SPLIT^3: SEPARATING TEST TIME TRAINING PLASTICITY FOR ANOMALY DETECTION AND LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series (MTS) anomaly diagnosis requires identifying both when anomalies occur and which variables are associated with them. Test Time Training (TTT) continuously adapts a predictive model at test time, but detection and localization can favor different levels of adaptation. We derive an exact recurrence describing how differences in TTT update rates propagate through the evolving fast state and relate the resulting prediction-error differences to the distinct ranking structures of detection and localization. To allow detection and localization to follow task-specific adaptation trajectories within the same trained predictive model, we propose Separated PLasticity In Test Time Training (SPLIT^3), which shares the encoder and learned initialization while maintaining separate fast states with task-specific update rates. Experiments on the SMD, SWaT, and WaDI datasets consistently show that detection favors lower update rates than localization. Across multiple MTS benchmarks, SPLIT^3 achieves strong localization performance while maintaining competitive detection performance.

1 INTRODUCTION

Multivariate Time Series (MTS) are widely used to monitor complex systems, including industrial control systems, servers, and spacecraft (Su et al., 2019). Anomaly diagnosis in these systems requires more than detecting when abnormal behavior occurs. It is also important to identify which variables are associated with the anomaly. Detection indicates when attention is needed, while localization narrows the set of variables that should be inspected (Abdulaal et al., 2021). A useful diagnosis system should therefore support both tasks.

Recent work has introduced Test Time Training (TTT) to MTS anomaly detection, allowing predictive models to adapt to patterns encountered in the test stream (Kim et al., 2024; 2026; Gong et al., 2025; Zhang et al., 2025). This adaptation creates an important tension for anomaly diagnosis. Prediction errors provide the signal used to update the model, but the same errors also provide the evidence used to detect and localize anomalies (Tian et al., 2023). As the model adapts, the diagnostic evidence itself can therefore change.

Existing test time adaptation methods for MTS have largely been developed from the perspective of anomaly detection. When the same adaptive predictor is also used for localization, however, it is unclear whether the adaptation behavior that preserves useful detection evidence is also suitable for identifying anomalous variables. This leads to a natural question: *if detection and localization use the same predictive model, should they also share the same adaptation state?*

We examine this question empirically by varying only the TTT update rate at test time while keeping the trained predictive model fixed. As shown in Figure 1, the resulting rate sweeps on the SMD, SWaT, and WaDI datasets consistently favor lower update rates for detection Average Precision (AP) than for localization Hit Rate (HR), revealing distinct task-specific adaptation preferences within the same model.

To understand this difference, we examine how test time adaptation unfolds through the evolving fast state. Each test time update is accumulated in the fast state and can influence predictions later in the stream. Changing the update rate can therefore change the adaptation trajectory and alter the prediction errors produced along that trajectory. Detection and localization then evaluate these errors through different ranking structures: detection compares anomaly scores across timestamps, whereas

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

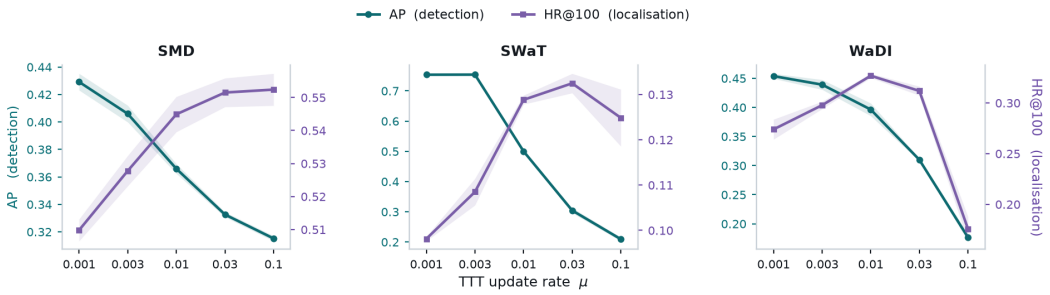


Figure 1: **Detection and localization respond differently to the TTT update rate.** Using the same trained predictive model, we vary only the test time update rate μ. Across SMD, SWaT, and WaDI, detection AP favors lower update rates than localization HR@100 within the evaluated range.

localization compares variables within each timestamp. The same change in test time adaptation can therefore have different consequences for the two diagnostic tasks.

Motivated by these task-specific adaptation preferences, we propose Separated PLasticity In Test Time Training (SPLIT^3). SPLIT^3 keeps the trained predictive components shared while maintaining separate fast states for detection and localization, allowing each task to follow its own adaptation trajectory under the same prediction objective and update rule. Our main contributions are summarized as follows:

- We empirically identify a consistent task dependent difference in test time update rate preferences across SMD, SWaT, and WaDI, with detection favoring lower rates than localization.
- We derive an exact recurrence for fast state differences under different update rates and relate the resulting prediction error differences to the distinct ranking structures of detection and localization.
- We introduce SPLIT^3, which maintains separate fast states for detection and localization while sharing the trained encoder and learned initialization, allowing the two tasks to use different update rates within one predictive model

2 Related Work

MTS Anomaly Detection and Localization. MTS anomaly diagnosis requires first determining whether the current observation deviates from normal behavior and then identifying which variables are most associated with the anomaly (Kong et al., 2027). Existing methods typically learn relationships among variables or temporal patterns from normal data and use deviations from these learned patterns as anomaly evidence. For example, SARAD (Dai et al., 2024) and CST-GL (Zheng et al., 2023) model relationships among variables and detect anomalies when these relationships change substantially. AERCA (Han et al., 2025b) and GCAD (Liu et al., 2025) use causal relationships to model how variables explain one another. CrossAD (Li et al., 2026) and CATCH (Wu et al., 2025) instead capture normal behavior across multiple temporal scales and in the frequency domain, respectively, and detect observations that deviate from these patterns.

For localization, the anomaly evidence is further resolved at the variable level. SARAD, CST-GL, and AERCA use changes in their learned relationships to identify variables associated with an anomaly, while ALoRa (Shimillas et al., 2026) ranks variables according to their contributions to the reconstructed outputs. These methods differ primarily in how they represent anomaly evidence and translate it into variable level localization scores.

Adaptation at Test Time for MTS Anomaly Detection. A static anomaly detector applies the normal patterns learned during training throughout the test stream. When the normal operating pattern changes, this mismatch can increase anomaly scores even for normal observations. Test time adaptation addresses this problem by updating the model using patterns observed during the test

stream (Gong et al., 2025; Zhang et al., 2025). M2N2 (Kim et al., 2024) combines trend estimation with self supervision during adaptation so that the model can follow changes in the test distribution. CANDI (Kim et al., 2026) instead focuses on which test observations should be used for adaptation, selecting samples based on anomaly scores and representation similarity. Related work similarly studies how to avoid harmful updates from unreliable test observations through sample selection, knowledge preservation, and stable online adaptation (Ma et al., 2025; Han et al., 2025a).

These approaches primarily use adaptation to maintain anomaly detection under distribution changes, whereas we study how the same online adaptation affects both anomaly detection and variable localization.

Fast Weights and TTT. Fast weights store information in parameters that evolve while a sequence is processed (Ba et al., 2016; Sun et al., 2024). Unlike recurrent activation states, these parameters directly modify how subsequent inputs are processed. Prior work connects such fast weights to linear attention and parameterized memory, where new observations update stored associations that are later used for prediction (Schlag et al., 2021; Wang et al., 2026; Yang et al., 2025).

This perspective is directly relevant to TTT, where inference time updates form an evolving fast state that can influence subsequent predictions (Zhang et al., 2026). Building on this view, we analyze how update rate differences propagate through the fast state and relate the resulting prediction error changes to the distinct ranking structures of detection and localization.

3 How TTT Affects Detection and Localization

3.1 Prediction and Fast State Updates

Let $\mathbf{x}_t \in \mathbb{R}^d$ denote the multivariate observation at time t. We denote the fast weight matrix before incorporating $\mathbf{x}_t$ by $\mathbf{W}_{t-1} \in \mathbb{R}^{d\times m}$ and refer to it as the fast state. The prediction, prediction error, and fast state update are

$$\begin{aligned} \hat{\mathbf{x}}_t &= \mathbf{W}_{t-1}\mathbf{k}_t, \qquad \mathbf{k}_t = \phi_\theta(\mathbf{x}_{t-p:t-1}), \\ \mathbf{e}_t^- &= \mathbf{x}_t - \hat{\mathbf{x}}_t, \qquad q_t = \left(\|\mathbf{k}_t\|_2^2 + \delta\right)^{-1}, \\ \mathbf{W}_t &= \mathbf{W}_{t-1} + \mu q_t \mathbf{e}_t^- \mathbf{k}_t^\top, \end{aligned} \tag{1}$$

where ϕ_θ is a fixed encoder, $\delta > 0$ is a stabilization constant, and $\mu \geq 0$ controls the magnitude of the fast state update. The prediction error $\mathbf{e}_t^-$ is computed before the current observation is used to update the fast state. Inference therefore follows *predict* $\rightarrow$ *score* $\rightarrow$ *update*, so the current observation is scored before it is incorporated into the fast state.

3.2 Accumulated Fast State Dynamics

A fast state update can influence predictions at later timestamps. To make this temporal effect explicit, consider a reference trajectory and another trajectory in which only the update at time s is omitted. Because the encoder is fixed and $\mathbf{k}_j$ depends only on the observed history, the two trajectories use the same subsequent $\mathbf{k}_j$ and q_j.

Let $\mathbf{d}_{s\rightarrow t}$ denote the difference between their predictions at a later time $t > s$. Then

$$\mathbf{d}_{s\rightarrow t} = \underbrace{\mu q_s \mathbf{e}_s^- \mathbf{k}_s^\top}_{\text{update at } s} \underbrace{\left[\prod_{j=s+1}^{t-1} \left(\mathbf{I} - \mu q_j \mathbf{k}_j \mathbf{k}_j^\top\right)\right]}_{\text{subsequent state transformations}} \underbrace{\mathbf{k}_t}_{\text{read at } t}. \tag{2}$$

The product is ordered in increasing time, with an empty product defined as the identity matrix. Equation 2 makes the temporal nature of test time adaptation explicit. An update introduced at time s is carried through subsequent fast state transformations and can later appear in the prediction at time t. Its future influence therefore depends not only on the original update, but also on how the fast state evolves afterward and how that state is read by future observations. The derivation is provided in Appendix A.1.

Changing the update rate can change how the effects of successive updates accumulate along the fast state trajectory. Consider two trajectories with update rates $\mu_h > \mu_\ell$, while keeping the encoder, observation sequence, and initial fast state identical. Define their fast state difference as

$$\mathbf{D}_t = \mathbf{W}_t^h - \mathbf{W}_t^\ell.$$

Subtracting the two update equations gives the exact recurrence

$$\mathbf{D}_t = \mathbf{D}_{t-1}\left(\mathbf{I} - \mu_h q_t \mathbf{k}_t \mathbf{k}_t^\top\right) + (\mu_h - \mu_\ell) q_t \mathbf{e}_t^{\ell,-} \mathbf{k}_t^\top, \tag{3a}$$

$$\mathbf{e}_t^{h,-} - \mathbf{e}_t^{\ell,-} = -\mathbf{D}_{t-1}\mathbf{k}_t. \tag{3b}$$

The first term in Equation 3a transforms the state difference accumulated from earlier updates, while the second introduces a new difference due to the update rates at time t. Equation 3b shows how the accumulated state difference appears in the current prediction error.

Changing the update rate therefore does not merely rescale individual updates. The accumulated state difference enters the prediction error difference through $-\mathbf{D}_{t-1}\mathbf{k}_t$, whose components need not be equal across variables. The complete unrolled recurrence is provided in Appendix A.2.

3.3 Consequences for Detection and Localization

The prediction error differences induced by Equation 3 are evaluated differently by detection and localization. Detection ranks anomaly scores across timestamps, whereas localization ranks variables within each timestamp.

Detection. To isolate the effect of prediction errors, consider a fixed diagonal normalization matrix $\mathbf{C}$ with positive diagonal entries and define

$$Q_t = (\mathbf{e}_t^-)^\top \mathbf{C} \mathbf{e}_t^-. \tag{4}$$

The detection score is a strictly increasing function of this quantity, so ranking by the score is equivalent to ranking by Q_t for fixed $\mathbf{C}$. Let $\mathbf{u}_t = \mathbf{e}_t^{h,-} - \mathbf{e}_t^{\ell,-}$. The change in detection energy is

$$\Delta Q_t = 2(\mathbf{e}_t^{\ell,-})^\top \mathbf{C}\mathbf{u}_t + \mathbf{u}_t^\top \mathbf{C}\mathbf{u}_t. \tag{5}$$

A change in prediction error at anomalous timestamps alone therefore does not determine the change in detection AP. Scores at normal timestamps can also change, and AP depends on the relative ranking of anomalous and normal timestamps.

