# Robot-WATCHDOG : failure detection through object-centric graph representation

**Anonymous Author(s)**
Affiliation
Address
email

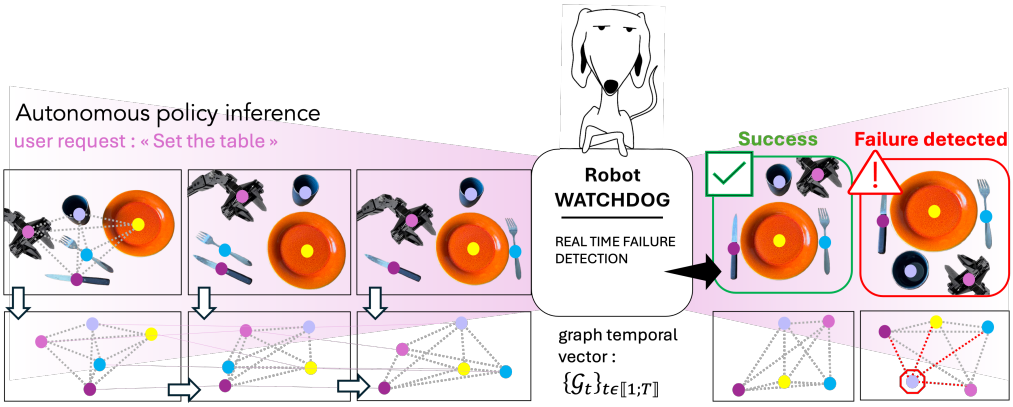


Figure 1: **Detecting failures through object-centric relational dynamics.** Robot-WATCHDOG represents manipulation scenes as spatio-temporal graphs enabling robust detection of relational and temporal errors beyond raw visual or trajectory-based methods.

**Abstract:**

Reliable real-time failure detection is critical for deploying learned robotic policies in real-world environments. However, explicitly enumerating failures is intractable in open-world settings, making failure detection from nominal demonstrations alone a necessary alternative. Existing approaches face key limitations: purely visual methods are highly sensitive to benign background variations, while kinematic monitors lack awareness of the surrounding environment. Crucially, both often fail to capture relational errors involving object identity, spatial arrangement, or task progression, such as grasping the wrong object or violating interaction structure. In this work, we introduce **Robot-WATCHDOG**, a self-supervised failure detection framework based on an object-centric representation of dynamic scenes as spatio-temporal graphs over tracked labeled objects and their interactions. Within this representation, the framework deploys two complementary methods to adapt across data regimes. For data-rich settings, we propose **GnnT**, a predictive graph transformer that identifies failures as deviations from learned object-centric spatio-temporal dynamics. To address data scarcity, we introduce **Trajectory Projection and Correlation (TPC)**, a non-parametric method that detects failures through geometric consistency and changes in object-relation dynamics in extreme few-shot scenarios. We evaluate the framework on a public benchmark and a newly collected dataset featuring relationally challenging manipulation tasks. Results show that modeling object-centric spatial and temporal interactions improves failure detection over existing baselines and enables reliable detection of failure modes that are difficult to capture using visual or kinematic monitoring alone.

Submitted to the 10th Conference on Robot Learning (CoRL 2026). Do not distribute.

**Keywords:** Robot Learning, Failure Detection, Object-Centric Representations, Graph Neural Networks

# 1 INTRODUCTION

Learned robotic policies provide no guarantees beyond the distribution induced by training demonstrations [1]. Detecting deviations from expected behavior is therefore essential for accountable deployment, and increasingly emphasized in regulation for high-risk AI systems such as the EU AI Act [2, 3]. As imitation learning and vision-language-conditioned policies increasingly enable long-horizon, closed-loop manipulation [4, 5, 6, 7, 8, 9], policies are executed autonomously in real time and without step-by-step human intervention, motivating dedicated monitoring systems that operate online, require no failure examples, and can identify anomalous behavior before errors propagate.

Explicitly enumerating all possible failures, however, is infeasible in open-world robotic settings. The standard framing therefore casts failure detection as a one-class problem [10, 11, 12]: a monitor trained exclusively on nominal demonstrations must identify, at runtime, deviations from expected behavior. Existing approaches divide broadly into two families. Visual failure detectors [13, 14, 15, 11] flag deviations in raw image space or learned visual representations; they are sensitive to benign background changes, lighting shifts, and visual clutter, producing frequent false positives that erode operational efficiency. Kinematic and action-based monitors [16, 17, 18] instead operate on trajectory or joint-space signals; they are robust to visual noise but structurally blind to the environment — they only verify whether the movement itself is familiar, failing to detect if the robot is interacting with the wrong object or operating in an environment that mismatches training.

This division points to a broader limitation. A substantial class of failures in robotic manipulation does not arise from abnormal motion or corrupted observations, but from incorrect interactions between otherwise familiar objects: grasping the wrong item, violating task-order constraints, or placing an object in an incorrect spatial relation to others [19, 20]. These failures emerge naturally in manipulation tasks whose success depends on object identity, relative placement, and sequential consistency, yet they remain difficult to capture when monitoring is restricted to either image appearance or motion trajectories alone.

The central idea of this work is to recast robotic failure detection as monitoring the evolution of object interactions over time. Rather than treating visual and kinematic signals as separate modalities, we represent the scene as a spatio-temporal graph of interacting objects, where nodes encode object states and edges encode their pairwise spatial relationships throughout execution. This object-centric formulation generalizes prior monitoring approaches: visual detectors can be interpreted as operating on node appearance, while kinematic monitors correspond to tracking motion along trajectories. By jointly modeling objects, their relations, and their temporal evolution in a unified representation, the proposed framework captures both the signals targeted by prior work and the relational structure required to detect task-relevant interaction failures.

Failure detection then becomes a well-posed problem: deviations from learned or reference relational dynamics constitute anomalies. We instantiate this principle in Robot-WATCHDOG, a self-supervised framework that operates in two complementary modes depending on data availability. When sufficient demonstrations are available, GnnT — a Graph Attention Network coupled with a Transformer encoder — learns to predict future object dynamics from a sliding observation window; failures are flagged as prediction errors exceeding a calibrated threshold. When data is scarce (fewer than 25 episodes), TPC (Trajectory Projection and Correlation) provides a non-parametric alternative, detecting anomalies through orthogonal projection distances to expert trajectories and deviations in pairwise object correlation structure. Both methods are calibrated via conformal prediction, which provides a mathematically guaranteed bound on the false positive rate without requiring manual threshold tuning or exposure to failure data.

We evaluate Robot-WATCHDOG on the public BotFails benchmark and on a newly introduced dataset featuring two relationally challenging manipulation tasks—structured table setting and waste

sorting on a moving conveyor belt—designed to challenge relational failure detection. Across both datasets and all metrics, our object-centric methods consistently outperform prior work: TPC and GnnT achieve AUPR scores of 0.69–0.78 and MCC scores of 0.67–0.72, whereas the strongest baseline reaches only 0.47 and 0.40. Crucially, methods that perform well on simple kinematic deviations struggle with relational failures involving object identity and spatial arrangement—the very settings Robot-WATCHDOG is designed to address.

Our main contributions are: (i) identifying relational anomalies as a key yet underexplored failure mode in robotic manipulation, arising from dependencies on object identity, spatial arrangement, and temporal consistency; (ii) introducing an object-centric spatio-temporal graph formulation for failure detection that unifies visual and kinematic monitoring while preserving task-relevant relations; (iii) presenting Robot-WATCHDOG, a dual-method framework combining predictive spatio-temporal modeling and geometric reasoning; and (iv) releasing a benchmark of relationally challenging manipulation tasks with annotated relational failures.

# 2 RELATED WORK

## 2.1 Graph oriented object representation

A growing line of work in robotics and visuomotor learning has explored object-centric graph representations, where nodes encode task-relevant entities and edges capture spatial, label-conditioned, or action-conditioned relations [21, 22, 23]. Such representations improve invariance to irrelevant visual variations and provide a structured way to model interactions between objects. Building on this paradigm, several approaches construct dynamic or action-conditioned scene graphs to model evolving environments, improving robustness and generalization in manipulation tasks [24, 25]. In parallel, graph neural networks have been leveraged for relational policy learning, planning, and dynamics prediction, demonstrating improved scalability in multi-object settings [26, 27]. More recently, Vosylius et al. [28] show that graph-structured representations can support in-context policy adaptation by modeling demonstrations over relational structures. Recent progress in vision-language models and object tracking has also made these representations increasingly practical in robotics, enabling automatic extraction of task-relevant objects and trajectories directly from RGB observations. To our knowledge, while object-centric graph representations have shown strong promise for policy learning and planning, they have received limited attention for online failure detection, motivating their investigation in this work.

## 2.2 Failure detection in autonomous robotics

Existing real-time monitors struggle to balance environmental awareness, robustness to benign variations, and sensitivity to task-relevant object interactions. Furthermore, many traditional failure classifiers [29, 30, 31] require collecting explicit failure data *a priori*, a time-consuming process that scales poorly to open-world robotic deployment. Consequently, recent efforts focus on one-class methods trained exclusively on nominal demonstrations. However, vision-based detectors like FIDeL [11] remain highly sensitive to visual clutter. Sentinel [32] attempts to bridge the modality gap by combining STAC—a statistical measure of kinematic action consistency—with a Vision-Language Model (VLM) for visual verification. Yet, STAC remains blind to environment-only failures, and the VLM fallback introduces severe inference latency unsuited for reactive continuous control. Similarly, FAIL-Detect [10] merges modalities through latent concatenation, while FIPER [12] relies on brittle logical conjunctions.

Crucially, these approaches often struggle to capture failures involving object identity, spatial arrangement, or task progression—such as grasping the wrong object, executing sub-actions out of order, or misplacing items. Such errors often produce nominal kinematic profiles and only subtle pixel-level changes, making them difficult to detect with action-consistency metrics or standard visual anomaly detectors.

Our framework, Robot-WATCHDOG, addresses these limitations through an object-centric representation of scene dynamics. By abstracting tracked entities into a spatio-temporal graph, the framework reduces sensitivity to irrelevant visual variation while preserving explicit environmental context unavailable to kinematic-only monitors. Representing the scene in a relational geometric space also enables anomaly detection based on changes in object interactions over time. Using a predictive GNN-Transformer architecture, the proposed method jointly models object identities, spatial configurations, and temporal dependencies. This enables real-time detection of relational failure modes that remain challenging for existing multimodal baselines, without requiring failure demonstrations during training.

# 3 Method

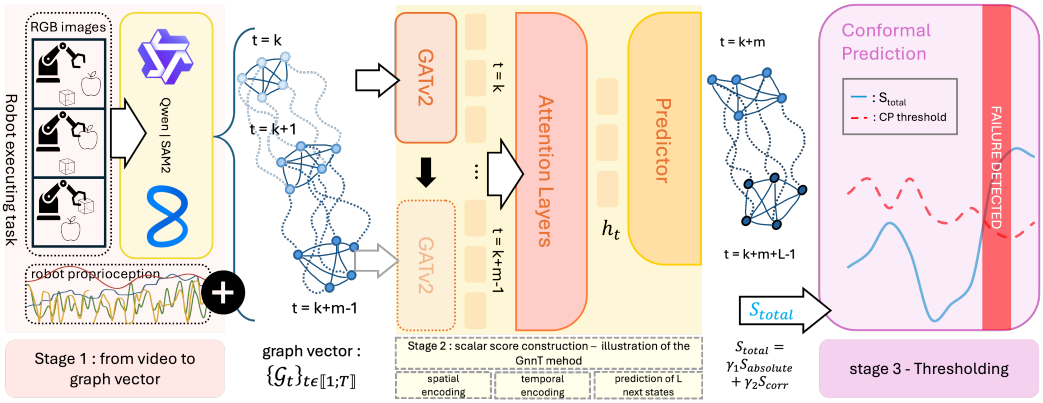


Figure 2: Overview of the Robot-WATCHDOG pipeline. **Stage 1:** object-centric graph extraction from RGB observations and robot proprioception. **Stage 2:** failure detection (illustrated with *GnnT*), which takes the graph as input and outputs a scalar failure score $S_{\text{total}} \in \mathbb{R}$ at each timestep. **Stage 3:** conformal prediction, where $S_{\text{total}}$ is compared to a threshold calibrated offline on demonstration data, yielding a binary decision (failure / no failure).

