PULSE: PROTOTYPE-BASED LATENT SURPRISE DETECTS PHYSICS ANOMALIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics anomalies are events in which a system stops following its usual physical dynamics. Such anomalies appear normal in a single frame and only become apparent through the temporal evolution of the scene. One approach is to use a world model to predict how a scene evolves and measure surprise when the system deviates from the prediction. However, such predictive approaches can assign high surprise to any unexpected change, whether or not the change is a deviation from *normal* dynamics. We propose PULSE, which learns a system's normal dynamics by training a model on normal videos in the feature space of a frozen encoder and scores frames using the resulting latent surprise. Feature-space world models typically regress future features, but when several futures are normal, regression can average them into a single intermediate representation, causing normal outcomes to receive high surprise scores. PULSE instead clusters normal features into prototypes and assigns a probability to each. These prototypes provide explicit normal references for anomaly scoring, while the predicted probabilities preserve distinct plausible futures. PULSE sets a new state of the art with the highest average AUROC on three physics anomaly benchmarks that cover real-world object interactions, industrial processes and video games, achieving 98.6 on Phys-AD, 76.5 on IPAD and 72.8 on TempGlitch. Our ablations show that using prototype-based prediction provides the largest methodological improvement at +6.3 AUROC points.

1 INTRODUCTION

The goal of this paper is to recognize *physics anomalies*: events in which a system departs from how it normally moves and changes. A hinge may stop short of its full opening angle (Li et al., 2025), an item on a conveyor belt may drift off-center (Liu et al., 2024), or a video game character may keep running without moving because of a software bug (Yu et al., 2026). In each case, what counts as normal is set by the system itself: the hinge should open fully, the conveyor should keep items centered, and the character should move as the game is programmed, even where the game's rules differ from real-world physics. Video world models, which are increasingly deployed to predict and plan in physical environments (Assran et al., 2025; Baldassarre et al., 2025), seem a natural fit for this task, since an anomaly should surprise a model that knows what comes next. However, even a perfect world model is also surprised by changes that are not anomalies. On a production line, for example, a change of lighting and a machine that runs slightly slower may both be unexpected, and only the slowdown is an anomaly. Recognizing physics anomalies therefore requires learning which changes from the system to ignore and which to flag.

Identifying physics anomalies is inherently challenging as a scene can appear plausible in isolation, abnormality emerges from their relationship to the *expected physical dynamics* of their preceding context. Hence, predicting the expected future based on past context, and measuring the deviation is one approach to detecting such anomalies. Traditional video anomaly detection approaches (Gong et al., 2019; Liu et al., 2023; Lv et al., 2021) use pixel space prediction. This representation is not well suited to physics anomalies, as it is sensitive to low-level background and appearance variations. Instead, we take inspiration from the feature space prediction approaches of world modeling which capture high level state changes. Several works in this paradigm assess physical plausibility using *latent surprise* (Bordes et al., 2025; Garrido et al., 2025): measuring the deviation between an observed outcome and its predicted future feature representation. However, while general-purpose

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.

representation learning and physics motivates world modeling, our objective is anomaly detection. Here, anomalies are related to deviations from the normal physics of a specific system.

Latent world models are usually trained by regression, which tends to average the possible futures (Boduljak et al., 2026; Porcher et al., 2026). If a ball may roll either left or right, a regressor predicts a blend of both outcomes that matches neither, and the normal future looks surprising. Our key insight is that surprise should be measured against normal states that the system actually visits. PULSE therefore trains a small world model on normal videos of the target system, in the feature space of a frozen encoder. It clusters these features into prototypes and predicts a probability for each prototype instead of regressing the features. The latent surprise of an observed patch is the negative log-probability of its assigned prototype. Prototype prediction is our key contribution, it improves over regression with 6.3 AUROC points on the challenging IPAD dataset. PULSE further achieves the highest average AUROC on Phys-AD (98.6), TempGlitch (72.8) and IPAD (76.5), with 1.9 to 8.9 points above the best prior supervised, zero-shot and unsupervised methods.

2 Related Work

Video anomaly detection. Unsupervised video anomaly detection learns regularities from normal training videos and identifies events that deviate from these learned patterns at test time. Early approaches predominantly model normality through reconstruction, using reconstruction error as an anomaly signal (Hasan et al., 2016). Subsequent methods improve this paradigm by restricting reconstruction to normal patterns through a memory of normal features, as in MemAE (Gong et al., 2019), MNAD (Park et al., 2020) and MPN (Lv et al., 2021), or by explicitly modeling the diversity of normal observations, as in DMAD (Liu et al., 2023). An alternative line of work formulates anomaly detection as future prediction: Liu et al. (2018) predict the next video frame from preceding observations and regard disagreement between the predicted and observed future as evidence of abnormality. Later approaches combine prediction with reconstruction or motion modeling to better capture temporal regularities (Liu et al., 2021). Prediction in feature space (Yan et al., 2023; Lee et al., 2026) and over discrete targets (Abati et al., 2019; Liu et al., 2026) has also been explored, but the former predict continuous features and the latter discretize a latent space they learn jointly with the predictor. PULSE instead predicts a distribution over k-means prototypes of frozen normal features, which act as a fixed memory of normal states, following self-supervised methods that train on cluster assignments (Caron et al., 2018; 2020), with centered, sharpened soft targets as in Caron et al. (2021).

Industrial video anomalies. Much of the recent progress in video anomaly detection targets human-centred benchmarks such as UCF-Crime (Sultani et al., 2018), ShanghaiTech (Luo et al., 2017) and Avenue (Lu et al., 2013), and hence successful methods rely on pose estimators and object detectors (Lee et al., 2026; Liu et al., 2026). The former assumes people, and the latter locates an object but says nothing about how it moves, such as a fan spinning at the wrong speed. Phys-AD (Li et al., 2025) and IPAD (Liu et al., 2024) poses this setting as benchmarks, and the methods that report on them take two approaches. IPAD (Liu et al., 2024) and INFP (Yu et al., 2025) build periodicity into the architecture. O-VAD (Yuan et al., 2026) and MT-PIVLM (Gu et al., 2026) adapt vision-language models such as Qwen3-VL (Bai et al., 2025) to anomaly detection. On an earlier industrial dataset, Kart & Cohen (2022) score a video by its nearest-neighbour distance to normal videos in frozen features. PULSE also builds on frozen features, but learns the normal dynamics of the scene with a small predictor trained on normal video, and needs no detector, prior, or language.

Latent prediction and world models. V-JEPA learns video representations by predicting masked or future content in feature space rather than in pixels (Bardes et al., 2024; Assran et al., 2025), and DINO-Foresight and DINO-World do the same on top of a frozen DINOv2 encoder (Karypidis et al., 2025; Baldassarre et al., 2025). Garrido et al. (2025) report that the prediction error of these models is larger on physically impossible videos (Garrido et al., 2025; Bordes et al., 2025). These works test whether a video is physically possible, whereas we test whether it is normal for a particular machine, process or game. PULSE keeps the same design but trains the predictor on a scene's normal videos, so its error measures how far the scene departs from its usual dynamics.