Localization. Localization involves a different comparison. At time t, consider a variable i in the localization ground truth set and a variable j outside the set. For $r \in \{\ell, h\}$, define their squared error margin as

$$M_{ij,t}^r = (e_{i,t}^{r,-})^2 - (e_{j,t}^{r,-})^2,$$

and define the signed reduction in squared error for variable i under the higher update rate as

$$A_{i,t} = (e_{i,t}^{\ell,-})^2 - (e_{i,t}^{h,-})^2.$$

The change in the margin is

$$\Delta M_{ij,t} = M_{ij,t}^h - M_{ij,t}^\ell = A_{j,t} - A_{i,t}. \tag{6}$$

Equation 6 shows that localization depends on the relative changes in prediction errors across variables. For example, the rank of a labeled variable can improve even when its own prediction error decreases, if the errors of competing variables decrease by larger amounts.

In summary, the analysis characterizes how update rate differences propagate through the fast state trajectory and how the resulting state differences are expressed in subsequent prediction errors. Detection evaluates these errors through the ranking of anomaly scores across timestamps, whereas localization evaluates them through the ranking of variables within each timestamp. The analysis does not determine which update rate is preferable for either task; we evaluate these task specific preferences empirically. Detailed derivations are provided in Appendix A.3.

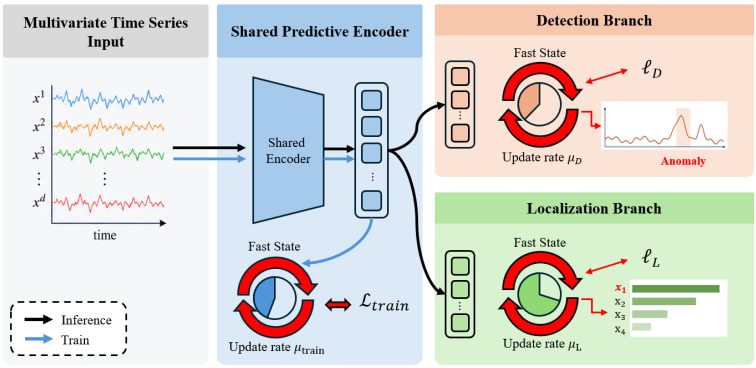


Figure 2: **Overview of SPLIT^3.** Detection and localization share the encoder and learned initialization, while maintaining separate fast states at test time.

4 SPLIT^3: Separated Plasticity in TTT

The analysis in Section 3 relates the dynamics of fast state differences to the distinct ranking structures of detection and localization. Motivated by the different update rate preferences observed in Figure 1, we propose SPLIT^3, which shares the encoder and learned initialization between detection and localization while maintaining a separate fast state for each task. The two fast states use the same prediction objective and update rule, with a task-specific update rate assigned to each state.

4.1 Shared Predictive Training

SPLIT^3 trains one predictive model using normal sequences. The encoder ϕ_θ and initial fast state $\mathbf{W}_0$ are learned jointly while a single fast state evolves along each training sequence according to Equation 1 with update rate μ_{train}. For a training chunk $\mathcal{B}$, the training objective is

$$\mathcal{L}_{\text{train}} = \frac{1}{|\mathcal{B}|d} \sum_{t \in \mathcal{B}} \|\mathbf{x}_t - \mathbf{W}_{t-1}\mathbf{k}_t\|_2^2 , \tag{7}$$

where d is the number of variables. Training uses only normal sequences and does not use a separate localization objective. The learned encoder and initialization are then shared by detection and localization. Architecture and optimization details are provided in Appendix B.2.

4.2 Separate Fast States at Test Time

At test time, the encoder and learned initialization are fixed. Detection and localization use the same key $\mathbf{k}_t$ and update normalization term q_t from Equation 1. The learned initialization is copied into two fast states:

$$\mathbf{W}_0^D = \mathbf{W}_0^L = \mathbf{W}_0. \tag{8}$$

For each task $\tau \in \{D, L\}$, the fast state is adapted using the same prediction objective,

$$\ell_t^\tau(\mathbf{W}) = \frac{1}{2} \|\mathbf{x}_t - \mathbf{W}\mathbf{k}_t\|_2^2 . \tag{9}$$

At time t, the prediction error for task τ is $\mathbf{e}_t^{\tau,-} = \mathbf{x}_t - \mathbf{W}_{t-1}^\tau \mathbf{k}_t$, and its fast state is updated as

$$\mathbf{W}_t^\tau = \mathbf{W}_{t-1}^\tau + \mu_\tau q_t \mathbf{e}_t^{\tau,-} \mathbf{k}_t^\top , \qquad \tau \in \{D, L\}, \tag{10}$$

where μ_D and μ_L are the update rates used for detection and localization, respectively. Thus, the two tasks share the encoder, initialization, prediction objective, and update rule, but maintain separate fast states with task-specific update rates.

Detection and localization scores. For detection, the anomaly score is

$$s_t^D = \log \left[\epsilon_s + (\mathbf{e}_t^{D,-})^\top \mathbf{C}_D \mathbf{e}_t^{D,-} \right], \tag{11}$$

where ϵ_s is a small constant for numerical stability and

$$\mathbf{C}_D = \mathrm{diag}(\sigma_1^{-2}, \ldots, \sigma_d^{-2})$$

is a diagonal normalization matrix constructed from prediction error scales estimated on normal validation data. For localization, each variable is scored directly by its squared prediction error,

$$s_{i,t}^L = \left(e_{i,t}^{L,-} \right)^2, \qquad i = 1, \ldots, d. \tag{12}$$

Variables are ranked in descending order according to these scores. No separate localization model is used. After scoring, both fast states are updated at every timestamp using their respective update rates.

Relation to independent TTT runs. Each fast state follows the same sequence as an independent TTT run with the corresponding update rate. Therefore,

$$\mathbf{W}_t^\tau = \mathbf{W}_t^{\mathrm{TTT}}(\mu_\tau), \qquad \tau \in \{D, L\}. \tag{13}$$

Thus, the detection fast state follows the sequence associated with μ_D, while the localization fast state follows the sequence associated with μ_L. SPLIT^3 allows these two fast states to evolve separately while sharing the same encoder and learned initialization. The proof is provided in Appendix A.4.

Computational cost. The shared encoder is evaluated once at each timestamp. Compared with TTT using a single fast state, SPLIT^3 stores one additional $d \times m$ fast state and performs one additional prediction and update for that state. Detailed parameter counts and arithmetic GFLOPs are reported in Appendix D.

5 Experiments

5.1 Overall Anomaly Diagnosis Performance

We evaluate detection on SMD (Su et al., 2019), SWaT (Mathur & Tippenhauer, 2016), WaDI (Ahmed et al., 2017), HAI (Shin et al., 2021), and MSL (Hundman et al., 2018), and localization on SMD, SWaT, and WaDI. Unless otherwise stated, we use $\mu_{\mathrm{train}} = 0.001$, $\mu_D = 0.001$, and $\mu_L = 0.01$. Detailed experimental settings, evaluation metrics, and rate selection are provided in Appendix B, with full results in Appendix C.1.

Tables 1 and 2 compare SPLIT^3 with existing methods for localization and detection. We use these comparisons to examine whether the different adaptation preferences of the two tasks can be accommodated within one trained predictive model while maintaining competitive performance on both tasks.

Localization. SPLIT^3 achieves the highest mean HR and NDCG among the compared methods on SMD, SWaT, and WaDI, and the highest mean IPS on SMD and SWaT. On WaDI, several baselines achieve higher IPS. These results show that the separated fast state design can provide strong localization performance while sharing the trained encoder and learned initialization with detection, without training a separate localization model.

Table 1: **Anomaly localization performance.** We report the mean HR@P, NDCG@P, and IPS@P over three runs. The best result for each metric is highlighted, and the second best is underlined.

Method	SMD						SWaT						WaDI					
	HR@P		NDCG@P		IPS@P		HR@P		NDCG@P		IPS@P		HR@P		NDCG@P		IPS@P	
	100	150	100	150	100	150	100	150	100	150	100	150	100	150	100	150	100	150
SARAD	52.78	65.91	56.28	63.75	58.55	73.08	3.08	14.77	3.13	10.31	7.82	11.52	19.45	23.51	19.47	21.98	15.87	26.19
CST-GL	44.30	57.13	47.66	54.99	57.18	70.35	1.69	1.87	1.76	1.87	7.61	8.02	22.10	26.03	22.93	25.36	28.17	33.33
AERCA	31.58	45.80	31.40	39.50	53.62	67.71	1.85	2.90	1.81	2.43	5.97	9.67	21.31	27.02	21.22	24.79	19.05	23.81
ALoRa	38.31	51.83	40.38	48.04	36.98	51.16	2.30	3.46	2.59	3.05	10.49	11.73	7.71	9.24	8.33	9.27	5.16	8.73
M2N2	50.81	64.20	53.77	61.40	70.26	84.93	7.49	11.97	7.71	10.45	22.63	37.45	30.03	42.88	30.20	38.13	30.95	30.95
CANDI	48.67	61.45	51.69	58.95	72.84	85.31	4.94	9.37	5.04	7.75	19.96	34.16	30.77	45.90	31.01	40.37	**33.33**	**34.52**
SPLIT^3	**54.49**	**68.47**	**57.25**	**65.26**	**74.91**	**86.92**	**12.88**	**17.94**	**13.68**	**16.73**	**38.27**	**52.26**	**32.68**	**46.04**	**32.57**	**40.80**	28.57	32.14

Table 2: **Anomaly detection performance.** We report the mean F1, AUROC, and AP over three runs. The best result for each metric is highlighted, and the second best is underlined.

Method	SMD			SWaT			WaDI			HAI			MSL		
	F1	AUROC	AP	F1	AUROC	AP	F1	AUROC	AP	F1	AUROC	AP	F1	AUROC	AP
ALoRa	51.87	84.89	48.02	74.66	83.55	73.81	40.44	80.31	41.19	65.70	81.88	58.60	21.19	56.89	14.06
SARAD	36.60	83.11	25.74	67.34	79.29	54.10	27.48	71.39	17.54	8.44	60.62	3.77	21.05	55.96	16.12
CST-GL	42.43	82.28	37.16	72.06	86.09	67.68	26.94	62.15	26.08	**71.85**	**96.14**	**72.62**	21.62	55.75	12.97
AERCA	45.03	82.18	40.25	73.07	81.78	69.65	35.58	74.99	34.31	38.55	82.83	23.15	21.61	56.83	13.57
GCAD	44.86	84.07	40.00	24.92	49.66	12.04	18.26	61.50	13.82	43.42	89.49	32.50	22.51	56.48	14.73
CrossAD	49.01	86.29	46.41	74.42	80.04	70.51	28.65	75.01	16.81	63.19	86.86	56.60	22.36	59.52	15.43
CATCH	44.30	87.47	38.94	36.69	51.37	31.00	31.12	73.87	17.19	65.63	86.57	60.84	**25.72**	**66.44**	**16.35**
M2N2	44.15	84.19	40.30	**75.67**	**87.67**	**75.79**	38.22	78.20	28.26	70.39	90.52	66.26	21.44	58.25	12.54
CANDI	**52.60**	**87.86**	**49.20**	73.02	81.37	66.38	38.40	78.04	36.84	41.11	88.24	22.79	21.53	55.89	12.80
SPLIT^3	48.74	85.74	42.91	75.41	86.62	75.35	**48.01**	**80.55**	**45.37**	70.37	89.78	69.40	23.19	60.08	13.82

Detection. SPLIT^3 maintains competitive detection performance across the five benchmarks. It achieves the highest F1, AUROC, and AP on WaDI, and the second highest result for all three metrics on SWaT. It also achieves the second highest AP on HAI and the second highest F1 and AUROC on MSL. Together, these results show strong localization and competitive detection performance using the same encoder and learned initialization.

5.2 Different Update Rate Preferences

We directly examine whether detection and localization favor different update rates within the same trained model while keeping the encoder and learned initialization unchanged. For this comparison, we denote the lower and higher test time update rates as $\mu_\ell = 0.001$ and $\mu_h = 0.01$, respectively. We compare a frozen state, a single shared state updated with either μ_ℓ or μ_h, and the separate states used by SPLIT^3. Table 3 shows a consistent tradeoff between detection and localization among the evaluated shared state settings. Increasing the shared update rate from μ_ℓ to μ_h decreases detection AP while increasing localization HR@100 on all three datasets. By separating the fast states, SPLIT^3 retains the detection performance of the μ_ℓ trajectory and the localization performance of the μ_h trajectory simultaneously. These results show distinct update rate preferences for detection and localization among the evaluated configurations within the same trained predictive model. The frozen comparison further shows that, over the full test stream, μ_ℓ improves detection over a frozen

Table 3: **Effect of the update rate and fast state separation.** AP measures detection; HR@100 measures localization.