As illustrated in Figure 2, we propose a self-supervised failure detection framework for multi-object robotic manipulation. By modeling scene dynamics through spatio-temporal graphs, we formulate failure detection as a future prediction problem. The model is trained solely on nominal demonstrations and flags failures as deviations from nominal object-centric spatial and temporal interaction patterns.

## 3.1 Problem Formulation and Graph Extraction

We consider robotic manipulation episodes involving $C$ distinct entities—task-relevant objects together with the robot end-effector—evolving in a fixed workspace. Using a vision-language initialization model and a tracking pipeline [33, 34], each episode is converted from RGB observations into a sequence of 2D trajectories over tracked labeled objects. This yields a temporal graph sequence $\{\mathcal{G}_t\}_{t=1}^{T}$. Given a sliding observation window of length $m$, the objective is to predict the future $L$ graph states; the resulting prediction error defines the real-time failure score.

At each time step $t$, the scene is abstracted as $\mathcal{G}_t = (V, E_t, X_t)$. The vertex features $X_t \in \mathbb{R}^{C\times d}$ of vertices V encode absolute positions $\mathbf{p}_{i,t}$ and instantaneous velocities: $\mathbf{x}_{i,t} = [\mathbf{p}_{i,t}, \mathbf{p}_{i,t} - \mathbf{p}_{i,t-1}]$. Edges $E_t$ encode the explicit pairwise spatial relations. For any pair $(i, j)$, the relative displacement vector $\mathbf{o}_{ij,t} = \mathbf{p}_{j,t} - \mathbf{p}_{i,t}$ yields a Euclidean distance $d_{ij,t}$ and an orientation $\theta_{ij,t}$. The edge feature is defined as : $\mathbf{e}_{ij,t} = \left[\, o^{(x)}_{ij,t},\ o^{(y)}_{ij,t},\ d_{ij,t},\ \cos(\theta_{ij,t}),\ \sin(\theta_{ij,t}) \,\right] \in \mathbb{R}^5$

This representation has three useful properties for failure detection: (i) translation invariance through relative edge encoding, reducing sensitivity to workspace shifts; (ii) explicit modeling of pairwise

object interactions, allowing failures involving object identity or spatial arrangement to be represented directly; and (iii) temporal consistency through velocity features and graph sequences, enabling detection of both instantaneous and sequential deviations.

### 3.2 GnnT: Predictive Spatio-Temporal Dynamics

**Spatial Encoding via Dynamic Graph Attention** Standard convolutional operators treat all neighbors equally. Instead, we employ GATv2 [35] to compute dynamic attention coefficients $\alpha_{ij,t}$, allowing the network to weight task-relevant interactions (e.g., an end-effector approaching a target object) more strongly than distant or weakly coupled entities. For a vertex $i$ and its neighbors $j \in \mathcal{N}_i$, the updated spatial embedding $\mathbf{h}_{i,t} \in \mathbb{R}^D$ is:

$$\mathbf{h}_{i,t} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij,t} \mathbf{W} \mathbf{x}_{j,t}\right) \tag{1}$$

where $\mathbf{W}$ is a learnable weight matrix applied to the input features, and $\sigma$ denotes a non-linear activation function. This operation yields a set of spatially contextualized embeddings $H_t = \{\mathbf{h}_{1,t}, \ldots, \mathbf{h}_{C,t}\}$, where $C$ is the total number of vertices in the graph.

**Spatio-Temporal Transformer & Future Prediction** To model action-reaction delays across time, we process a sliding window of the past $m$ spatial embeddings $\mathcal{S}_{in} = \{H_{t-m+1}, \ldots, H_t\}$. We flatten $\mathcal{S}_{in}$ into a sequence of $m \times C$ tokens. This fine-grained tokenization enables modeling dependencies between specific objects at specific past timesteps. We inject structural awareness via a composite positional encoding (PE) that accounts for relative time $\tau$ within the $m$-window, object ID $i$, and global episode progress:

$$\mathbf{z}_{i,\tau}^{(0)} = \mathbf{h}_{i,t-(m-1)+\tau} + \text{PE}_{\text{time}}(\tau) + \text{PE}_{\text{obj}}(i) + \text{PE}_{\text{global}}\left(\frac{t}{T_{\max}}\right) \tag{2}$$

After processing through $N$ Transformer encoder layers, the tokens corresponding to the current step $t$ are projected via an MLP to predict the future sequence of positions over horizon $L$: $\widehat{\mathbf{Y}}_{i,t} = \text{MLP}(\widehat{\mathbf{h}}_{i,t}) \in \mathbb{R}^{L \times d_{out}}$.

**Training Objective** The network is optimized solely on nominal demonstrations using a dual-loss objective. We minimize the Mean Squared Error (MSE) of the predicted trajectories ($\mathcal{L}_{\text{pos}}$) and add an edge-consistency term ($\mathcal{L}_{\text{edge}}$) encouraging agreement between predicted and observed pairwise spatial relations: $\mathcal{L} = \lambda_p \mathcal{L}_{\text{pos}} + \lambda_e \mathcal{L}_{\text{edge}}$.

### 3.3 Extreme Few-Shot Fallback: Trajectory Projection and Correlation

As shown in Figure 8 of the Appendix, while GnnT excels at capturing complex predictive dynamics given sufficient demonstration data, data-scarce scenarios (e.g., $< 25$ episodes) can cause predictive models to overfit. To address this trade-off, we additionally formulate **Trajectory Projection and Correlation (TPC)**, a non-parametric alternative. TPC evaluates individual kinematic deviations by computing the time-weighted orthogonal projection distance to the closest expert trajectory. To capture deviations in object interaction structure, it incorporates a correlation penalty by measuring the $\ell_1$ divergence between the pairwise Pearson correlation matrices of the objects in the observed scene versus the nominal dataset. This provides a robust, purely geometric failure detector for few-shot regimes. Details of this method can be found in the Appendix, section B.3.

### 3.4 Uncertainty Calibration via Conformal Prediction

A fundamental challenge in failure detection is defining the decision threshold without relying on arbitrary, task-specific manual tuning or exposing the model to OOD failure data. We address this using Conformal Prediction (CP) [36, 37]. Using a held-out calibration set of nominal episodes $\mathcal{Z}_N$, we compute the non-conformity scores (i.e., the prediction error for GnnT, or the geometric distance

for TPC). For a user-specified tolerance level $\alpha \in (0, 1)$, the decision threshold is strictly set as the $(1 - \alpha)$-quantile of these calibration scores. Under the assumption of data exchangeability, CP provides a mathematically guaranteed upper bound on the False Positive Rate (FPR), ensuring it does not exceed $\alpha$. During online deployment, any observation yielding a score above this dynamically calibrated threshold is immediately flagged as a failure, triggering a safe-stop protocol.

# 4 Evaluation

## 4.1 Dataset

We evaluate Robot-WATCHDOG on two complementary datasets: (i) a public benchmark from prior work and (ii) a new dataset designed to stress-test relational and dynamic failure detection.

**BotFails.** We evaluate on BotFails [11], a recent benchmark for robotic failure detection collected with LeRobot [38]. The full dataset spans **10 manipulation tasks** across domestic and industrial settings with diverse annotated anomaly types. To align with our object-centric formulation, we retain **4 tasks** involving rigid and distinguishable objects: **Table-setting**, **Dish storing**, **Vegetable sorting**, and **Groceries sorting**. Tasks involving deformable or fluid interactions are excluded, as they violate the discrete object abstraction underlying our method. Additional details are provided in Appendix C.1.

**Ours.** To complement BotFails, we introduce a new dataset comprising two manipulation tasks with increased ralational and dynamic complexity. The first task is a **table-setting** scenario requiring structured object placement and ordering. The second is a **waste-sorting** task on a conveyor belt, where objects exhibit continuous motion, leading to variability in initial states and interaction timing. This dataset serves two purposes: (i) providing a larger number of demonstrations for training data-driven models, and (ii) evaluating robustness under increased variability and less constrained dynamics. Further details are provided in Appendix C.1.

## 4.2 Baselines

We compare Robot-WATCHDOG against a diverse set of anomaly detection methods spanning three complementary categories.

**Trajectory matching.** These methods detect kinematic and geometric deviations by comparing observed trajectories to expert demonstrations. To ensure invariance to execution speed and temporal misalignment, AP-RMS and AP-Fréchet rely on arc-length resampling and evaluate prefix-based distances using Root Mean Square error and the discrete Fréchet distance [39], respectively. Soft-DTW instead computes a global alignment cost using a differentiable soft-minimum formulation [40], making it more robust to local temporal distortions.

**Learned representations.** We evaluate density-based and reconstruction-based approaches trained exclusively on nominal data. Specifically, we consider Continuous Normalizing Flow models (logpZO, lopO) [10], which estimate the likelihood of observations either in latent space or via exact log-likelihood computation, and an AutoEncoder (AE-Recon) [41, 42, 43], which uses reconstruction error as an anomaly score.

**Visual-semantic alignment.** We include FIDeL [11], a recent two-stage method that combines patch-level feature alignment via optimal transport with a vision-language model to filter semantically irrelevant anomalies.

**Ours.** We compare against both variants of our framework: (i) **TPC**, a non-parametric method based on trajectory projection and relational correlation, and (ii) **GnnT**, a predictive spatio-temporal graph model that captures object interactions and temporal dependencies. Additional implementation details are provided in the Appendix C.2.

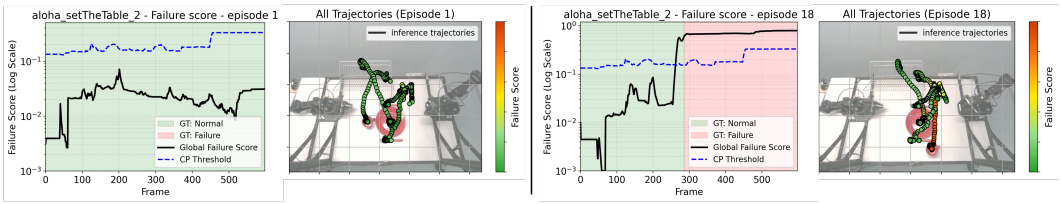


Figure 3: **Qualitative failure detection on the Table-setting task**. (Left) Nominal execution (Ep. 1): The global score remains safely below the CP threshold. (Right) Relational failure (Ep. 18): The cup is spatially misplaced. Due to natural spatial variability, absolute kinematics cannot detect this relational error. However, our object-centric graph captures the relational positional inconsistency, triggering a global score spike above the threshold that aligns with the misplaced cup's elevated local anomaly score (red trajectory points).

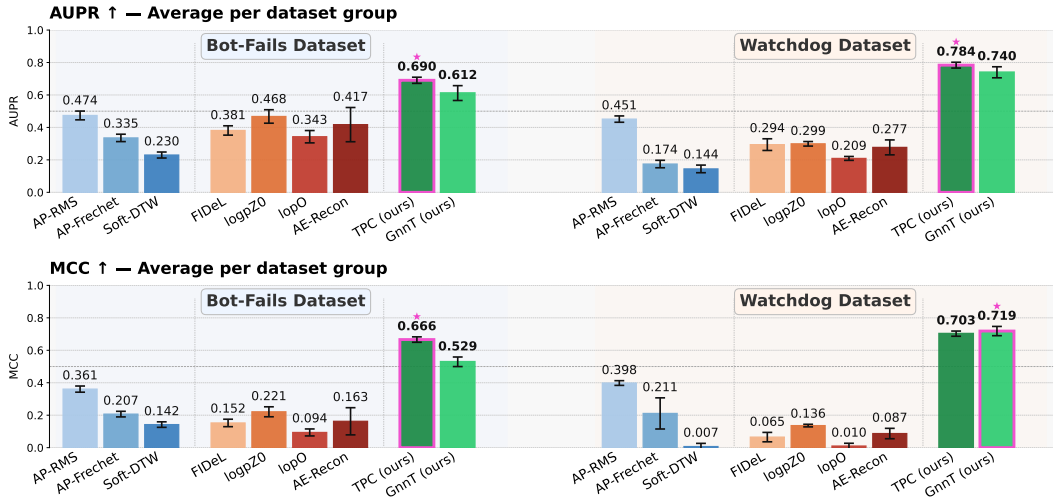


Figure 4: **Threshold-independent discriminative performance.** Average Area Under the Precision-Recall Curve (AUPR) and maximum Matthews Correlation Coefficient (MCC) across the Bot-Fails and Watchdog datasets. Error bars denote the standard deviation across task subsets. Best results framed in purple with a star.