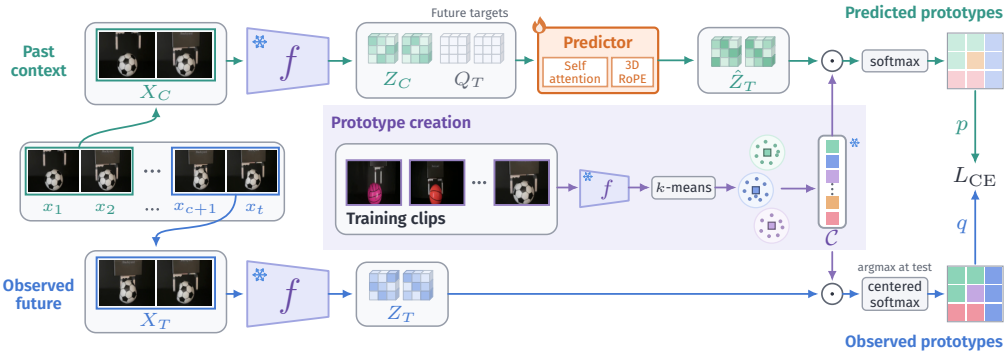


Figure 1: **Overview of PULSE.** Given a video clip, context and future frames are encoded by a frozen encoder f into Z_C and Z_T. Patch tokens from normal training clips are clustered into a set of prototypes C representing normal features. From the context Z_C and future queries Q_T a predictor outputs future features $\hat{Z}_T$. Predicted and observed features are mapped to a distribution over prototypes. The predictor is trained by a cross-entropy loss between the predicted and observed prototype distributions. Their discrepancy gives a latent surprise score.

3 METHOD

Our goal is to train a predictor that learns a *categorical distribution* over plausible future feature states, called *prototypes*. This distribution captures the normal dynamics observed in the training videos. At test time, physics anomalies produce latent surprise when the observed future is unlikely under the predicted distribution. To learn this prediction task, as shown by Figure 1, we sample video clips and divide each clip into context and target frames. We encode the frames densely using a pretrained vision foundation model to preserve spatial information. We then cluster the resulting features to construct high-level prototypes that represent common patterns in the feature space, corresponding to visual content such as backgrounds, objects, and object parts. Finally, we train a predictor to use the context features to predict the target frames in this prototype space. We call our method PULSE: **P**rototype-Based **U**nsupervised **L**atent **S**urprise **E**stimation for Physics Anomaly Detection. We describe each component in detail below.

3.1 FEATURE EXTRACTION AND PROTOTYPE CONSTRUCTION

Given a video clip $X = (x_1, \ldots, x_{c+t})$, where x_i denotes the frame at time step i, we divide it into a context clip $X_C = (x_1, \ldots, x_c)$ and a target clip $X_T = (x_{c+1}, \ldots, x_{c+t})$. A frozen visual encoder f independently maps each frame x_i to N patch features of dimension D:

$$f(x_i) = Z_i \in \mathbb{R}^{N \times D}. \tag{1}$$

We stack the frame-level features to form the context and target representations:

$$Z_C = [Z_1; \ldots; Z_c] \in \mathbb{R}^{c \times N \times D}, \qquad Z_T = [Z_{c+1}; \ldots; Z_{c+t}] \in \mathbb{R}^{t \times N \times D}. \tag{2}$$

Frames are encoded independently so that features of the context frames cannot contain information from future frames. This yields temporally isolated context and target features, which we next organize into a prototype space of normal states.

To obtain prototypes, we encode all training frames using a frozen encoder f, and run k-means on the resulting features, giving K prototypes $\mathcal{C} = \{c_1, \ldots, c_K\}$. The encoder and prototypes are frozen during training so that the normal feature space remains fixed while the predictor learns future dynamics.

3.2 FUTURE PROTOTYPE PREDICTION

Direct regression of features can collapse distinct plausible futures into a single continuous prediction, from which deviations are ambiguous as an anomaly signal. Instead, we propose to train with a prototype objective and evaluate predictions relative to multiple representative normal feature states.

As our predictor g_ϕ we use a transformer whose blocks receive Z_C along with a learnable query token $Q_T \in \mathbb{R}^{t \times N \times D}$. The transformer processes context features and future queries jointly using

self attention (Dosovitskiy et al., 2020): $\hat{Z}_T = g_\phi(Z_C; Q_T) \in \mathbb{R}^{t\times N\times D}$ where $\hat{Z}_T$ denotes the outputs corresponding to the future queries. The spatial and temporal location of context and queries is encoded by 3D axial rotary positional embeddings (Su et al., 2024). This allows the predictor to jointly model context and future using the spatial and temporal positions of both. Encoding spatio-temporal position is important for physics anomalies which are related to deformation and discontinuity in both.

For any feature vector z, we define its affinity to prototype c_k as the logit $\ell_k(z) = -\|z - c_k\|_2^2$. For each predicted future feature $\hat{z}_t$ and observed target feature z_t, the predicted and target distributions over K prototypes are:

$$p(k \mid \hat{z}_t) = \underset{k}{\text{softmax}}\left(\ell_k(\hat{z}_t)/\tau_p\right), \qquad q(k \mid z_t) = \underset{k}{\text{softmax}}\left((\ell_k(z_t) - \mu_k)/\tau_q\right). \tag{3}$$

where p and q are the predicted and target prototype distributions with softmax temperatures τ_p and τ_q.

If the target is less concentrated than the prediction, all prototypes get equal distribution, but that would make predictions less decisive about specific prototypes that come next. Hence we use $\tau_q < \tau_p$, so that the target assignment is more concentrated than the prediction. To avoid frequently occurring prototypes from dominating the target distribution, we use centering with μ_k, a running mean of the logit for prototype k (Caron et al., 2021). The predictor is optimized with cross-entropy over all $t \times N$ future features,

$$\mathcal{L} = -\frac{1}{t\times N}\sum_{i=1}^{t\times N}\sum_{k=1}^{K} q(k \mid z_i)\,\log p(k \mid \hat{z}_i). \tag{4}$$

The predictor thus learns which normal prototype assignments are expected from context, which we use next to score the surprise of observed futures.

3.3 Latent Surprise Scoring

At inference, the observed future is realized as a single feature rather than a distribution. To reflect this, we assign each observed future feature z_t to a single prototype using a hard version of Equation 3. We fix the learned centering statistics μ, and find the probability assigned to that prototype by:

$$k_t^\star = \arg\max_k \left[\ell_k(z_t) - \mu_k\right], \qquad a_t = -\log p_\phi(k_t^\star \mid \hat{z}_t). \tag{5}$$

where a_t is the surprise score of future token t. This assigns low scores to future distributions the model expects, while unexpected futures receive high scores. In order to obtain frame-level scores, patch feature scores are averaged spatially. As a frame may appear as a prediction target in multiple clips, we average its anomaly scores across these predictions to obtain a single frame-level score. To obtain video-level scores, we simply average frame level scores. Formally, let $\mathcal{W}(i)$ denote the set of clips that predict frame i, and let $a_{w,i,n}$ be the score of patch n for frame i in clip w. For a video with T scored frames,

$$\text{Score}_{\text{frame}}(i) = \frac{1}{|\mathcal{W}(i)|}\sum_{w\in\mathcal{W}(i)}\frac{1}{N}\sum_{n=1}^{N} a_{w,i,n}, \qquad \text{Score}_{\text{video}} = \frac{1}{T}\sum_{i=1}^{T}\text{Score}_{\text{frame}}(i). \tag{6}$$

These aggregations convert local patch-level surprise into frame- and video-level anomaly scores.

