Adaptive World-Models for Anomaly Recognition (AWARE): A Framework for Robot Dynamics Prediction and Introspection

Luke Beddow **George Mavroghenis** **Rodrigo Chacon Quesada** **Kamil Dreczkowski**
Cong Sun **Jiankai Wang** **Oscar Kwong-Fai Pang** **Antoine Cully**

BOLD, Imperial College London

Project page: https://adaptive-intelligent-robotics.github.io/aware/
Code: https://github.com/adaptive-intelligent-robotics/aware

Abstract:
Unrecognized anomalies in critical infrastructure can rapidly cascade into catastrophic physical or operational failures. Current anomaly detection approaches often rely on direct access to the monitored system by utilizing onboard sensors, such as motor encoders, to detect anomalous states. However, legacy hardware often lacks these sensors, or strict security protocols may prohibit access altogether. We therefore present Adaptive World-model for Anomaly REcognition (AWARE), a world model framework that is capable of anomaly detection using only external camera monitoring via system dynamics estimation and state prediction. Evaluated across 240 real-world robotic crane trajectories, including payload collisions and motor degradations, AWARE demonstrates a 26% improvement in anomaly classification over our strongest baseline under noisy, camera-only state estimation. Furthermore, the inferred dynamics parameter can be decoded to identify the source of faults, localizing the degrading motor with 88% accuracy.

Keywords: anomaly detection, world model, dynamics estimation

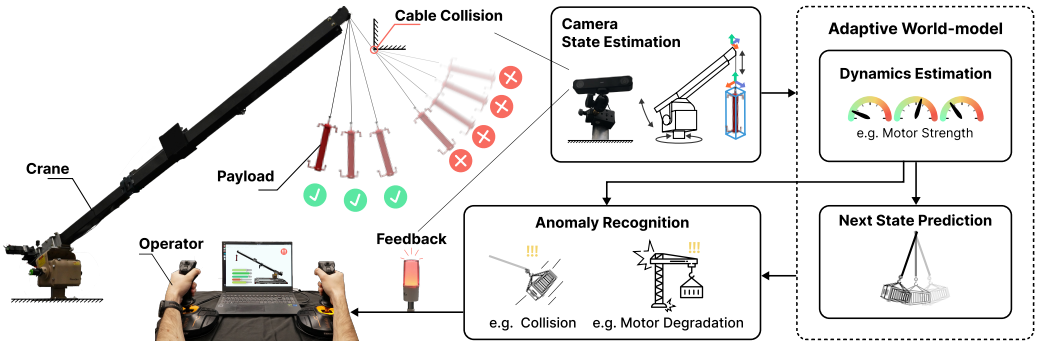


Figure 1: System and model overview, showing human-operated robotic crane used for anomaly detection experiments. AWARE operates as follows: 1) camera-based state estimation; 2) estimation of system hidden dynamics parameters; 3) prediction of future trajectory; 4) anomaly recognition via examination of both the estimated dynamics parameters and the trajectory prediction error.

1 Introduction

Critical infrastructure, such as ports, power plants, and factories, heavily relies on robotic platforms to perform tasks [1]. Disruption to these systems, such as through unauthorized operation

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

or cyberattacks, can be physically dangerous to nearby workers, and cause significant economic disruption. Consequently, monitoring these systems for anomalous behavior is crucial to ensure safe and continuous nominal operation.

Traditional learning-based anomaly detection approaches focus either on spatial reconstruction methods for detecting unexpected features in high-dimensional state data (e.g. via variational auto-encoders), or temporal sequence methods for forecasting and measuring divergence from predicted states (e.g. via dynamics models) [2, 3]. However, these methods require precise estimation of the current robot state, typically via access to onboard sensors, such as motor encoders [4]. Many infrastructure applications prevent this low-level access due to lacking the required sensors or because of strict security protocols. Such applications, necessitate a physical 'air-gap' between the state estimation system and the observed hardware, such as through an external camera. This form of external passive state estimation injects noise throughout the entire anomaly detection pipeline and prevents sufficient system identification, drastically degrading the performance of learned models.

Foundation model-based approaches, (e.g. LLMs, VLMs) offer a promising alternative [5, 6] . Instead of relying on precise state data, they leverage a more general semantic understanding of the scene (i.e. directly from vision or text descriptions) to detect anomalies. However this comes with a clear trade-off: sacrificing a precise understanding of temporal dynamics for semantic generalizability, which may prohibit detection of subtle, mechanics-based anomalies.

To bridge this gap, we introduce **Adaptive World-Models for Anomaly Recognition (AWARE)**: a framework for anomaly detection under high-uncertainty camera-only state estimation. AWARE jointly trains a latent dynamics estimator and world model. The estimator provides a latent representation of the robot's intrinsic parameters, such as motor gain or mass, while the world model uses this representation, along with an intended action, to produce an accurate prediction of the next robot state under the assumed dynamics. AWARE directly addresses the three prior challenges of passive monitoring by introducing noise resilience via a novel dual-signal architecture, relaxing system identification requirements through domain randomization and world model dynamics conditioning, and providing explainable anomaly detection by identifying the fault source. Figure 1 shows the AWARE system overview.

The key contributions of this work are:

- **AWARE framework**. An adaptive world model conditioned on a latent dynamics estimator designed for anomaly detection under high noise state estimation. AWARE is trained fully in simulation using domain randomization, without requiring perfect system identification or explicit priors on anomaly types.
- **Robust detection and identification**. We demonstrate that a novel dual-signal architecture of world model prediction error and latent dynamics deviation improves the detection of in and out-of-distribution anomalies on a challenging real world platform. AWARE shows a modest, but statistically significant, 3% improvement over the next best baseline under low-noise state estimation, and a significant 26% performance increase while identifying the source of in-distribution anomalies in 88% of cases under high-noise state estimation.
- **Rigorous real-world experimental analysis**. We evaluate AWARE over 240 real world robotic crane trajectories driven by 5 different human operators. We perform a detailed analysis and ablation of the AWARE components and compare with a comprehensive selection of baselines including simulators, world models, foundation models, dynamics estimators and established anomaly detection approaches.

2 Related Work

Anomaly detection in robotics involves discerning unexpected states or behaviors [1]. Statistical approaches [7] and reconstruction techniques [8] score observations by their likelihood under a reference distribution. While effective for single-state anomalies, infrastructure monitoring must additionally capture temporal dynamics changes and distinguish anomalies from valid operator

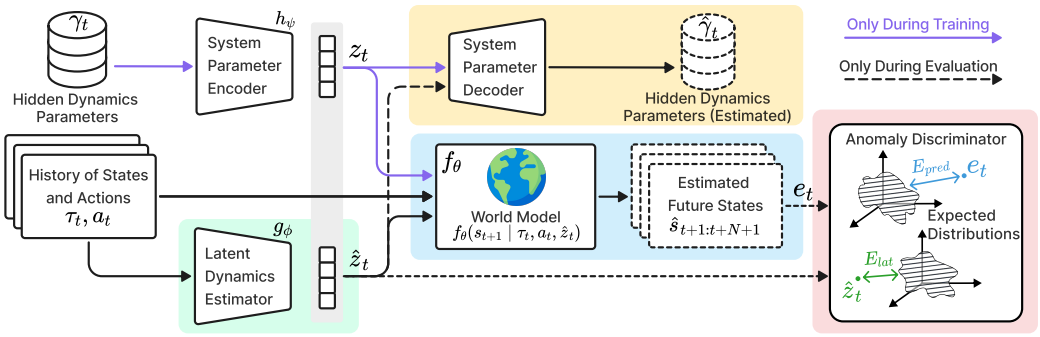


Figure 2: AWARE framework. Our model consists of two key components. First, a **Latent Dynamics Estimator** (green) reconstructs a latent encoding of system dynamics ($\hat{z}_t$) using a history of state-action pairs. Second, a **World Model** (blue) predicts the next state, given this encoding, the same history, and the current action. This integrated framework enables **introspection** (yellow) and **anomaly detection** (red) by comparing dynamics estimations and state prediction errors (e_t) against their expected distributions under normal operation.

actions. Recent spatiotemporal methods address this for driving, cyber-attacks, and navigation failures [9, 10, 11, 12, 13, 14], but assume labeled anomaly data that is unavailable in semi-supervised infrastructure settings [15]. Foundation-model approaches offer zero-shot semantic understanding to derive anomaly through visual or language prompts [5, 6, 16, 17]. However, these methods reason over semantic scene features and are not sensitive to subtle perturbations in system dynamics (e.g. gradual motor degradation or shifts in inertial properties) that weakly manifest in visual appearance. Predictive temporal models instead learn nominal dynamics and flag deviations [18, 4], but prior work assumes dense onboard sensing. To handle sensing uncertainty without onboard access, robotics uses estimator-based consistency statistics [19, 20] and distribution-free calibration [21, 22]. Unlike prior work, AWARE addresses passive monitoring with no onboard sensing, combining a dynamics-aware world model with latent parameter estimation to detect anomalies that are invisible to appearance-based methods yet critical for safe operation.