### 4.3 Evaluation Protocol

We evaluate Robot-WATCHDOG at the step level in two complementary settings: **(i)** threshold-independent anomaly scoring and **(ii)** end-to-end failure detection after calibration. For threshold-independent scoring, we report Area Under the Precision–Recall Curve (AUPR) and maximum Matthews Correlation Coefficient (MCC). We favor AUPR due to strong class imbalance, while MCC provides a balanced summary across all confusion matrix terms. For deployment evaluation, anomaly scores are converted into binary decisions using Conformal Prediction (CP). We then report calibrated MCC together with True Positive Rate (TPR) and True Negative Rate (TNR), capturing respectively failure interception and robustness to benign variability. Formal metric definitions are provided in Appendix C.3.

### 4.4 Results and discussion

Figure 4 summarizes threshold-independent performance. Across both datasets, our object-centric methods consistently outperform prior work. **TPC** achieves the highest MCC on BotFails (0.666), where limited training data favors its non-parametric formulation. **GnnT** performs best on the more dynamic Watchdog dataset (0.719 MCC), showing that predictive spatio-temporal modeling benefits continuous and delayed interactions. In contrast, trajectory metrics (AP-RMS, Fréchet, Soft-DTW)

perform substantially worse, as they evaluate trajectories independently and fail to capture relational deviations between objects. Likewise, density, reconstruction, and visual baselines (logpZO, lopO, AE-Recon, FIDeL) show less consistent performance, suggesting that implicit latent representations struggle to encode structured object interactions robustly. Overall, both TPC and GnnT demonstrate clear benefits from explicitly modeling inter-object relations.

### 4.4.1 End-to-End Failure Detection

We next evaluate the full deployment pipeline, where Conformal Prediction (CP) converts anomaly scores into binary alerts without manual threshold tuning. As shown in Figure 5, both variants outperform prior methods by a clear margin. GnnT achieves the highest MCC on the Watchdog dataset (0.657), while TPC performs best on Bot-Fails (0.483); competing methods such as FIDeL and FAIL-Detect remain below 0.15.

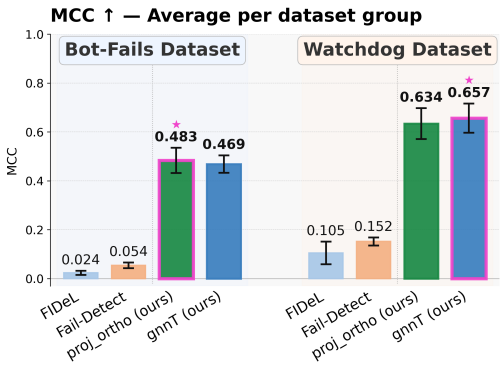


Figure 5: End-to-end failure detection performance (MCC) after conformal calibration, averaged across tasks, seeds, and configurations.

These results show that improvements in raw anomaly scoring transfer reliably to calibrated online detection. The same trend also holds across datasets: TPC performs best in low-data regimes, whereas GnnT benefits from larger datasets and more complex dynamics. An ablation study in Appendix A further isolates the contribution of the graph representation, relational edge modeling, and predictive formulation.

## 5 Conclusion

We introduced Robot-WATCHDOG, a self-supervised framework for robotic failure detection based on spatio-temporal object graphs. To adapt across data regimes, the framework combines a predictive graph-transformer (GnnT) for data-rich settings with a non-parametric trajectory projection method (TPC) for few-shot scenarios. Across both a public benchmark and a newly collected dataset, Robot-WATCHDOG improves anomaly detection performance over existing baselines, particularly on failures involving object interactions and task structure. More broadly, these results highlight object-centric relational representations as a promising direction for scalable real-time monitoring of learned robotic policies.

## 6 Limitations

Robot-WATCHDOG relies on a discrete object-centric abstraction, which limits applicability to tasks involving highly deformable materials, fluids, or interaction domains without clear object decomposition (e.g., painting or soldering). Performance also depends on the upstream vision-language tracking pipeline: severe occlusions, missed detections, or identity switches can degrade graph quality and propagate to the anomaly detector. Finally, our current formulation models object centroids only, and therefore cannot capture failures involving orientation or fine-grained pose. Extending the graph representation with reliable 6D pose estimates is a natural direction for future work.

# References


[1] C. Agia. Deployment-time reliability of learned robot policies, 2026. URL https://arxiv.org/abs/2603.11400.

[2] A. Herrera-Poyatos, J. D. Ser, M. L. de Prado, F.-Y. Wang, E. Herrera-Viedma, and F. Herrera. A framework for responsible ai systems: Building societal trust through domain definition, trustworthy ai design, auditability, accountability, and governance. 2025. URL https://api.semanticscholar.org/CorpusID:276885181.

[3] M. Valdenegro-Toro and R. Stoykova. The dilemma of uncertainty estimation for general purpose ai in the eu ai act, 2024. URL https://arxiv.org/abs/2408.11249.

[4] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL https://arxiv.org/abs/2303.04137.

[5] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL https://arxiv.org/abs/2304.13705.

[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In arXiv preprint arXiv:2307.15818, 2023.

[7] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246, 2024.

[8] P. Intelligence, B. Ai, A. Amin, R. Aniceto, A. Balakrishna, G. Balke, K. Black, G. Bokinsky, S. Cao, T. Charbonnier, V. Choudhary, F. Collins, K. Conley, G. Connors, J. Darpinian, K. Dhabalia, M. Dhaka, J. DiCarlo, D. Driess, M. Equi, A. Esmail, Y. Fang, C. Finn, C. Glossop, T. Godden, I. Goryachev, L. Groom, H. Habeeb, H. Hancock, K. Hausman, G. Hussein, V. Hwang, B. Ichter, C. Jacobsen, S. Jakubczak, R. Jen, T. Jones, G. Kammerer, B. Katz, L. Ke, M. Khadikov, C. Kuchi, M. Lamb, D. LeBlanc, B. LeCount, S. Levine, X. Li, A. Li-Bell, V. Lialin, Z. Liang, W. Lim, Y. Lu, E. Luo, V. Mano, N. Marwaha, A. Mongush, L. Murphy, S. Nair, T. Patterson, K. Pertsch, A. Z. Ren, G. Schelske, C. Sharma, B. Shi, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, W. Stoeckle, J. Tang, J. Tanner, S. Tekeste, M. Torne, K. Vedder, Q. Vuong, A. Walling, H. Wang, J. Wang, X. Wang, C. Whalen, S. Whitmore, B. Williams, C. Xu, S. Yoo, L. Yu, W. Zhang, Z. Zhang, and U. Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026. URL https://arxiv.org/abs/2604.15483.

[9] NVIDIA, J. Bjorck, N. C. Fernando Castañeda, X. Da, R. Ding, L. J. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In ArXiv Preprint, March 2025.

[10] C. Xu, T. K. Nguyen, E. Dixon, C. Rodriguez, P. Miller, R. Lee, P. Shah, R. Ambrus, H. Nishimura, and M. Itkina. Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies, 2025. URL https://arxiv.org/abs/2503.08558.

[11] Q. Rolland, F. Mayran de Chamisso, and J.-B. Mouret. Failure identification in imitation learning via statistical and semantic filtering. In IEEE International Conference on Robotics and Automation (ICRA), 2026.

[12] R. Römer, A. Kobras, L. Worbis, and A. P. Schoellig. Failure prediction at runtime for generative robot policies, 2025. URL https://arxiv.org/abs/2510.09459.

[13] S. Thoduka, J. Gall, and P. G. Ploger. Using visual anomaly detection for task execution monitoring. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), page 4604–4610. IEEE, 2021. doi:10.1109/iros51168.2021.9636133. URL http://dx.doi.org/10.1109/IROS51168.2021.9636133.

[14] Q. Gu, Y. Ju, S. Sun, I. Gilitschenski, H. Nishimura, M. Itkina, and F. Shkurti. Safe: Multitask failure detection for vision-language-action models. arXiv preprint arXiv:2506.09937, 2025.

[15] S. Zhou, B. Zhu, J. Yang, X. Zhao, J. Chen, and Y.-G. Jiang. Rc-nf: Robot-conditioned normalizing flow for real-time anomaly detection in robotic manipulation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2026.

[16] E. D. Lello, M. Klotzbücher, T. D. Laet, and H. Bruyninckx. Bayesian time-series models for continuous fault detection and recognition in industrial robotic tasks. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 5827–5833, 2013. URL https://api.semanticscholar.org/CorpusID:15269134.

[17] N. Grambow, L.-M. Fenner, F. Kempkes, P. Hotz, D. Wan, J. Krüger, and K. Haninger. Anomaly detection for generic failure monitoring in robotic assembly, screwing and manipulation, 2026. URL https://arxiv.org/abs/2509.26308.

[18] A. Vemuri, M. Polycarpou, and S. Diakourtis. Neural network based fault detection in robotic manipulators. IEEE Transactions on Robotics and Automation, 14(2):342–348, 1998. doi:10.1109/70.681254.

[19] K. M. K. G. e. a. Khalastchi, E. Online data-driven anomaly detection in autonomous robots, 2015.

[20] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. ACM Comput. Surv., 41(3), July 2009. ISSN 0360-0300. doi:10.1145/1541880.1541882. URL https://doi.org/10.1145/1541880.1541882.

[21] Y. Zhu, A. Lim, P. Stone, and Y. Zhu. Vision-based manipulation from single human video with open-world object graphs, 2025. URL https://arxiv.org/abs/2405.20321.

[22] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel. Neural relational inference for interacting systems, 2018. URL https://arxiv.org/abs/1802.04687.

[23] P. W. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu. Interaction networks for learning about objects, relations and physics, 2016. URL https://arxiv.org/abs/1612.00222.

[24] H. Jiang, B. Huang, R. Wu, Z. Li, S. Garg, H. Nayyeri, S. Wang, and Y. Li. Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation, 2024. URL https://arxiv.org/abs/2402.15487.

[25] H. Huang, M. Cen, K. Tan, X. Quan, G. Huang, and H. Zhang. Graphcot-vla: A 3d spatial-aware reasoning vision-language-action model for robotic manipulation with ambiguous instructions, 2025. URL https://arxiv.org/abs/2508.07650.

[26] Y. Huang, A. Conkey, and T. Hermans. Planning for multi-object manipulation with graph neural network relational classifiers, 2023. URL https://arxiv.org/abs/2209.11943.

[27] R. Vakhitov, L. Ugadiarov, and A. Panov. Object-Centric World Models Meet Monte Carlo Tree Search, page 481–491. Springer Nature Switzerland, Dec. 2025. ISBN 9783032136121. doi:10.1007/978-3-032-13612-1_42. URL http://dx.doi.org/10.1007/978-3-032-13612-1_42.

[28] V. Vosylius and E. Johns. Instant policy: In-context imitation learning via graph diffusion, 2025. URL https://arxiv.org/abs/2411.12633.

[29] C. Gokmen, D. Ho, and M. Khansari. Asking for help: Failure prediction in behavioral cloning through value approximation, 2023. URL https://arxiv.org/abs/2302.04334.

[30] H. Liu, S. Dass, R. Martín-Martín, and Y. Zhu. Model-based runtime monitoring with interactive imitation learning, 2023. URL https://arxiv.org/abs/2310.17552.

[31] 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, 2024. URL https://arxiv.org/abs/2410.22689.

[32] C. Agia, R. Sinha, J. Yang, Z. ang Cao, R. Antonova, M. Pavone, and J. Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress, 2024. URL https://arxiv.org/abs/2410.04640.

[33] Qwen, :, A. Yang, and B. Y. et al. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[34] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714, 2024. URL https://arxiv.org/abs/2408.00714.

[35] S. Brody, U. Alon, and E. Yahav. How attentive are graph attention networks? In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=F72ximsx7C1.

[36] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman. Distribution-free predictive inference for regression, 2017. URL https://arxiv.org/abs/1604.04173.

[37] J. Diquigiovanni, M. Fontana, and S. Vantini. The importance of being a band: Finite-sample exact distribution-free prediction sets for functional data, 2021. URL https://arxiv.org/abs/2102.06746.

[38] R. Cadene, S. Alibert, A. Soare, Q. Gallouedec, A. Zouitine, and T. Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

[39] T. Eiter and H. Mannila. Computing discrete fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, 1994.