4 Experiments

4.1 Setup

Datasets. We evaluate our method on three video anomaly detection benchmarks containing physics and temporal anomalies. ***Phys-AD*** (Li et al., 2025) contains 6,400 videos of motors and grippers interacting with 22 objects, spanning 47 anomaly types including abnormal motion, collisions, and deformation. ***TempGlitch*** (Yu et al., 2026) contains 1,500 gameplay videos with five simulated glitch types: blinking, shooting error, velocity bug, stuck-in-place, and frozen animation. ***IPAD*** (Liu et al., 2024) contains 1,068 clips from 12 simulated and 4 real-world industrial devices. IPAD evaluates anomalies in structured, repetitive industrial processes, where deviations may affect motion, timing,

Table 1: **Phys-AD results** in Video-level AUROC (↑). Best in **bold**, second-best underline. † indicates our evaluation of author-provided code. PULSE outperforms all unsupervised and zero-shot alternatives, and even improves over the supervised MT-PIVLM by +1.9.

	Zero-shot			Supervised	Unsupervised				
	JEPA IntPhys†	Qwen3-VL-32B†	O-VAD	MT-PIVLM	MNAD.r	MemAE	DMAD	DINO-Foresight†	**Ours**
Car	58.5	49.2	57.5	99.9	94.4	52.3	38.1	**100.0**	**100.0**
Fan	70.1	56.6	66.9	91.2	54.2	37.1	36.5	59.1	**95.7**
Rolling Bearing	98.1	41.8	65.0	**100.0**	80.0	4.4	89.8	**100.0**	**100.0**
Spherical Bearing	95.3	48.3	64.1	**100.0**	81.3	9.2	64.9	99.8	**100.0**
Servo	23.1	49.2	49.5	92.6	87.8	44.5	96.9	95.9	**100.0**
Clip	0.8	44.2	42.3	97.4	33.3	44.3	56.0	68.3	**100.0**
U Disk	43.8	57.8	60.6	96.4	94.0	61.7	78.0	94.9	**100.0**
Hinge	50.1	58.9	52.6	81.8	89.5	87.0	85.0	**100.0**	**100.0**
Sticky Roller	67.1	53.3	81.1	**100.0**	93.6	96.7	97.8	**100.0**	**100.0**
Caster Wheel	61.2	53.3	30.1	**100.0**	50.8	52.3	71.6	**100.0**	**100.0**
Screw	52.7	45.0	70.4	98.1	54.7	56.7	70.9	**100.0**	**100.0**
Lock	26.0	38.8	78.0	**100.0**	64.1	59.7	63.7	88.6	**100.0**
Gear	64.4	64.4	87.9	99.1	69.4	51.0	83.5	82.1	**99.9**
Clock	36.9	58.1	66.9	**88.6**	58.7	54.2	62.3	68.6	80.5
Slide	26.9	52.9	41.5	**100.0**	78.4	96.2	99.3	**100.0**	**100.0**
Zipper	66.3	51.7	38.8	94.5	42.1	59.2	64.6	92.2	**100.0**
Button	14.4	50.6	39.3	98.7	56.8	36.5	88.2	65.2	**100.0**
Liquid	2.8	83.3	77.6	**100.0**	83.1	70.0	91.1	**100.0**	**100.0**
Rubber Band	51.8	76.7	72.1	95.8	30.7	36.8	89.7	96.2	**96.7**
Ball	78.2	52.8	57.9	97.3	68.7	38.3	82.9	79.1	**100.0**
Magnet	82.1	59.2	57.6	**100.0**	43.8	46.4	40.3	95.7	96.4
Toothpaste	50.8	57.8	68.9	–	63.1	88.9	58.3	81.6	**99.7**
Average	51.0	54.7	58.4	96.7	66.9	53.8	73.1	89.4	**98.6**

or logic (such as color-based sorting). These datasets provide complementary evaluations: Phys-AD and IPAD span diverse objects and physics interactions, while TempGlitch contains game bugs in simulated gameplay. We use the official train/test splits for Phys-AD and IPAD. TempGlitch does not provide an official split, so we use 80% of the normal videos for training and evaluate on the remaining 20% together with all anomalous videos. We do not evaluate on human-centred benchmarks such as UCF-Crime, ShanghaiTech and Avenue, whose anomalies are semantic events such as fights or intrusions rather than deviations in how a system moves.

Evaluation. We report video-level and frame-level AUROC by comparing surprise scores per frame and video against the ground-truth binary anomaly labels. Video-level AUROC measures discrimination between normal and anomalous videos, while frame-level AUROC measures anomaly localization within a video. For IPAD, we report frame-level AUROC only as it provides frame-level annotations.

Baselines. We compare against state-of-the-art physics anomaly detection methods, including pixel prediction methods IPAD (Liu et al., 2024) and INFP (Yu et al., 2025) and the VLM-based methods O-VAD (Yuan et al., 2026) and MT-PIVLM (Gu et al., 2026). We additionally compare with future latent prediction methods, which are closest to our approach: DINO-Foresight (Karypidis et al., 2025) and zero-shot JEPA Intuitive Physics (Garrido et al., 2025). We train and evaluate DINO-Foresight on our datasets. Finally, we compare against established state of the art general VLMs and video anomaly detection methods DMAD (Liu et al., 2023), MemAE (Gong et al., 2019) and Qwen3 (Bai et al., 2025). Results are taken from the original papers where available; otherwise, we evaluate released implementations under each benchmark's protocol, with VLM prompts following Phys-AD (Li et al., 2025) and TempGlitch (Yu et al., 2026). Additional baselines and prompt details are reported in Appendix section 8.

Training details. Following the unsupervised setting, we train only on normal videos. The frozen encoder is DINOv2 ViT-S/14, and the predictor is a transformer of 4 blocks 6 heads, which predicts a distribution over $K{=}32,768$ prototypes. Inputs are $224{\times}224$ on IPAD and $448{\times}448$ on Phys-AD and TempGlitch. We use a learning rate of $1\mathrm{e}{-5}$ on IPAD and Phys-AD and $3\mathrm{e}{-5}$ on TempGlitch, and train for 120 epochs on all three datasets. At native frame rates, consecutive frames are nearly identical, so predicting the next one is trivially learnable by copying the current frame. We therefore sample at lower frame rates of 15 fps for Phys-AD, IPAD ∼6 fps and 5 fps on TempGlitch.



Figure 2: **Qualitative examples demonstrating the effect of latent surprise.** We visualize latent surprise for a blinking anomaly from TempGlitch, in which the large vehicle in the upper-left region disappears and reappears. Heatmaps overlay the per-token latent surprise on the observed frames. Brighter colors indicate higher surprise. While JEPA IntPhys produces a scattered response, PULSE concentrates surprise near the affected object during disappearance and reappearance.

Table 2: **TempGlitch results** in Video-level AUROC (↑). Best in **bold**, second-best underline. † indicates our evaluation of author-provided code. PULSE improves over all alternatives, including the zero-shot JEPA IntPhys by +8.9. It captures speed-change anomalies particularly well.