World models predict future states given current state and action, and are widely used in model-based RL [23, 24, 25, 26, 27]. The Dreamer series established the Recurrent State-Space Model (RSSM) [28] as a prominent backbone; Robotic World Model (RWM) [29] extended this to output task-relevant quantities alongside state predictions. Existing applications to anomaly detection [30, 3] demonstrate success through state prediction and reconstruction. AWARE goes further through the latent dynamics estimator, providing an explicit prior on the assumed dynamics for the world model and an additional signal, the latent deviation, for more robust anomaly detection. Estimation of such dynamics parameters via domain randomization in simulation has been demonstrated in many RL control policies [31, 32, 33]. Regularised Online Adaptation (ROA) [34] trains a teacher–student estimator whose latent dynamics predictions condition a control policy. AWARE adapts this for anomaly detection by learning a teacher encoder through world model prediction gradients and a more informative representation by removing teacher regularization and latent bottleneck whilst including a decoder for introspection.

3 Adaptive World-Models for Anomaly Recognition (AWARE)

We present **AWARE**, an adaptive world model framework, in Figure 2. A latent dynamics estimator predicts an encoding of the system dynamics, which conditions a world model to predict the next state. The latent dynamics estimation and next state prediction error are jointly used to recognize anomalies by comparing them with their expected distribution under normal operation.

3.1 Problem Formulation

We model the environment as a parametrized stochastic dynamical system, defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{U})$, where $\mathcal{S} \subseteq \mathbb{R}^{n_s}$ and $\mathcal{A} \subseteq \mathbb{R}^{n_a}$ denote continuous state and action spaces, and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$

is the dynamics context. Each context $u = (\gamma, \zeta) \in \mathcal{U}$ comprises modeled dynamics parameters γ, such as motor characteristics, and unmodeled dynamics ζ, such as disturbance forces and unexpected contacts. The probabilistic transition function is defined as $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{U} \to \Delta(\mathcal{S})$, and transitions at timestep t are sampled according to $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t, \gamma_t, \zeta_t)$.

We define anomalies as when the dynamics context changes from an initial value $u_0 = (\gamma_0, \zeta_0)$ to a new value $u_t = (\gamma_t, \zeta_t) \neq u_0$. Changes in γ_t are detected by estimating it via latent representation $\hat{z}_t$. Since ζ_t cannot be estimated directly, changes in it are instead detected through errors in predicting the next state $\hat{s}_{t+1}$.

AWARE detects anomalies by taking as input an H-step history of states and actions, $\tau_t = (s_{t-H:t}, a_{t-H:t-1})$, along with the current action a_t and predicting two quantities: $\hat{z}_t$ and $\hat{s}_{t+1}$. Each prediction is converted to an anomaly score based on its error against a reference distribution characterizing normal operation. Figure 2 illustrates the AWARE architecture.

3.2 Dynamics Parameter Estimation

Identifying true robotic system dynamics requires complex system identification [35], which is particularly challenging when only noisy state estimates are available. AWARE instead learns to estimate a representation of the dynamics from a history of states and actions using a method based on *Regularized Online Adaptation* (ROA) [34].

Training occurs in simulation, by varying a predefined set of modeled dynamics parameters, γ, such as motor strength or body inertia. Such parameters may be unknown on the real system, but can typically be bounded within a range $\gamma \in [\gamma_{min}, \gamma_{max}]$, with a wider range reflecting greater uncertainty.

During training, γ is randomly sampled and applied to the physics simulation. Following the teacher-student method of ROA, a teacher encoder h_ψ maps the true parameters to a latent representation $z_t = h_\psi(\gamma_t)$, shaped by the downstream world model. A student latent dynamics estimator g_ϕ then learns to predict this representation from a history of states and actions $\hat{z}_t = g_\phi(s_{t-H:t}, a_{t-H:t})$ by minimizing the following loss:

$$\mathcal{L}_\phi = ||g_\phi(s_{t-H:t}, a_{t-H:t}) - h_\psi(\gamma_t)||^2 \tag{1}$$

We make three key changes to adapt ROA for anomaly detection. First, the downstream model that shapes z is a world model rather than a policy, meaning z is optimized for dynamics prediction rather than control (Sec. 3.3). Second, we add a System Parameter Decoder that recovers γ from $\hat{z}$. This supports introspection into which aspect of the dynamics has changed. Third, we remove both compression and regularization of z used in the original ROA method, to preserve as much information as possible in the latent representation.

3.3 World Model State Prediction

We train a world model f_θ to approximate the true transition function $\mathcal{P}$. Conditioned on the history τ_t, the current action a_t and the latent dynamics z ($\hat{z}_t$ at deployment), it outputs a Gaussian over the next state with diagonal covariance, whose mean is the prediction $\hat{s}_{t+1}$:

$$f_\theta(s_{t+1} \mid \tau_t, a_t, z) = \mathcal{N}\big(\hat{s}_{t+1},\, \mathrm{diag}(\hat{\sigma}^2_{t+1})\big). \tag{2}$$

To show that AWARE is architecture-agnostic, we evaluate two backbones: a Transformer, which attends over τ_t directly, and an RSSM [28] , which summarizes the history in a recurrent hidden state h_t. Both are trained by minimizing the negative log-likelihood $\mathcal{L}_\theta = -\log f_\theta(s_{t+1} \mid \tau_t, a_t, z)$.

3.4 Anomaly Decision

An anomaly score is derived from the estimated latent dynamics parameters $\hat{z}_t$ as well as the prediction of the next state $\hat{s}_{t+1}$. The latter, is converted into an error e_t against the observed next state s_{t+1} and averaged over a prediction window of N time-steps. The prediction error is normalized by the

predicted standard deviation $\hat{\sigma}$, reducing the contribution of prediction errors to the anomaly score when the world model expresses higher uncertainty:

$$e_t = \frac{1}{N} \sum_{k=t-N}^{t-1} |s_{k+1} - \hat{s}_{k+1}| \oslash \hat{\sigma}_{k+1}. \tag{3}$$

During initialization, a set of reference trajectories is collected to characterize the distribution of nominal system operation with respect to the dynamics parameters $\hat{z}_0$ and model error e_0. The reference trajectories may be collected prior to initialization as long as the robot context dynamics remain fixed up to the current operating point.

At test-time, the Mahalanobis distance [36] between the state error, dynamics latent, and their respective reference distributions, is computed to give two anomaly scores E_{lat} and E_{pred}. E_{lat} arises from the latent dynamics estimation $\hat{z}_t$ which is intended to detect changes in the modeled dynamics γ. E_{pred} derives from the prediction error e_t and aims to detect changes in unmodeled dynamics ζ. Additionally, by conditioning the world model on assumed nominal dynamics $\hat{z}_0$ (as opposed to $\hat{z}_t$) we provide E_{pred} with additional sensitivity to changes in γ. Anomaly decisions use two independent thresholds for E_{lat} and E_{pred}, where each score represents the distance in standard deviations that an incoming data point is from the reference distribution. Thresholds are calibrated during nominal operation considering specific operational sensitivity requirements.

4 Experimental Evaluation

Experimental Procedure and Task Design. The models were trained using data generated from a MuJoCo simulation of the crane. The simulation was calibrated to the real platform through a grid search over parameters γ. We then generated data using an exploratory policy with uniform domain randomization centered over the ground-truth values. This randomization therefore removes the requirement for perfect identification of γ. Appendix A.2 details the parameterized values and domain randomization ranges.

Our robotic crane is an under-actuated system with three actuated joints, slew (q_1, revolute), luff (q_2, revolute), hoist (q_5, prismatic), and four free joints, formed by the suspended payload and the attached cable, with (q_3, q_4) at the boom captures pendulum swing and (q_6, q_7) at the payload captures double-pendulum effects when the payload is not aligned with the cable (Figure 3).

We collected human-in-the-loop crane trajectories with one of five human operators continuously moving the payload through three randomly sampled waypoints. Two state-estimation pipelines ran in parallel to produce the robot state s_t and operator action a_t at each timestep: one using high-quality motion capture, and the other using RGB cameras. To make sure AWARE is evaluated on system anomalies other than perception failures (e.g. sensor blockage, markers or payload leaving the field of view), we screened the dataset during collection to ensure such episodes are absent. In total, we collected 240 trajectories, comprising 80 nominal runs and 160 anomalous runs.

We consider two anomaly families. *Payload collision* (40 trajectories) introduces an unmodeled disturbance ζ via a single impact to the payload or cable during the anomaly window. *Motor degradation* (120 trajectories) scales down the executed velocity of one of the three motors, representing anomalies in modeled dynamics γ. For each motor, 40 trajectories are evenly split across $\{75, 50, 25, 0\}\%$ of nominal velocity. Since training randomizes motor strength within $50 - 150\%$, 75% and 50% are in-distribution while 25% and 0% are out-of-distribution. We compute anomaly scores as a continuous (per-timestep) classification signal and adopt a trajectory-level criterion following prior work [18]: a nominal trajectory must contain no anomaly decisions, while an anomalous trajectory must trigger at least one anomaly decision within the anomaly window.

State Estimation. Robot joint positions ($q_1, \ldots, q_7$) and operator actions (a_1, a_2, a_3), were measured for every trajectory using two different state estimation pipelines, all at 20 Hz.