[40] M. Cuturi and M. Blondel. Soft-dtw: a differentiable loss function for time-series. In Proceedings of the 34th International Conference on Machine Learning, pages 894–903. PMLR, 2017.

[41] Y. Shi, J. Yang, and Z. Qi. Unsupervised anomaly segmentation via deep feature reconstruction. Neurocomputing, 2021.

[42] M. Hasan et al. Learning temporal regularity in video sequences. In Proc. of CVPR, 2016.

[43] T. Wang et al. Generative neural networks for anomaly detection in crowded scenes. IEEE TIFS, 2018.

[44] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

# Appendix


**A Ablation Study** **13**

A.1 Impact of Object-Centric Graph Representation . . . . . 13

A.2 Impact of Relational Modeling . . . . . 14

A.3 Impact of Spatio-Temporal Modeling . . . . . 15

A.4 Prediction vs Density Estimation . . . . . 16

**B Method Details** **17**

B.1 Trajectory Extraction . . . . . 17

B.2 GnnT - Model Architecture . . . . . 21

B.3 Trajectory Projection and Correlation (TPC) Formulation . . . . . 22

B.4 Conformal Prediction . . . . . 23

**C Experimental Details** **24**

C.1 Datasets . . . . . 24

C.2 Baselines . . . . . 26

C.3 Evaluation Metrics . . . . . 28

**D Additional results** **29**

D.1 Impact of the number of demonstration on the results . . . . . 29

D.2 Results details per task . . . . . 29

# A Ablation Study

We conduct a series of ablation experiments to quantify the contribution of each component of the proposed framework. Our analysis focuses on (i) the input representation, (ii) relational modeling, (iii) the spatio-temporal architecture, (iv) the predictive formulation.

## A.1 Impact of Object-Centric Graph Representation

We evaluate whether representing the scene as an object-centric graph provides a measurable advantage over alternative representations.

In table 1, we compare three configurations:

**Vision feature baseline.** To isolate the contribution of our Spatio-Temporal Graph Neural Network (GnnT) and its object-centric inductive bias, we introduce a purely sequential vision-based baseline that bypasses the graph construction phase. In this ablation, the demonstration episode is modeled as a flat sequence of global observations. At each timestep $t$, dense visual features extracted by a pre-trained encoder $\mathcal{F}_\theta$ (Resnet18 [44]) are concatenated with proprioceptive data to form a state vector $\mathbf{x}_t \in \mathbb{R}^{d_{in}}$. These states are linearly projected into a hidden dimension $D$, augmented with sinusoidal positional encodings, and processed by a multi-layer Transformer Encoder to capture global temporal dependencies across the sequence. The resulting output tokens are then mapped back to the original feature space via an MLP head to reconstruct the input states $\mathbf{x}_t$. Trained exclusively on nominal data using a Mean Squared Error (MSE) objective, this auto-encoding baseline utilizes the state-wise reconstruction error as its failure score during inference. Results are presented in table 1 under the name *Vision features*.

**Trajectory-only representation.** We evaluate a graph-free ablation model, denoted as the Trajectory Transformer. This baseline deliberately removes the Graph Neural Network (GNN) encoder and all explicit edge-feature computations. Specifically, for each object $c \in \{1, \dots, C\}$ at time step $t \in \{1, \dots, m\}$, its kinematic state $x_c^t \in \mathbb{R}^2$ is concatenated with a learned semantic embedding $e_c \in \mathbb{R}^{d_{sem}}$ and linearly projected to a hidden token representation: $h_c^t = \mathbf{W}[x_c^t \parallel e_c] + \mathbf{b}$, where $\mathbf{W} \in \mathbb{R}^{d_h \times (2+d_{sem})}$ is the projection weight matrix and $\parallel$ denotes concatenation. Following the addition of standard temporal positional encodings, the flattened sequence of independent object tokens is processed by a multi-head Transformer encoder. The future trajectories $\hat{Y} \in \mathbb{R}^{L \times C \times 2}$ are regressed directly from the final sequence embeddings via a Multi-Layer Perceptron (MLP) head. Unlike the full model, this baseline is optimized exclusively using a Mean Squared Error (MSE) loss on absolute coordinates, without any topological constraints. By stripping away structural priors, this ablation forces the self-attention mechanism to independently discover and model inter-object physical correlations solely from raw trajectories.

**Graph-based representation (ours).** Object trajectories are structured as spatio-temporal graphs and processed by the proposed GNN-Transformer architecture, see section **3-Method** of the paper for more details.

The results in Table 1 demonstrate the critical role of the graph representation in our framework. The global vision feature baseline struggles significantly, particularly on the more complex "Table-setting" task, indicating that unstructured, dense visual features lack the precise physical and spatial inductive biases required for robust failure detection. While extracting explicit object trajectories yields a substantial performance jump over raw vision features, this graph-free approach still falls short. By structuring the scene as a spatio-temporal graph, our full model successfully captures the essential physical interactions and topological constraints between objects. This explicit relational modeling provides a modest but consistent improvement in the "Waste-sorting" task and leads to massive gains in the "Table-setting" task, increasing the True Positive Rate by nearly 30% and the Matthews Correlation Coefficient by over 35% compared to the independent trajectory baseline.

| Representation | Waste-sorting | | | | Table-setting | | | |
|---|---|---|---|---|---|---|---|---|
| | TPR ↑ | TNR ↑ | AUPR ↑ | MCC ↑ | TPR ↑ | TNR ↑ | AUPR ↑ | MCC ↑ |
| Vision features | 29.7 ± 0.2 | 84.3 ± 1.7 | 29.7 ± 2.4 | 16.7 ± 1.8 | 30.6 ± 0.6 | 0.0 ± 0.0 | 23.3 ± 1.2 | 0.0 ± 0.0 |
| Trajectories | 58.7 ± 1.2, | 92.9 ± 0.8 | 54.7 ± 2.5 | 56.5 ± 1.9 | 54.5 ± 1.6 | 90.9 ± 1.2 | 54.5 ± 1.7 | 49.3 ± 1.7 |
| **Graph (Ours)** | **61.0** ± 0.9 | **93.1** ± 1.1 | **57.0** ± 2.2 | **58.7** ± 1.7 | **84.0** ± 0.7 | **98.1** ± 0.2 | **90.9** ± 1.4 | **85.0** ± 1.0 |

Table 1: Quantitative evaluation of different input representations on failure detection performance. We compare our proposed object-centric graph representation against a global vision feature baseline and a graph-free trajectory-only model across the WATCHDOG dataset. Metrics reported include True Positive Rate (TPR), True Negative Rate (TNR), Area Under the Precision-Recall Curve (AUPR), and Matthews Correlation Coefficient (MCC). All values are reported as percentages, with higher values (↑) indicating better performance. The best results for each metric are highlighted in bold.

### A.2 Impact of Relational Modeling

We evaluate the importance of explicitly modeling interactions between objects for both the predictive (GnnT) and non-parametric (TPC) formulations.

**For the GnnT method** To assess the contribution of relational inductive bias, we construct a randomized graph baseline by perturbing the edge structure while preserving node features. Formally, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{E}| = E$, we sample a randomized edge set $\tilde{\mathcal{E}}$ by drawing $E$ pairs $(i, j)$ uniformly from $\mathcal{V} \times \mathcal{V}$, yielding a graph $\tilde{\mathcal{G}} = (\mathcal{V}, \tilde{\mathcal{E}})$. The node representations are then computed as

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}_{\tilde{\mathcal{G}}}(i)} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}\right),$$

where $\alpha_{ij}^{(l)}$ denotes attention coefficients as in GATv2 [35]. This procedure preserves the computational structure of the model while destroying meaningful spatial correlations, enabling a controlled evaluation of the role of relational structure.

**For the TPC method** To disentangle the contribution of geometric versus relational cues in TPC, we evaluate three variants of the anomaly score:

- **Orthogonal projection only.** Each object trajectory is compared independently to expert demonstrations using the weighted orthogonal projection distance. This captures local kinematic deviations but ignores interactions between objects.
- **Correlation only.** We discard absolute trajectories and instead compare the pairwise correlation structure between object motions. Concretely, for each pair of objects, we compute the Pearson correlation of their $x$ and $y$ trajectories, and penalize deviations from the correlation patterns observed in expert demonstrations. This captures relational consistency but is invariant to global geometric shifts.
- **Full TPC (projection + correlation).** The final score combines both terms, aggregating per-object projection errors with a global correlation penalty. This jointly captures absolute trajectory deviations and interaction-level inconsistencies.

**Results and Interpretation** The results in Table 2 highlight distinct roles of geometric and relational signals in failure detection, as well as their task-dependent interplay. For GnnT, removing the edge model consistently degrades performance, most notably in TNR and MCC, indicating that relational structure is essential for maintaining robustness to benign variations and for preserving discriminative power; modeling objects independently leads to increased sensitivity to noise and reduced reliability. For TPC, the orthogonal projection term alone achieves strong performance on *Waste-sorting*, suggesting that failures in this task are predominantly characterized by deviations from nominal trajectories. In contrast, the correlation-only variant attains the highest TPR

| Method | Waste-sorting | | | | Table-setting | | | |
|---|---|---|---|---|---|---|---|---|
| | TPR ↑ | TNR ↑ | AUPR ↑ | MCC ↑ | TPR ↑ | TNR ↑ | AUPR ↑ | MCC ↑ |
| **GnnT** | 61.0 ± 0.9 | **93.1** ± 1.1 | **57.0** ± 2.2 | **58.7** ± 1.7 | **84.0** ± 0.7 | **98.1** ± 0.2 | **90.9** ± 1.4 | **85.0** ± 1.0 |
| w/o Edge model | **73.7** ± 5.6 | 80.9 ± 8.3 | 48.5 ± 5.2 | 49.0 ± 6.3 | 81.8 ± 6.9 | 86.8 ± 1.0 | 59.3 ± 9.3 | 61.9 ± 6.3 |
| Orthogonal projection only | **89.2** ± 0 | **87.3** ± 0 | **67.7** ± 0 | **61.0** ± 0 | 83.7 ± 0 | 95.1 ± 0 | 92.5 ± 0 | 78.2 ± 0 |
| Correlation only | 82.1 ± 0 | 86.1 ± 0 | 65.1 ± 0 | 53.9 ± 0 | **94.9** ± 0 | 91.2 ± 0 | 90.8 ± 0 | 81.3 ± 0 |
| Both **(TPC)** | 86.1 ± 0 | 85.9 ± 0 | 63.9 ± 0 | 55.2 ± 0 | **94.9** ± 0 | **95.2** ± 0 | **92.8** ± 0 | **85.3** ± 0 |

Table 2: Ablation study on relational modeling for both GnnT and TPC. For GnnT, removing or corrupting edge-based interactions degrades performance, confirming that structured object relationships are important for robust prediction and failure detection. For TPC, geometric (projection) and relational (correlation) components capture complementary aspects of anomalies: projection is sufficient for trajectory-level deviations in simpler tasks (*Waste-sorting*), while correlation improves detection of interaction-level errors in more relational settings (*Table-setting*). Combining both signals yields the most balanced performance in relationally complex tasks, but does not uniformly dominate in all regimes, highlighting the task-dependent role of relational modeling.

on *Table-setting*, indicating that many failures arise from incorrect coordination between objects despite locally plausible motions. Combining both signals yields the best overall performance on *Table-setting*, improving MCC while maintaining high TPR and TNR, which confirms that geometric deviations and relational inconsistencies provide complementary information. However, this combination does not consistently outperform projection-only on *Waste-sorting*, suggesting that in simpler tasks dominated by spatial accuracy, relational modeling may introduce limited additional benefit. Overall, these results indicate that effective failure detection requires capturing both absolute trajectory deviations and interaction-level structure, with their relative importance depending on the semantic complexity of the task.