	Zero-shot		Unsupervised						
	JEPA IntPhys†	Qwen3-VL-32B†	MNAD.r†	MPN†	MemAE†	SVM†	DMAD†	DINO-Foresight†	**Ours**
Blinking	74.4	52.9	49.5	34.1	39.0	52.7	27.9	58.6	**76.8**
Frozen animation	45.5	49.7	65.1	45.6	41.3	51.7	63.8	60.6	**72.6**
Shooting error	57.7	48.6	41.8	50.5	**71.3**	50.7	66.2	60.7	62.5
Stuck in place	**72.6**	48.3	40.2	67.5	45.3	47.3	44.9	35.3	61.1
Velocity bug	69.5	48.1	38.4	40.1	46.9	49.3	54.1	63.5	**90.9**
Average	63.9	49.5	47.0	47.6	48.8	50.3	51.4	55.7	**72.8**

4.2 Benchmark Results

Phys-AD results. We provide results for Phys-AD in Table 1. PULSE achieves 98.6 video-level AUROC and obtains the highest per-class performance on 20 of 22 object categories, a +1.9 AUROC improvement over the strongest *supervised* VLM MT-PIVLM (96.7) and +9.2 gain over future-prediction baseline DINO-Foresight (89.4). PULSE achieves near-perfect AUROC on anomalies with mechanical dynamics (bearings, servos, screws, gears) and deformation/flow (liquid, sticky roller, toothpaste). On subtle small rotational and velocity-related anomalies (lock, fan, gear), PULSE improves by between +11.4 and +36.6 AUROC on the closest future-prediction baseline DINO-Foresight, which uses the same DINOv2 encoder family as PULSE. PULSE is weakest on Clock (80.5), where the supervised MT-PIVLM (88.6) is best. Clock anomalies are a pointer that stops or vibrates (Li et al., 2025); a stopped pointer changes little between frames and can look like a normal one, a limitation we discuss in section 5. Full per-baseline comparisons are in Appendix Table 6 and Appendix Table 7.

TempGlitch results. We report results for TempGlitch in Table 2. PULSE achieves 72.8 average AUROC, improving over the strongest baseline, JEPA IntPhys, by 8.9 points, and leads on three of five anomaly types. Its largest gain is on velocity bug (+21.4 points over JEPA IntPhys), followed by frozen animation (+7.5 over MNAD.r) and blinking (+2.4 over JEPA IntPhys). These results support the effectiveness of prototype-based prediction for detecting deviations in motion and temporal continuity. Figure 2 illustrates this behavior on a blinking anomaly. In the remaining categories, PULSE is worse on incorrect projectile trajectory errors (-8.8 compared to MemAE) and stuck in place (-11.5 compared to JEPA IntPhys). Incorrect projectile trajectories may produce brief, localized changes that reconstruction-based scoring in MemAE preserves more effectively than the

Table 3: **IPAD results** in AUROC (↑). Best results are in **bold**, second best underline. † indicates our evaluation of author-provided code. PULSE improves over all methods, including INFP designed for industrial anomaly detection.

	Zero-shot		Unsupervised						
	O-VAD	JEPA IntPhys†	MNAD.r†	DINO-Foresight†	MemAE	DMAD	IPAD	INFP	**Ours**
Synthetic cases									
S01	42.4	47.4	42.2	22.7	63.2	55.9	69.5	77.2	**78.8**
S02	44.4	60.2	54.1	28.7	50.6	55.3	**63.9**	62.8	60.0
S03	59.8	50.6	68.5	47.5	65.6	47.9	70.6	70.6	**92.0**
S04	37.0	30.5	51.5	16.9	49.5	47.9	58.3	56.0	**69.5**
S05	59.1	42.4	85.4	59.2	78.8	69.3	86.2	89.7	**98.2**
S06	41.1	34.2	51.0	27.5	45.9	61.0	61.2	**73.4**	57.3
S07	56.8	32.8	59.5	25.8	57.9	66.9	60.6	**78.1**	73.8
S08	50.0	36.9	75.7	34.4	84.7	87.5	85.6	**96.3**	81.4
S09	65.4	49.9	63.5	43.4	65.7	69.7	**71.2**	66.6	70.6
S10	43.8	43.1	78.0	40.5	59.9	67.0	62.2	71.9	**83.7**
S11	49.8	43.1	49.7	26.3	49.4	56.0	60.9	56.9	**69.5**
S12	61.1	36.6	**74.8**	27.3	50.7	55.8	67.1	60.9	71.3
Average	50.9	42.3	62.8	33.4	60.2	61.7	68.1	71.7	**75.5**
Real cases									
R01	59.5	40.2	86.7	81.5	77.9	79.5	84.4	86.5	**86.9**
R02	51.2	39.6	79.8	59.9	65.0	68.5	75.4	77.9	**94.4**
R03	54.5	46.1	**65.3**	47.7	41.6	43.1	43.5	48.4	56.4
R04	57.2	53.7	74.8	66.0	70.7	63.1	76.7	75.3	**80.0**
Average	55.6	44.9	76.6	63.8	63.8	63.6	70.0	72.0	**79.4**
Overall	51.8	43.0	66.3	41.0	61.1	62.2	68.6	71.8	**76.5**

spatial averaging of prototypes in PULSE. For stuck in place, the context consists entirely of stuck frames, which may become locally predictable despite remaining abnormal. Overall, PULSE captures diverse physics anomalies in TempGlitch using a general prototype-based prediction objective, without requiring anomaly-specific modeling for each category.

IPAD results. We report frame-AUROC for IPAD in Table 3. On IPAD, PULSE reaches 76.5 on average and ranks first on 9 of 16 devices, outperforming prior state of the art INFP by 4.7 points. Its strongest gains over each best competing method occur on S03 (+21.4), R02 (+14.6), and S04 (+11.2). PULSE also leads on three of the four real-world devices, achieving a real-world average of 79.4, compared with 76.6 for MNAD.r. These results demonstrate that prototype-based future prediction captures physics dynamics across both synthetic and real industrial settings.

PULSE shows its largest gains from broad-motion devices where anomalies occur over many feature patches at once (S03 (92.0 vs. IPAD 70.6, +21.4), R02 (94.4 vs. MNAD.r 79.8, +14.6), S05 (+8.5), S10 (+5.7). On classes with small localized temporal anomalies, such as an object slowing down briefly on a conveyor belt (S06, S07, and S08), INFP performs better. Its explicit frequency-domain modelling and weighting of motion trajectories may serve as strong inductive biases to expose deviations that are concentrated in small regions. R03, which contains forklift jitter, remains challenging for both prior state-of-the-art methods and PULSE, with AUROC scores of 43.5 for IPAD, 48.4 for INFP, and 56.4 for PULSE, highlighting the difficulty of detecting this type of motion anomaly. The gap between PULSE at 76.5 and other latent-prediction baselines (JEPA IntPhys 43.0, DINO Foresight 41.0) is large on IPAD, as Frame AUROC requires temporal localization of anomalies within a video. Continuous feature regression produces per-frame residuals that stay smooth across anomalous windows rather than spiking on the specific anomalous frames, as shown in Figure 3. Overall, PULSE maintains strong performance across both simulated and real-world industrial devices using a general future-prediction objective.

4.3 Ablations

In this section, we ablate the key design choices underlying PULSE on IPAD. Unless stated otherwise, we use DINOv2-S/14 as the frozen encoder with $K{=}32,768$ prototypes and follow the training configuration described in section 4. All ablation runs are trained for 120 epochs. For full per class results of ablations we refer to Appendix section 10.

