that tracks semantic task changes from one that is present only along successful executions.

Table 11 reports Frame ROC-AUC after stratifying trajectories by final task outcome. To keep the comparison compact, we show representative causal temporal and distributional signals.

For Flow Matching, all eight tasks contain both successful and failed rollouts in both context regimes. The full-context analysis contains 108 successful and 292 failed episodes, while current-only contains 117 successful and 283 failed episodes.

Across both architectures, failure trajectories do not exhibit a systematic collapse in frame-level separability. Some disagreement signals are stronger in successful episodes and others in failed episodes. We therefore do not interpret the direction of individual success–failure differences as evidence that either outcome is intrinsically easier to detect. The relevant result is that task-relevant plan revision remains observable even when the policy ultimately fails to complete the task.

A.7 Detector feature representation used in RQ2

The purpose of RQ1 is not to identify a single universally optimal scalar metric. Rather, the screening study establishes that multiple forms of plan revision carry useful information and that many individual candidates are strongly redundant. These observations guide the compact detector representation used in the pretrained-VLA experiments of RQ2.

These findings motivate constructing a compact representation for the subsequent VLA experiments. We retain nine complementary plan-disagreement signals from the RQ1 screening and add one VLA-specific visual-representation change feature. The resulting 10-dimensional representation is detailed in Appendix A.7 and evaluated on pretrained VLA policies in RQ2.