### A.3 Impact of Spatio-Temporal Modeling

To rigorously evaluate the contribution of our combined GNN–Transformer architecture, we conduct an ablation study isolating the spatial and temporal inductive biases. We compare the full model against three degraded baselines:

- **Full model (GnnT):** Our proposed architecture, which uses an Edge-aware Graph Attention Network (EGAT) to explicitly model spatial interactions and physical constraints between objects, followed by a Transformer encoder to capture long-term temporal dependencies.
- **Transformer-only (Temporal without Spatial Priors):** The graph encoder is completely removed. Raw object coordinates and semantic embeddings are linearly projected into a hidden space and fed directly to the Transformer. This baseline tests whether the self-attention mechanism alone is sufficient to implicitly discover inter-object physical correlations without the structural priors provided by the GNN.
- **GNN-only (Spatial without Temporal Attention):** The temporal sequence modeling is replaced by a static feed-forward network. Spatial embeddings are still extracted frame-by-frame via the GNN, but they are subsequently concatenated along the temporal axis and mapped to future predictions via a Multi-Layer Perceptron (MLP). This isolates the value of the Transformer's sequence-awareness.
- **MLP (No explicit priors):** A naive baseline where both the GNN and the Transformer are removed. The kinematic data of the scene over the observation window is flattened and processed by a simple MLP. This serves as a lower bound, representing a purely data-driven regression without structural or sequential inductive biases.

**Results and Discussion** The empirical results, summarized in Table 3, demonstrate a clear performance hierarchy: the naive MLP baseline consistently yields the lowest metrics across the board, while the full GnnT architecture achieves the highest. This substantial performance gap directly

Table 3: **Quantitative results of the spatio-temporal architecture ablation.** Metrics are reported as mean ± standard deviation over multiple random seeds. The full GnnT architecture consistently provides the best balance between detection sensitivity and false positive rejection, particularly reflected in the Area Under the Precision-Recall curve (AUPR) and Matthews Correlation Coefficient (MCC).

| | Waste-sorting | | | | Table-setting | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | TPR ↑ | TNR ↑ | AUPR ↑ | MCC ↑ | TPR ↑ | TNR ↑ | AUPR ↑ | MCC ↑ |
| **GnnT** | 61.0 ± 0.9 | **93.1** ± 1.1 | **57.0** ± 2.2 | **58.7** ± 1.7 | 84.0 ± 0.7 | **98.1** ± 0.2 | **90.9** ± 1.4 | **85.0** ± 1.0 |
| Transformer only | 71.7 ± 5.9 | 85.8 ± 4.0 | 53.5 ± 4.7 | 53.1 ± 2.8 | 82.2 ± 6.7 | 86.8 ± 2.1 | 54.7 ± 6.3 | 62.3 ± 6.7 |
| GNN only | 70.8 ± 5.6 | 82.9 ± 5.3 | 51.8 ± 1.4 | 48.6 ± 4.0 | 83.3 ± 3.0 | 92.8 ± 2.8 | 78.1 ± 6.0 | 73.2 ± 4.2 |
| MLP | **75.3** ± 12.8 | 70.6 ± 13.1 | 40.5 ± 2.9 | 39.3 ± 3.6 | **88.7** ± 6.6 | 46.2 ± 7.3 | 31.4 ± 3.7 | 28.8 ± 5.1 |

validates our core hypothesis that explicitly modeling spatio-temporal priors is essential for robust anomaly detection in multi-object physical manipulation.

Crucially, the ablation reveals that isolated spatial and temporal inductive biases exhibit complementary strengths depending on the task's primary failure modes. In the highly structured *Table-setting* task, anomalies frequently stem from incorrect geometric configurations or collisions. Without explicit relational modeling, the Transformer-only baseline struggles to capture these spatial constraints and severely underperforms the GNN-only model (e.g., AUPR of $54.7\%$ vs. $78.1\%$). Conversely, in the *Waste-sorting* task, the topological structure is less critical, and anomalies often manifest as purely kinematic or timing deviations (e.g., Missing an object). In this regime, the temporal blindness of the GNN-only baseline restricts its performance, allowing the sequence-aware Transformer-only model to surpass it (MCC of $53.1\%$ vs. $48.6\%$). By effectively fusing both modalities, the GnnT architecture seamlessly adapts to both spatial and temporal failure regimes, consistently securing the highest detection sensitivity and false-positive rejection across all scenarios.

### A.4 Prediction vs Density Estimation

We compare the predictive formulation with a likelihood-based alternative. Results are presented in table 4.

- **GnnT, Prediction-based (ours).** The model predicts future trajectories and uses prediction error as the failure score.
- **Density-based.** A normalizing flow is trained on latent representations to estimate likelihood.

#### A.4.1 Latent Density Estimation via Normalizing Flows

Instead of directly predicting the future trajectories of objects, we formulate the failure detection task as estimating the exact likelihood of the observed spatio-temporal dynamics. Let $\mathbf{h}_t \in \mathbb{R}^D$ be the latent representation of the scene at time $t$, extracted by the Spatio-Temporal Transformer from the observation window. The distribution of nominal interactions $p_H(\mathbf{h}_t)$ is highly complex and intractable.

To model this density, we employ a Normalizing Flow (NF) parameterized by $\theta$. The flow applies a sequence of invertible, differentiable transformations $f_\theta$ to map the complex latent space $\mathbf{h}_t$ into a simple, tractable base distribution, typically an isotropic multivariate Gaussian $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Using the change of variables formula, the exact log-likelihood of a nominal representation is given by:

$$\log p_H(\mathbf{h}_t) = \log p_Z(f_\theta(\mathbf{h}_t)) + \log \left| \det \left( \frac{\partial f_\theta(\mathbf{h}_t)}{\partial \mathbf{h}_t} \right) \right| \tag{3}$$

where $p_Z$ is the standard Gaussian density function, and the second term is the log-determinant of the Jacobian of the transformation, accounting for the change in volume induced by $f_\theta$.

**Training Objective.** The model is trained end-to-end to maximize the log-likelihood of the nominal data representations. Since $\mathbf{z}_t = f_\theta(\mathbf{h}_t)$ follows a standard normal distribution, maximizing $\log p_H(\mathbf{h}_t)$ is equivalent to minimizing the negative log-likelihood loss $\mathcal{L}_{\text{NF}}$:

$$\mathcal{L}_{\text{NF}} = \frac{1}{2}\|f_\theta(\mathbf{h}_t)\|_2^2 - \log \left| \det \left( \frac{\partial f_\theta(\mathbf{h}_t)}{\partial \mathbf{h}_t} \right) \right| \tag{4}$$

**Inference and failure Scoring.** During deployment, the Normalizing Flow acts as an explicit density estimator. For a new observation window, the model computes the latent representation $\mathbf{h}_t$ and its corresponding negative log-likelihood. The failure score $S_t$ is directly defined as $S_t = \mathcal{L}_{\text{NF}}(\mathbf{h}_t)$. A high score $S_t$ indicates that the observed spatio-temporal interaction pattern has a low probability under the learned nominal distribution.

| | Waste-sorting | | | | Table-setting | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | TPR ↑ | TNR ↑ | AUPR ↑ | MCC ↑ | TPR ↑ | TNR ↑ | AUPR ↑ | MCC ↑ |
| **GnnT (prediction based)** | 61.0 ± 0.9 | **93.1** ± 1.1 | 57.0 ± 2.2 | **58.7** ± 1.7 | 84.0 ± 0.7 | **98.1** ± 0.2 | **90.9** ± 1.4 | **85.0** ± 1.0 |
| Density based | **73.4** ± 4.2 | 84.4 ± 3.2 | **57.5** ± 5.9 | 52.4 ± 4.9 | **87.2** ± 3.5 | 90.4 ± 6.9 | 78.2 ± 11.7 | 72.9 ± 9.24 |

Table 4: Quantitative comparison of failure detection formulations: predictive modeling (GnnT) versus latent density estimation (Normalizing Flows). Mean and standard deviation are reported across 5 different seeds. The predictive approach demonstrates superior robustness (higher MCC and AUPR) and stability (lower variance), effectively minimizing false positives (higher TNR) compared to the density-based baseline.

As shown in Table 4, the predictive formulation (GnnT) substantially outperforms the density estimation baseline in overall robustness and stability. While the density-based approach yields a marginally higher True Positive Rate (TPR), it suffers from a pronounced degradation in True Negative Rate (TNR) and exhibits significantly higher variance across both tasks. This suggests that fitting a normalizing flow to high-dimensional, complex spatio-temporal latents struggles to establish tight decision boundaries, making the model overly sensitive and prone to false positives. Conversely, framing anomaly detection as future trajectory prediction enforces a strict structural prior on nominal behavior. By directly computing errors in the geometric observation space rather than relying on abstract latent likelihoods, our predictive method effectively isolates genuine relational failures from benign variations, resulting in superior MCC and highly stable performance across different environments.

# B Method Details

## B.1 Trajectory Extraction

Object trajectories are extracted from RGB observations using a two-stage pipeline combining vision-language-based object detection and segmentation-based multi-object tracking. The pipeline is designed to be fully automated, requiring only a natural language description of the task to identify and track all task-relevant entities throughout an episode.

**Stage 1: Object detection via VLM (Qwen3-VL)** The first frame of each episode is passed to a vision-language model (Qwen3-VL-8B-Instruct [33]) together with a task-specific prompt that enumerates the expected task-relevant objects. For each object, the model is asked to predict a bounding box in normalised coordinates $[0, 1000]^2$, a fine-grained semantic label, and a coarse-grained label. To improve robustness to occasional parsing failures, the model is queried up to seven times with a temperature annealing schedule ($T_k = 0.10 + (k-1) \times 0.15$, $k \in \{1, \ldots, 7\}$). A response is accepted as valid only if all expected objects are returned with non-degenerate bounding boxes and non-empty labels. If the maximum number of attempts is exceeded, missing detections are flagged. This initialisation step is performed *offline* prior to deployment and therefore imposes no latency on real-time execution. The prompts used for each task are provided below.

**Stage 2: Multi-object tracking via SAM2** The bounding boxes produced in Stage 1 initialise a SAM2 video tracking session [34]. At each subsequent frame, the segmentation masks are propagated using the SAM2 recurrent memory mechanism, and the 2D centroid of each object's mask is recorded as its spatial state $\mathbf{p}_{i,t} \in \mathbb{R}^2$. When the mask of an object degenerates (empty support), the last valid centroid is carried forward (*forward-fill* imputation) so that all trajectories remain temporally aligned, while a validity label is switch from 1 to 0 to notify the missing object. If the object reappears later, the trajectory is interpolated between the last known position and the new position. The robot end-effector is treated as an additional object and tracked through the same pipeline from an initial bounding box provided by the VLM. Illustration of an example of the process is presented figure 6.

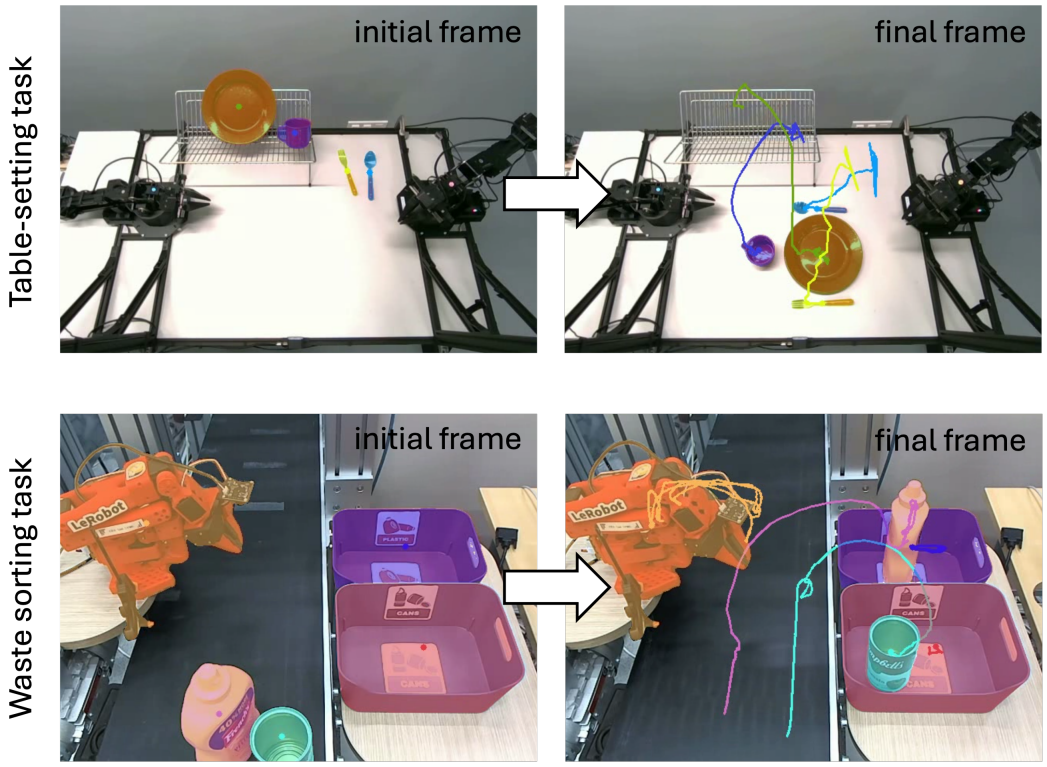


Figure 6: **Trajectory extraction pipeline.** (Left) The initial frame of an episode where task-relevant objects and the robot end-effector are identified using a Vision-Language Model (Stage 1). (Right) The final frame displaying the continuous 2D spatial trajectories extracted by tracking the objects' mask centroids throughout the episode via SAM2 (Stage 2). The top row illustrates the structured *Table-setting* task, while the bottom row demonstrates the dynamic *Waste sorting* task on a moving conveyor belt.