Firstly, *Motion Capture Quality Estimation* captures high quality data. This case uses motion tracking cameras to measure the payload position relative to the crane boom, and extract the underactuated

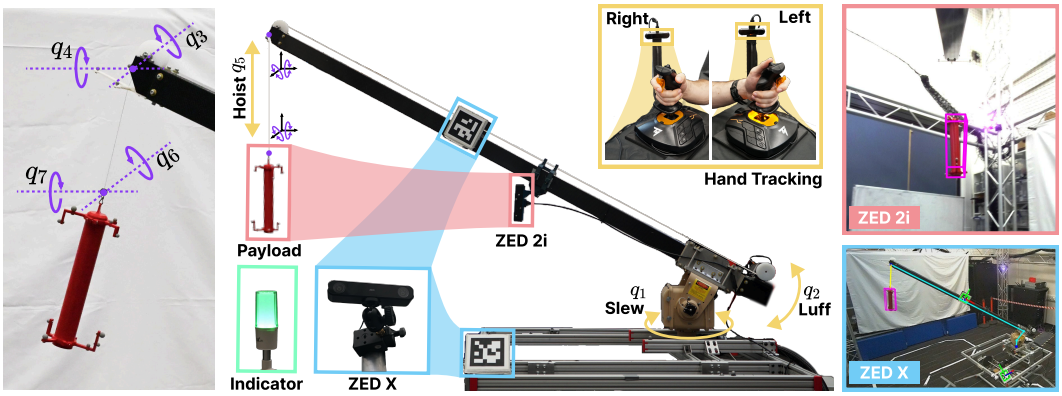


Figure 3: Deployment Method **(1) Crane platform**: The system is modeled by three actuated joints: slew (q_1), luff (q_2), and hoist (q_5). The under-actuated dynamics are captured by a double pendulum model: two angles (q_3, q_4) defining the movement at the hoist connection point (cable swing), and two angles (q_6, q_7) defining the relative orientation of the payload with respect to the cable. **(2) Camera-Based State Estimation**: Operator actions are estimated by two hand-tracking cameras. Crane states q_1 and q_2 are derived from ArUco marker poses using a ZED X camera. The remaining states (q_3–q_7) are estimated by a boom-mounted ZED 2i camera utilizing a Convolutional Neural Network (CNN). **(3) Operator Interface**: A multi-color indicator lamp provides real-time visual feedback, signaling nominal operation (green light) or system alerts (red light).

joint angles and hoist length ($q_3, \ldots, q_7$). The remaining joints are measured using proprioception, motor encoders for slew and luff joints (q_1, q_2), and the exact joystick position for operator actions (a_1, a_2, a_3). The resultant state estimation approaches the ground truth, but noise and measurable calibration error arising from the motion capture pipeline reduce data quality.

Secondly, *Camera Only Estimation*: a fully passive state estimation pipeline using only exteroception. This data contains considerably higher levels of noise, and was motivated by the application of infrastructure monitoring, which can require entirely exteroceptive monitoring because proprioceptive sensors may be blocked due to security requirements or constraints of legacy equipment.

Four cameras estimate state and action: a ZED X for slew and luff joints (q_1, q_2) via ArUco markers, a boom-mounted ZED 2i with a CNN (EfficientNet-B2 backbone [37]) for payload joints ($q_3, \ldots, q_7$), and two Leap Motion cameras for operator actions via an MLP. Joint velocities are derived by finite difference after outlier filtering.

Metrics. We define the anomaly detection experiment as trajectory level classification, derived from per-timestep anomaly decisions. This results in a binary label for each of the 240 trajectories in the experiment. Performance was evaluated using Average Precision (AP), a threshold-independent measure, and F1 Score. To standardize dual-threshold comparison, we fix the threshold grid at 32×32 and use interpolated AP, which constructs a monotonic Pareto front treating single and dual threshold methods identically. Note that a random signal achieves 0.17 AP. Generalization is assessed via F1 score using stratified 5-fold cross validation, with thresholds selected on validation folds to prevents dual threshold overfitting. All metrics were averaged over 10 repeats, each with a random sample of 10 anomaly free trajectories as the reference distribution for calculating the Mahalanobis distance. Statistical significance was assessed using a two-tailed Wilcoxon signed-rank test over 10 random model seeds for both AP and F1 score. Given the two metrics and two state estimation conditions, we applied the Bonferroni correction for an adjusted significance level of $\alpha = 0.05/4 = 0.0125$.

Model Baselines. We demonstrate that AWARE is model agnostic by evaluating with both a Transformer and RSSM world model, in line with previous work [29, 38, 39]. We also evaluate both as stand-alone world model baselines (without the full AWARE method) alongside the fine-tuned MuJoCo model from Appendix A.2.

We select two non-temporal, non-model based anomaly detection baselines: Isolation Forest [40] and Statistical Anomaly Detection. The former is an unsupervised tree-based method that aims to isolate

Table 1: Anomaly recognition results over 240 real robotic trajectories, averaged over 10 seeds per trained model. Metrics detail classification performance for identifying anomalies at the correct times within trajectories, under two different passive state estimation schemes. Bold entries are statistically significant improvement over the best non-AWARE model. *AnyAnomaly is only evaluated on the nominal and payload collision set, see Appendix C.3 for more detail.

Method	Motion Capture Quality State Estimation (Low Noise)		Camera Only State Estimation (High Noise)	
	Avg. Precision ± std	F1 Score ± std	Avg. Precision ± std	F1 Score ± std
Randomly Generated Signal	0.17 ± 0.03	0.26 ± 0.03	0.17 ± 0.03	0.26 ± 0.03
Isolation Forest [40]	0.29 ± 0.00	0.36 ± 0.01	0.16 ± 0.00	0.25 ± 0.01
Statistical Anomaly Detection	0.40 ± 0.01	0.40 ± 0.01	0.22 ± 0.01	0.31 ± 0.02
AESOP [6]	0.47 ± 0.00	0.56 ± 0.00	0.24 ± 0.00	0.41 ± 0.00
AnyAnomaly* [5]	0.37 ± 0.07	0.56 ± 0.06	0.37 ± 0.07	0.56 ± 0.06
MuJoCo [41]	0.77 ± 0.00	0.77 ± 0.00	0.45 ± 0.00	0.44 ± 0.00
RSSM [23]	0.86 ± 0.01	0.80 ± 0.01	0.53 ± 0.01	0.56 ± 0.01
Transformer [42]	0.84 ± 0.01	0.79 ± 0.01	0.41 ± 0.02	0.45 ± 0.01
ROA [34]	0.34 ± 0.06	0.39 ± 0.04	0.29 ± 0.04	0.35 ± 0.03
RWM [29]	0.90 ± 0.03	0.85 ± 0.03	0.61 ± 0.05	0.59 ± 0.03
RSSM AWARE (ours)	0.91 ± 0.00	0.85 ± 0.01	0.76 ± 0.01	$\mathbf{0.69 \pm 0.01}$
Transformer AWARE (ours)	$\mathbf{0.93 \pm 0.01}$	$\mathbf{0.87 \pm 0.01}$	$\mathbf{0.77 \pm 0.01}$	0.68 ± 0.01

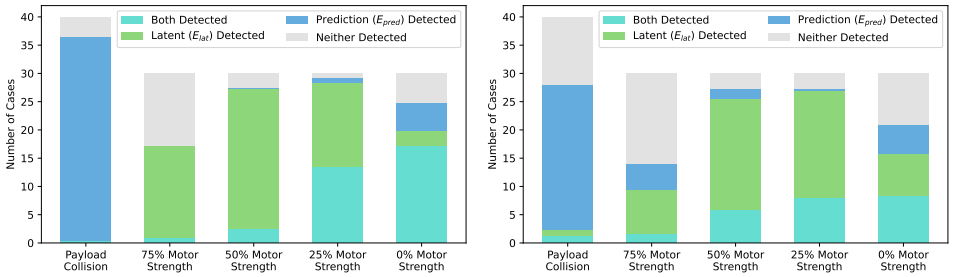
anomalous data-points via random partitioning and the latter measures the Mahalanobis distance of each data-point to a reference distribution. Both work using raw states and actions. We evaluate two foundation model based methods: AESOP [6] and AnyAnomaly [5]. AESOP generates an embedding from text-descriptive states and actions using a small LLM, computing anomaly signals via the Mahalanobis distance to a reference distribution. AnyAnomaly uses a VLA to infer if an anomaly is occurring in a batch of image frames, requiring the type anomaly to be specified in the prompt. As AnyAnomaly only operates over a short sequence of frames, we only evaluate it on the payload collision anomaly set. See Appendix C.3 for the implementation details.

We use the ROA [34] method as a baseline, training a latent estimator with γ as the privileged information and identical domain randomization. This allows for latent dynamics anomaly scores. We train it via learning a policy to navigate the crane payload to specific targets in MuJoCo. Finally, we implement RWM [29], as the closest related world model to our method. We set their model's privileged information to γ, which whilst outside of the scope of the in-domain learning presented by their method, is valid for architectural comparison. This results in both prediction error and dynamics estimation anomaly scores.

5 Results and Discussion

Anomaly Recognition. AWARE (Transformer backbone) achieves the highest performance in anomaly recognition (Table 1), yielding statistically significant improvements over the strongest baseline, RWM, for both AP and F1 across low and high noise state estimation (Appendix Table 4). Specifically, AWARE reaches 0.93 AP (low noise) and 0.77 AP (high noise), outperforming RWM by 3% and 26%, respectively. Methods relying solely on state prediction error E_{pred} (RSSM, Transformer, MuJoCo) perform well under low noise but suffer an average 44% AP degradation on high noise data. Standard anomaly detection methods (Isolation Forest, Statistical Anomaly Detection) fail to reliably detect anomalies, and foundation-model baselines (AESOP, AnyAnomaly) perform 38% worse on average than dynamics-based methods. Finally, the standard ROA latent estimator performs poorly compared to our world model prediction objective. Appendix B.1 provides performance of E_{pred} and E_{lat} individually for each method.