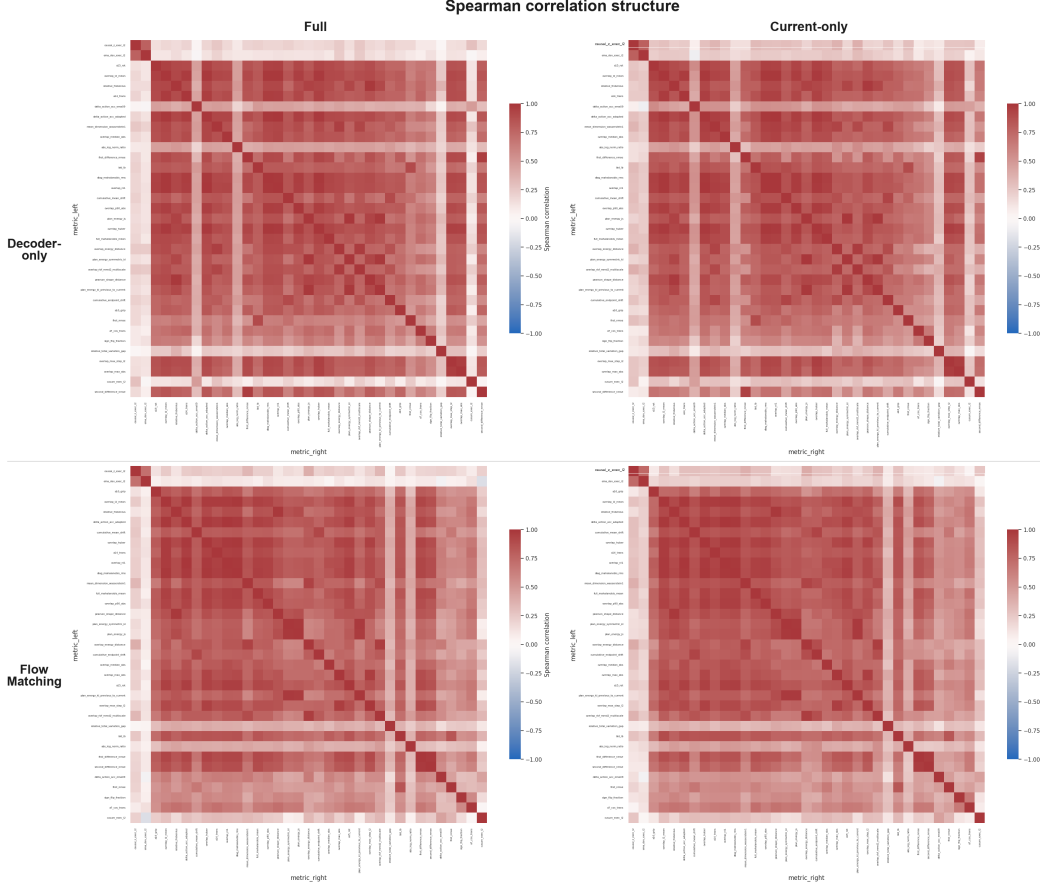


Figure 7: **Spearman correlation structure of the candidate disagreement metrics.** Rows correspond to decoder-only and Flow-Matching policies, and columns correspond to full-context and current-only conditioning. Pairwise correlations are computed within episodes and aggregated equally across the eight tasks. The same metric ordering and correlation scale are used in all panels, allowing direct comparison of the dependency structure across architectures and context regimes.

The RQ2 detector uses the following 10 base signals:

Figure 8 shows the within-episode Spearman structure of the compact feature set for the single-task artifacts where each signal is available.

B Additional Analysis for RQ2

This appendix complements Section 4.2. The main text reports that overlap-disagreement features transfer to pretrained VLAs and support online keyframe detection. Here we record the protocol, detector setup, full per-regime tables, the shared-task comparison, the class-balance effect on PR-AUC, and limitations needed to interpret those results. Notation follows Section 3: chunk horizon K , execution stride H , and replanning query q . In the RQ2 experiments, $K = 8$ and $H \in \{1, 4\}$, written RH1 and EXEC4.

B.1 Experimental setup and evaluation

Experiments use the MIKASA-Robo-VLA (Cherepanov et al., 2026b), which evaluates manipulation under partial observability and delayed temporal dependencies. Simulator phase transitions define the binary keyframe label. As in Appendix A.1, a phase boundary between frames $b - 1$ and b yields the event window $\mathcal{I}_e = [b - 1, b]$. Detectors never observe these labels at inference, they are used only to score held-out episodes. Two pretrained backbones are evaluated. OpenVLA-OFT is run on

Table 11: Frame ROC-AUC stratified by trajectory outcome for representative disagreement measures. Values are medians across the eight tasks.

Policy	Context	Metric	Success	Failure
Decoder-only	Full	Causal Z-score	0.589	0.700
		EMA deviation	0.577	0.616
		Energy distance	0.572	0.510
		Wasserstein-1	0.561	0.502
Decoder-only	Current-only	Causal Z-score	0.538	0.691
		EMA deviation	0.585	0.622
		Energy distance	0.596	0.648
		Wasserstein-1	0.611	0.604
Flow Matching	Full	Causal Z-score	0.642	0.627
		EMA deviation	0.667	0.590
		Energy distance	0.643	0.694
		Wasserstein-1	0.670	0.732
Flow Matching	Current-only	Causal Z-score	0.633	0.669
		EMA deviation	0.696	0.643
		Energy distance	0.618	0.703
		Wasserstein-1	0.640	0.741

Table 12: Base detector features used in RQ2. The VLM top-view feature is available only for pretrained VLA policies and is therefore not part of the 35-metric single-task screening.

Feature	Role
Overlap L2	Direct magnitude of aligned action disagreement
BiD lower bound	Horizon-weighted overlap disagreement
VLM top-view L2	Visual representation change between consecutive observations
Causal Z-score	Current first-action revision relative to its causal baseline
CUSUM	Accumulated one-sided temporal change signal
EMA deviation	Deviation of the current revision from its recent moving average
A7 cosine translation	Directional disagreement in translational action components
A14 translation	Translation disagreement in controller-physical space
A15 rotation	Geodesic rotational disagreement
A16 gripper	Normalized gripper disagreement

all 22 tasks. $\pi_{0.5}$ is run on 19 shared tasks (the ShellGame environments are excluded because the corresponding assets were unavailable). For each task we collect 50 independent episodes and split at the episode level into approximately 70% train, 10% validation, and 20% test, so that detector calibration cannot leak frames from the same trajectory. Policies generate chunks of length $K = 8$. Two execution regimes are compared:

- RH1 ($H = 1$): one action is executed, then the policy is replanned at the next query q , providing step-wise visual feedback.
- EXEC4 ($H = 4$): four actions are executed open-loop before the next chunk is generated.