**Real-time deployment characteristics** The VLM initialisation is performed entirely offline; once bounding boxes are obtained, the real-time tracking pipeline relies solely on SAM2. In its optimised streaming configuration, SAM2 operates at approximately 44 fps on an NVIDIA A100 GPU with 5 tracked objects, well above the 30 fps acquisition rate of the camera used in our experiments. The resulting end-to-end perception latency is therefore bounded by the single-frame forward pass of SAM2 and does not accumulate over an episode.

**Task-specific VLM prompts** For reproducibility, we report the exact prompts supplied to Qwen3-VL for each task in our benchmark. Each prompt follows the same schema: a brief scene description, the user task string, a numbered list of the expected objects with their semantic categories, and an instruction to return a strictly formatted JSON response.

Table 5 summarises the object lists used per task. Full prompt strings are reproduced verbatim in the supplementary material.

Table 5: Object lists supplied to Qwen3-VL per task.

| Task | $N_{\text{obj}}$ | Object list |
|---|---|---|
| `domotic_groceriesSorting` | 8 | 4 sweets, 4 vegetables |
| `domotic_vegetablesAndFruitsSorting` | 6 | 3 fruits, 3 vegetables |
| `domotic_dishTidyUp` | 8 | 1 whisk, 3 plates, 3 glasses, 1 cup |
| `domotic_setTheTable` | 4 | 1 glass, 1 plate, 1 fork, 1 knife |
| `sort_trash_conveyor` | 2 | 1 plastic waste, 1 metal can |
| `fiper_sort` | 5 | blue cube, red cube, blue frame, red frame, robot flange |
| `fiper_stacking` | 5 | blue, red, green, yellow cube, robot flange |

All prompts share the following preamble and JSON schema:

**Common preamble and output schema**

```
A robot arm is executing a task.
User task:  "<TASK_DESCRIPTION>"
Please identify this <N> objects in the image and provide their bounding box
as (x1, y1, x2, y2) in image coordinates.
The <N> objects:
  - [object list -- see below per task]
Return your answer strictly in this JSON format and NOTHING ELSE,
make sure to keep objects in the correct order:

{
  "objects":  [
    {
      "fine_label":  "...",
      "coarse_label":  "...",
      "bbox":  [x1, y1, x2, y2],
      "is_related":  true
    }
  ]
}

Ensure the JSON is valid and contains no extra text or explanations.
```

**`domotic_groceriesSorting` (8 objects)**

```
A robot arm is executing a task.
User task:  "<TASK_DESCRIPTION>"
Please identify this 8 objects in the image and provide their bounding box
as (x1, y1, x2, y2) in image coordinates.
The 8 objects:
  - 1.  sweet.
  - 2.  sweet.
  - 3.  sweet.
  - 4.  sweet.
  - 5.  vegetal.
  - 6.  vegetal.
  - 7.  vegetal.
  - 8.  vegetal.
Return your answer strictly in this JSON format and NOTHING ELSE,
make sure to keep objects in the correct order.
```

domotic_vegetablesAndFruitsSorting (6 objects)

```
A robot arm is executing a task.
User task:  "<TASK_DESCRIPTION>"
Please identify this 6 objects in the image and provide their bounding box
as (x1, y1, x2, y2) in image coordinates.
The 6 objects:
  - 1.  fruit.
  - 2.  fruit.
  - 3.  fruit.
  - 4.  vegetable.
  - 5.  vegetable.
  - 6.  vegetable.
Return your answer strictly in this JSON format and NOTHING ELSE,
make sure to keep objects in the correct order.
```

domotic_dishTidyUp (8 objects)

```
A robot arm is executing a task.
User task:  "<TASK_DESCRIPTION>"
Please identify this 8 objects in the image and provide their bounding box
as (x1, y1, x2, y2) in image coordinates.
The 8 objects:
  - 1.  whisk.
  - 2.  plate.
  - 3.  plate.
  - 4.  plate.
  - 5.  glass.
  - 6.  glass.
  - 7.  glass.
  - 8.  cup.
Return your answer strictly in this JSON format and NOTHING ELSE,
make sure to keep objects in the correct order.
```

domotic_setTheTable / aloha_setTheTable_2 (4 objects)

```
A robot arm is executing a task.
User task:  "<TASK_DESCRIPTION>"
Please identify this 4 objects in the image and provide their bounding box
as (x1, y1, x2, y2) in image coordinates.
The 4 objects:
  - 1.  glass.
  - 2.  plate.
  - 3.  fork.
  - 4.  knife.
Return your answer strictly in this JSON format and NOTHING ELSE,
make sure to keep objects in the correct order.
```

sort_trash_conveyor (2 objects)

```
A robot arm is executing a task.
User task:  "<TASK_DESCRIPTION>"
Please identify this 2 objects in the image and provide their bounding box
as (x1, y1, x2, y2) in image coordinates.
The 2 objects:
  - 1.  a plastic waste -- fine_label=plastic.
  - 2.  a can, metal waste -- fine_label=metal.
Return your answer strictly in this JSON format and NOTHING ELSE,
make sure to keep objects in the correct order.
```

`fiper_sort` (5 objects)

```
A robot arm is executing a task.
User task:  "<TASK_DESCRIPTION>"
Please identify this 5 objects in the image and provide their bounding box
as (x1, y1, x2, y2) in image coordinates.
The 5 objects:
  - 1.  blue_cube.
  - 2.  red_cube.
  - 3.  blue_frame.
  - 4.  red_frame.
  - 5.  robot_flange.
Return your answer strictly in this JSON format and NOTHING ELSE,
make sure to keep objects in the correct order.
```

`fiper_stacking` (5 objects)

```
A robot arm is executing a task.
User task:  "<TASK_DESCRIPTION>"
Please identify this 5 objects in the image and provide their bounding box
as (x1, y1, x2, y2) in image coordinates.
The 5 objects:
  - 1.  blue_cube.
  - 2.  red_cube.
  - 3.  green_cube.
  - 4.  yellow_cube.
  - 5.  robot_flange.
Return your answer strictly in this JSON format and NOTHING ELSE,
make sure to keep objects in the correct order.
```

### B.2 GnnT - Model Architecture

The spatial encoder is implemented using a Graph Attention Network (GATv2), while temporal dependencies are modeled using a multi-layer Transformer encoder. The prediction head consists of a shared MLP applied independently to each object embedding.

**Background** Spatio-temporal forecasting in multi-agent or multi-object scenarios requires simultaneously resolving relational structures and temporal dynamics. While standard Graph Neural Networks (GNNs) effectively capture spatial topologies, they lack mechanisms for long-term sequence modeling. Conversely, Transformers excel at sequential processing but lack structural priors for geometric arrangements. The *GnnT* (Graph Neural Network - Transformer) architecture marries these paradigms, utilizing an edge-aware spatial encoder to independently process intra-frame geometry, followed by a sequence model to capture inter-frame kinematics.

**Hypothesis** We hypothesize that nominal robotic manipulation is governed by strict, predictable physical laws and interaction patterns (e.g., collisions, grasping semantics, kinematic constraints). By explicitly forcing a network to predict future states from an observation window, it should internalize these underlying rules. The dynamic attention mechanism of the GATv2 is expected to learn to attend heavily to interacting entities (e.g., the end-effector approaching a target), while the Transformer learns temporal delays, inertia, and task phases. Consequently, mechanical or procedural failures will intrinsically violate these learned spatio-temporal priors, manifesting as a sharp spike in the prediction error ($\mathcal{L}_{\text{pos}}$ and $\mathcal{L}_{\text{edge}}$), which serves as a highly discriminative anomaly signal.

**Implementation details** The model takes as input a sequence of $m$ graphs containing $C$ objects. Let the raw kinematic state of object $i$ at time $t$ be $\mathbf{x}_{i,t} \in \mathbb{R}^2$. To provide semantic identity, each object's discrete category $c_i \in \{1, \dots, C\}$ is mapped to a continuous embedding $\mathbf{e}_i \in \mathbb{R}^{16}$. During training, we inject Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ with $\sigma = 0.01$ to the coordinates to prevent over-

fitting and encourage the learning of robust physical dynamics rather than trivial identity mappings. The initial node feature vector is thus the concatenation:

$$\mathbf{v}_{i,t}^{(0)} = [\,(\mathbf{x}_{i,t} + \epsilon) \parallel \mathbf{e}_i\,] \in \mathbb{R}^{18}$$

The spatial encoder employs a two-layer Edge-aware GATv2 (EGAT). Let $\mathbf{V}_t^{(l)}$ and $\mathbf{E}_t$ denote the set of node and edge features at time $t$ for layer $l$. The spatial embedding is refined iteratively:

$$\mathbf{v}_{i,t}^{(1)} = \text{ELU}\left(\text{EGAT}^{(1)}\left(\mathbf{v}_{i,t}^{(0)}, \mathbf{E}_t\right)\right), \quad \mathbf{v}_{i,t}^{(2)} = \text{ELU}\left(\text{EGAT}^{(2)}\left(\mathbf{v}_{i,t}^{(1)}, \mathbf{E}_t\right)\right)$$

where $\text{ELU}(\cdot)$ is the Exponential Linear Unit activation used to maintain non-linear representational capacity while preventing dying gradients, and $\mathbf{v}_{i,t}^{(2)} \in \mathbb{R}^{d_h}$ (with $d_h = 128$) represents the spatially contextualized node embedding.

To model sequence dynamics, standard sinusoidal temporal positional encodings $\mathbf{p}_t \in \mathbb{R}^{d_h}$ are added to the spatial embeddings: $\tilde{\mathbf{h}}_{i,t} = \mathbf{v}_{i,t}^{(2)} + \mathbf{p}_t$. The sequence of $m$ spatial graphs is then flattened into a single contiguous token sequence:

$$\mathcal{S}_{in} = \left[\tilde{\mathbf{h}}_{1,t-m+1}, \ldots, \tilde{\mathbf{h}}_{C,t-m+1}, \ldots, \tilde{\mathbf{h}}_{1,t}, \ldots, \tilde{\mathbf{h}}_{C,t}\right] \in \mathbb{R}^{mC \times d_h}$$

This flattened topology permits unrestricted, fine-grained self-attention across both the temporal and spatial axes. The sequence is processed by a 4-layer Transformer Encoder to yield $\mathcal{S}_{out} = \text{Transformer}(\mathcal{S}_{in})$.

Finally, we extract only the $C$ tokens corresponding to the most recent observation at timestep $t$ (i.e., the last $C$ elements of $\mathcal{S}_{out}$). Let $\mathbf{z}_{i,t} \in \mathbb{R}^{d_h}$ denote this output token for object $i$. The future trajectory over horizon $L$ is directly regressed via a two-layer Multi-Layer Perceptron (MLP) with a ReLU activation:

$$\widehat{\mathbf{Y}}_{i,t} = \mathbf{W}_2 \max(0, \mathbf{W}_1 \mathbf{z}_{i,t} + \mathbf{b}_1) + \mathbf{b}_2 \quad \in \mathbb{R}^{L \times 2}$$

where $\mathbf{W}_1 \in \mathbb{R}^{2d_h \times d_h}$ and $\mathbf{W}_2 \in \mathbb{R}^{2L \times 2d_h}$ are the shared projection weights applied independently to each object token.

**Hyperparameters** The architecture and training routine are governed by the hyperparameters detailed in Table 6. These values were selected empirically to balance the temporal receptive field, representational capacity, and computational efficiency.