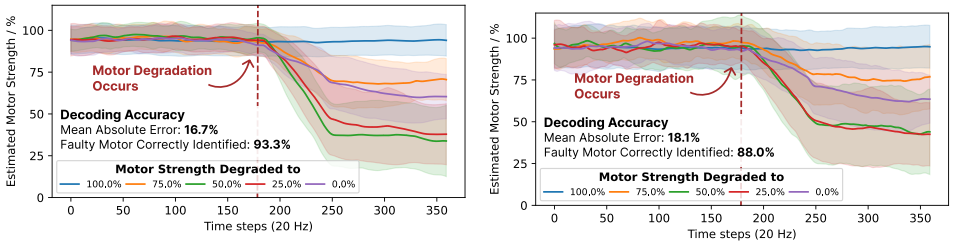
Figure 4 compares the effectiveness of E_{lat} and E_{pred} across all the anomaly cases and shows how each of the two measurement signals contributed to solving the anomaly recognition task. E_{lat}



(a) Low Noise State Estimation (b) High Noise State Estimation

Figure 4: Contribution of Latent and Predictor to Anomaly Recognition across different Impairments types using the *Transformer AWARE* model under (a) *Motion Capture Quality Estimation* (low noise) and (b) *Camera Only Estimation* (high noise).

mostly detects in distribution anomalies (75%, 50% motor strength) and becomes less effective out-of-distribution (0%, 25% motor strength, payload collision). E_{pred} continuously detected more cases as the amount of motor degradation increased, and was far more effective at detecting payload anomalies. These results support our hypothesis that E_{lat} detects changes in modeled dynamics, γ, whilst E_{pred} detects changes in the unmodeled dynamics, ζ. This effect is even more significant in the low noise setting.



(a) Low Noise State Estimation (b) High Noise State Estimation

Figure 5: Estimated Motor Strength using the *Transformer AWARE* model under (a) Low Noise State Estimation Data and (b) High Noise State Estimation Data. The solid curves and shaded bands represent the mean and standard deviation over multiple environments respectively. Decoding Accuracy is the rate of correctly detecting the impaired motor within a $\pm 12.5\%$ tolerance.

System Introspection. Figure 5 summarizes results from the introspection experiment, where we evaluate if we can recover the degraded motor strength from the estimated dynamics latent. Estimation accuracy is quantified by the mean absolute error (MAE) between the true and decoded motor strength, averaged over the final 100 timesteps of each trajectory, and aggregated across motor degradation conditions ($0\% - 75\%$ of nominal strength). Under high noise estimation, the MAE is 18.1%, with 88.0% accuracy in identifying which motor is impaired. Under low noise estimation, the MAE is 16.7% and fault identification accuracy 93.3%.

Discussion. AWARE outperforms all baselines in both noise conditions and are the most resilient to degrading data quality. The key factor is the dual-signal architecture (E_{pred}, E_{lat}). E_{pred}, detects unmodeled anomalies ζ, and can be made more sensitive to modeled anomalies when fixed on nominal dynamics $\hat{z}_0$ (Table 5). However, E_{pred} is inherently noise sensitive and suffers a large performance drop under camera-only state estimation (Figure 6). State prediction error only methods (RSSM, Transformer, MuJoCo) therefore see poor performance when operating in this noise setting. E_{lat} is crucial for detecting modeled anomalies γ that are missed by E_{pred} (Figure 4). Additionally, E_{lat} provides superior noise resiliency (Figure 6, Table 3) as it is referenced against an implicitly denoised $\hat{z}_0$. The unrestricted, world-model-informed dynamics representation learned

by AWARE demonstrates significantly better performance compared to the control-informed representation learned by ROA and raw parameter prediction from RWM.

Foundation model based methods (AESOP and AnyAnomaly) demonstrate worse performance than dynamics-based methods across both noise levels. This confirms the importance of accurate temporal dynamics understanding over semantic generalization for robotic anomaly detection. The results for these methods are discussed further in Appendix C.3. A full discussion on the optimal way to combine E_{pred}, E_{lat} and different distance measures to the reference distribution is detailed in Appendix D.1.

The introspection results (Figure 5) confirm the model learns to infer motor strength from state-action history. Estimates stabilize $\sim$80 timesteps after anomaly onset, roughly one full history window. In-distribution degradations (75%, 50%, motor strength) are decoded more accurately than out-of-distribution cases (25%, 0%), consistent with the limits of domain randomization.

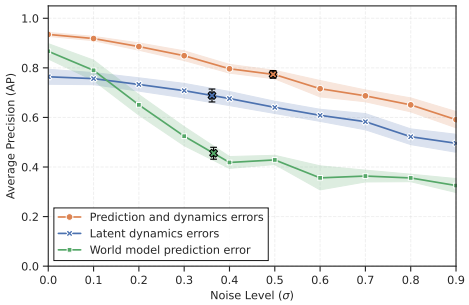


Figure 6: Average Precision (AP) of Transformer AWARE across three anomaly detection modes as increasing levels of noise, sampled from a Student's t-distribution, are injected into the Motion Capture Quality dataset. Noise is scaled relative to the standard deviation (σ) of each joint. The noise level of the Camera Only Quality data relative to the Motion Capture Quality data is estimated by inverse interpolation of the AP score, with the cross indicating the interpolated σ corresponding to the Camera Only Quality data.

6 Limitations

A core limitation of the AWARE method is its assumption of fixed operating conditions under our reference distribution. Whilst AWARE is robust to the initial distribution choice, moving to a new non-anomalous operating point (e.g. a mobile robot traversing varying terrain) may trigger false positives. Future work could address this through periodic updates of the reference distribution or defining separate intrinsic and extrinsic reference distributions to separate harmless changes in the environment from true intrinsic failures.

We acknowledge AWARE's proven generalization is limited by our evaluation focusing on a single platform. However, AWARE is designed to make few platform specific assumptions by using domain randomization and requiring no priors on the type of anomalies that occur. Nonetheless, validation on other systems represents a valuable direction for future work. We hope that releasing our code base will help others extend this work and contribute towards safer robtic systems.

7 Conclusions

We present AWARE, a world modeling framework for robust anomaly detection under high noise state estimation. It enables passive, camera only monitoring of critical infrastructure through a novel dual-signal architecture without requiring precise system identification or any anomaly priors. On a real world robotic crane dataset of 240 trajectories, AWARE achieves a 93% AP with low-noise data, and a 77% AP under camera-only estimation, showing a 26% improvement over baselines. AWARE also correctly isolates the faulty motor in 88% of cases. By combining state prediction with latent dynamics estimation, AWARE bridges the gap between noise resilience and dynamic precision, providing a reliable, introspective safety layer for passive infrastructure monitoring.

Acknowledgments

This work is funded by the Defense Advanced Research Projects Agency (DARPA) Learning Introspective Control (LINC) program; and the Engineering and Physical Sciences Research Council [grant number UKRI5149] as well as other sponsors of BOLD (www.bold-lab.ai/sponsors).

References

[1] S. C. Nandakumar. Anomaly detection methods in autonomous robotic missions. *Sensors*, 24 (4):1330, 2024. doi:10.3390/s24041330.

[2] D. Bogdoll, M. Nitsche, and J. M. Zollner. Anomaly Detection in Autonomous Driving: A Survey . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4487–4498, Los Alamitos, CA, USA, 2022. IEEE Computer Society. doi:10.1109/CVPRW56347.2022.00495.

[3] F. Domberg and G. Schildbach. World models for anomaly detection during model-based reinforcement learning inference. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4182–4187, 2025. doi:10.1109/IROS60139.2025.11245876.

[4] H. Lu, M. Du, K. Qian, X. He, and K. Wang. GAN-based data augmentation strategy for sensor anomaly detection in industrial robots. *IEEE Sensors Journal*, 22(18):17464–17474, 2022. doi:10.1109/JSEN.2021.3069452.

[5] S. Ahn, Y. Jo, K. Lee, S. Kwon, I. Hong, and S. Park. Anyanomaly: Zero-shot customizable video anomaly detection with lvlm. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3026–3035, March 2026.

[6] R. Sinha, A. Elhafsi, C. Agia, M. Foutter, E. Schmerling, and M. Pavone. Real-time anomaly detection and reactive planning with large language models. In *Robotics: Science and Systems*, 2024.

[7] T. Schnell, C. Plasberg, L. Puck, T. Buettner, C. Eichmann, G. Heppner, A. Roennau, and R. Dillmann. Robot health estimation through unsupervised anomaly detection using gaussian mixture models. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1037–1042, 2020. doi:10.1109/CASE48305.2020.9217025.

[8] H. Nehme, C. Aubry, R. Rossi, and R. Boutteau. An anomaly detection approach to monitor the structured-based navigation in agricultural robotics. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 1111–1117, 2021. doi:10.1109/CASE49439.2021.9551568.

[9] M. B. Ammar, A. Mendoza, N. Belkhir, A. Manzanera, and G. Franchi. Foundation models and transformers for anomaly detection: A survey. *Information Fusion*, 126:103517, 2026. ISSN 1566-2535. doi:https://doi.org/10.1016/j.inffus.2025.103517.

[10] T. Kang, B.-J. You, J. Park, and Y. Lee. A real-time anomaly detection method for robots based on a flexible and sparse latent space. *Engineering Applications of Artificial Intelligence*, 158:111310, 2025. URL `https://www.sciencedirect.com/science/article/abs/pii/S0952197625013120`.