The overlap used for disagreement is the temporally aligned residual of the previous plan versus the matching prefix of the new plan (Eqs. (1)). At query q this yields paired sequences of length $K - H$ (7 under RH1, 4 under EXEC4). OpenVLA-OFT uses a single checkpoint in both regimes. $\pi_{0.5}$ uses the native weights associated with each execution regime. Absolute success rates should therefore not be read as a controlled ablation of H alone for $\pi_{0.5}$.

B.2 Features and detector architectures

At each query q we compute a 10-dimensional feature vector from the complementary distributional, cumulative, and causal temporal families identified in Section 4.1 and Appendix A. This is the frozen-policy screening vector used in RQ2. It is not identical to the SFT detector in Section 3,

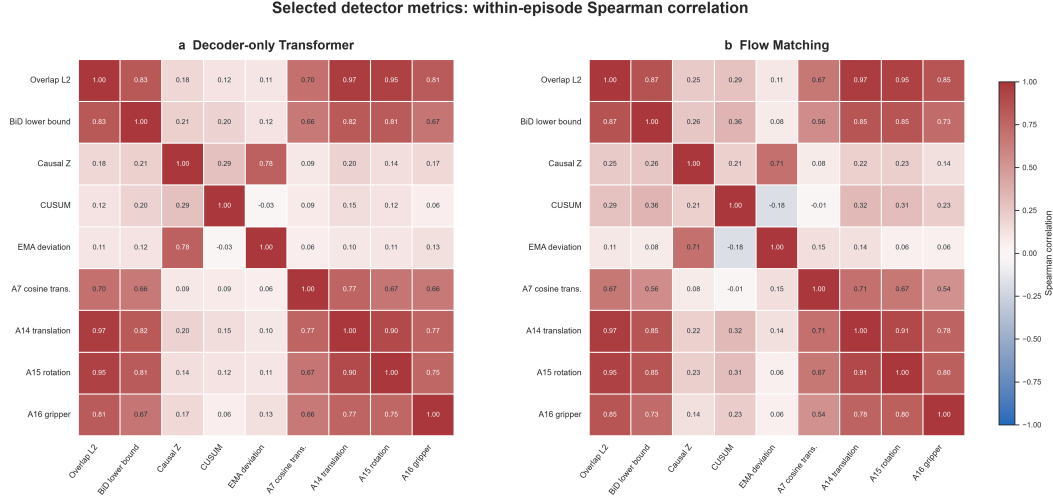


Figure 8: **Correlation structure of the detector representation used in RQ2.** Within-episode Spearman correlations are shown for decoder-only and Flow-Matching policies under full-context conditioning and are aggregated equally across the eight MIKASA-Robo-VLA tasks.

Table 13: OpenVLA-OFT: policy success and keyframe detectors under RH1 ($H = 1$) vs. EXEC4 ($H = 4$) on 22 tasks, $K = 8$. Companion to Section 4.2.

Regime	Mean SR	Classifier	PR-AUC	ROC-AUC	F1	pos.rate
RH1 ($H = 1$)	17.7%	XGBoost	0.932	0.982	0.841	0.197
RH1 ($H = 1$)	17.7%	CatBoost	0.929	0.981	0.831	0.197
RH1 ($H = 1$)	17.7%	LSTM	0.962	0.989	0.874	0.197
EXEC4 ($H = 4$)	18.7%	XGBoost	0.996	0.997	0.966	0.450
EXEC4 ($H = 4$)	18.7%	CatBoost	0.995	0.996	0.967	0.450
EXEC4 ($H = 4$)	18.7%	Causal LSTM	0.984	0.990	0.964	0.503