Configuration	SMD		SWaT		WaDI	
	AP	HR@100	AP	HR@100	AP	HR@100
Frozen	38.19	47.89	70.44	4.76	44.50	26.13
Shared μ_ℓ	**42.91**	50.98	**75.35**	9.81	**45.37**	27.39
Shared μ_h	36.59	**54.49**	50.02	**12.88**	39.63	**32.68**
SPLIT^3	**42.91**	**54.49**	**75.35**	**12.88**	**45.37**	**32.68**

Table 4: **Metric changes when the TTT update rate increases from μ_ℓ to μ_h.** The same prediction error scales are used for both detection trajectories. All values denote changes in percentage points.

	Detection (ΔAP)			Localization (ΔHR@100)		
Dataset	Anomalous	Normal	Total	Labeled vars.	Remaining vars.	Total
SMD	-14.81	$+8.12$	$\mathbf{-6.69}$	-2.04	$+5.55$	$\mathbf{+3.51}$
SWaT	-41.97	$+14.38$	$\mathbf{-27.59}$	-16.01	$+19.08$	$\mathbf{+3.07}$
WaDI	-10.46	$+3.90$	$\mathbf{-6.57}$	-3.98	$+9.27$	$\mathbf{+5.29}$

fast state, while μ_h improves localization on all three datasets. We next examine why changing the update rate affects detection and localization differently. Additional robustness analyses across training update rates and rate selection are provided in Appendices C.2 and C.3.

5.3 How Adaptation Affects Detection and Localization

Changes in detection and localization. We first examine how increasing the test time update rate from μ_ℓ to μ_h changes detection and localization. For detection, we separate the change in AP into contributions associated with anomalous and normal timestamps. For localization, we separate the change in HR@100 into contributions associated with prediction error changes for labeled and remaining variables. The detailed decomposition procedure is provided in Appendix C.4.

Table 4 shows different patterns for the two tasks. For detection, the positive contribution associated with normal timestamps is outweighed by the negative contribution associated with anomalous timestamps, resulting in lower AP on all three datasets. For localization, the positive contribution associated with the remaining variables is larger than the negative contribution associated with the labeled variables, resulting in higher HR@100.

These results are consistent with the different ranking structures analyzed in Section 3.3. Detection compares anomaly scores across timestamps to distinguish anomalous observations from normal observations, making the separation between the two groups important. Localization instead compares variables within the same timestamp, so its performance depends more directly on how prediction errors change relative to one another across variables than on their absolute magnitudes.

Adaptation within anomaly events. We next examine how the update rate affects prediction error within anomaly events. TTT continues to update its fast state using observations encountered during anomaly events. To examine the effect of these updates, we replay each labeled anomaly event from the same fast state while varying the update rate used during the event. Detailed procedures are provided in Appendix C.5.

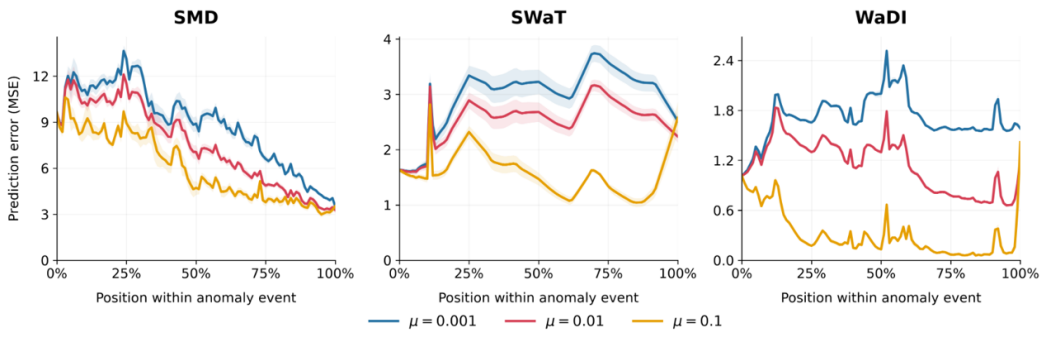


Figure 3: **Prediction error within labeled anomaly events under different update rates.** For each event, all replay conditions start from the same pre-event fast state, and the horizontal axis shows the relative position within the event.

Table 5: **Effect of anomaly event updates on localization.** Event updates use 0, μ_ℓ, or μ_h. We report HR@100.

Dataset	Rate before event	Update rate during event		
		0	μ_ℓ	μ_h
SMD	μ_ℓ	50.34	50.98	**53.60**
	μ_h	52.00	52.59	**54.49**
SWaT	μ_ℓ	9.83	9.81	**9.97**
	μ_h	12.22	12.24	**12.88**
WaDI	μ_ℓ	28.42	27.39	**28.76**
	μ_h	31.93	32.12	**32.68**

The aggregated event replay curves in Figure 3 show generally lower prediction errors on subsequent anomalous observations at larger update rates. Because detection uses prediction error as anomaly evidence, this reduction can weaken the distinction between anomalous and normal timestamps. This is consistent with the negative contribution from anomalous timestamps in Table 4.

Localization gains at the higher event update rate. Localization depends on the relative ordering of variables rather than on the overall prediction error magnitude. We therefore evaluate how updates performed during anomaly events affect localization. Using labeled event boundaries only for this analysis, we replay each event from a fixed fast state while changing the update rate used during the event. Detailed replay procedures and additional event analysis are provided in Appendix C.6.

Table 5 shows that, for both fast states used to initialize the event, μ_h gives the highest HR@100 on all three datasets. Thus, among the evaluated rates, localization consistently favors the higher update rate over suppressing updates during the anomaly event. Together with Table 4, this shows that a higher update rate can improve localization even when prediction errors decrease within anomaly events.

Taken together, the results show different responses to stronger test time adaptation across the two tasks. For detection, the event replay shows generally lower prediction errors at larger update rates, while the output decomposition shows that the negative contribution associated with anomalous timestamps outweighs the positive contribution associated with normal timestamps. For localization, μ_h yields the highest HR@100 across all evaluated datasets and pre-event states in the controlled replay. These analyses help explain the distinct update rate preferences observed for detection and localization. For completeness, additional event replay results for detection and a score based update suppression analysis are provided in Appendices C.6 and C.7.

6 Conclusion

In this work, we analyzed how TTT affects detection and localization in MTS anomaly diagnosis. We derived a recurrence for fast state differences under different update rates and related the resulting prediction error differences to the ranking structures of detection and localization. Using SPLIT^3, we compared a shared fast state with separate fast states while keeping the encoder and learned initialization unchanged. Across SMD, SWaT, and WaDI datasets, the evaluated update rate sweeps consistently favored lower rates for detection than for localization. Under the evaluated shared state settings, this difference creates a tradeoff between the two tasks. In contrast, separating the fast states allows the detection performance of the lower rate trajectory and the localization performance of the higher rate trajectory to be retained simultaneously. These results show that detection and localization need not share a single adaptation state, even when they share the same trained predictive model.

AI Assistance

Large language models were used to assist with writing and language polishing, including improving the clarity, organization, and presentation of the manuscript.

REFERENCES

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2485–2494, 2021.

Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. Wadi: A water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks*, pp. 25–28, 2017.

Jimmy Ba, Geoffrey E Hinton, Volodymyr Mnih, Joel Z Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. *Advances in neural information processing systems*, 29, 2016.

Zhihao Dai, Ligang He, Shuang-Hua Yang, and Matthew Leeke. Sarad: Spatial association-aware anomaly detection and diagnosis for multivariate time series. *Advances in Neural Information Processing Systems*, 37:48371–48410, 2024.

Peiliang Gong, Mohamed Ragab, Min Wu, Zhenghua Chen, Yongyi Su, Xiaoli Li, and Daoqiang Zhang. Augmented contrastive clustering with uncertainty-aware prototyping for time series test time adaptation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 390–401, 2025.

Jisu Han, Jaemin Na, and Wonjun Hwang. Ranked entropy minimization for continual test-time adaptation. *arXiv preprint arXiv:2505.16441*, 2025a.

Xiao Han, Saima Absar, Lu Zhang, and Shuhan Yuan. Root cause analysis of anomalies in multivariate time series through granger causal discovery. In *International Conference on Learning Representations*, volume 2025, pp. 45137–45153, 2025b.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018.

Dongmin Kim, Sunghyun Park, and Jaegul Choo. When model meets new normals: Test-time adaptation for unsupervised time-series anomaly detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 13113–13121, 2024.

HyunGi Kim, Jisoo Mok, Hyungyu Lee, Juhyeon Shin, and Sungroh Yoon. Candi: Curated test-time adaptation for multivariate time-series anomaly detection under distribution shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15018–15026, 2026.

Hyunmin Kong, Chunsoo Ahn, Seung Ho Jung, Seung Yeol Yoo, Donghwi Shin, and Jitae Shin. Probabilistic dynamic causal graphs for root cause analysis with distributional forecasting. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pp. 128–144, Cham, 2027. Springer Nature Switzerland. ISBN 978-3-032-37654-1.

Beibu Li, Qichao Shentu, Yang Shu, Hui Zhang, Ming Li, Ning Jin, Bin Yang, and Chenjuan Guo. Crossad: Time series anomaly detection with cross-scale associations and cross-window modeling. *Advances in Neural Information Processing Systems*, 38:137959–137985, 2026.

Zehao Liu, Mengzhou Gao, and Pengfei Jiao. Gcad: Anomaly detection in multivariate time series from the perspective of granger causality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19041–19049, 2025.

Jing Ma, Hanlin Li, and Xiang Xiang. Ptta: Purifying malicious samples for test-time model adaptation. In *Forty-second International Conference on Machine Learning*, 2025.

Aditya P Mathur and Nils Ole Tippenhauer. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pp. 31–36. IEEE, 2016.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pp. 9355–9366. PMLR, 2021.

Charalampos Shimillas, Kleanthis Malialis, Konstantinos Fokianos, and Marios Polycarpou. Low rank transformer for multivariate time series anomaly detection and localization. In *International Conference on Learning Representations*, volume 2026, pp. 59972–60005, 2026.

Hyeok-Ki Shin, Woomyo Lee, Jeong-Han Yun, and Byung-Gil Min. Two ics security datasets and anomaly detection contest on the hil-based augmented ics testbed. In *Proceedings of the 14th cyber security experimentation and test workshop*, pp. 36–40, 2021.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Zhiwen Tian, Ming Zhuo, Leyuan Liu, Junyi Chen, and Shijie Zhou. Anomaly detection using spatial and temporal information in multivariate time series. *Scientific Reports*, 13(1):4400, 2023.

Ke Alexander Wang, Jiaxin Shi, and Emily B Fox. Test-time regression: a unifying framework for designing sequence models with associative memory. *Journal of Machine Learning Research*, 27 (169):1–41, 2026.

Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. Catch: Channel-aware multivariate time series anomaly detection via frequency patching. In *International conference on learning representations*, volume 2025, pp. 17017–17045, 2025.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, volume 2025, pp. 29687–29707, 2025.

Huanyu Zhang, Yi-Fan Zhang, Jian Liang, Zhang Zhang, and Liang Wang. Spectral decomposition and adaptation for non-stationary time series anomaly detection. *Neurocomputing*, pp. 131978, 2025.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William Freeman, and Hao Tan. Test-time training done right. In *International Conference on Learning Representations*, volume 2026, pp. 157604–157638, 2026.

Yu Zheng, Huan Yee Koh, Ming Jin, Lianhua Chi, Khoa T Phan, Shirui Pan, Yi-Ping Phoebe Chen, and Wei Xiang. Correlation-aware spatial–temporal graph learning for multivariate time-series anomaly detection. *IEEE transactions on neural networks and learning systems*, 35(9):11802–11816, 2023.

A Mathematical Derivations and Proofs

This appendix provides the detailed derivations underlying the fast state analysis in Section 3 and the equivalence between the separate fast states of SPLIT^3 and independent TTT runs described in Section 4.2. The analysis considers the linear fast state predictor $\mathbf{W}_t$ used in this work with a fixed encoder ϕ_θ.

Throughout the analysis, compared trajectories process the same observation sequence and start from the same initial fast state unless stated otherwise. Because the key

$$\mathbf{k}_t = \phi_\theta(\mathbf{x}_{t-p:t-1}) \tag{14}$$

depends only on the observed history, trajectories that differ only in their fast state updates use the same $\mathbf{k}_t$ at each timestamp. They therefore also share the same normalization term

$$q_t = \left(\|\mathbf{k}_t\|_2^2 + \delta\right)^{-1}. \tag{15}$$

In particular, when the fast state update at time s is omitted, the observation $\mathbf{x}_s$ itself remains in the sequence and continues to appear in the history used to construct subsequent keys.