[11] J. Fang, J. Qiao, J. Bai, H. Yu, and J. Xue. Traffic accident detection via self-supervised consistency learning in driving scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):9601–9614, 2022. doi:10.1109/TITS.2022.3157254.

[12] A. Zekry, A. Sayed, M. Moussa, and M. Elhabiby. Anomaly detection using iot sensor-assisted convlstm models for connected vehicles. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pages 1–6, 2021. doi:10.1109/VTC2021-Spring51267.2021.9449086.

[13] T. Ji, A. N. Sivakumar, G. Chowdhary, and K. Driggs-Campbell. Proactive anomaly detection for robot navigation with multi-sensor fusion. *IEEE Robotics and Automation Letters*, 7(2): 4975–4982, 2022. doi:10.1109/LRA.2022.3153989.

[14] S. Rahman, R. DiPietro, D. Kedarisetti, and V. Kulathumani. Large-scale indoor mapping with failure detection and recovery in slam. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12294–12301. IEEE, 2024.

[15] J. T. Brockmann, M. Rudolph, B. Rosenhahn, and B. Wandt. The voraus-ad dataset for anomaly detection in robot applications. *IEEE Transactions on Robotics*, 40:438–451, 2024. doi:10.1109/TRO.2023.3332224.

[16] C. Agia, R. Sinha, J. Yang, Z. Cao, R. Antonova, M. Pavone, and J. Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 689–723. PMLR, 2025.

[17] H. Liu, Y. Zhang, V. Betala, E. Zhang, J. Liu, C. Ding, and Y. Zhu. Multi-task interactive robot fleet learning with visual world models. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 4286–4313. PMLR, 2025.

[18] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 387–395, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi:10.1145/3219819.3219845.

[19] N. Darabi, S. Tayebati, S. Senthilkumar, D. Jayasuriya, S. Ravi, T. Tulabandhula, and A. R. Trivedi. Starnet: Sensor trustworthiness and anomaly recognition via lightweight likelihood regret for robust edge autonomy. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

[20] O. A. Hafez, M. Joerger, and M. Spenko. How safe is particle filtering-based localization for mobile robots? an integrity monitoring approach. *IEEE Transactions on Robotics*, 40: 3372–3387, 2024. doi:10.1109/TRO.2024.3420798.

[21] R. Luo, S. Zhao, J. Kuck, B. Ivanovic, S. Savarese, E. Schmerling, and M. Pavone. Sample-efficient safety assurances using conformal prediction. *The International Journal of Robotics Research*, 43(9):1409–1424, 2024. doi:10.1177/02783649231221580.

[22] C. Xu, T. K. Nguyen, E. Dixon, C. Rodriguez, P. Miller, R. Lee, P. Shah, R. A. Ambrus, H. Nishimura, and M. Itkina. Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies. In *Robot Evaluation for the Real World*, 2025.

[23] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, Apr 2025. ISSN 1476-4687. doi:10.1038/s41586-025-08744-2.

[24] D. Hafner, W. Yan, and T. Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.

[25] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg. Daydreamer: World models for physical robot learning. In *6th Annual Conference on Robot Learning*, 2022.

[26] L. Grillotti, L. Coiffard, O. Pang, M. Faldor, and A. Cully. From tabula rasa to emergent abilities: Discovering robot skills via real-world unsupervised quality-diversity. In *Conference on Robot Learning*, pages 2519–2535. PMLR, 2025.

[27] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun. Navigation world models, 2025.

[28] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565. PMLR, 09–15 Jun 2019.

[29] C. Li, A. Krause, and M. Hutter. Robotic world model: A neural network simulator for robust policy optimization in robotics. *arXiv preprint arXiv:2501.10100*, 2025.

[30] D. Bogdoll, L. Bosch, T. Joseph, H. Gremmelmaier, Y. Yang, and J. M. Zöllner. Exploring the potential of world models for anomaly detection in autonomous driving. In *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 488–495. IEEE, 2023.

[31] A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2021.

[32] H. Sanghvi, S. Folk, and C. J. Taylor. Occam: Online continuous controller adaptation with meta-learned models. In *Conference on Robot Learning*, pages 25–47. PMLR, 2025. URL https://proceedings.mlr.press/v270/sanghvi25a.html.

[33] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450, 2024. doi:10.1109/ICRA57147.2024.10610200.

[34] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023.

[35] T. Lee, J. Kwon, P. M. Wensing, and F. C. Park. Robot model identification and learning: A modern perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 7, 2023.

[36] P. C. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

[37] M. Tan and Q. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.

[38] E. Alonso, A. Jelley, V. Micheli, A. Kanervisto, A. Storkey, T. Pearce, and F. Fleuret. Diffusion for world modeling: visual details matter in atari. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

[39] N. Hansen, H. Su, and X. Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024.

[40] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, page 413–422, USA, 2008. IEEE Computer Society. ISBN 9780769535029. doi:10.1109/ICDM.2008.17.

[41] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi:10.1109/IROS.2012.6386109.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[43] Q. Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

A AWARE Architecture

A.1 Robotic Platform

Our target platform is an under-actuated robot crane (Figure 3) with three actuated joints, $\mathcal{A} \in \mathbb{R}^3$, and four free joints, $\mathcal{S} \in \mathbb{R}^7$. The actuated joints are slew (q_1, revolute), rotating about the vertical axis of the crane base, luff (q_2, revolute), raising and lowering the boom, and hoist (q_5, prismatic), controlling cable length. The suspended payload is modeled with two 2D spherical joints at the cable endpoints: (q_3, q_4) at the boom captures pendulum swing and (q_6, q_7) at the payload captures double-pendulum effects when the payload is not aligned with the cable.

The action a_t is a velocity command for the slew, luff, and hoist joints. We simulate the system in MuJoCo and define a hidden dynamics parameter γ to capture motor properties and payload mass: for each motor, an inertial term (`armature` in MuJoCo) and a maximum velocity term (`ctrlrange` in MuJoCo), plus payload mass, yielding $\gamma \in \mathbb{R}^7$. We focus on the motors because they dominate system dynamics and the maximum velocity term can be accurately known on the real system during experiments given privileged access. This enables the analysis of decoded motor strength in $\hat{\gamma}_t$ against the ground truth.

The control architecture and hardware interface are implemented within ROS 2 Humble. Actuation commands and telemetry are interfaced with the crane's control bus through a dedicated Raspberry Pi 4, allowing for necessary signal reading and writing. The perception pipeline employs two Stereolabs cameras: a ZED X[1] mounted off the crane and a ZED 2i[2] rigidly attached to the crane structure. Both cameras interface with an NVIDIA Jetson AGX Orin, which in turn is connected to the host via Ethernet, managing state estimation and all primary processes within our ROS framework.

A.2 Model Training and Hyperparameters

The models were trained using data generated from a MuJoCo XLA simulation of the crane. To serve as a baseline and support analysis of true dynamics parameters, extensive fine-tuning of γ was conducted using motion capture data on the real crane with a grid search over the six motor parameters. The uncovered values represent the ground truth γ^*.

We then apply uniform domain randomization centered on γ^* with a randomized scale factor within the ranges $[\gamma_{min}, \gamma_{max}]$. The payload mass parameter was randomized absolutely. These wide ranges reflect the typical deployment case where fine-tuning is not possible, and there is high uncertainty. Table 2 summarizes the randomization scales across the dynamics parameters.

Training data was generated in MuJoCo using an exploratory policy which aimed to maximize coverage of the state action space. The policy would independently operate each actuated joint using one of four sub-policies: random, random-walk, sinusoid and no-motion with smoothing between action steps, using a timestep between observations of $t = 0.05$ s to reflect deployment at 20Hz. We model the real sensor noise distribution in simulation during training, combining gaussian noise with time correlated noise, and including artifacts such as repeated readings, zero error offsets, and sensor spikes. For velocity, we add noise arising from the differentiation of position. The dynamics parameters γ_t had only gaussian noise applied, and were randomly changed during trajectories. All models were trained using a single NVIDIA L40S 48GB GPU, taking under 48 hrs.

The models received $H = 64$ timesteps of history, and made predictions over $N = 20$ future timesteps during anomaly detection. The Transformer and RSSM world model backbones have 600,000 and 400,000 parameters respectively, with both latent dynamics estimators for the AWARE models having 500,000 parameters.

[1] https://www.stereolabs.com/en-gb/products/zed-x

[2] https://www.stereolabs.com/en-gb/products/zed-2i

Table 2: Parameters for Domain Randomization

Parameter	Config in MuJoco	Range
Motor Strength Scale Factor	`ctrlrange`	(0.5, 1.5)
Motor Rotor Inertia Scale Factor	`armature`	(0.1, 2.0)
Payload Mass / kg		(0.1, 2.0)

Table 3: Anomaly recognition results organized by anomaly decision input. Bold entries are statistically significant improvement over the best non-AWARE model. *AnyAnomaly is only evaluated on the nominal and payload collision set.