which additionally uses temporal differences, EMA deviations, a visual-change term, and a causal Mahalanobis score (Eqs. (2)–(4)). Detectors include tabular gradient-boosting models (XGBoost, CatBoost), trained frame-wise, and a causal LSTM that consumes the feature sequence within an episode. Decision thresholds maximize F1 on the validation split. Held-out test episodes are scored with PR-AUC, ROC-AUC, F1, positive-class rate (pos.rate), and event_recall@1 (at least one detection within ± 1 step of a phase boundary). We also report mean policy success (Mean SR).

B.3 OpenVLA-OFT under RH1 vs. EXEC4

Table 13 reports the OpenVLA-OFT numbers with positive rate included. Mean SR is similar across regimes (17.7% vs. 18.7%): reducing replan frequency by $4\times$ does not degrade manipulation. Detection remains strong in both settings. Under RH1, all classifiers reach event_recall@1 = 1.0, with ROC-AUC 0.981–0.989 and PR-AUC 0.929–0.962. Under EXEC4, PR-AUC is 0.984–0.996 and F1 rises from about 0.84 to about 0.96.

B.4 Keyframe detection for $\pi_{0.5}$

Table 14 reports the corresponding $\pi_{0.5}$ numbers on the 19 shared tasks. Mean SR falls from 13.2% under RH1 to 8.8% under EXEC4, consistent with accumulated positioning error without step-wise visual correction. Detectors nevertheless remain accurate: RH1 PR-AUC 0.924–0.933, ROC-AUC 0.984–0.985, F1 0.854–0.859, EXEC4 PR-AUC 0.981–0.983 and F1 0.917–0.942.

Table 14: $\pi_{0.5}$: policy success and keyframe detectors under RH1 ($H = 1$) vs. EXEC4 ($H = 4$) on 19 shared tasks, $K = 8$. Companion to Section 4.2.

Regime	Mean SR	Classifier	PR-AUC	ROC-AUC	F1	pos.rate
RH1 ($H = 1$)	13.2%	XGBoost	0.928	0.984	0.854	0.180
RH1 ($H = 1$)	13.2%	CatBoost	0.924	0.984	0.854	0.180
RH1 ($H = 1$)	13.2%	Causal LSTM	0.933	0.985	0.859	0.183
EXEC4 ($H = 4$)	8.8%	XGBoost	0.982	0.988	0.917	0.388
EXEC4 ($H = 4$)	8.8%	CatBoost	0.981	0.987	0.921	0.388
EXEC4 ($H = 4$)	8.8%	Causal LSTM	0.983	0.988	0.942	0.463

B.5 Class balance and PR-AUC across execution strides

The PR-AUC increase at $H = 4$ is partly a change in class balance. Under RH1, a query q occurs at every step, so trajectory tables are long and phase events are relatively rare (pos.rate ≈ 0.18 – 0.20). Under EXEC4, the number of queries per episode drops by about $4\times$ while the number of phase transitions is unchanged, and the positive rate rises to 0.39 – 0.50 . Because a PR curve’s baseline depends on prevalence, PR-AUC is not comparable across values of H without this context. ROC-AUC, which is invariant to prevalence, stays near 0.98 in both regimes and is the fairer comparison across strides. Cross-horizon detector comparisons should therefore report positive rate together with ROC-AUC.

B.6 Cross-model comparison on shared tasks

Restricting OpenVLA-OFT to the same 19 tasks as $\pi_{0.5}$ preserves a large control gap: OpenVLA-OFT completes 18.8% of episodes under RH1 and 18.0% under EXEC4, versus 13.2% and 8.8% for $\pi_{0.5}$. Under EXEC4 the success advantage is more than twofold. Detection quality does not track this gap. Under EXEC4, XGBoost PR-AUC is 0.996 for OpenVLA-OFT vs. 0.982 for $\pi_{0.5}$, and the causal LSTM yields 0.984 vs. 0.983. Plan disagreement therefore marks semantic phase boundaries even when the policy fails at end-effector positioning: external condition changes still reorganize the predicted chunk.