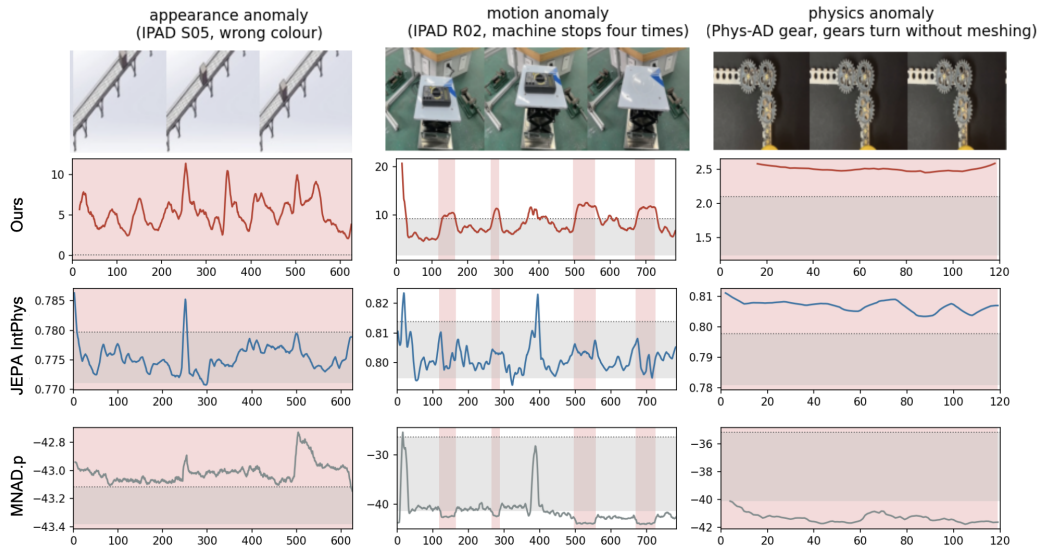


Figure 3: **Qualitative examples of latent surprise.** Surprise scores are visualized from IPAD and PhysAD video samples for our method and two alternatives: JEPA IntPhys and MNAD.p. Columns show an appearance anomaly, a machine that stops, and a gear that has shifted and no longer drives its neighbor. Red shading marks anomalous frames. PhysAD contains video level labels hence all frames of the video are shown as anomalous. PULSE's surprise rises during anomalous intervals and stays elevated above the normal range (grey) throughout the appearance and gear anomalies. The latent prediction baseline (JEPA IntPhys) shows no consistent correspondence between surprise and anomalous frames. The pixel prediction baseline (MNAD.p) is particularly insensitive to motion anomalies: its surprise decreases while the machine is stopped as static scenes are trivially predictable.

Prediction objective. We first investigate the effect of supervising future prediction through prototypes. Table 4a compares prototype-based prediction with direct feature regression, showing that prototype prediction outperforms directly regressing features by 6.3% on average. This supports our hypothesis that anchoring predictions to normal feature regions, rather than exact future features, provides a more effective objective for modeling normal dynamics.

Prototype granularity. We report the performance by varying the number of prototypes K in Table 4b. Performance increases from 71.2 frame-level AUROC at $K = 512$ to 76.5 at $K = 32,768$, increasing to $K = 65,536$ yields a comparable 76.0. PULSE therefore remains stable across a broad range of prototype sizes. This can be attributed to a finer prototype space preserving more distinctions between normal future representations. However it also increases prediction difficulty, leading to diminishing returns at larger K.

Temporal context and prediction horizon. Table 4c shows the impact of context length and prediction horizon. Longer context windows provide more evidence about the preceding dynamics, while increasing the number of target frames requires the model to capture evolution over longer timescales. Both parameters influence computation efficiency by introducing more context tokens and future queries. Increasing context from 2 to 4 frames improves AUROC from 75.0 to 76.5, while extending the prediction horizon from 2 to 4 frames further improves performance to 77.3. Increasing context to 6 frames adds only 0.2 (77.5), suggesting that more temporal information is useful only up to the timescale relevant to the dataset dynamics.

Robustness to feature representation. Table 4d compares DINOv2-S/14 and DINOv3-S/16 as frozen visual encoders. The two encoders obtain similar performance, with DINOv2 slightly outperforming DINOv3 by 1.4% in the current configuration. This suggests that PULSE is not tied to a single pretrained feature representation. Table 4e shows that with larger encoders performance improves, suggesting that performance will profit from future progress on pretrained encoders.

Predictor design. We additionally compare predictor depth and attention type in Table 4f and Table 4g. Across predictor layer depths $L \in \{2, 4, 8\}$ performance improves 1.3% from depth 2 to 4 and 0.1% more at depth 8. Replacing self-attention with cross-attention between the context and the

Table 4: **Ablation study on IPAD.** We evaluate the training objective (4a), prototype size (4b), temporal prediction window (4c), feature encoder (4d), encoder size (4e), predictor depth (4f), and attention type (4g) using frame-level AUROC (↑). For each ablation the best result is shown in **bold**. All ablations are trained for 120 epochs. The purple row, shows our default setup. Prototypes provide an effective prediction target, improving over direct feature regression by 6.3 AUROC points. Performance generally improves with finer prototype vocabularies and is stable across predictor architectures and encoder representations. Additional gains from longer temporal windows and larger encoders suggest that richer temporal context and visual features complement prototype-based supervision.

(a) Train objective and clustering.

Clustering	Loss type	AUROC
×	ℓ_1	70.2
✓	CE	**76.5**

(b) Prototype size K.

Prototypes K	AUROC
512	71.2
1,024	72.4
4,096	74.5
8,192	75.4
16,384	76.2
32,768	**76.5**
65,536	76.0

(c) Context/prediction window.

Context	Target	AUROC
2	2	75.0
4	2	76.5
4	4	77.3
6	4	**77.5**

(d) Encoder.

Encoder	AUROC
DINOv2-S/14	**76.5**
DINOv3-S/16	75.1

(e) Encoder size.

Encoder size	AUROC
DINOv2-S/14	76.5
DINOv2-B/14	76.7
DINOv2-L/14	**77.7**

(f) Predictor depth.

Depth L	AUROC
L=2	75.2
L=4	76.5
L=8	**76.6**

(g) Attention type.

Attention	AUROC
Cross	**76.5**
Self	**76.5**

future queries leaves the mean unchanged at 76.5. Each of these choices moves the mean by at most 1.3%, against 5.3% between the smallest and the largest prototype vocabulary (Table 4b). Accuracy is therefore driven by the prototype vocabulary rather than by the predictor architecture.

5 Conclusion

We present PULSE, a method for detecting physics anomalies in video by learning from normal videos to predict future prototype assignments and scoring frames using the resulting latent surprise. PULSE achieves the highest average AUROC among the methods evaluated on Phys-AD, IPAD, and TempGlitch, leading on 20 of 22, 9 of 16, and 3 of 5 categories, respectively. Its strengths span identifying abnormal mechanical interactions and deformations in Phys-AD, disruptions in motion continuity of processes in IPAD, and velocity and object impermanence anomalies in simulated game physics in TempGlitch. Subtle, localized anomalies and persistent anomalies that become locally predictable may remain challenging, motivating future work on motion-sensitive representations and longer-term references for normal behavior. Our ablations identify prototype-based prediction as the largest source of improvement over feature regression, supporting explicit normal references as an effective basis for physics anomaly detection.

6 AI Usage Statement

In this work, we used generative AI tools for feedback on code implementation, prose drafting and refinement, table and figure formatting, writing proofs, and surfacing related literature. We have not used generative AI tools to generate synthetic data, form hypotheses, or interpret results. We have critically reviewed all AI-assisted work: code was manually reviewed by authors, proofs are manually cross-checked, tables and figures originate from a reproducible pipeline and are cross-checked against source metrics, LLM-assisted literature search was complemented by manual search, and all writing was reviewed and revised. All LLM-sourced related work have been manually reviewed. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 481–490, 2019.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: DINO as a foundation for video world models. *arXiv preprint arXiv:2507.19468*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research (TMLR)*, 2024.