Fast state update. The update in Equation 1 is a normalized gradient step on the prediction loss

$$\ell_t(\mathbf{W}) = \frac{1}{2}\|\mathbf{x}_t - \mathbf{W}\mathbf{k}_t\|_2^2. \tag{16}$$

At the current fast state $\mathbf{W}_{t-1}$,

$$\nabla_{\mathbf{W}}\ell_t(\mathbf{W}_{t-1}) = -\mathbf{e}_t^- \mathbf{k}_t^\top. \tag{17}$$

Applying the normalized step size μq_t therefore gives

$$\begin{aligned}\mathbf{W}_t &= \mathbf{W}_{t-1} - \mu q_t \nabla_{\mathbf{W}}\ell_t(\mathbf{W}_{t-1}) \\ &= \mathbf{W}_{t-1} + \mu q_t \mathbf{e}_t^- \mathbf{k}_t^\top,\end{aligned} \tag{18}$$

which is exactly the fast state update used in the main text.

For later derivations, define the state transformation matrix associated with update rate μ at time j as

$$\mathbf{R}_j(\mu) = \mathbf{I} - \mu q_j \mathbf{k}_j \mathbf{k}_j^\top \in \mathbb{R}^{m\times m}. \tag{19}$$

This matrix describes how an existing difference between two fast states is transformed by the update at time j.

Because these matrices do not generally commute, their temporal order must be preserved. We therefore define the ordered product

$$\prod_{j=a}^{b} \mathbf{R}_j(\mu) := \mathbf{R}_a(\mu)\mathbf{R}_{a+1}(\mu)\cdots\mathbf{R}_b(\mu), \tag{20}$$

with an empty product defined as $\mathbf{I}$. When the update rate is clear from context, we write $\mathbf{R}_j$ for $\mathbf{R}_j(\mu)$. For the higher update rate μ_h, we similarly use $\mathbf{R}_j^h := \mathbf{R}_j(\mu_h)$.

A.1 Influence of a Single Fast State Update

This subsection derives Equation 2 in Section 3.2. Let $\mathbf{W}_j$ denote the reference fast state trajectory and $\bar{\mathbf{W}}_j$ a trajectory in which only the update at time s is omitted. Both trajectories observe the same sequence and follow the same update rule after time s, although their subsequent prediction errors and update values can differ because their fast states are no longer identical.

Define their state difference as

$$\boldsymbol{\Delta}_j = \mathbf{W}_j - \bar{\mathbf{W}}_j. \tag{21}$$

The two trajectories are identical before time s. Immediately after the update at s, their difference is

$$\boxed{\boldsymbol{\Delta}_s = \mu q_s \mathbf{e}_s^- \mathbf{k}_s^\top}. \tag{22}$$

For any $j > s$, the difference between their prediction errors before the update is

$$\begin{aligned}\mathbf{e}_j^- - \bar{\mathbf{e}}_j^- &= (\mathbf{x}_j - \mathbf{W}_{j-1}\mathbf{k}_j) - \left(\mathbf{x}_j - \bar{\mathbf{W}}_{j-1}\mathbf{k}_j\right) \\ &= -\boldsymbol{\Delta}_{j-1}\mathbf{k}_j. \end{aligned} \tag{23}$$

Subtracting their fast state updates gives

$$\begin{aligned}\boldsymbol{\Delta}_j &= \boldsymbol{\Delta}_{j-1} + \mu q_j \left(\mathbf{e}_j^- - \bar{\mathbf{e}}_j^-\right)\mathbf{k}_j^\top \\ &= \boldsymbol{\Delta}_{j-1}\left(\mathbf{I} - \mu q_j \mathbf{k}_j \mathbf{k}_j^\top\right). \end{aligned} \tag{24}$$

Using

$$\mathbf{R}_j = \mathbf{I} - \mu q_j \mathbf{k}_j \mathbf{k}_j^\top, \tag{25}$$

repeated application of Equation 24 yields

$$\boldsymbol{\Delta}_{t-1} = \mu q_s \mathbf{e}_s^- \mathbf{k}_s^\top \prod_{j=s+1}^{t-1} \mathbf{R}_j. \tag{26}$$

For a future time $t > s$, define the prediction difference as the prediction of the reference trajectory minus the prediction obtained after omitting the update at time s:

$$\mathbf{d}_{s\to t} := \hat{\mathbf{x}}_t - \bar{\hat{\mathbf{x}}}_t = \boldsymbol{\Delta}_{t-1}\mathbf{k}_t. \tag{27}$$

Substituting Equation 26 gives

$$\boxed{\mathbf{d}_{s\to t} = \mu q_s \mathbf{e}_s^- \mathbf{k}_s^\top \left[\prod_{j=s+1}^{t-1} \left(\mathbf{I} - \mu q_j \mathbf{k}_j \mathbf{k}_j^\top\right)\right] \mathbf{k}_t}. \tag{28}$$

Equation 28 shows that the future influence of an update is determined by three components: the state change introduced at time s, its transformation by subsequent fast state updates, and the readout of the resulting state difference through the future key $\mathbf{k}_t$. The magnitude of the original update alone is therefore insufficient to determine its future predictive influence.

A.2 Difference Between Update Rate Trajectories

We next consider two trajectories with update rates $\mu_h > \mu_\ell$, while keeping the encoder, observation sequence, and initial fast state identical. Define their fast state difference as

$$\mathbf{D}_t = \mathbf{W}_t^h - \mathbf{W}_t^\ell, \qquad \mathbf{D}_0 = \mathbf{0}. \tag{29}$$

Their prediction error difference before the update is

$$\begin{aligned}\mathbf{e}_t^{h,-} - \mathbf{e}_t^{\ell,-} &= \left(\mathbf{x}_t - \mathbf{W}_{t-1}^h \mathbf{k}_t\right) - \left(\mathbf{x}_t - \mathbf{W}_{t-1}^\ell \mathbf{k}_t\right) \\ &= -\mathbf{D}_{t-1}\mathbf{k}_t. \end{aligned} \tag{30}$$

Thus,

$$\boxed{\mathbf{e}_t^{h,-} - \mathbf{e}_t^{\ell,-} = -\mathbf{D}_{t-1}\mathbf{k}_t}. \tag{31}$$

Subtracting the two fast state updates gives

$$\begin{aligned}\mathbf{D}_t &= \mathbf{D}_{t-1} + q_t\left(\mu_h \mathbf{e}_t^{h,-} - \mu_\ell \mathbf{e}_t^{\ell,-}\right)\mathbf{k}_t^\top \\ &= \mathbf{D}_{t-1} + q_t\left[\mu_h\left(\mathbf{e}_t^{\ell,-} - \mathbf{D}_{t-1}\mathbf{k}_t\right) - \mu_\ell \mathbf{e}_t^{\ell,-}\right]\mathbf{k}_t^\top \\ &= \mathbf{D}_{t-1}\left(\mathbf{I} - \mu_h q_t \mathbf{k}_t \mathbf{k}_t^\top\right) + (\mu_h - \mu_\ell) q_t \mathbf{e}_t^{\ell,-}\mathbf{k}_t^\top. \end{aligned} \tag{32}$$

Therefore,

$$\boxed{\mathbf{D}_t = \mathbf{D}_{t-1}\left(\mathbf{I} - \mu_h q_t \mathbf{k}_t \mathbf{k}_t^\top\right) + (\mu_h - \mu_\ell) q_t \mathbf{e}_t^{\ell,-}\mathbf{k}_t^\top}. \tag{33}$$

This is the exact recurrence between the two trajectories. The first term transforms the state difference accumulated from previous timestamps, while the second introduces a new difference caused by the two update rates at the current timestamp.

Define

$$\mathbf{R}_t^h = \mathbf{I} - \mu_h q_t \mathbf{k}_t \mathbf{k}_t^\top. \tag{34}$$

Unrolling Equation 33 gives

$$\boxed{\mathbf{D}_t = \sum_{s=1}^{t} (\mu_h - \mu_\ell) q_s \mathbf{e}_s^{\ell,-} \mathbf{k}_s^\top \left[\prod_{j=s+1}^{t} \mathbf{R}_j^h \right]}. \tag{35}$$

Using Equation 31, the prediction error difference at time t becomes

$$\boxed{\mathbf{e}_t^{h,-} - \mathbf{e}_t^{\ell,-} = -\sum_{s=1}^{t-1} (\mu_h - \mu_\ell) q_s \mathbf{e}_s^{\ell,-} \mathbf{k}_s^\top \left[\prod_{j=s+1}^{t-1} \mathbf{R}_j^h \right] \mathbf{k}_t}. \tag{36}$$

Equations 35 and 36 show that changing the update rate does not simply rescale an isolated update. New state differences can be introduced throughout the sequence, previously accumulated differences are transformed by later updates, and the resulting state difference is read into future predictions. Although $\mu_h - \mu_\ell$ is a scalar, historical prediction errors generally differ across variables, so their future prediction errors need not change by the same amount or proportion.

Different update rates also do not imply different predictions at every timestamp. For example, the new difference introduced at time t vanishes when $\mathbf{e}_t^{\ell,-} = \mathbf{0}$. Moreover, two different fast states produce the same prediction at time t whenever

$$\mathbf{D}_{t-1} \mathbf{k}_t = \mathbf{0}. \tag{37}$$

The analysis therefore establishes that changing the update rate *can* alter subsequent predictions rather than requiring a difference at every timestamp.

A.3 Effects on Detection and Localization

The previous derivations show how changing the update rate can alter the fast state and prediction error trajectories. The resulting detection and localization performance cannot be determined from prediction error magnitude alone because the two tasks evaluate these errors through different ranking structures.

A.3.1 Detection

To isolate the effect of prediction errors, consider a fixed diagonal normalization matrix $\mathbf{C}$ with positive diagonal entries, and define the detection energy as

$$Q_t = (\mathbf{e}_t^-)^\top \mathbf{C} \mathbf{e}_t^-. \tag{38}$$

The detection score is a strictly increasing function of Q_t. Therefore, for fixed $\mathbf{C}$, ranking timestamps by Q_t is equivalent to ranking them by the detection score.

In the main benchmark, the normalization matrix is constructed from prediction error scales estimated on normal validation data as described in Appendix B. For controlled comparisons that isolate changes in prediction errors, the same normalization matrix is applied to both trajectories.

Let

$$\mathbf{u}_t = \mathbf{e}_t^{h,-} - \mathbf{e}_t^{\ell,-}. \tag{39}$$

Since

$$\mathbf{e}_t^{h,-} = \mathbf{e}_t^{\ell,-} + \mathbf{u}_t, \tag{40}$$

the change in detection energy is

$$\begin{aligned} \Delta Q_t &= (\mathbf{e}_t^{h,-})^\top \mathbf{C} \mathbf{e}_t^{h,-} - (\mathbf{e}_t^{\ell,-})^\top \mathbf{C} \mathbf{e}_t^{\ell,-} \\ &= 2(\mathbf{e}_t^{\ell,-})^\top \mathbf{C} \mathbf{u}_t + \mathbf{u}_t^\top \mathbf{C} \mathbf{u}_t. \end{aligned} \tag{41}$$

Thus,

$$\boxed{\Delta Q_t = 2(\mathbf{e}_t^{\ell,-})^\top \mathbf{C}\mathbf{u}_t + \mathbf{u}_t^\top \mathbf{C}\mathbf{u}_t}. \tag{42}$$

Detection AP depends on how anomalous and normal timestamps are ranked relative to one another. For an anomalous timestamp a and a normal timestamp n, define

$$\Delta s_t = s_t^h - s_t^\ell. \tag{43}$$

The change in their score margin is

$$\boxed{\Delta(s_a - s_n) = \Delta s_a - \Delta s_n}. \tag{44}$$

Consequently, a decrease or increase in scores at anomalous timestamps alone is insufficient to determine the change in AP. Scores at normal timestamps can change simultaneously, and AP depends on their relative ordering. This motivates the output decomposition in Appendix C.4, which separately examines score changes at anomalous and normal timestamps.

A.3.2 LOCALIZATION

Localization evaluates relative prediction errors across variables within the same timestamp. At an anomalous timestamp t, consider a variable i in the localization ground truth set and a remaining variable j outside that set. A remaining variable is not assumed to be physically normal or unaffected by the anomaly; the distinction is defined only by the available variable level annotations.

For rate $r \in \{\ell, h\}$, define their squared prediction error margin as

$$M_{ij,t}^r = (e_{i,t}^{r,-})^2 - (e_{j,t}^{r,-})^2. \tag{45}$$

A positive margin indicates that variable i receives a larger squared prediction error than variable j and therefore ranks above j under the localization score.

Define the signed reduction in squared prediction error for variable i as

$$A_{i,t} = (e_{i,t}^{\ell,-})^2 - (e_{i,t}^{h,-})^2. \tag{46}$$

A positive $A_{i,t}$ indicates that the squared prediction error of variable i is smaller under the higher update rate.

The change in the pairwise margin is

$$\begin{aligned} M_{ij,t}^h - M_{ij,t}^\ell &= \left[(e_{i,t}^{h,-})^2 - (e_{i,t}^{\ell,-})^2\right] - \left[(e_{j,t}^{h,-})^2 - (e_{j,t}^{\ell,-})^2\right] \\ &= A_{j,t} - A_{i,t}. \end{aligned} \tag{47}$$

Therefore,

$$\boxed{M_{ij,t}^h - M_{ij,t}^\ell = A_{j,t} - A_{i,t}}. \tag{48}$$

Equation 48 shows that localization depends on relative changes in prediction errors across variables. For example, even if the squared error of a labeled variable decreases under the higher update rate, its relative rank can improve when the squared error of a competing variable decreases by an even larger amount.