B.7 Qualitative localization

Figure 9 shows within-episode profiles on RememberColor5 (OpenVLA-OFT: episode_006, $\pi_{0.5}$ exec4: episode_008). Normalized disagreement stays near zero while the cue is visible. At the transition to manipulation, metrics such as `a14_trans` and `overlap_12_mean` form peaks more than three standard deviations above background, and XGBoost, CatBoost, and LSTM probabilities rise to nearly 1.0. After the arm settles into steady motion, scores return to baseline and detector probabilities drop to zero. Under RH1 ($H = 1$, panels a,c), ground-truth windows are compact and align with peaks near steps 4 and 8. Under EXEC4 ($H = 4$, panels b,d), a query exists only every four steps, so consecutive positive ground-truth ticks can visually merge into a wider band. This should not be read as one long continuous phase. Across both models and both regimes, detectors fire at early phase boundaries and remain quiet during later steady motion, consistent with Tables 13 and 14.

B.8 Summary

Overlap disagreement transfers from the single-task screening to pretrained OpenVLA-OFT and $\pi_{0.5}$. Detection remains accurate under both $H = 1$ and $H = 4$ and does not require high task success. These results motivate using a compact surprise score as the write gate for slot memory during VLA fine-tuning (RQ3).

C Complete Registry of Disagreement Metrics

The single-task screening uses 35 disagreement signals. Let $P, Q \in \mathbb{R}^{L \times d}$ denote the temporally aligned overlapping parts of two consecutive plans.

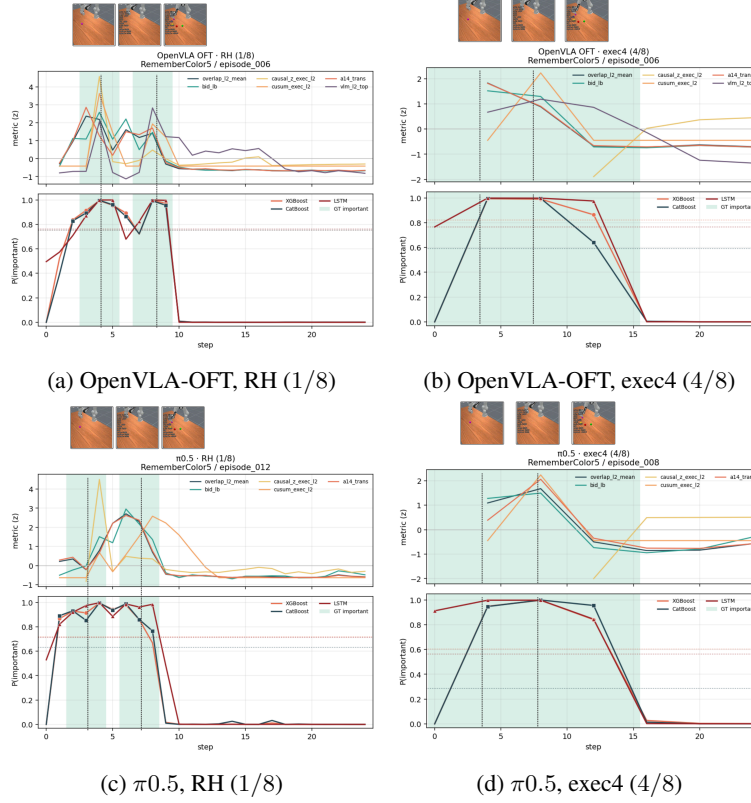


Figure 9: Episode timelines on RememberColor5 (OpenVLA-OFT: episode_006, $\pi 0.5$ exec4: episode_008). Each panel shows within-episode z -scored surprise metrics (top) and $P(\text{important})$ from XGBoost / CatBoost / LSTM (bottom). Green shading is ground-truth important (windows around simulator phase transitions).