Gabrijel Boduljak, Yushi Lan, Christian Rupprecht, and Andrea Vedaldi. VFMF: Dense forecasting by generating foundation model features. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Florian Bordes, Quentin Garrido, Justine T Kao, Adina Williams, Michael Rabbat, and Emmanuel Dupoux. IntPhys 2: Benchmarking intuitive physics understanding in complex synthetic environments. *arXiv preprint arXiv:2506.09849*, 2025.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision (ECCV)*, pp. 139–156, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 9912–9924, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Quentin Garrido, Nicolas Ballas, Mahmoud Assran, Adrien Bardes, Laurent Najman, Michael Rabbat, Emmanuel Dupoux, and Yann LeCun. Intuitive physics understanding emerges from self-supervised pretraining on natural videos. *arXiv preprint arXiv:2502.11831*, 2025.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1705–1714, 2019.

Yao Gu, Xiaohao Xu, and Yingna Wu. Multi-turn physics-informed vision-language model for physics-grounded anomaly detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2026.

Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 733–742, 2016.

Laura Kart and Niv Cohen. Approaches toward physical and general video anomaly detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1785–1789, 2022.

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyridon Gidaris, and Nikos Komodakis. DINO-foresight: Looking into the future with DINO. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:163779–163811, 2025.

Kijung Lee, Youngwan Jo, Sunghyun Ahn, and Sanghyun Park. Unified video anomaly detection model for detecting different anomaly types. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 784–794, 2026.

Wenqiao Li, Yao Gu, Xintao Chen, Xiaohao Xu, Ming Hu, Xiaonan Huang, and Yingna Wu. Towards visual discrimination and reasoning of real-world physical dynamics: Physics-grounded anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30409–30419, 2025.

Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Dongheng Lin, Mengxue Qu, Kunyang Han, Jianbo Jiao, Xiaojie Jin, and Yunchao Wei. A unified reasoning framework for holistic zero-shot video anomaly analysis. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:29713–29744, 2025.

Jinfan Liu, Yichao Yan, Junjie Li, Weiming Zhao, Pengzhi Chu, Xingdong Sheng, Yunhui Liu, and Xiaokang Yang. IPAD: Industrial process anomaly detection dataset. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(1):380–393, 2024.

Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection—a new baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6536–6545, 2018.

Wenrui Liu, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Diversity-measurable anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12147–12156, 2023.

Yang Liu, Boan Chen, Yuanyuan Meng, Jing Liu, Zhengliang Guo, Wei Zhou, Peng Sun, and Hong Chen. Motion semantics guided normalizing flow for privacy-preserving video anomaly detection. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2026.

Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13568–13577, 2021.

Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 FPS in MATLAB. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2720–2727, 2013.

Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked RNN framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 341–349, 2017.

Hui Lv, Chen Chen, Zhen Cui, Chunyan Xu, Yong Li, and Jian Yang. Learning normal dynamics in videos with meta prototype network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15420–15429, 2021.

Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14360–14369, 2020.

Francois Porcher, Nicolas Carion, Karteek Alahari, and Shizhe Chen. Flow matching in feature space for stochastic world modeling. *arXiv preprint arXiv:2606.29059*, 2026.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6479–6488, 2018.

Cheng Yan, Shiyu Zhang, Yang Liu, Guansong Pang, and Wenjun Wang. Feature prediction diffusion model for video anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5504–5514, 2023.

Muchao Ye, Weiyang Liu, and Pan He. VERA: Explainable video anomaly detection via verbalized learning of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8679–8688, 2025.

Qianzi Yu, Kai Zhu, Yang Cao, and Yu Kang. INFP: INdustrial video anomaly detection via frequency prioritization. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2296–2304, 2025.

Yakun Yu, Ashley Wiens, Adrián Barahona-Ríos, Benedict Wilkins, Saman Zadtootaghaj, Nabajeet Barman, and Cor-Paul Bezemer. TempGlitch: Evaluating Vision-Language models for temporal glitch detection in gameplay videos. *arXiv preprint arXiv:2605.21443*, 2026.

Mei Yuan, Qi Long, Qifeng Wu, Zhenyang Li, Yizhou Zhao, Lei Wang, Yang Liu, and Min Xu. O-VAD: Industrial video anomaly detection through object-centric tracking and reasoning. In *European Conference on Computer Vision (ECCV)*, 2026.

Yuanhan Zhang, Bo Li, Haotian Liu, Yong Jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. LLaVA-NeXT: A strong zero-shot video understanding model. https://llava-vl.github.io/blog/2024-04-30-llava-next-video/, 2024.

APPENDIX

7 IMPLEMENTATION DETAILS

We provide the hyperparameters and compute settings used for evaluation on each dataset. Entries marked "same" match the settings of the IPAD column.

Table 5: **Training configuration and compute.** Settings used for every reported PULSE run.

	IPAD	Phys-AD	TempGlitch
Encoder (frozen)	DINOv2 ViT-S/14	same	same
Resolution	224	448	448
Frames / context / horizon	6 / 4 / 2	same	same
Frame step / stride	4 / 1	4 / 2	12 / 4
Predictor	L=4, 6 heads, width 384 MLP ratio 4, self-attention	same	same
Prototypes	K=32768, frozen, per property	same	same
Temperatures	τ_p=0.1, τ_q=0.05, centering EMA 0.9	same	same
Optimizer	AdamW	same	same
Learning rate	1e−5, constant	same	3e−5, constant
Weight decay	0	same	same
Gradient clip	3	same	same
Batch size	16	same	same
Epochs	120	same	same
Precision	bf16 mixed	same	same
Hardware	1× RTX A5000	same	4× H100
Seed	123	same	same

8 VLM PROMPT DETAILS

For TempGlitch, we prompt with:

> *You are evaluating keyframes from one gameplay video. Decide whether the video shows a real gameplay bug/glitch based only on the frames. Do not use any outside knowledge. Use only visible evidence from the provided frames. These images are keyframes from a single gameplay video, in chronological order. Score the level of the possibility of the anomalies from 0 to 1, where 0 stand for 'Absolutely no abnormalities' and 1 stand for 'Absolutely abnormalities'. Try not to score precisely at 0.5. Only output the score as the answer and no other reply.*

This is the same prompt as used by TempGlitch (Yu et al., 2026), with the exception being that to retrieve video AUROC scores, we change the last line to request a numerical score between 0 to 1 without scoring precisely at 0.5 following Li et al. (2025). We follow the same prompts as PhysAD for PhysAD VLM results.

9 FULL BENCHMARK TABLES

We provide comparison against additional baselines, including established video anomaly detection methods.

10 EXTENDED ABLATION TABLES

We report full per-class results for all ablations on the IPAD dataset.

11 COMPUTATIONAL COST

We report parameter count for each baseline and for our method in Table 15.

Table 6: **Video-level AUROC on Phys-AD and TempGlitch (↑).** Best results are in **bold**, second best in underline. † indicates our evaluated baselines. We evaluate all baselines on TempGlitch using officially released implementations. PULSE improves over the strongest baseline in this table (DINO-Foresight) by $+9.2$ on average in Phys-AD and $+17.1$ on average in TempGlitch.