A larger pairwise margin does not by itself imply a change in HR@100. HR@100 depends on the complete variable ordering and, in particular, on which variables cross the top K boundary. The pairwise relation therefore explains how relative error changes can alter localization rankings, but it does not determine the direction of the resulting HR@100 change.

Hypothetical immediate re-evaluation. The preceding analysis concerns differences accumulated through the evolving fast state. To distinguish these effects from directly fitting the current observation, consider hypothetically evaluating the same observation again immediately after its fast state update. Using Equation 1,

$$\begin{aligned} \mathbf{e}_t^+ &= \mathbf{x}_t - \mathbf{W}_t \mathbf{k}_t \\ &= \mathbf{x}_t - \left(\mathbf{W}_{t-1} + \mu q_t \mathbf{e}_t^- \mathbf{k}_t^\top\right) \mathbf{k}_t \\ &= (1 - \alpha_t)\, \mathbf{e}_t^-, \end{aligned} \tag{49}$$

where

$$\alpha_t = \mu q_t \|\mathbf{k}_t\|_2^2. \tag{50}$$

Thus,

$$\boxed{\mathbf{e}_t^+ = (1-\alpha_t)\mathbf{e}_t^-} \tag{51}$$

and

$$\|\mathbf{e}_t^+\|_2^2 = (1-\alpha_t)^2 \|\mathbf{e}_t^-\|_2^2. \tag{52}$$

The total squared prediction error strictly decreases when

$$0 < \alpha_t < 2, \qquad \mathbf{e}_t^- \neq \mathbf{0}. \tag{53}$$

For each variable,

$$(e_{i,t}^+)^2 = (1-\alpha_t)^2 (e_{i,t}^-)^2. \tag{54}$$

When $\alpha_t \neq 1$, all variable scores are therefore multiplied by the same positive scalar, so their ordering within the timestamp is preserved. When $\alpha_t = 1$, all re-evaluated squared errors become zero and the ordering collapses to a tie.

This hypothetical immediate re-evaluation cannot by itself explain an improvement in within timestamp localization metrics such as HR or NDCG. More importantly, the actual inference procedure follows *predict* $\rightarrow$ *score* $\rightarrow$ *update*. The current update therefore does not alter the score already assigned to the current observation; its effects appear through subsequent fast states and predictions.

A.4 Equivalence to Independent TTT Runs

We finally show that each task specific fast state in SPLIT^3 follows the same trajectory as an independent TTT run with the corresponding update rate. For task $\tau \in \{D, L\}$, let $\mathbf{W}_t^\tau$ denote its fast state in SPLIT^3. Let $\mathbf{W}_t^{\mathrm{TTT}}(\mu_\tau)$ denote the fast state obtained by independently running TTT with the same encoder, observation sequence, initial state, update rule, and update rate μ_τ.

At initialization,

$$\mathbf{W}_0^\tau = \mathbf{W}_0^{\mathrm{TTT}}(\mu_\tau) = \mathbf{W}_0. \tag{55}$$

Assume at time $t-1$ that

$$\mathbf{W}_{t-1}^\tau = \mathbf{W}_{t-1}^{\mathrm{TTT}}(\mu_\tau). \tag{56}$$

Because the two runs process the same observation history using the same encoder, they use the same $\mathbf{k}_t$ and q_t. Their prediction errors before the update are therefore identical:

$$\mathbf{e}_t^{\tau,-} = \mathbf{e}_t^{\mathrm{TTT},-}(\mu_\tau). \tag{57}$$

Applying the same update rate μ_τ and the same recurrence gives

$$\begin{aligned} \mathbf{W}_t^\tau &= \mathbf{W}_{t-1}^\tau + \mu_\tau q_t \mathbf{e}_t^{\tau,-} \mathbf{k}_t^\top \\ &= \mathbf{W}_{t-1}^{\mathrm{TTT}}(\mu_\tau) + \mu_\tau q_t \mathbf{e}_t^{\mathrm{TTT},-}(\mu_\tau) \mathbf{k}_t^\top \\ &= \mathbf{W}_t^{\mathrm{TTT}}(\mu_\tau). \end{aligned} \tag{58}$$

By induction,

$$\boxed{\mathbf{W}_t^\tau = \mathbf{W}_t^{\mathrm{TTT}}(\mu_\tau), \qquad \mathbf{e}_t^{\tau,-} = \mathbf{e}_t^{\mathrm{TTT},-}(\mu_\tau), \qquad \forall t}. \tag{59}$$

Each task specific fast state in SPLIT^3 therefore follows the same fast state and prediction error trajectory as an independent TTT run with the corresponding update rate. Separating the states does not construct an intermediate or averaged trajectory.

When $\mu_D = \mu_L$, the two fast states have the same initialization and follow the same recurrence. Consequently,

$$\boxed{\mathbf{W}_t^D = \mathbf{W}_t^L, \qquad \forall t}. \tag{60}$$

In this case, the two fast state trajectories are identical and reduce to a single shared fast state, although detection and localization can still apply their respective scoring functions to the resulting prediction errors.

This equivalence concerns the fast state trajectory for a fixed update rate and does not imply that any particular update rate is optimal.

Scope. The derivations in this appendix characterize the linear fast state and normalized update rule used in this work under a fixed encoder and a shared observation sequence. They establish that individual updates can influence future predictions, that changing the update rate can alter the accumulated fast state and prediction error trajectories, and that detection and localization evaluate the resulting errors through different ranking structures.

They do not establish a universal monotonic relationship between the update rate and AP or HR, nor do they determine an optimal update rate. The empirical directions of the metric changes are therefore evaluated separately in Figure 1 and Section 5.3, with additional controlled analyses provided in Appendix C.

B Experimental Details

B.1 Datasets and Preprocessing

We evaluate anomaly detection on SMD (Su et al., 2019), SWaT (Mathur & Tippenhauer, 2016), WaDI (Ahmed et al., 2017), HAI (Shin et al., 2021), and MSL (Hundman et al., 2018), and localization on SMD, SWaT, and WaDI.

For SMD, SWaT, HAI, and MSL, we follow the data partition and preprocessing protocol of Shimillas et al. (2026). Each variable is standardized using the mean and standard deviation computed from the normal training data, and the same statistics are applied to the validation and test sequences. Following Shimillas et al. (2026), SWaT and HAI are downsampled to one minute resolution by averaging observations over consecutive intervals of 60 seconds.

WaDI. WaDI contains 14 days of normal operation followed by two days containing 15 attack scenarios. After excluding the time, row index, and label fields, the raw data contain 127 sensor and actuator variables. We remove four variables with no observations throughout the sequence and use the remaining 123 variables as model inputs. No additional filtering based on variable names or constant values is applied. WaDI is evaluated at 10 second resolution, and each variable is standardized using statistics estimated from the normal training data. The same train and validation partitioning rule used for the other datasets is applied to the normal WaDI sequence.

Timestamp level anomaly labels are constructed from the provided attack intervals. For localization, the affected variables specified in the attack metadata are used as variable level labels during the corresponding attack intervals. These annotations indicate variables associated with each attack and are used for localization evaluation; they are not treated as an exhaustive set of physical root causes.

Use of validation data and labels. Normal validation data are used for checkpoint selection, detection score normalization, and, where explicitly stated, score threshold selection. Ground truth labels are used for benchmark evaluation, for the development analysis of test time update rates described in Appendix B.4, and for the controlled analyses in Appendix C. They are not used in the training objective or during standard online prediction, scoring, and fast state updates.

B.2 Model Architecture and Training

Shared encoder and fast state. We use a history length of $p = 10$ for all datasets. For each variable, its previous ten observations are processed independently by a per variable MLP whose parameters are shared across variables:

$$\mathrm{Linear}(10, 32) \to \mathrm{GELU} \to \mathrm{Linear}(32, 4). \tag{61}$$

The resulting four dimensional representations of all d variables are concatenated, and a constant component of one is appended to form the key

$$\mathbf{k}_t \in \mathbb{R}^m, \qquad m = 4d + 1. \tag{62}$$

For SMD, SWaT, WaDI, HAI, and MSL, the variable dimensions are

$$d = 38,\ 51,\ 123,\ 78,\ 55, \tag{63}$$

respectively, giving key dimensions

$$m = 153,\ 205,\ 493,\ 313,\ 221. \tag{64}$$

The key is constructed only from past observations $\mathbf{x}_{t-p:t-1}$ and does not contain the current observation $\mathbf{x}_t$. Unavailable history at the beginning of a sequence is padded with zeros in the standardized coordinate system.

The initial fast state

$$\mathbf{W}_0 \in \mathbb{R}^{d\times m} \tag{65}$$

is a learnable parameter initialized to zero. The stabilization constant in the fast state update is fixed to $\delta = 1$ for all experiments.

Training objective. The encoder and initial fast state are jointly optimized on normal training sequences while the fast state updates are explicitly executed during training. Let $\mathcal{B}$ denote the set of stream and timestamp indices contained in a training chunk. We minimize the prediction loss computed before each fast state update:

$$\mathcal{L}_{\text{train}} = \frac{1}{|\mathcal{B}|d} \sum_{(b,t)\in\mathcal{B}} \|\mathbf{x}_{b,t} - \mathbf{W}_{b,t-1}\mathbf{k}_{b,t}\|_2^2 . \tag{66}$$

The stream index b is included because multiple contiguous substreams are processed in parallel during training. This is the batched form of the prediction objective in Equation 7.

Training uses only normal sequences and does not use anomaly labels or a localization objective. The fast state update rate during training is $\mu_{\text{train}} = 0.001$ in the main configuration. This update rate is distinct from the optimizer learning rate for the learnable parameters. The encoder and $\mathbf{W}_0$ are optimized using Adam with learning rate 2×10^{-3} and $(\beta_1, \beta_2) = (0.9, 0.999)$.

For a training sequence of length N_{tr}, we process

$$B = \max\left(16, \left\lceil \frac{N_{\text{tr}} - p}{1000} \right\rceil\right) \tag{67}$$

contiguous substreams in parallel. Each substream is initialized from $\mathbf{W}_0$ at the beginning of an epoch.

We use truncated backpropagation through time with a maximum unrolling length of 128 timestamps. Fast state values are carried across successive chunks, while the computation graph is detached at chunk boundaries. The gradient norm is clipped to 1.0 before each optimizer step.

We train our predictive model for 15 epochs. We retain the checkpoint with the lowest prediction MSE on the normal validation sequence after excluding the first p timestamps. No separate static predictor pretraining is used. All experiments are repeated with random seeds $\{0, 1, 2\}$. The model is implemented in PyTorch, and all experiments are conducted on a single NVIDIA A100 80GB GPU.

B.3 Test Time Inference and Score Normalization

Validation and test sequences are initialized independently from the learned initial state $\mathbf{W}_0$. The final fast state reached during training is not transferred to evaluation, and no additional normal sequence is used to warm up the test stream.

At test time, the learned encoder and initialization are fixed. The detection and localization fast states are initialized as

$$\mathbf{W}_0^D = \mathbf{W}_0^L = \mathbf{W}_0. \tag{68}$$

Only the two fast states evolve online during test time.

For the standard SPLIT^3 inference used in the main benchmark, each timestamp follows

$$\text{predict} \rightarrow \text{score} \rightarrow \text{update}. \tag{69}$$

The current observation is therefore incorporated into the fast state only after its score has been computed. Standard inference does not use anomaly labels, event boundaries, or an anomaly dependent update rule. The event replay and score based update suppression experiments in Appendix C are separate controlled analyses.

Detection score normalization. Let $\mathcal{T}_{\text{val}}$ denote the valid timestamps in the normal validation sequence after excluding the first p timestamps. For variable i, we estimate the prediction error scale as

$$\sigma_i = \max\left(\text{std}_{t\in\mathcal{T}_{\text{val}}}[e_{i,t}^{D,-}], 0.05\right). \tag{70}$$

The detection normalization matrix is

$$\mathbf{C}_D = \text{diag}\left(\sigma_1^{-2}, \ldots, \sigma_d^{-2}\right). \tag{71}$$

The detection and localization scores are

$$s_t^D = \log\left(10^{-8} + \sum_{i=1}^{d} \frac{(e_{i,t}^{D,-})^2}{\sigma_i^2}\right), \qquad s_{i,t}^L = (e_{i,t}^{L,-})^2. \tag{72}$$

No additional variable specific normalization is applied to the localization scores.

For the standard benchmark evaluation, the detection normalization statistics are estimated from normal validation prediction errors corresponding to the evaluated test time update rate. In the output decomposition analysis of Appendix C.4, the normalization statistics estimated at $\mu = 0.001$ are instead fixed and applied to both trajectories. This controlled choice isolates changes in the prediction error trajectories from changes in the normalization statistics.