Table 15: Complete registry of the 35 disagreement signals.

#	Metric	Definition / interpretation	Space
1	overlap_l2_mean	$\frac{1}{L} \sum_i \ P_i - Q_i\ _2$	raw action
2	bid_lb	$\sum_i 0.5^i \ P_i - Q_i\ _2$	raw action
3	causal_z_exec_l2	Causal Z-score of the change in the first policy output	raw first output
4	cusum_exec_l2	One-sided CUSUM of the change in the first policy output	raw first output
5	ema_dev_exec_l2	Deviation of the first-output change from its causal EMA	raw first output
6	a7_cos_trans	Mean cosine distance between translation components	raw action
7	a14_trans	Translation disagreement in controller-physical space	controller physical
8	a15_rot	Geodesic rotation disagreement in $SO(3)$	controller physical
9	a16_grip	Normalized gripper disagreement	raw action
10	first_nmse	Normalized MSE of the first aligned action pair	normalized action
11	overlap_nll	Mean normalized L1 overlap error	normalized action

#	Metric	Definition / interpretation	Space
12	<code>overlap_huber</code>	Mean Huber loss over the overlap	normalized action
13	<code>overlap_median_abs</code>	Median absolute component-wise error	normalized action
14	<code>overlap_p90_abs</code>	90th percentile of absolute component-wise error	normalized action
15	<code>overlap_max_abs</code>	Maximum absolute component-wise error	normalized action
16	<code>overlap_max_step_l2</code>	Maximum per-step L2 disagreement	normalized action
17	<code>relative_frobenius</code>	$\frac{\ P - Q\ _F}{0.5(\ P\ _F + \ Q\ _F) + \epsilon}$	normalized action
18	<code>abs_log_norm_ratio</code>	$\left \log \frac{\ P\ _F + \epsilon}{\ Q\ _F + \epsilon} \right $	normalized action
19	<code>diag_mahalanobis_rms</code>	RMS disagreement normalized by training-set standard deviations	raw / train std
20	<code>full_mahalanobis_mean</code>	Mean Mahalanobis distance using the training covariance	raw / train covariance
21	<code>delta_action_acc_adapted</code>	Absolute disagreement normalized by the training action range	raw / train range
22	<code>delta_action_acc_ema09</code>	EMA-smoothed range-normalized disagreement	raw / train range
23	<code>pearson_shape_distance</code>	$1 - \text{corr}(\text{vec } P, \text{vec } Q)$	normalized action
24	<code>plan_energy_kl_previous_to_current</code>	$D_{\text{KL}}(p q)$ between normalized plan-energy distributions	normalized action
25	<code>plan_energy_symmetric_kl</code>	Symmetrized KL divergence between plan-energy distributions	normalized action
26	<code>plan_energy_js</code>	Jensen–Shannon divergence between plan-energy distributions	normalized action
27	<code>sign_flip_fraction</code>	Fraction of components with different signs in P and Q	raw action
28	<code>first_difference_nmse</code>	MSE between first temporal differences of the two plans	normalized action
29	<code>second_difference_nmse</code>	MSE between second temporal differences of the two plans	normalized action
30	<code>relative_total_variation_gap</code>	Relative difference in total variation between the two plans	normalized action
31	<code>cumulative_mean_drift</code>	Mean norm of accumulated disagreement along the prediction horizon	normalized action
32	<code>cumulative_endpoint_drift</code>	Norm of the final accumulated displacement disagreement	normalized action
33	<code>overlap_energy_distance</code>	Empirical multivariate Energy distance	normalized action
34	<code>mean_dimension_wasserstein1</code>	Mean one-dimensional Wasserstein-1 distance across action dimensions	normalized action
35	<code>overlap_rbf_mmd2_multiscale</code>	Multiscale RBF MMD ² over the overlapping plan segments	normalized action