Table 6: Hyperparameters for the *GnnT* architecture and training routine.

| **Category** | **Parameter** | **Value** |
|---|---|---|
| *Sequence* | Observation window ($m$) | 20 |
| | Prediction horizon ($L$) | 15 |
| *Architecture* | Hidden dimension ($d_h$) | 128 |
| | Spatial Encoder layers (GATv2) | 2 |
| | Spatial Encoder attention heads | 8 |
| | Temporal Encoder layers (Transformer) | 4 |
| | Temporal Encoder attention heads | 8 |
| | Tokenization format | Batch-first |
| *Optimization & Training* | Optimizer | AdamW |
| | Learning rate | $10^{-3}$ |
| | Weight decay | $10^{-4}$ |
| | Batch size | 256 |
| | Total epochs | 100 |

### B.3 Trajectory Projection and Correlation (TPC) Formulation

The TPC method evaluates kinematic and relational anomalies by comparing inference trajectories against a set of nominal expert demonstrations. Let a tracked object's trajectory of length $T$ be

defined as $P = \{p_1, \ldots, p_T\}$ with $p_t \in \mathbb{R}^2$, and let the set of nominal expert trajectories for the corresponding object class be $\mathcal{E} = \{V^{(1)}, \ldots, V^{(K)}\}$. Prior to distance computation, trajectories are filtered to remove static segments, ensuring the evaluation focuses strictly on kinematic progression.

**1. Projection-Based Kinematic Scoring:** The spatial deviation of a point $p_t$ from an expert trajectory $V$ (modeled as a piecewise linear polyline) is defined by the orthogonal point-to-polyline distance, denoted $d(p_t, V)$. The spatial failure at time $t$ is taken as the distance to the nearest expert:

$$d^*(p_t) = \min_{V \in \mathcal{E}} d(p_t, V) \tag{5}$$

To account for the inherent natural variability of the task, these distances are standard-normalized using the empirical mean $\mu$ and standard deviation $\sigma$ of the pairwise projection distances computed within the expert set itself:

$$\hat{d}(p_t) = \frac{|d^*(p_t) - \mu|}{\sigma + \epsilon} \tag{6}$$

where $\epsilon$ is a small constant for numerical stability. The individual projection score aggregates these normalized deviations over time. To penalize errors that compound towards the end of the execution, temporal weights $w_t$ (e.g., derived from an exponential decay function) are applied:

$$S_{\text{proj}} = \sum_{t=1}^{T} w_t \hat{d}(p_t) \tag{7}$$

**2. Relational Correlation Penalty:** To capture higher-level interaction failures, the method evaluates the synchronized movement between objects. For any pair of objects $i$ and $j$ resampled to a uniform length $T$, let $\rho(x^i, x^j)$ and $\rho(y^i, y^j)$ be the Pearson correlation coefficients of their respective spatial coordinates over time. The pairwise spatial correlation is averaged across dimensions:

$$C_{i,j} = \frac{1}{2}\big(\rho(x^i, x^j) + \rho(y^i, y^j)\big) \tag{8}$$

The relational failure penalty is computed as the $\ell_1$ distance between the inference correlation matrix ($C^{\text{inf}}$) and the reference correlation matrix ($C^{\text{exp}}$) derived from the expert dataset:

$$S_{\text{corr}} = \sum_{i=1}^{N} \sum_{j>i} \left| C_{i,j}^{\text{inf}} - C_{i,j}^{\text{exp}} \right| \tag{9}$$

where $N$ is the number of tracked objects.

**3. Final failure Score:** Depending on the configuration, the method can evaluate purely projection-based errors, purely relational errors, or a combination of both. The combined failure score is the sum of the individual projection scores and the relational penalty:

$$S_{\text{total}} = S_{\text{proj}} + S_{\text{corr}} \tag{10}$$

### B.4 Conformal Prediction

#### B.4.1 Calibration Data Quality and Curation

Calibrating the decision threshold $\eta$ requires a reference set of $N$ successful, nominal policy rollouts, denoted as $\mathcal{Z}_N = \{\tau^{(1)}, \ldots, \tau^{(N)}\}$. The qualitative integrity of this calibration data is paramount. Trajectories in which the task ultimately succeeds but exhibits undesirable or unsafe intermediate behavior (e.g., severe kinematic jitter, suboptimal grasps, or minor environmental collisions) must be strictly filtered out. Retaining marginally successful trajectories would falsely widen the bounds of the acceptable nominal distribution, thereby degrading the detector's test-time sensitivity and increasing the risk of false negatives. Furthermore, we construct $\mathcal{Z}_N$ entirely from hold-out human expert demonstrations rather than autonomous policy rollouts. While calibrating directly on policy-induced rollouts is possible, it often introduces covariate shift—the policy's self-induced states inherently inflate the expected prediction errors, which artificially raises the threshold and degrades sensitivity.

### B.4.2 Sequential Threshold Construction

To adapt CP for sequential, real-time failure detection over a finite-horizon task, we construct a one-sided, scalar prediction band. Let $S_t^{(i)}$ denote the non-conformity score produced by our framework (either GnnT or TPC) at timestep $t$ for the $i$-th trajectory in $\mathcal{Z}_N$. Because our objective is to flag anomalies, we are exclusively concerned with high scores indicating out-of-distribution (OOD) behavior.To ensure the false positive guarantee holds over the entire duration of a rollout rather than just at a marginal timestep, we evaluate the maximum deviation observed per calibration rollout. For each trajectory $i$, we extract the maximum score over its full length $T$:

$$S_{max}^{(i)} = \max_{t \in [1,T]} S_t^{(i)}$$

The global decision threshold $\eta$ is then computed as the empirical $(1-\alpha)$-quantile of the maximum score collection $\{S_{max}^{(1)}, \ldots, S_{max}^{(N)}\}$. By evaluating the quantile over the maximum trajectory scores, we adapt the bandwidth to safely encapsulate the entirety of non-extreme, nominal functional behaviors.

### B.4.3 Online Deployment and Evaluation

At test time, given a new ongoing trajectory $\tau_{test}$ evolving up to the current timestep $t$, the framework continuously computes the instantaneous anomaly score $S_t$. We define the sequential decision rule as an indicator function $\mathbb{1}(S_t > \eta)$. If this condition is met, the trajectory breaches the established prediction band, and a failure is flagged at the earliest possible timestep. By nature of the conformal guarantees, the probability that a perfectly nominal, in-distribution trajectory is falsely flagged as a failure at any point during its execution is statistically bounded to a maximum of $\alpha$.

# C Experimental Details

## C.1 Datasets

We evaluate Robot-WATCHDOG on two datasets: the BotFails benchmark [11] and a new dataset collected for this work. Together, they provide complementary coverage of controlled failure scenarios and more realistic, dynamically varying environments.

**BotFails.** BotFails is a recently introduced dataset for robotic failure detection, collected via teleoperation on a real robotic platform. It includes multimodal observations (vision, proprioception, and language) across 10 tasks spanning domestic and industrial settings, with annotations for both general anomalies and task-relevant failures.

In our evaluation, we select a subset of tasks compatible with our object-centric assumptions. Specifically, we retain tasks involving rigid, non-deformable, and visually distinguishable objects, enabling consistent tracking and relational reasoning. Tasks involving fluids or deformable materials (e.g., pouring) are excluded, as they violate the assumption of persistent object identity.

**Ours.** We introduce a new dataset, see figure 7, composed of two manipulation tasks designed to increase both relational and dynamic complexity:

- **Table-setting:** a structured assembly task where the robot must place objects (e.g., plates, utensils) in predefined spatial configurations and correct temporal order. This task emphasizes relational correctness (object identity, placement location, ordering).
- **Waste sorting:** a dynamic task where the robot sorts objects on a moving conveyor belt. Unlike static tabletop setups, object positions evolve continuously, introducing variability in initial conditions and interaction timing.

For each task, we collect two types of datasets: (i) *expert demonstrations*, used for training, and (ii) *failure datasets*, used for evaluation and containing both nominal and failure trajectories.

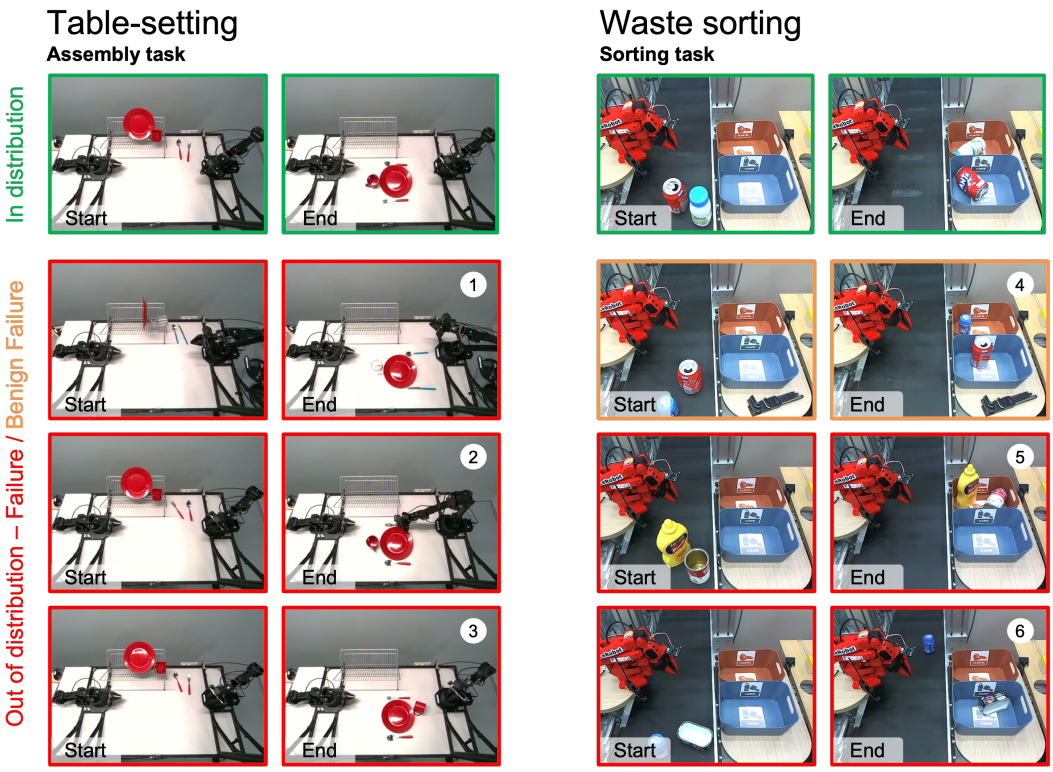


Figure 7: **Overview of the Robot-WATCHDOG evaluation dataset.** The dataset comprises a structured Table-setting task (left) and a dynamic Waste sorting task (right). The top row illustrates in-distribution, nominal expert demonstrations. The subsequent rows depict out-of-distribution scenarios, categorized into critical task failures (red borders) and benign visual variations (orange borders). Specific evaluation scenarios include: (1) Spatial misplacement, where objects are arranged with an incorrect offset; (2) Object inversion, where the target locations of items are swapped; (3) Grasping failure, resulting in a dropped object; (4) Benign visual variation featuring an intrusive distractor object—this highlights our method's robustness to visual clutter, which typically triggers false positives in baseline anomaly detectors; (5) identity sorting error, where an item is placed into the incorrect bin; and (6) Missed object, where the robot fails to make a decision or initiate a grasp.

Table 7: Statistics of our dataset.

| Task | Split | Data | Robot | # Episodes | # Frames | FPS | Cameras | Action dim |
|---|---|---|---|---|---|---|---|---|
| Table-setting | Expert | Video RGB/Proprio/text | ALOHA | 100 | 67,341 | 15 | 4 views | 9 |
| Table-setting | Test | Video RGB/Proprio/text | ALOHA | 21 | 13,471 | 15 | 4 views | 9 |
| Waste sorting | Expert | Video RGB/Proprio/text | SO-100 | 100 | 59,562 | 30 | Top view | 6 |
| Waste sorting | Test | Video RGB/Proprio/text | SO-100 | 51 | 30,378 | 30 | Top view | 6 |

**Data collection and modalities.** The table-setting task is performed using an ALOHA robot, while the waste-sorting task uses a SO-100 platform. Observations include RGB video streams and proprioceptive states (joint positions), along with corresponding action commands. The table-setting setup includes multiple camera viewpoints (external and wrist-mounted), whereas the sorting task uses a top-down camera.