B.4 Update Rate Selection

We distinguish the fast state update rate used during training from those used during test time. For the training rate sensitivity analysis, models are trained with

$$\mu_{\text{train}} \in \{0.001,\ 0.01,\ 0.1\}. \tag{73}$$

Each trained model is evaluated over the test time update rate grid

$$\mathcal{U} = \{0,\ 0.001,\ 0.003,\ 0.01,\ 0.03,\ 0.1,\ 0.3\}. \tag{74}$$

Here, $\mu = 0$ denotes a frozen test time setting in which the fast state is not updated while the encoder and learned initialization remain unchanged.

Different test time update rates can be applied to the same learned encoder and initial fast state. The model therefore does not need to be retrained for different (μ_D, μ_L) combinations.

During development, we examined how detection AP and localization HR@100 change across the test time update rate grid. Based on these task responses, the configuration used in the main experiments is fixed across all datasets and random seeds as

$$\boxed{\mu_{\text{train}} = 0.001, \qquad \mu_D = 0.001, \qquad \mu_L = 0.01.} \tag{75}$$

Appendix C.3 further examines task specific rate selection while excluding the evaluation dataset from the selection calculation.

B.5 Evaluation Metrics

Anomaly detection. We evaluate anomaly detection using F1, AUROC, and AP. AUROC and AP are computed directly from the continuous anomaly scores. F1 is reported using the threshold that maximizes F1 on the test scores, and the same evaluation procedure is applied to all methods. No point adjustment is applied. All anomaly scores are aligned with their corresponding observation timestamps before evaluation.

The threshold used to report F1 is an evaluation threshold and is distinct from the normal validation score threshold used in the score based update suppression analysis in Appendix C.7. The latter determines whether a detection fast state update is performed and does not use test labels.

Localization. We evaluate localization using HR, NDCG, and IPS for $P \in \{100, 150\}$. Let G_t denote the set of anomaly labeled variables at an anomalous timestamp t. The number of candidate variables is

$$K_t^{(P)} = \min\left(d, \left\lceil |G_t| \frac{P}{100} \right\rceil\right). \tag{76}$$

Let $\Gamma_t^{(P)}$ denote the top $K_t^{(P)}$ variables according to the localization score. Then

$$\mathrm{HR}_t@P = \frac{|G_t \cap \Gamma_t^{(P)}|}{|G_t|}. \tag{77}$$

For $P = 100$,

$$K_t^{(100)} = |G_t|. \tag{78}$$

Thus, HR@100 evaluates whether the variables in the ground truth set are recovered within a candidate set of the same size.

NDCG additionally accounts for the positions of labeled variables within the ranking. Let $r_j \in \{0, 1\}$ indicate whether the variable at rank j belongs to G_t. Then

$$\mathrm{DCG}_t@P = \sum_{j=1}^{K_t^{(P)}} \frac{r_j}{\log_2(j+1)} \tag{79}$$

and

$$\mathrm{NDCG}_t@P = \frac{\mathrm{DCG}_t@P}{\sum_{j=1}^{|G_t|} \frac{1}{\log_2(j+1)}}. \tag{80}$$

HR and NDCG are averaged over anomalous timestamps for which variable level annotations are available.

IPS evaluates localization over anomaly segments rather than individual timestamps. For an anomaly segment $\mathcal{S}$, the score of variable i is

$$s_{i,\mathcal{S}} = \max_{t \in \mathcal{S}} s_{i,t}^L. \tag{81}$$

Variables are ranked according to these segment level scores, and the same $P \in \{100, 150\}$ candidate rule is applied. IPS is obtained by averaging the recovery rate of anomaly labeled variables over anomaly segments. Thus, HR and NDCG evaluate variable rankings at individual timestamps, whereas IPS evaluates rankings after aggregating localization evidence over an anomaly segment.

B.6 Baselines and Evaluation Protocol

For localization, we compare against SARAD (Dai et al., 2024), CST-GL (Zheng et al., 2023), AERCA (Han et al., 2025b), ALoRa (Shimillas et al., 2026), M2N2 (Kim et al., 2024), and CANDI (Kim et al., 2026). For detection, we compare against ALoRa (Shimillas et al., 2026), SARAD (Dai et al., 2024), CST-GL (Zheng et al., 2023), AERCA (Han et al., 2025b), GCAD (Liu et al., 2025), CrossAD (Li et al., 2026), CATCH (Wu et al., 2025), M2N2 (Kim et al., 2024), and CANDI (Kim et al., 2026).

Whenever available, we use official or publicly available implementations and follow the settings of the corresponding methods. For methods that support both detection and localization, localization results are obtained using the localization procedure defined by the original method.

For window based methods, scores are aligned with the observation timeline using the causal information available to each method. Scores are generally assigned to the rightmost observation of the input window. Whether a method evaluates reconstruction or prediction error over the full window or only at its final timestamp follows the scoring definition of the corresponding method. All methods are evaluated using the metric implementations described above.

C Additional Experimental Results and Analyses

This appendix provides additional results and controlled analyses supporting the experiments in Section 5. We first report full benchmark results with standard deviations and examine robustness to the training update rate and test time rate selection. We then provide the output decomposition procedure, the anomaly event prediction error analysis, a controlled event replay for detection and localization, and detection score based update suppression.

Unless otherwise stated, the model architecture, training procedure, preprocessing, scoring functions, and evaluation protocol follow Appendix B. Analyses that directly compare two test time update rates use

$$\mu_\ell = 0.001, \qquad \mu_h = 0.01. \tag{82}$$

C.1 Full Benchmark Results

Tables 6 and 7 provide the full results corresponding to Tables 1 and 2 in the main text. While the main text reports the mean for readability, the tables below report the mean and standard deviation over three runs.

C.2 Sensitivity to the Training Update Rate

The main experiments use models trained with $\mu_{\text{train}} = 0.001$. We examine whether the different responses of detection and localization to the test time update rate depend on this training setting.

We separately train models with

$$\mu_{\text{train}} \in \{0.001,\ 0.01,\ 0.1\}. \tag{83}$$

For each model, we increase the test time update rate from μ_ℓ to μ_h and measure the resulting changes in detection AP and localization HR@100.

Table 8 shows that increasing the test time update rate from μ_ℓ to μ_h decreases detection AP and increases localization HR@100 under every training condition. The different responses of the two tasks are therefore not specific to training with $\mu_{\text{train}} = 0.001$.

Detection AP in this analysis follows the standard evaluation protocol, in which normalization statistics are estimated separately for each evaluated test time update rate. The resulting ΔAP values therefore need not match the totals in Table 4, where the normalization statistics are fixed across trajectories to isolate changes in prediction errors.

C.3 Leave One Dataset Out Selection of Test Time Update Rates

The main experiments use a single update-rate pair shared across datasets, selected from the rate-sensitivity analysis during development. We further evaluate whether allowing detection and localization to use separately selected update rates remains beneficial when the evaluation dataset is excluded from rate selection.

We fix $\mu_{\text{train}} = 0.001$ and perform leave one dataset out selection over SMD, SWaT, and WaDI. For each held out dataset, the remaining two datasets are used to select the test time update rates. We choose the rate with the highest mean detection AP as μ_D and the rate with the highest mean

Table 6: **Full anomaly localization results.** We report mean ± standard deviation over three runs for HR@P, NDCG@P, and IPS@P. The best result for each metric is highlighted, and the second best is underlined.

Method	SMD						SWaT						WaDI					
	HR@P		NDCG@P		IPS@P		HR@P		NDCG@P		IPS@P		HR@P		NDCG@P		IPS@P	
	100	150	100	150	100	150	100	150	100	150	100	150	100	150	100	150	100	150
SARAD	52.78 ±0.37	65.91 ±0.21	56.28 ±0.48	63.75 ±0.38	58.55 ±0.78	73.08 ±0.68	3.08 ±1.71	14.77 ±18.50	3.13 ±1.67	10.31 ±11.81	7.82 ±0.71	11.52 ±3.77	19.45 ±4.75	23.51 ±3.78	19.47 ±4.68	21.98 ±4.06	15.87 ±4.51	26.19 ±0.00
CST-GL	44.30 ±0.45	57.13 ±0.92	47.66 ±0.52	54.99 ±0.75	57.18 ±1.54	70.35 ±1.39	1.69 ±0.34	1.87 ±0.33	1.76 ±0.22	1.87 ±0.21	7.61 ±1.43	8.02 ±1.23	22.10 ±0.47	26.03 ±4.56	22.93 ±0.59	25.36 ±3.17	28.17 ±1.37	33.33 ±3.57
AERCA	31.58 ±1.00	45.80 ±0.93	31.40 ±1.07	39.50 ±1.03	53.62 ±1.83	67.71 ±1.40	1.85 ±0.32	2.90 ±0.26	1.81 ±0.39	2.43 ±0.35	5.97 ±2.02	9.67 ±1.78	21.31 ±2.62	27.02 ±3.58	21.22 ±2.68	24.79 ±3.14	19.05 ±5.46	23.81 ±4.12
ALoRa	38.31 ±2.41	51.83 ±3.29	40.38 ±2.27	48.04 ±2.66	36.98 ±3.82	51.16 ±6.13	2.30 ±1.17	3.46 ±0.52	2.59 ±1.56	3.05 ±1.32	10.49 ±4.90	11.73 ±4.82	7.71 ±1.90	9.24 ±1.86	8.33 ±2.17	9.27 ±2.12	5.16 ±1.82	8.73 ±1.82
M2N2	50.81 ±0.74	64.20 ±0.59	53.77 ±0.74	61.40 ±0.60	70.26 ±0.37	84.93 ±0.24	7.49 ±0.59	11.97 ±0.93	7.71 ±0.56	10.45 ±0.71	22.63 ±3.97	37.45 ±0.71	30.03 ±2.38	42.88 ±3.86	30.20 ±2.36	38.13 ±3.30	30.95 ±0.00	30.95 ±0.00
CANDI	48.67 ±1.12	61.45 ±1.36	51.69 ±0.91	58.95 ±1.05	72.84 ±0.64	85.31 ±0.15	4.94 ±0.67	9.37 ±0.84	5.04 ±0.51	7.75 ±0.45	19.96 ±3.77	34.16 ±1.55	30.77 ±2.83	45.90 ±0.84	31.01 ±2.93	40.37 ±1.73	**33.33** ±2.06	**34.52** ±3.15
SPLIT^5	**54.49** ±0.53	**68.47** ±0.31	**57.25** ±0.63	**65.26** ±0.50	**74.91** ±0.56	**86.92** ±0.17	**12.88** ±0.10	**17.94** ±0.55	**13.68** ±0.09	**16.73** ±0.40	**38.27** ±2.14	**52.26** ±3.11	**32.68** ±0.21	**46.04** ±1.18	**32.57** ±0.36	**40.80** ±0.93	28.57 ±0.00	32.14 ±0.00

Table 7: **Full anomaly detection results.** We report mean ± standard deviation over three runs for F1, AUROC, and AP. The best result for each metric is highlighted, and the second best is underlined.