Model	Anomaly Decision Input	Motion Capture Quality State Estimation (Low Noise)		Camera Only State Estimation (High Noise)	
		Avg. Precision ± std	F1 Score ± std	Avg. Precision ± std	F1 Score ± std
Randomly Generated Signal	Raw sensor data	0.17 ± 0.03	0.26 ± 0.03	0.17 ± 0.03	0.26 ± 0.03
Isolation Forest [40]		0.29 ± 0.00	0.36 ± 0.01	0.16 ± 0.00	0.25 ± 0.01
Statistical Anomaly Detection		0.40 ± 0.01	0.40 ± 0.01	0.22 ± 0.01	0.31 ± 0.02
AESOP [6]		0.47 ± 0.00	0.56 ± 0.00	0.24 ± 0.00	0.41 ± 0.00
AnyAnomaly* [5]		0.37 ± 0.07	0.56 ± 0.00	0.37 ± 0.07	0.56 ± 0.00
MuJoCo [41]	World model prediction error, E_{pred}	0.77 ± 0.00	0.77 ± 0.00	0.45 ± 0.00	0.44 ± 0.00
RSSM [23]		0.86 ± 0.01	0.80 ± 0.01	0.53 ± 0.01	0.56 ± 0.01
Transformer [42]		0.84 ± 0.01	0.79 ± 0.01	0.41 ± 0.02	0.45 ± 0.01
RWM [29]		0.79 ± 0.06	0.73 ± 0.07	0.41 ± 0.03	0.44 ± 0.03
RSSM AWARE (ours)		0.90 ± 0.01	0.85 ± 0.01	0.59 ± 0.01	0.61 ± 0.01
Transformer AWARE (ours)		0.87 ± 0.03	0.82 ± 0.03	0.46 ± 0.02	0.50 ± 0.02
ROA [34]	Latent dynamics estimation error, E_{lat}	0.34 ± 0.06	0.39 ± 0.04	0.29 ± 0.04	0.35 ± 0.03
RWM [29]		0.75 ± 0.04	0.70 ± 0.04	0.53 ± 0.05	0.53 ± 0.03
RSSM AWARE (ours)		0.70 ± 0.02	0.63 ± 0.02	0.61 ± 0.02	0.59 ± 0.01
Transformer AWARE (ours)		0.76 ± 0.03	0.71 ± 0.03	0.69 ± 0.03	0.64 ± 0.02
RWM [29]	Prediction and dynamics errors, $[E_{pred}, E_{lat}]$	0.90 ± 0.03	0.85 ± 0.03	0.61 ± 0.05	0.59 ± 0.03
RSSM AWARE (ours)		0.91 ± 0.00	0.85 ± 0.01	0.76 ± 0.01	$\mathbf{0.69 \pm 0.01}$
Transformer AWARE (ours)		$\mathbf{0.93 \pm 0.01}$	$\mathbf{0.87 \pm 0.01}$	$\mathbf{0.77 \pm 0.01}$	0.68 ± 0.01

B Additional Results

B.1 Full Results Table

Table 3 shows the results of the anomaly detection experiment organized by the anomaly decision signal. The combination of prediction and dynamics errors, $[E_{pred}, E_{lat}]$ achieve the best results. E_{lat}-only methods demonstrate the best noise resilience when moving from low to high noise regimes with an average 17% drop across all models. Despite this, E_{lat} alone isn't sufficient for accurate anomaly detection, combining with E_{pred} gives AWARE models an average 18% AP improvement in the high noise setting.

B.2 Statistical Significance Tests

Table 4 reports two-sided Wilcoxon signed-rank tests over 10 paired seeds comparing **Transformer AWARE** and **RSSM AWARE**, with every baseline across four metrics (Motion Capture Quality State Estimation / Camera Only State Estimation $\times$ AP/F1). For each baseline, we apply a Bonferroni correction across these four tests, using $\alpha = 0.05/4 = 0.0125$. The results show strong evidence that our method outperforms the baselines: for nearly all baseline to metric pairs, the reported p-values take the minimum discrete value observed in our setting, $p = 0.00195$, indicating highly consistent per-seed improvements under both low-noise and high-noise state estimation. A small number of exceptions occur for Transformer AWARE against RSSM AWARE, where we observe weaker evidence on the high-noise metrics (e.g., $p = 0.02734$ and $p = 0.1934$), suggesting less

Table 4: Per-metric raw p-values from two-sided Wilcoxon signed-rank tests over 10 paired seeds. Significance is assessed within each baseline using a Bonferroni-corrected threshold over four metrics (two datasets $\times$ AP/F1), i.e., $\alpha = 0.05/4 = 0.0125$.

	Baseline	Motion Capture Quality (Low Noise)		Camera Only (High Noise)	
		AP (p)	F1 (p)	AP (p)	F1 (p)
Transformer AWARE	RSSM AWARE (double)	0.00195	0.00195	0.02734	0.19336
	RWM (double)	0.00195	0.01172	0.00195	0.00195
	Transformer AWARE (latent)	0.00195	0.00195	0.00195	0.00195
	RSSM AWARE (latent)	0.00195	0.00195	0.00195	0.00195
	RWM (latent)	0.00195	0.00195	0.00195	0.00195
	ROA (latent)	0.00195	0.00195	0.00195	0.00195
	Transformer AWARE (predictor)	0.00195	0.00195	0.00195	0.00195
	RSSM AWARE (predictor)	0.00195	0.00586	0.00195	0.00195
	RWM (predictor)	0.00195	0.00195	0.00195	0.00195
	Transformer (predictor)	0.00195	0.00195	0.00195	0.00195
	RSSM (predictor)	0.00195	0.00195	0.00195	0.00195
RSSM AWARE	Transformer AWARE (double)	0.00195	0.00195	0.02734	0.19336
	RWM (double)	0.01562	0.86523	0.00195	0.00195
	Transformer AWARE (latent)	0.00195	0.00195	0.00195	0.00195
	RSSM AWARE (latent)	0.00195	0.00195	0.00195	0.00195
	RWM (latent)	0.00195	0.00195	0.00195	0.00195
	ROA (latent)	0.00195	0.00195	0.00195	0.00195
	Transformer AWARE (predictor)	0.00195	0.01953	0.00195	0.00195
	RSSM AWARE (predictor)	0.00195	0.01367	0.00195	0.00195
	RWM (predictor)	0.00195	0.00195	0.00195	0.00195
	Transformer (predictor)	0.00195	0.00195	0.00195	0.00195
	RSSM (predictor)	0.00195	0.00195	0.00195	0.00195

consistent improvements under Camera Only State Estimation for RSSM AWARE. Similarly, for RSSM AWARE against RWM, we also observe weaker evidence on the F1 score metric in the low-noise setting ($p = 0.86523$).

B.3 World Model Prediction Error

We investigate the world model prediction accuracy by comparing foretasted predictions to the real world crane over a 100 step horizon, equivalent to 5 seconds, in Figure 7. We test initializing the model on both low and high noise data, with the errors being calculated in both cases using the low noise motion capture data of the crane. We find the AWARE enhanced world models perform better than their non-AWARE counterpart, demonstrating the impact of the encoded dynamics parameter latent in making accurate predictions. We additionally see that all the learned world models are more accurate than MuJoCo over a short horizon, showing our domain randomization procedure helps effectively cross the sim-to-real gap. The RSSM AWARE model is the strongest learned predictor and when conditioned on noisy data is better than MuJoCo over the full 100 step horizon, likely due to it's recurrent hidden state architecture favoring dynamics prediction, as seen in prior works [39, 28].

C Ablation Studies

C.1 AWARE Components

We ablate over the changes we make to the Real Time Online Adaptation method [34] for learning an latent dynamics estimator. We reintroduced each of three core changes we made from the original method: regularizing the encoder, removing the decoder and reintroducing a latent bottleneck. Figure 8 shows the training curves across the ablation over 1.5 training steps for the RSSM. We see a key improvement in AWARE is removing the latent bottleneck, suggesting a large latent space is

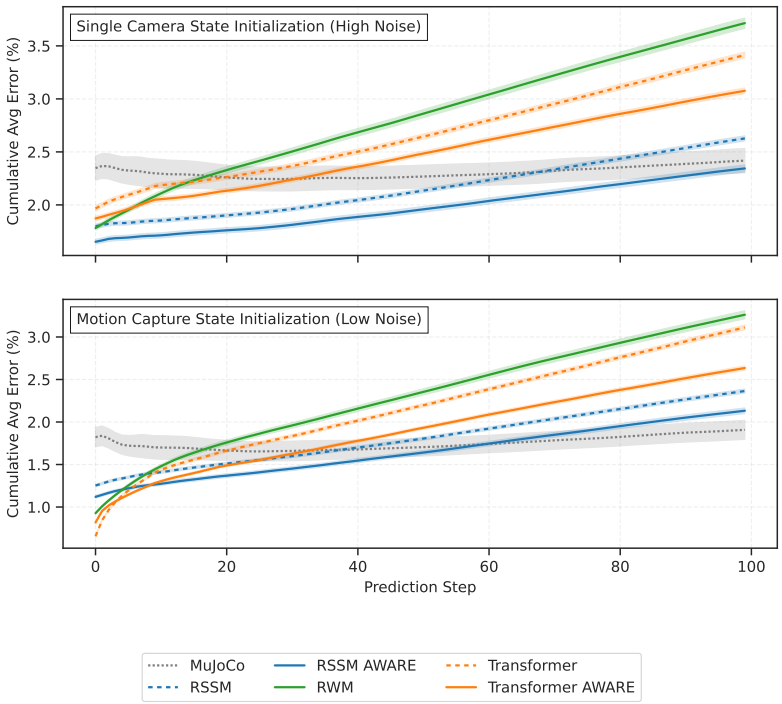


Figure 7: World model prediction error over a 100 step horizon. We condition the model on either Single Camera Quality Data or Motion Capture Quality Data, computing the error relative the Motion Capture data in both cases. The error is computed as a fraction of the joint's total range, averaged over all joint dimensions then as a running cumulative error averaged over all experiments.

important for encoding information useful for dynamics prediction. Encoder regularization, which is designed to improve the performance of the student latent dynamics estimator has a negligible effect. We argue removing it is therefore beneficial as it simplifies the method, requiring fewer parameters to tune. Removing the decoder sees little performance gain, coming at the cost of introspection.