	Zero-shot	**Unsupervised**							
	JEPA IntPhys†	MPN	MemAE	SVM	MNAD.p	MNAD.r	DMAD	DINO-Foresight	**Ours**
Phys-AD									
Car	58.5	22.9	52.3	58.7	49.2	94.4	38.1	**100.0**	**100.0**
Fan	70.1	81.1	37.1	50.0	81.0	54.2	36.5	59.1	**95.7**
Rolling Bearing	98.1	35.3	4.4	93.3	35.2	80.0	89.8	**100.0**	**100.0**
Spherical Bearing	95.3	11.3	9.2	65.0	96.2	81.3	64.9	99.8	**100.0**
Servo	23.1	36.4	44.5	50.0	97.5	87.8	96.9	95.9	**100.0**
Clip	0.8	53.5	44.3	50.0	63.0	33.3	56.0	68.3	**100.0**
U Disk	43.8	24.0	61.7	50.0	60.9	94.0	78.0	94.9	**100.0**
Hinge	50.1	76.9	87.0	50.0	70.5	89.5	85.0	**100.0**	**100.0**
Sticky Roller	67.1	96.7	96.7	50.0	45.1	93.6	97.8	**100.0**	**100.0**
Caster Wheel	61.2	36.4	52.3	50.0	27.1	50.8	71.6	**100.0**	**100.0**
Screw	52.7	52.2	56.7	50.0	68.0	54.7	70.9	**100.0**	**100.0**
Lock	26.0	56.3	59.7	73.3	43.0	64.1	63.7	88.6	**100.0**
Gear	64.4	52.9	51.0	50.0	65.2	69.4	83.5	82.1	**99.9**
Clock	36.9	34.0	54.2	50.0	39.5	58.7	62.3	68.6	**80.5**
Slide	26.9	51.7	96.2	50.0	91.7	78.4	99.3	**100.0**	**100.0**
Zipper	66.3	81.5	59.2	50.0	82.9	42.1	64.6	92.2	**100.0**
Button	14.4	85.3	36.5	50.0	66.0	56.8	88.2	65.2	**100.0**
Liquid	2.8	18.4	70.0	50.0	67.1	83.1	91.1	**100.0**	**100.0**
Rubber Band	51.8	37.4	36.8	56.7	36.6	30.7	89.7	96.2	**96.7**
Ball	78.2	54.3	38.3	50.0	72.8	68.7	82.9	79.1	**100.0**
Magnet	82.1	67.1	46.4	50.0	69.1	43.8	40.3	95.7	**96.4**
Toothpaste	50.8	58.7	88.9	50.0	52.0	63.1	58.3	81.6	**99.7**
Average	51.0	51.1	53.8	54.4	62.7	66.9	73.1	89.4	**98.6**
TempGlitch									
Blinking	74.4	34.1	39.0	52.7	51.7	49.5	27.9	58.6	**76.8**
Frozen animation	45.5	45.6	41.3	51.7	54.5	65.1	63.8	60.6	**72.6**
Shooting error	57.7	50.5	**71.3**	50.7	34.9	41.8	66.2	60.7	62.5
Stuck in place	**72.6**	67.5	45.3	47.3	24.7	40.2	44.9	35.3	61.1
Velocity bug	69.5	40.1	46.9	49.3	59.0	38.4	54.1	63.5	**90.9**
Average	63.9	47.6	48.8	50.3	45.0	47.0	51.4	55.7	**72.8**

Table 7: **Comparison with VLM baselines on Phys-AD and TempGlitch.** Video-level AUROC (↑) is reported per category and averaged across each benchmark. Best and second-best results are shown in **bold** and underline, respectively. ⋆ indicates a constant anomaly score across all videos in the category. †MT-PIVLM does not report results for the Toothpaste class; its Phys-AD average is computed over the remaining 21 classes. O-VAD and MT-PIVLM do not report on TempGlitch. We evaluate all baselines on TempGlitch using officially released implementations. With orders of magnitude fewer parameters (7.4M trainable and 34.5M frozen vs. 7-33B), PULSE improves on supervised and weakly supervised VLMs by +1.9 on Phys-AD and outperforms TempGlitch baselines.

	Video-LLaVA-7B	Qwen3-VL-8B	Video-LLaMA3-7B	LLaVA-NeXT-Video-7B	Qwen3-VL-32B	O-VAD	MT-PIVLM†	**Ours**
Phys-AD								
Car	52.2	51.3	76.8	⋆50.0	49.2	57.5	99.9	**100.0**
Fan	61.1	57.0	50.9	⋆50.0	56.6	66.9	91.2	**95.7**
Rolling Bearing	50.0	48.3	33.3	⋆50.0	41.8	65.0	**100.0**	**100.0**
Spherical Bearing	50.0	48.3	41.7	⋆50.0	48.3	64.1	**100.0**	**100.0**
Servo	46.4	39.7	43.3	⋆50.0	49.2	49.5	92.6	**100.0**
Clip	45.8	29.4	36.0	⋆50.0	44.2	42.3	97.4	**100.0**
U Disk	50.0	45.4	43.8	⋆50.0	57.8	60.6	96.4	**100.0**
Hinge	50.0	33.3	⋆50.0	⋆50.0	58.9	52.6	81.8	**100.0**
Sticky Roller	46.7	48.3	⋆50.0	⋆50.0	53.3	81.1	**100.0**	**100.0**
Caster Wheel	50.0	⋆50.0	⋆50.0	⋆50.0	53.3	30.1	**100.0**	**100.0**
Screw	55.0	50.0	⋆50.0	⋆50.0	45.0	70.4	98.1	**100.0**
Lock	50.0	20.8	47.5	⋆50.0	38.8	78.0	**100.0**	**100.0**
Gear	51.7	43.9	34.0	49.9	64.4	87.9	99.1	**99.9**
Clock	50.0	⋆50.0	⋆50.0	⋆50.0	58.1	66.9	**88.6**	80.5
Slide	17.9	50.4	⋆50.0	⋆50.0	52.9	41.5	**100.0**	**100.0**
Zipper	50.0	51.3	⋆50.0	⋆50.0	51.7	38.8	94.5	**100.0**
Button	36.0	⋆50.0	51.7	⋆50.0	50.6	39.3	98.7	**100.0**
Liquid	21.7	⋆50.0	58.3	⋆50.0	83.3	77.6	**100.0**	**100.0**
Rubber Band	45.0	56.7	46.7	⋆50.0	76.7	72.1	95.8	**96.7**
Ball	53.3	52.2	63.2	51.1	52.8	57.9	97.3	**100.0**
Magnet	40.0	50.8	48.3	⋆50.0	59.2	57.6	**100.0**	96.4
Toothpaste	46.7	44.4	48.6	⋆50.0	57.8	68.9	–	**99.7**
Average	46.3	46.4	48.8	50.0	54.7	58.4	96.7	**98.6**
TempGlitch								
Blinking	⋆50.0	58.2	57.0	⋆50.0	52.9	–	–	**76.8**
Frozen animation	52.2	55.3	52.7	50.4	49.7	–	–	**72.6**
Shooting error	⋆50.0	55.1	52.3	⋆50.0	48.6	–	–	**62.5**
Stuck in place	59.7	53.0	51.0	55.4	48.3	–	–	**61.1**
Velocity bug	⋆50.0	47.9	39.0	⋆50.0	48.1	–	–	**90.9**
Average	52.4	53.9	50.4	51.2	49.5	–	–	**72.8**

Table 8: **Predictor depth ablation on IPAD.** Increasing predictor depth improves performance initially but has diminishing returns at much higher depth: L=2 → L=4 improves +1.3 but at L=4 → L=8 improvement is only +0.1.

Depth	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	R01	R02	R03	R04	Mean
L=2	67.7	58.6	91.6	**70.2**	98.3	52.0	71.6	79.3	69.8	82.5	**70.6**	**71.8**	**89.8**	93.9	57.3	78.2	75.2
L=4	**78.8**	**60.0**	**92.0**	69.5	98.2	57.3	73.8	**81.4**	**70.6**	**83.7**	69.5	71.3	86.9	94.4	56.4	**80.0**	76.5
L=8	78.4	59.5	91.0	69.1	**98.4**	**57.5**	**74.8**	81.0	70.1	82.8	69.8	70.5	88.3	**94.9**	**60.1**	78.8	**76.6**

Table 9: **Encoder ablation on IPAD.** PULSE remains robust to differing encoder representations. DINOv2 is better (+1.4 average improvement) over DINOv3 encodings.