	SMD			SWaT			WaDI			HAI			MSL		
Method	F1	AUROC	AP	F1	AUROC	AP	F1	AUROC	AP	F1	AUROC	AP	F1	AUROC	AP
ALoRa	51.87 ±0.01	84.89 ±0.09	48.02 ±0.11	74.66 ±0.06	83.55 ±0.17	73.81 ±0.34	40.44 ±0.32	80.31 ±0.08	41.19 ±1.06	65.70 ±0.90	81.88 ±0.22	58.60 ±0.33	21.19 ±0.33	56.89 ±0.25	14.06 ±0.13
SARAD	36.60 ±0.54	83.11 ±0.23	25.74 ±0.42	67.34 ±0.18	79.29 ±0.50	54.10 ±1.20	27.48 ±1.27	71.39 ±1.18	17.54 ±1.02	8.44 ±0.06	60.62 ±0.18	3.77 ±0.09	21.05 ±0.60	55.96 ±1.44	16.12 ±0.46
CST-GL	42.43 ±1.29	82.28 ±0.18	37.16 ±1.28	72.06 ±2.67	86.09 ±0.86	67.68 ±7.71	26.94 ±0.75	62.15 ±2.88	26.08 ±0.48	**71.85** ±1.42	**96.14** ±0.20	**72.62** ±0.55	21.62 ±0.85	55.75 ±0.89	12.97 ±0.43
AERCA	45.03 ±0.26	82.18 ±1.44	40.25 ±0.49	73.07 ±0.73	81.78 ±1.29	69.65 ±2.90	35.58 ±2.29	74.99 ±1.89	34.31 ±0.67	38.55 ±1.53	82.83 ±1.52	23.15 ±1.63	21.61 ±0.43	56.83 ±1.65	13.57 ±0.08
GCAD	44.86 ±0.37	84.07 ±0.21	40.00 ±0.54	24.92 ±1.12	49.66 ±2.67	12.04 ±0.68	18.26 ±0.60	61.50 ±0.76	13.82 ±0.29	43.42 ±1.22	89.49 ±1.31	32.50 ±1.41	22.51 ±1.29	56.48 ±1.75	14.73 ±0.56
CrossAD	49.01 ±0.09	86.29 ±0.21	46.41 ±0.18	74.42 ±0.05	80.04 ±0.03	70.51 ±0.03	28.65 ±0.54	75.01 ±0.08	16.81 ±0.10	63.19 ±0.57	86.86 ±0.02	56.60 ±0.21	22.36 ±0.05	59.52 ±0.10	15.43 ±0.04
CATCH	44.30 ±0.46	87.47 ±0.07	38.94 ±0.64	36.69 ±1.04	51.37 ±1.11	31.00 ±0.18	31.12 ±0.83	73.87 ±0.53	17.19 ±0.42	65.63 ±0.78	86.57 ±0.77	60.84 ±0.64	**25.72** ±0.04	**66.44** ±0.23	**16.35** ±0.20
M2N2	44.15 ±0.12	84.19 ±0.27	40.30 ±0.33	**75.67** ±0.04	**87.67** ±0.14	**75.79** ±0.11	38.22 ±1.09	78.20 ±0.46	28.26 ±0.69	70.39 ±1.29	90.52 ±0.42	66.26 ±0.13	21.44 ±0.08	58.25 ±0.22	12.54 ±0.10
CANDI	**52.60** ±0.46	**87.86** ±0.05	**49.20** ±0.65	73.02 ±0.18	81.37 ±0.10	66.38 ±0.21	38.40 ±0.46	78.04 ±0.13	36.84 ±0.55	41.11 ±0.16	88.24 ±0.21	22.79 ±0.11	21.53 ±0.06	55.89 ±0.11	12.80 ±0.05
SPLIT^3	48.74 ±0.41	85.74 ±0.25	42.91 ±0.60	75.41 ±0.22	86.62 ±0.32	75.35 ±0.18	**48.01** ±1.29	**80.55** ±0.13	**45.37** ±0.43	70.37 ±1.18	89.78 ±0.74	69.40 ±1.11	23.19 ±0.17	60.08 ±0.91	13.82 ±0.33

Table 8: **Sensitivity to the training update rate.** ΔAP and ΔHR@100 denote percentage point changes when the test time update rate is increased from μ_ℓ to μ_h.

Dataset	μ_{train}	ΔAP	ΔHR@100
SMD	.001	-6.32	$+3.51$
	.01	-6.25	$+3.73$
	.1	-4.87	$+3.76$
SWaT	.001	-25.33	$+3.07$
	.01	-25.02	$+3.22$
	.1	-27.53	$+3.64$
WaDI	.001	-5.75	$+5.29$
	.01	-5.81	$+4.57$
	.1	-4.16	$+4.49$

localization HR@100 as μ_L. The two selection datasets receive equal weight, and no constraint requires μ_D and μ_L to differ.

As shown in Table 9, holding out either SMD or SWaT selects $(\mu_D, \mu_L) = (0.001, 0.01)$. When WaDI is held out, $\mu_D = 0.001$ is again selected for detection, while the selected localization rate changes to $\mu_L = 0.03$. Thus, the selection procedure does not always recover the rate pair used in the main experiments, and different rates are not enforced by design.

We next apply the selected rates to the corresponding held out dataset. As a reference, we compare SPLIT^3 with a shared μ_D configuration that uses the selected detection rate for both tasks.

Table 10 shows that detection AP is unchanged because both configurations use the same selected detection rate. Allowing localization to use the separately selected μ_L improves HR@100 by $+3.51$, $+3.07$, and $+3.80$ percentage points on SMD, SWaT, and WaDI, respectively.

These results show that the benefit of allowing localization to follow its separately selected update rate is retained when the evaluation dataset is excluded from the rate selection calculation. The WaDI result further shows that this benefit does not depend on always selecting the fixed pair $(0.001, 0.01)$ used in the main experiments.

C.4 Decomposition of Metric Changes

Table 4 separates the metric change from μ_ℓ to μ_h into two components. Here we describe how these components are computed.

This analysis operates on model outputs that have already been produced. Ground truth labels are used only to divide those outputs into groups and are not used for prediction or fast state updates. The decomposition is therefore an output level analysis rather than a causal intervention on particular timestamps or variables.

Detection. For detection, timestamps are divided into anomalous and normal groups. We independently replace the scores of the two groups between the low rate and high rate trajectories.

Table 9: **Update rates selected by leave one dataset out analysis.** AP and HR@100 from the held out dataset are not used for rate selection.

Held out dataset	Datasets used for selection	μ_D	μ_L
SMD	SWaT + WaDI	.001	.01
SWaT	SMD + WaDI	.001	.01
WaDI	SMD + SWaT	.001	.03

Table 10: **Evaluation with leave one dataset out rate selection.** Shared μ_D uses the selected detection rate for both tasks, whereas SPLIT^3 uses the separately selected μ_D and μ_L.

	Shared μ_D		SPLIT^3		ΔHR
Dataset	AP	HR@100	AP	HR@100	(%p)
SMD	42.91	50.98	42.91	54.49	+3.51
SWaT	75.35	9.81	75.35	12.88	+3.07
WaDI	45.37	27.39	45.37	31.19	+3.80

$F_{\ell\ell}$ uses low rate scores for both groups. $F_{h\ell}$ uses high rate scores at anomalous timestamps and low rate scores at normal timestamps. $F_{\ell h}$ reverses this assignment, and F_{hh} uses high rate scores for both groups. The same normalization statistics estimated at μ_ℓ are used in all four cases.

Because AP is nonlinear, the contribution can depend on which group is replaced first. We therefore average the marginal change over the two possible orders:

$$\phi_{\text{anom}} = \frac{1}{2}\left[(F_{h\ell} - F_{\ell\ell}) + (F_{hh} - F_{\ell h})\right], \tag{84}$$

$$\phi_{\text{normal}} = \frac{1}{2}\left[(F_{\ell h} - F_{\ell\ell}) + (F_{hh} - F_{h\ell})\right]. \tag{85}$$

By construction,

$$\phi_{\text{anom}} + \phi_{\text{normal}} = F_{hh} - F_{\ell\ell}. \tag{86}$$

Localization. For localization, variables at each anomalous timestamp are divided into labeled and remaining variables. Their prediction errors are independently replaced between the two trajectories.

$H_{\ell\ell}$ uses low rate errors for both groups. $H_{h\ell}$ uses high rate errors for labeled variables and low rate errors for the remaining variables. $H_{\ell h}$ uses the opposite combination, and H_{hh} uses high rate errors for both groups.

Their contributions are

$$\phi_{\text{labeled}} = \frac{1}{2}\left[(H_{h\ell} - H_{\ell\ell}) + (H_{hh} - H_{\ell h})\right], \tag{87}$$

$$\phi_{\text{remaining}} = \frac{1}{2}\left[(H_{\ell h} - H_{\ell\ell}) + (H_{hh} - H_{h\ell})\right], \tag{88}$$

with

$$\phi_{\text{labeled}} + \phi_{\text{remaining}} = H_{hh} - H_{\ell\ell}. \tag{89}$$

The remaining variables are those outside the localization ground truth set; they are not assumed to be physically normal or unaffected by the anomaly. The resulting contributions are reported in Table 4 and are not repeated here.

C.5 Prediction Error Within Anomaly Events

We examine how the update rate affects prediction error while an anomaly event is unfolding. To do so, we first process the full test stream from the learned initialization $\mathbf{W}_0$ using μ_ℓ. For each labeled anomaly event, we save the fast state immediately before the first observation of the event is incorporated. The saved state is copied to initialize all replay conditions for that event, and the event is replayed using

$$\mu_{\text{event}} \in \{\mu_\ell,\ \mu_h,\ 0.1\}. \tag{90}$$

Table 11: **Controlled replay during labeled anomaly events.** The rate before the event determines the fast state used to initialize the replay, while the event rate controls updates during replay. Detection AP is computed using a fixed outside-event reference trajectory; HR@100 is computed from localization scores within the replayed events.

Dataset	Rate before event	Detection AP			Localization HR@100		
		Update rate during event			Update rate during event		
		0	μ_ℓ	μ_h	0	μ_ℓ	μ_h
SMD	μ_ℓ	**45.53**	42.91	31.04	50.34	50.98	**53.60**
	μ_h	**38.86**	36.45	27.20	52.00	52.59	**54.49**
SWaT	μ_ℓ	**76.07**	75.35	35.72	9.83	9.81	**9.97**
	μ_h	**68.14**	67.61	30.16	12.22	12.24	**12.88**
WaDI	μ_ℓ	**46.43**	45.37	37.37	28.42	27.39	**28.76**
	μ_h	**42.53**	41.65	35.01	31.93	32.12	**32.68**

Each event is replayed independently. The replayed final state is discarded, and each event is initialized from its corresponding saved state on the original full-stream μ_ℓ trajectory. Thus, for a given event, the observations and initialization state are identical across replay conditions, and only the update rate used within the event differs.

At each timestamp, prediction error is measured before the current observation is incorporated into the fast state:

$$\mathrm{MSE}_t = \frac{1}{d} \left\| \mathbf{e}_t^- \right\|_2^2 . \tag{91}$$

Because anomaly events have different lengths, timestamps are represented by their relative positions within the corresponding event. The relative event range is divided into common position intervals, and prediction errors are aggregated across events within each interval. The resulting curves in Figure 3 are averaged over the three random seeds.

All replay conditions begin from the same pre-event fast state. They therefore produce the same prediction at the first timestamp of an event. Differences at later timestamps arise from fast state updates accumulated during the event. As shown in Figure 3, larger update rates generally reduce prediction error as the event progresses. This provides direct evidence that the fast state continues to adapt to observations within anomaly events and can make subsequent anomalous observations easier to predict.

C.6 Controlled Replay During Anomaly Events

We next isolate the effect of updates performed during labeled anomaly events. This experiment is a controlled oracle replay used for analysis. It is not the inference policy used by SPLIT^3 in the main benchmark.

We first process the full test stream using either μ_ℓ or μ_h and save the fast state immediately before each labeled anomaly event. Starting from each saved state, we replay the same event while changing only the update rate used during the event:

$$\mu_{\text{event}} \in \{0,\ \mu_\ell,\ \mu_h\}. \tag{92}$$

Each event is replayed independently with the same observations. For a fixed pre-event state, changing μ_{event} therefore isolates the effect of updates accumulated within that event.

Detection evaluation. Detection requires scores over both anomalous and normal timestamps to compute AP. To vary only the scores produced during the replayed anomaly events, we construct a full test score sequence by replacing the scores inside labeled anomaly intervals with the corresponding replay scores. Scores outside those intervals are fixed to the original μ_ℓ trajectory. The same normalization statistics estimated from the normal validation sequence at μ_ℓ are used in every replay condition. AP is then computed over this constructed full score sequence.

This construction is deliberately different from running an end to end test stream with a conditional oracle policy. Its purpose is to hold the scores outside the labeled events fixed while changing the fast state updates inside the events. Accordingly, the detection AP values in Table 11 should be interpreted as controlled intervention results, not as the performance of a deployable detector or as the AP of a standard full-stream run at the corresponding event rate.

Results. For either pre-event state, suppressing updates during the event gives the highest detection AP among the three evaluated event rates on SMD, SWaT, and WaDI. In contrast, $\mu_{\text{event}} = \mu_h$ gives the highest localization HR@100 for both pre-event states on all three datasets. The contrast for detection is particularly clear when comparing $\mu_{\text{event}} = \mu_h$ with smaller event update rates; the difference between 0 and μ_ℓ is more modest in several conditions. We therefore interpret the result as evidence that stronger updates within known anomaly events can weaken detection evidence, rather than as evidence that zero update is universally optimal.

Importantly, this result does not imply that detection should remain frozen throughout the test stream. Table 3 shows that, over the full test stream, detection with μ_ℓ outperforms a frozen fast state on SMD, SWaT, and WaDI. The controlled replay retains the adaptation accumulated before each event and fixes the outside-event detection scores; it changes only the updates performed inside the labeled anomaly interval. On SMD, SWaT, and WaDI datasets, full stream adaptation at μ_ℓ improves detection AP over a frozen state, whereas using μ_h rather than μ_ℓ within labeled events lowers AP in the controlled replay.

The pre-event state also affects both tasks. Even under the same event update rate, changing the state accumulated before the event changes the resulting AP and HR@100. This is consistent with the accumulated fast state analysis in Section 3.2.

Because the replay uses ground truth event boundaries, it cannot be used as standard test time inference. The result instead isolates what would happen if updates inside a known anomaly interval could be controlled independently. The next subsection examines whether a simple label free decision based on the current detection score can reproduce this detection benefit.