C.2 Prediction Anomaly Score

We investigate the method discussed in subsection 3.4 to use the world model's predictions to detect anomalies by seeing if the model's state predictions vary from the observed predictions, aiming to detect changes in the unmodeled dynamics ζ. Table 5 shows these results for the RSSM AWARE method. We first test the value of encoding the dynamics parameters to a latent representation $\hat{z}$, as opposed to directly using parameters γ as in Robotic World Model [29]. We observed a significant performance degradation in anomaly detection when not encoding the parameters, suggesting we are able to learn a more useful latent representation of the parameters that enables better quality prediction. We then test not normalizing the model error by the variance predicted by the model (Equation 3). We hypothesized that this would reduce the contribution of errors to the anomaly score when the model is uncertain. We see marginal reduction under low noise data but more significant reduction under high noise data when the model is more likely to be uncertain given it has been conditioned on noise which supports our hypothesis. Finally, we test not conditioning the model on fixed median nominal latent $\hat{z}_0$, instead conditioning the model on the moving latent representation of the dynamics parameters $\hat{z}_t$. This significantly reduces performance, suggesting our model is better able to capture anomalies when solely conditioned on the nominal case, as opposed to using a potential anomalous latent.

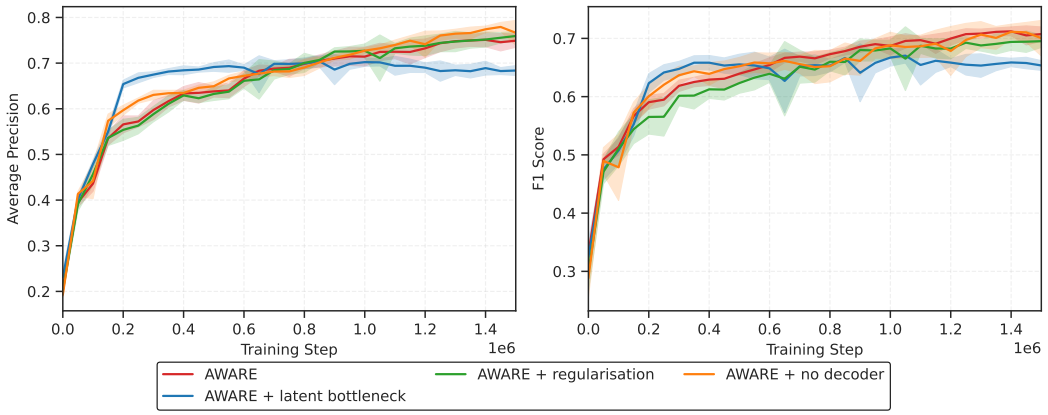


Figure 8: Training curves for RSSM AWARE. We ablate by reintroducing components from *Regularized Online Adaptation* [34] to AWARE, including: regularization of the target encoder that the latent dynamics estimator learns to follow; removing the latent decoder and reintroducing the latent bottleneck. We plot both average precision and F1 over 1.5 million training steps on the Single Camera Quality dataset.

Table 5: RSSM AWARE results using only the world model E_{pred} to detect anomalies. We ablate over the enhancements we make to improve predictor performance, including learning a latent representation, $\hat{z}$, of the hidden dynamics parameter, γ; conditioning the world model on the fixed median nominal latent, $\hat{z}_0$; and normalizing the computed error by the world model's variance.

Encode dynamics param to $\hat{z}_t$	Dynamics latents fixed to median $\hat{z}_0$	Error normalized by variance σ_t	Motion Capture Quality State Estimation (Low Noise)		Single Camera Quality State Estimation (High Noise)	
			AP	F1	AP	F1
	✓	✓	0.71 ± 0.02	0.65 ± 0.02	0.50 ± 0.01	0.54 ± 0.01
✓	✓		0.88 ± 0.01	0.83 ± 0.01	0.55 ± 0.01	0.58 ± 0.01
✓		✓	0.81 ± 0.01	0.75 ± 0.01	0.50 ± 0.01	0.54 ± 0.01
✓	✓	✓	0.90 ± 0.01	0.85 ± 0.01	0.59 ± 0.01	0.61 ± 0.01

C.3 Foundation Models

Foundation models are promising for anomaly detection under passive state estimation as they aim to provide a semantic level understanding of the scene and thus not requiring a noisy estimate of the robot state. We test two methods: LLM based AESOP [6] and VLM based AnyAnomaly [5] on our robotic crane dataset.

AESOP. AESOP use a small LLM conditioned on a prompt that describes the task overview, observations [6]. For our dataset, we use the estimated joint states and actions as observation inputs that provide dynamics context of the scene. The language model then outputs an latent embedding which is used to compare with a precomputed reference via a same Mahalanobis distance measurement. In our experiments, we note that the model is struggling to encode differentiated latents from raw observations, as it provides limited dynamic semantics. To resolve this, we translate observations into qualitative text descriptions before filling the prompt templates. Joint velocities are mapped into five categorical linguistic bins("very slow", "slow", "moderate", "fast", and "very fast"), defined by the percentile ranges computed across the nominal training dataset. For double pendulum joints (q_3, q_4, q_6, q_7), we compute the L2 norm and then use it for categorization. We provide an example prompt as follows:

Example Prompt

```
Task: A crane is transporting a payload to the target drop zone.
Observations: slew: commanded hard left, slewing left slowly; luff:
commanded down, luffing down slowly; hoist: no hoist command, hoist
stationary; boom tip swinging slowly; payload stable.
```

The anomaly detection results are compared in Table 1. To explain why AESOP underperforms (-46%) relative to AWARE, we propose a two-fold argument. Firstly, AESOP's capacity is capped by the use of a lightweight language model (MPNet-110M) mandated by real-time inference constraints. Crucially, however, the model encodes textual descriptions rather than visual data. The reliance on a reduced representation results in a substantial loss of the rich semantics inherent in the original raw images.

AnyAnomaly. AnyAnomaly is a zero-shot anomaly detection method that uses a VLM and guided prompt to determine if an anomaly specified in the prompt is occurring in the given batch of frames. The method operates temporally over a short sequence of 4 key frames. Given this limited window, evaluating the method on our motor degradation dataset, which requires long-horizon dynamics understanding, was out of the intended scope of this work. We therefore only choose to apply our payload collision dataset alongside the nominal cases. We utilize the official implementation and prompt template, filling in with our specified task:

Example Prompt

```
Task: Evaluate whether the given image includes a person manipulating the
red payload or the invisible string that it is vertically suspended from on a
scale from 0 to 1.
```

Using Qwen3-VL-8B-Instruct [43] as the VLM backbone, AnyAnomaly achieves AP of 0.37 and F1 of 0.56. Whilst being the only model that is designed for external camera state estimation, it struggled on our dataset with a high false negative rate ($21/40$). We found the VLM paradigm faced two core challenges in this domain: spatial misidentification resulting in background objects being identified as the crane and a lack of precise physical scene understanding, resulting in some subtle visible manipulations not to be flagged by the model. Furthermore, we found performance to be highly sensitive to prompt phrasing, requiring multiple iterations to achieve reasonable results. These limitations highlight the motivation behind AWARE. Explicitly learning the temporal dynamics enables higher reliability and sensitivity to anomalies than semantic visual interpretation.

D Anomaly Recognition

D.1 Measurement Signal Processing

We calculate the anomaly measurement by computing the Mahalanobis distance of the prediction error $E_{i,t}$ against reference distribution $\mathcal{P}_{ref} = (\mu_i, \sum_i)_{ref}$:

$$M_{i,t} = \sqrt{(E_{i,t} - \mu_{ref})^T \sum^{-1} (E_{i,t} - \mu_{ref})} \tag{4}$$

where $i = lat, pred$.

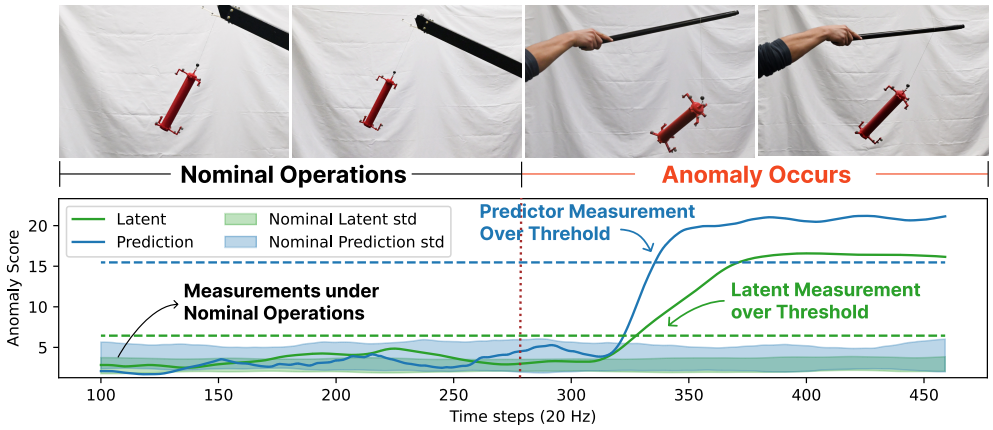
The Mahalanobis distance is appealing, being statistically justified, versatile, and simple to implement. It provided the better performance even than hand-tuned normalization, likely due to accounting for the full covariance matrix. As shown in Figure 6 and Table 1, the latent signal is general more stable and more resilient to noise that the world model prediction error, as it takes a longer term view and aims to uncover the pattern rather than analyzing short horizon behavior. It therefore follows that long term degradations, such as motor impairment, are well detected by the latent signal, whereas, short term instantaneous events, such as the payload collisions, are best detected by the world model error.