Encoder	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	R01	R02	R03	R04	Mean
DINOv2	**78.8**	60.0	**92.0**	69.5	**98.2**	**57.3**	**73.8**	**81.4**	70.6	83.7	69.5	**71.3**	**86.9**	**94.4**	56.4	**80.0**	**76.5**
DINOv3	68.4	**60.9**	85.8	**74.0**	97.0	55.6	**73.8**	75.6	**75.6**	**84.4**	**71.8**	67.9	81.7	91.2	**60.5**	76.7	75.1

Table 10: **Attention-type ablation on IPAD.** Self- and cross-attention between the context features and the future queries are equivalent at our recipe: both reach 76.5 mean frame AUROC, with cross-attention ahead on 5 devices, behind on 8, and tied on 3.

Attention	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	R01	R02	R03	R04	Mean
Self	**78.8**	**60.0**	**92.0**	**69.5**	98.2	**57.3**	73.8	**81.4**	70.6	83.7	69.5	71.3	86.9	**94.4**	56.4	**80.0**	76.5
Cross	78.0	59.0	91.9	68.8	98.2	55.2	73.8	80.9	**74.0**	**84.8**	**69.6**	**72.3**	**90.0**	93.1	56.4	77.8	76.5

Table 11: **Prototype size ablation on IPAD.** Frame AUROC improves monotonically with codebook size up to K=32768 and plateaus beyond: $+5.3$ points from K=512 to K=32768, and $\Delta < 0.5$ at K=65536. Larger codebooks add prototype memory without further accuracy gains.

K	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	R01	R02	R03	R04	Mean
512	63.0	56.2	82.3	67.5	97.1	40.0	63.1	74.2	66.9	80.4	67.1	68.6	88.6	90.5	**60.9**	72.3	71.2
1024	71.7	57.3	86.3	67.4	97.4	43.9	66.5	77.3	65.8	80.9	67.5	65.3	**88.7**	91.8	56.7	74.7	72.4
4096	74.1	57.1	88.6	68.1	97.7	51.9	72.8	80.7	68.8	82.3	67.1	65.5	85.9	93.3	60.6	77.1	74.5
8192	76.2	57.9	89.3	69.7	97.9	57.1	**74.0**	80.9	69.4	82.6	68.0	67.6	86.3	94.0	57.0	78.2	75.4
16384	77.4	**60.2**	90.2	**69.9**	98.0	56.6	73.7	**82.5**	70.2	83.4	68.6	70.6	88.4	94.2	56.6	78.5	76.2
32768	**78.8**	60.0	**92.0**	69.5	**98.2**	**57.3**	73.8	81.4	**70.6**	**83.7**	**69.5**	71.3	86.9	**94.4**	56.4	**80.0**	**76.5**
65536	77.8	59.0	91.6	69.7	98.2	57.2	71.2	80.2	70.4	83.7	69.4	**71.8**	86.6	93.8	55.7	79.8	76.0

Table 12: **Training-objective ablation on IPAD.** Frame-level AUROC reported. Best per device column in **bold**.

Clustering	Loss	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	R01	R02	R03	R04	Mean
×	ℓ_1	49.3	54.5	84.4	63.1	**98.5**	43.0	65.4	72.7	68.7	82.7	63.0	**71.7**	**88.7**	90.8	**56.6**	70.5	70.2
✓	CE	**78.8**	**60.0**	**92.0**	**69.5**	98.2	**57.3**	**73.8**	**81.4**	**70.6**	**83.7**	**69.5**	71.3	86.9	**94.4**	56.4	**80.0**	**76.5**

Table 13: **Context size and prediction horizon ablation on IPAD.** Longer context and prediction horizons help up to c=4, p=4. As context size increases from 2 to 4 (at p=2), performance improves by $+1.5$ points (75.0 $\rightarrow$ 76.5). As prediction horizon increases from 2 to 4 (at c=4), performance increases by $+0.8$ points (76.5 $\rightarrow$ 77.3). As context size increases further from 4 to 6 (at p=4), performance does not significantly improve on average ($\Delta = +0.2$). For S06 and R03, extending the prediction horizon yields larger gains ($+2.1$ and $+3.5$ points respectively), suggesting these classes benefit most from modeling longer temporal dependencies.

c, p	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	R01	R02	R03	R04	Mean
c=2, p=2	74.3	57.1	91.3	71.4	98.0	54.2	72.5	80.8	69.4	81.2	68.9	67.5	85.0	91.9	56.9	80.1	75.0
c=4, p=2	**78.8**	**60.0**	92.0	69.5	**98.2**	57.3	73.8	81.4	70.6	83.7	69.5	71.3	86.9	94.4	56.4	80.0	76.5
c=4, p=4	78.3	59.4	92.0	**71.8**	97.8	59.4	**74.8**	**81.9**	**70.9**	84.0	72.2	71.6	**87.8**	94.9	59.9	**80.3**	77.3
c=6, p=4	78.3	59.9	**92.5**	70.2	97.6	**61.3**	74.7	81.1	70.4	**84.6**	**72.3**	**73.8**	87.0	**96.3**	**60.9**	78.8	**77.5**

Table 14: **Encoder-size ablation on IPAD.** Performance scales with encoder size: DINOv2-S $\rightarrow$ B improves by $+0.2$ points, and DINOv2-B $\rightarrow$ L improves by $+1.0$ points.

Encoder	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	R01	R02	R03	R04	Mean
DINOv2-S/14	**78.8**	60.0	**92.0**	69.5	**98.2**	57.3	73.8	81.4	70.6	83.7	69.5	71.3	86.9	**94.4**	56.4	**80.0**	76.5
DINOv2-B/14	70.3	58.8	89.5	69.6	98.1	**61.0**	**76.4**	80.6	75.1	86.3	**71.9**	72.7	83.3	90.5	**64.6**	78.1	76.7
DINOv2-L/14	68.4	**63.6**	91.8	**70.1**	97.9	60.8	75.6	**81.5**	**80.3**	**87.5**	71.8	**74.5**	**89.0**	93.9	59.2	78.1	**77.7**

Table 15: **Parameter count comparison.** PULSE is orders of magnitude more efficient than the VLM baselines while achieving higher AUROC. With 7.4 M trainable and 34.5 M frozen parameters (41.9 M total at K=32768), it is an efficient approach for state-of-the-art physics anomaly detection.

Method	Params
MPN (Lv et al., 2021)	13.3 M
MemAE (Gong et al., 2019)	6.5 M
MNAD.p (Park et al., 2020)	15.7 M
MNAD.r (Park et al., 2020)	17.2 M
SVM (Sultani et al., 2018)	–
DMAD (Liu et al., 2023)	5.7 M
INFP (Yu et al., 2025)	–
IPAD (Liu et al., 2024)	263.5 M
JEPA Intuitive Phys. (Garrido et al., 2025)	∼325 M
Qwen3-VL-32B (Bai et al., 2025)	33 B
URF-ZS-HVAA (Lin et al., 2025)	22 B
VERA (Ye et al., 2025)	8 B
O-VAD (Yuan et al., 2026)	33 B
LLaVA-NeXT-Video-7B (Zhang et al., 2024)	7 B
Video-LLaVA-7B (Lin et al., 2024)	7 B
Ours	41.9 M (7.4 M train.)