We also verify the replay implementation by matching the event rate to the rate used to form the pre-event state. When both rates are μ_ℓ, the replay reproduces the corresponding interval of the original μ_ℓ trajectory. Likewise, when both rates are μ_h, it reproduces the original μ_h trajectory within numerical precision. Since inference follows *predict* $\rightarrow$ *score* $\rightarrow$ *update*, replays initialized from the same pre-event state also have identical predictions at the first timestamp of the event; subsequent differences arise from updates accumulated during replay.

C.7 Detection Score Based Update Suppression

The controlled replay above uses true anomaly event boundaries. Standard inference does not have access to this information. We therefore test whether a simple decision based only on the current detection score can obtain a similar benefit.

Let τ_ρ denote the ρ-th percentile of detection scores on the normal validation sequence. We use $\rho = 99$ and suppress the detection fast state update when the current score exceeds this threshold:

$$\mathbf{W}_t^D = \begin{cases} \mathbf{W}_{t-1}^D, & s_t^D > \tau_{99}, \\ \mathbf{W}_{t-1}^D + \mu_D q_t \mathbf{e}_t^{D,-} \mathbf{k}_t^\top, & s_t^D \le \tau_{99}. \end{cases} \tag{93}$$

The threshold is determined only from normal validation data. No test labels or anomaly event boundaries are used. Because the resulting fast state is carried forward, each update suppression decision can also affect subsequent predictions. The localization fast state is unchanged and follows the standard SPLIT^3 trajectory.

As shown in Table 12, this simple rule does not consistently improve detection. Relative to standard SPLIT^3, AP decreases on SMD and SWaT and increases only slightly on WaDI. Thus, the score based suppression rule evaluated here does not consistently reproduce the detection benefit observed in the oracle replay.

The controlled replay and score based experiment serve different purposes. The former isolates the effect of updates within known anomaly events, whereas the latter evaluates one simple way

Table 12: **Detection AP with score based update suppression.** We report mean AP over three runs using the 99th percentile of normal validation scores as the threshold.

Dataset	SPLIT[3]	Score based suppression
SMD	**42.91**	41.57
SWaT	**75.35**	70.44
WaDI	45.37	**45.83**

of deciding whether to update without those boundaries. The evaluated rule does not consistently improve detection AP over the standard detection trajectory.

C.8 Robustness to Score Normalization

The main evaluation uses variable-wise normalization for detection and raw squared prediction error for localization, as defined in Equation 72. We further examine whether the different update-rate responses of detection and localization depend on this scoring choice. Keeping the trained model, fast-state dynamics, and test-time update rule unchanged, we evaluate the full update-rate grid in Equation 74 using both raw and variable-wise normalized scores for each task.

For detection, we compare

$$s^D_{t,\mathrm{raw}} = \log\left(\epsilon_s + \sum_{i=1}^{d} (e^{D,-}_{i,t})^2\right) \tag{94}$$

with the normalized score used in the main evaluation,

$$s^D_{t,\mathrm{norm}} = \log\left(\epsilon_s + \sum_{i=1}^{d} \frac{(e^{D,-}_{i,t})^2}{\sigma_i^2}\right), \tag{95}$$

where σ_i is estimated from normal validation prediction errors as in Equation 70. For localization, we compare the main raw score

$$s^L_{i,t,\mathrm{raw}} = (e^{L,-}_{i,t})^2 \tag{96}$$

with

$$s^L_{i,t,\mathrm{norm}} = \frac{(e^{L,-}_{i,t})^2}{\sigma_i^2}. \tag{97}$$

This comparison also extends the ranking analysis in Section 3.3. For localization, let $c_i = \sigma_i^{-2}$. The normalized pairwise margin between variables i and j is

$$\widetilde{M}^r_{ij,t} = c_i(e^{r,-}_{i,t})^2 - c_j(e^{r,-}_{j,t})^2, \qquad r \in \{\ell, h\}. \tag{98}$$

Using the definition of $A_{i,t}$ from Section 3.3, its change between the higher- and lower-rate trajectories is

$$\widetilde{M}^h_{ij,t} - \widetilde{M}^\ell_{ij,t} = c_j A_{j,t} - c_i A_{i,t}. \tag{99}$$

Thus, variable-wise normalization changes the relative weighting of prediction errors across variables, but localization still evaluates their relative ordering within each timestamp. Likewise, the detection analysis already allows a positive diagonal weighting matrix $\mathbf{C}$, with raw and normalized scoring corresponding to different choices of that matrix.

Table 13 shows that normalization changes absolute performance, but the separation in update-rate responses remains consistent. On WaDI, the best observed detection rate is 0 with raw scoring and 0.001 with normalized scoring, whereas localization is highest at 0.01 under both scoring rules. On SMD, detection is highest at 0 or 0.001, while localization is highest at 0.1 under both raw and normalized scoring. On SWaT, detection is highest in the lower-rate region, at 0.001 with raw scoring and 0.003 with normalized scoring, whereas localization is highest at 0.03 under both. Across all three datasets, the best observed localization rate is therefore higher than the corresponding detection rate under either scoring rule.

Normalization also affects the absolute localization performance. It improves HR@100 over much of the evaluated range on WaDI and SWaT, whereas raw scoring gives higher HR@100 on SMD. Thus,

Table 13: **Robustness to score normalization across test-time update rates.** Detection AP and localization HR@100 are evaluated with raw and variable-wise normalized prediction errors. Bold values indicate the best observed result within each dataset and scoring column.

Dataset	Update rate	Detection AP		Localization HR@100	
		Raw	Normalized	Raw	Normalized
SMD	0	**41.46**	38.19	47.88	46.31
	0.001	41.18	**42.91**	50.98	49.34
	0.003	37.79	40.59	52.78	50.99
	0.01	33.09	36.59	54.49	52.53
	0.03	30.71	33.26	55.15	53.49
	0.1	29.47	31.52	**55.24**	**53.53**
	0.3	28.52	30.37	54.66	53.19
SWaT	0	72.24	70.44	4.76	5.56
	0.001	**76.33**	75.35	9.81	10.21
	0.003	67.95	**75.38**	10.85	11.23
	0.01	36.01	50.02	12.88	13.40
	0.03	22.07	30.44	**13.25**	**14.21**
	0.1	16.96	20.94	12.49	13.08
	0.3	14.03	16.47	11.57	11.61
WaDI	0	**27.14**	44.50	26.13	27.90
	0.001	25.10	**45.37**	27.39	29.63
	0.003	21.94	43.90	29.78	30.92
	0.01	18.27	39.63	**32.68**	**33.82**
	0.03	12.85	31.00	31.19	31.07
	0.1	8.25	17.66	17.58	24.75
	0.3	6.90	12.59	9.30	20.04

the results do not identify raw squared error as a universally preferable localization score. Rather, variable-wise normalization is an additional scoring choice that can change localization performance while leaving the qualitative task-specific plasticity pattern intact.

The main benchmark retains the raw localization score defined in Equation 72, which is also the score used in the localization ranking analysis of Section 3.3. The normalized variant is evaluated here as an alternative calibration of the same prediction errors rather than as a change to the main evaluation protocol. Likewise, the dataset-wise maxima in Table 13 are descriptive and are not used to retune the main update rates. The main configuration remains the fixed cross-dataset setting in Equation 75. Appendix C.3 separately shows that task-specific rate selection remains beneficial when the evaluation dataset is excluded from the selection calculation. Together, these results show that the observed separation between detection and localization is not tied to a particular choice of variable-wise score normalization.

D Computational Details

Table 14: **Inference efficiency.** Learned parameter counts and arithmetic GFLOPs per observation across datasets. Dashes indicate unprofiled variants.

Method	SMD	SWaT	WaDI	HAI	MSL
(a) Learned parameters (real scalars)					
ALoRa	3,176,726	3,183,395	3,220,331	3,197,246	3,185,447
SARAD	9,568,674	9,588,746	9,699,914	9,630,434	9,594,922
CST-GL (detection)	524,481	606,849	1,063,041	207,201	189,537
CST-GL (localization)	176,481	606,849	1,063,041	–	–
AERCA	44,874,440	56,586,010	182,711,290	91,720,840	60,870,250
GCAD	253,235	1,272,378	3,051,066	1,939,386	606,386
CrossAD	1,328,459	5,277,387	5,277,387	5,277,387	1,327,979
CATCH	229,708,784	229,708,992	229,710,144	229,709,424	229,709,056
M2N2 (detection)	47,062	509,057	2,821,625	1,157,570	96,914
M2N2 (localization)	47,062	509,057	2,821,625	–	–
CANDI (detection)	3,909,518	4,249,058	6,896,354	5,090,438	4,362,130
CANDI (localization)	3,909,518	4,249,058	6,896,354	–	–
SPLIT[3]	**6,298**	**10,939**	**61,123**	**24,898**	**12,639**
(b) Arithmetic GFLOPs per observation					
ALoRa	1.2928×10^{-1}	1.2955×10^{-1}	1.3106×10^{-1}	1.3011×10^{-1}	1.2963×10^{-1}
SARAD	1.4605×10^{0}	1.9683×10^{0}	4.8567×10^{0}	3.0364×10^{0}	2.1254×10^{0}
CST-GL (detection)	1.1148×10^{-1}	1.6575×10^{-1}	6.1528×10^{-1}	4.0268×10^{-2}	2.5782×10^{-2}
CST-GL (localization)	1.6479×10^{-2}	1.6576×10^{-1}	6.1530×10^{-1}	–	–
AERCA	8.9669×10^{-2}	1.1309×10^{-1}	3.6534×10^{-1}	1.8336×10^{-1}	1.2166×10^{-1}
GCAD	5.5488×10^{-1}	6.5282×10^{-1}	3.7544×10^{0}	1.5168×10^{0}	1.9216×10^{0}
CrossAD	4.4800×10^{0}	2.2453×10^{1}	5.4152×10^{1}	3.4340×10^{1}	3.6666×10^{0}
CATCH	1.8703×10^{1}	2.5205×10^{1}	6.2170×10^{1}	3.8878×10^{1}	2.7216×10^{1}
M2N2 (detection)	1.4262×10^{-4}	1.2388×10^{-3}	6.8675×10^{-3}	2.8169×10^{-3}	2.9398×10^{-4}
M2N2 (localization)	1.4258×10^{-4}	1.2373×10^{-3}	6.8638×10^{-3}	–	–
CANDI (detection)	2.2274×10^{0}	1.1209×10^{0}	4.7007×10^{0}	8.1141×10^{-1}	5.7418×10^{0}
CANDI (localization)	6.8845×10^{-1}	1.0955×10^{0}	5.0783×10^{0}	–	–
SPLIT[3]	9.2727×10^{-5}	1.5097×10^{-4}	7.1833×10^{-4}	3.1513×10^{-4}	1.7161×10^{-4}

FFT estimate: $5N\log_2 N$ (complex) and $2.5N\log_2 N$ (real), per transform.
Nonlinearities, fused normalization, SVD, comparisons, sorting, and indexing are excluded.
Radar radii use $\log_{10}(1+\text{parameters})$ and $\log_{10}(1+\text{FLOPs})$, with common axes per panel.

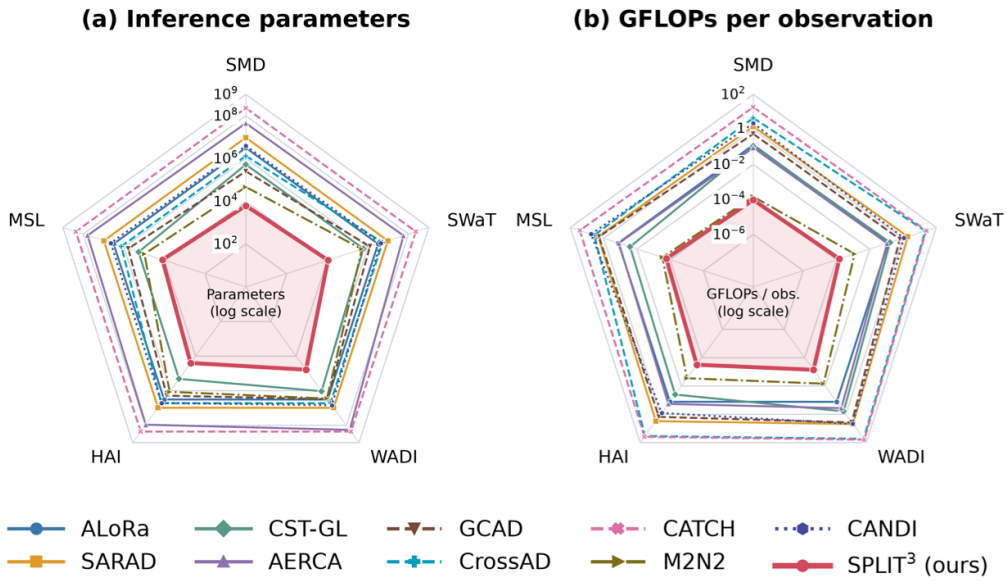


Figure 4: Learned parameter counts and arithmetic GFLOPs per observation across datasets.