Table 6 justifies our decision to use the Mahalanobis distance. We test multiple distance measures to our reference distribution and different operators to combine our two anomaly signals E_{pred} and E_{lat}. Notably, we see significant performance increase when moving from L1/L2 measurements to L2 standardized and Mahalanbois measures. The demonstrates the importance of normalizing by the diagonal of the covariance matrix and reduces noise from scale invariance between the features. We see a marginal improvement moving from L2 standardized to Mahalanbois, suggesting the impact of using the full covariance matrix (and therefore accounting for feature correlations) is small.

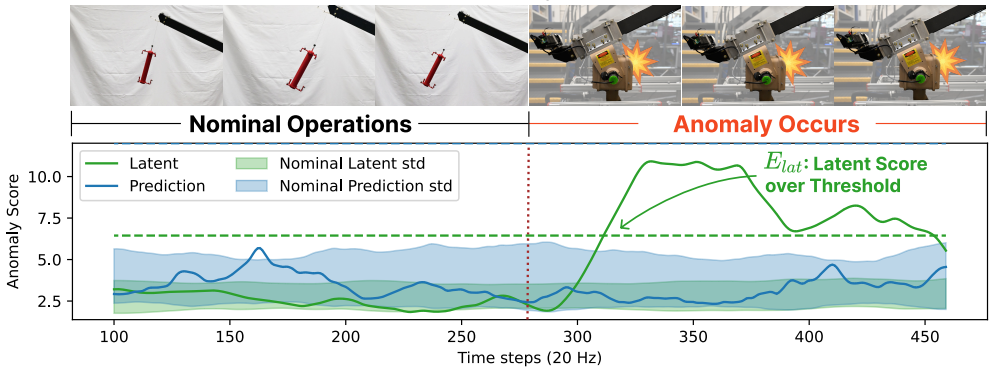
Figure 4 compares the effectiveness of E_{lat} and E_{pred} across all the anomaly cases and shows how each of the two measurement signals in the AWARE model, from the world model error (predictor) and the latent estimation (latent), contributed to solving the anomaly recognition task. It illustrates that E_{lat} became less reliable out-of-distribution. The number of cases correctly detected by E_{lat} as

Table 6: Anomaly detection results averaged across AWARE methods using different distance measures to the nominal reference distribution and operators to combine the prediction and latent signal E_{pred} and E_{lat}. 'L1' and 'L2' use the Manhattan and Euclidean distance to the reference distribution mean. 'L2 standardized' normalizes the L2 distance by the diagonal covariance.

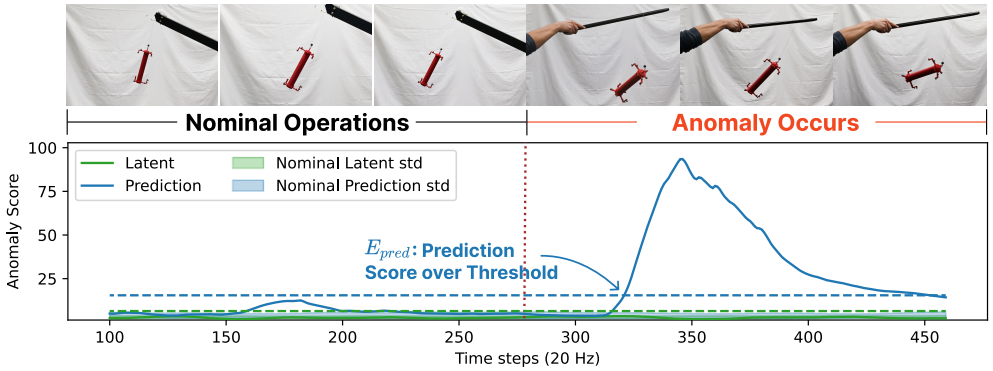
Distance Measure	Operator	Motion Capture Quality State Estimation (Low Noise)		Camera Only State Estimation (High Noise)	
		Avg. Precision ± std	F1 Score ± std	Avg. Precision ± std	F1 Score ± std
L1	OR	0.86 ± 0.02	0.80 ± 0.02	0.69 ± 0.03	0.63 ± 0.04
	AND	0.78 ± 0.04	0.72 ± 0.03	0.66 ± 0.01	0.59 ± 0.04
	SUM	0.78 ± 0.09	0.72 ± 0.07	0.50 ± 0.04	0.55 ± 0.04
L2	OR	0.83 ± 0.03	0.78 ± 0.02	0.67 ± 0.04	0.60 ± 0.04
	AND	0.75 ± 0.05	0.71 ± 0.03	0.65 ± 0.02	0.59 ± 0.04
	SUM	0.64 ± 0.18	0.61 ± 0.14	0.42 ± 0.04	0.49 ± 0.06
L2 standardized	OR	0.92 ± 0.02	0.86 ± 0.03	0.75 ± 0.03	0.67 ± 0.02
	AND	0.88 ± 0.02	0.83 ± 0.02	0.69 ± 0.02	0.61 ± 0.03
	SUM	0.91 ± 0.01	0.86 ± 0.02	0.63 ± 0.03	0.66 ± 0.03
Mahalanobis	OR	**0.92 ± 0.01**	0.86 ± 0.02	**0.77 ± 0.02**	**0.68 ± 0.01**
	AND	0.88 ± 0.02	0.83 ± 0.03	0.70 ± 0.02	0.62 ± 0.03
	SUM	0.92 ± 0.01	**0.87 ± 0.01**	0.66 ± 0.02	0.67 ± 0.02



(a) Both Caught



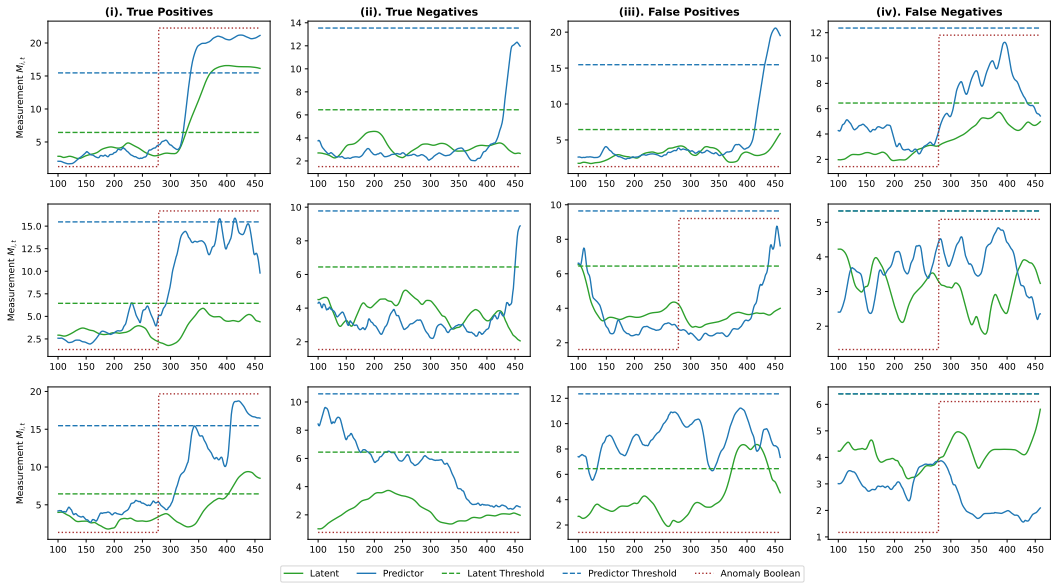
(b) Latent Caught



(c) Predictor Caught

Figure 9: Examples of Anomaly Recognition by a pivoting measurement signal, where a successful detection is classified as either measurement signal exceeds its corresponding threshold. Nominal std gives the standard deviation of both measurement signals during nominal operations across all environments.

motor strength reduces from $25\% \rightarrow 0\%$ decreases, even though 0% represents the more significant anomaly. In contrast, E_{pred} continuously detected more cases as the amount of motor degradation increased, and was far more effective at detecting payload anomalies. These results support our hypothesis that E_{lat} detects changes in modeled dynamics, γ, whilst E_{pred} detects changes in the unmodeled dynamics, ζ. This effect is even more significant in the low noise setting.



(a) Motion Capture Quality Estimation



(b) Camera Only Estimation

Figure 10: Examples of predictor and latent-estimator measurement signals with and without anomalies. (a) *Motion Capture Quality Estimation* (low noise); (b) *Camera Only Estimation* (high noise). From left to right, the four columns correspond to True Positive, True Negative, False Positive, and False Negative cases (based on whether an anomaly was present and whether the model raised an alarm). Solid blue and green curves show the predictor and latent-estimator signals, respectively, while dashed blue and green lines denote their anomaly thresholds. The red dashed vertical line marks the time at which the anomaly is injected.