**Discussion.** Our dataset complements BotFails in two key aspects. First, it provides a larger number of expert demonstrations, which is beneficial for training predictive models of scene dynamics. Second, it introduces more challenging conditions for failure detection, particularly in the sorting task where object motion induces continuous distribution shift. This setup allows us to evaluate

Table 8: Overview of the anomaly detection baselines compared in this work, including our proposed **TPC** (few-shot) and **GnnT** (predictive graph) approaches.

| Method | Input Space | Core Mechanism | Spatial Rel. | Temporal Align. | Scoring Strategy |
|---|---|---|---|---|---|
| AP-Fréchet | Trajectories (2D) | Trajectory Matching | Independent | Arc-length Resampling | Min. Fréchet Distance |
| Soft-DTW | Trajectories (2D) | Trajectory Matching | Independent | Soft-DTW | Min. Alignment Cost |
| AP-RMS | Trajectories (2D) | Trajectory Matching | Independent | Arc-length Resampling | Min. Prefix RMS Error |
| FIDeL [11] | Patch Features (RGB) | Optimal Transport | Implicit (Patches) | Min. Cost over $T$ | OT Cost (+ VLM Filter) |
| logpZO [10] | Observations | Density Estimation | Implicit (Network) | ODE Forward Pass | Latent Space $\ell_2$-norm |
| lopO [10] | Observations | Density Estimation | Implicit (Network) | ODE Integration | Exact Log-likelihood |
| AE-Recon [41] | Observations | Reconstruction | Implicit (Network) | Frame-wise (None) | Reconstruction Error |
| **TPC (Ours)** | Trajectories (2D) | Trajectory Matching | **Explicit (Pearson)** | Fixed Resampling | Proj. Dist. + Corr. Penalty |
| **GnnT (Ours)** | **Spatio-Temporal Graphs** | **Future Prediction** | **Explicit (GATv2)** | **Sliding Window (Attn)** | **Prediction Error** |

robustness to both relational or identity errors (e.g., wrong object or placement) and dynamic deviations (e.g., delayed or missed interactions).

### C.2 Baselines

We compare our methods against classical and learning-based failure detection methods, which caracteristics are presented in table 8.

#### C.2.1 Arc-Length Prefix RMS (AP-RMS)

The AP-RMS baseline detects kinematic deviations by evaluating the cumulative spatial error of a trajectory's prefix relative to a set of nominal expert trajectories. To ensure the comparison is invariant to execution speed and sampling rates, all trajectories are spatially aligned via arc-length parameterization.

**1. Arc-Length Resampling:** Let a raw trajectory consisting of $N$ points be defined as $P = \{p_1, \dots, p_N\}$ with $p_i \in \mathbb{R}^2$. The cumulative arc-length $s_i$ at step $i$ is computed as:

$$s_i = \sum_{j=2}^{i} \|p_j - p_{j-1}\|_2, \quad \text{with } s_1 = 0 \tag{11}$$

The sequence is parameterized by normalized arc-length $\bar{s}_i = s_i / s_N \in [0, 1]$. We construct a continuous trajectory function $f(\bar{s})$ via linear interpolation. The trajectory is then uniformly resampled into a fixed length $T$, producing the spatially aligned trajectory $\tilde{P} = \{\tilde{p}_1, \dots, \tilde{p}_T\}$, where $\tilde{p}_\tau = f\left(\frac{\tau-1}{T-1}\right)$. This procedure is applied identically to the set of expert trajectories to form the aligned expert set $\tilde{\mathcal{E}} = \{\tilde{V}^{(1)}, \dots, \tilde{V}^{(K)}\}$.

**2. Prefix RMS Distance:** To evaluate the deviation of the execution up to a given progression step $t \in [1, T]$, we extract the trajectory prefix $\tilde{P}_{1:t} = \{\tilde{p}_1, \dots, \tilde{p}_t\}$. The Root Mean Square (RMS) distance between the observed prefix and the corresponding prefix of an expert $\tilde{V}^{(k)}_{1:t}$ is defined as:

$$d_{\text{RMS}}\big(\tilde{P}_{1:t}, \tilde{V}^{(k)}_{1:t}\big) = \sqrt{\frac{1}{t} \sum_{\tau=1}^{t} \big\|\tilde{p}_\tau - \tilde{v}^{(k)}_\tau\big\|_2^2} \tag{12}$$

**3. Final failure Score:** At each progression step $t$, the spatial failure is quantified as the distance to the closest expert prefix. This local deviation is given by:

$$d^*(t) = \min_{\tilde{V} \in \tilde{\mathcal{E}}} d_{\text{RMS}}\big(\tilde{P}_{1:t}, \tilde{V}_{1:t}\big) \tag{13}$$

The overall failure score for the complete trajectory is the temporal average of these minimal prefix deviations, capturing the continuous divergence from the nominal behavior manifold:

$$S_{\text{total}} = \frac{1}{T} \sum_{t=1}^{T} d^*(t) \tag{14}$$

### C.2.2 Arc-Length Prefix Fréchet (AP-Fréchet) Formulation

The AP-Fréchet baseline detects structural and geometric anomalies by evaluating the shape discrepancy of trajectory prefixes relative to nominal expert demonstrations. Similar to the AP-RMS method, this approach enforces temporal invariance by spatially aligning trajectories via arc-length parameterization.

**1. Arc-Length Resampling:** We apply the identical arc-length normalization procedure defined for AP-RMS. Let the uniformly resampled observed trajectory of length $T$ be $\tilde{P} = \{\tilde{p}_1, \dots, \tilde{p}_T\}$, and the corresponding aligned expert set be $\tilde{\mathcal{E}} = \{\tilde{V}^{(1)}, \dots, \tilde{V}^{(K)}\}$.

**2. Discrete Fréchet Distance:** To evaluate the geometric divergence up to a progression step $t \in [1, T]$, we extract the trajectory prefixes $\tilde{P}_{1:t}$ and $\tilde{V}^{(k)}_{1:t}$. The Discrete Fréchet distance considers all possible ordered couplings between the points of the two polygonal curves. Let an order-preserving coupling sequence $L$ be a sequence of index pairs $(i, j)$ such that $(i_1, j_1) = (1, 1)$, $(i_{|L|}, j_{|L|}) = (t, t)$, and successive pairs transition by $(+1, 0)$, $(0, +1)$, or $(+1, +1)$.

The Discrete Fréchet distance is defined as the minimum over all valid coupling sequences of the maximum Euclidean distance between coupled points:

$$d_{\mathrm{F}}\big(\tilde{P}_{1:t}, \tilde{V}^{(k)}_{1:t}\big) = \min_{L} \max_{(i,j)\in L} \big\|\tilde{p}_i - \tilde{v}^{(k)}_j\big\|_2 \tag{15}$$

In practice, this distance is computed efficiently utilizing dynamic programming.

**3. Prefix failure and Normalization:** At each progression step $t$, the geometric failure is evaluated against the closest expert prefix:

$$d^*(t) = \min_{\tilde{V}\in\tilde{\mathcal{E}}} d_{\mathrm{F}}\big(\tilde{P}_{1:t}, \tilde{V}_{1:t}\big) \tag{16}$$

To account for the natural geometric variance of the specific object's class, $d^*(t)$ is standardized. Let $\mu$ and $\sigma$ be the empirical mean and standard deviation of the pairwise Fréchet distances evaluated among the full-length resampled experts in $\tilde{\mathcal{E}}$. The normalized failure score at step $t$ is formulated as:

$$\hat{d}(t) = \frac{d^*(t) - \mu}{\sigma + \epsilon} \tag{17}$$

where $\epsilon$ is a small constant ensuring numerical stability. The final per-step scores form a trajectory of geometric divergence tracking the severity of the failure over the execution sequence.

### C.2.3 Soft Dynamic Time Warping (Soft-DTW) Formulation

The Soft-DTW baseline detects anomalies by assessing the global spatio-temporal dissimilarity between a full observed trajectory and a set of nominal expert trajectories. By utilizing dynamic time warping, this approach is highly resilient to temporal scaling and non-linear shifts in execution speed.

**1. Trajectory Preprocessing:** Let the observed trajectory be $P = \{p_1, \dots, p_N\}$ and the set of expert trajectories be $\mathcal{E} = \{V^{(1)}, \dots, V^{(K)}\}$. To ensure computational tractability and enable batched parallel execution on GPU hardware, all trajectories are linearly downsampled to a maximum fixed length $T_{max}$ (if their original length exceeds it) and zero-padded.

**2. Soft-DTW Distance:** For a given test trajectory $P$ and an expert trajectory $V$ (both of length $T \le T_{max}$), we define the pairwise Euclidean distance matrix $\Delta \in \mathbb{R}^{T\times T}$, where $\Delta_{i,j} = \|p_i - v_j\|_2^2$. Let $\mathcal{A}(T, T)$ represent the set of all valid binary alignment matrices (paths) between two sequences of length $T$. The standard DTW distance minimizes the inner product $\langle A, \Delta\rangle$ over all $A \in \mathcal{A}$.

Soft-DTW introduces a smoothing parameter $\gamma > 0$ and replaces the non-differentiable $\min$ operator with a soft-minimum, defined for a set of values $\{x_i\}$ as:

$$\min{}^{\gamma}(x_1, \ldots, x_n) = -\gamma \log \sum_{i=1}^{n} e^{-x_i/\gamma} \tag{18}$$

The Soft-DTW distance between $P$ and $V$ aggregates the costs of all possible alignment paths:

$$d_{\text{sDTW}}(P, V) = \min{}^{\gamma}_{A \in \mathcal{A}(T,T)} \langle A, \Delta \rangle \tag{19}$$

This softening ensures that the distance metric is less brittle to localized noise compared to the standard, strictly optimal DTW path.

**3. Minimum Expert Distance and Normalization:** The raw trajectory failure is quantified as the minimum Soft-DTW distance to any expert demonstration in the nominal set:

$$d^*(P) = \min_{V \in \mathcal{E}} d_{\text{sDTW}}(P, V) \tag{20}$$

To determine statistical significance, this raw distance is standard-normalized. Let $\mu$ and $\sigma$ be the empirical mean and standard deviation of all pairwise Soft-DTW distances computed exclusively within the expert set $\mathcal{E}$:

$$\hat{d}(P) = \frac{|d^*(P) - \mu|}{\sigma + \epsilon} \tag{21}$$

where $\epsilon$ is a small scalar added for numerical stability. The resulting scalar $\hat{d}(P)$ serves as the final failure score for the evaluated trajectory.

### C.3 Evaluation Metrics

We assess our framework in two stages: evaluating failure scores independently of decision thresholds, and evaluating the operational end-to-end binary failure detection. Unlike some prior works that evaluate failure detection solely at the episode level, our predictions are evaluated at a granular step level, where labels $y \in \{0, 1\}$ correspond to nominal behavior and task failure, respectively. Let $TP$, $TN$, $FP$, and $FN$ denote the number of true positives, true negatives, false positives, and false negatives.

**Raw Failure Scoring.** To evaluate the intrinsic discriminative power of the continuous anomaly scores $S_t$, we report the Area Under the Precision-Recall Curve (AUPR) and the maximum Matthews Correlation Coefficient (MCC). We explicitly favor AUPR because it is highly sensitive to false positives, making it a strictly more informative metric for highly imbalanced datasets where anomalies are rare compared to nominal executions. Furthermore, to capture the optimal threshold-independent balance between precision and recall, we report the maximum MCC achieved across all possible thresholds. MCC evaluates the quality of binary classifications on skewed distributions, yielding a value between $-1$ and $1$ (with $0$ indicating random correlation and $1$ indicating perfect correlation):

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{22}$$

**End-to-End Detection Pipeline.** To evaluate the practical deployment readiness of the complete pipeline, we assess the binary decisions produced by applying the Conformal Prediction (CP) thresholds. This measures the entire system's ability to safely flag critical failures while remaining robust to benign variations. We report the True Positive Rate (TPR), True Negative Rate (TNR), and the threshold-calibrated MCC:

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{TNR} = \frac{TN}{TN + FP} \tag{23}$$

These metrics isolate distinct operational requirements: TPR (Recall) evaluates system safety by measuring the proportion of failures successfully intercepted, while TNR (Specificity) measures operational efficiency by quantifying the ability to ignore benign noise without triggering false alarms.

# D Additional results

## D.1 Impact of the number of demonstration on the results

TPC vs GnnT methods MCC score when gradually increasing the number of demonstrations.

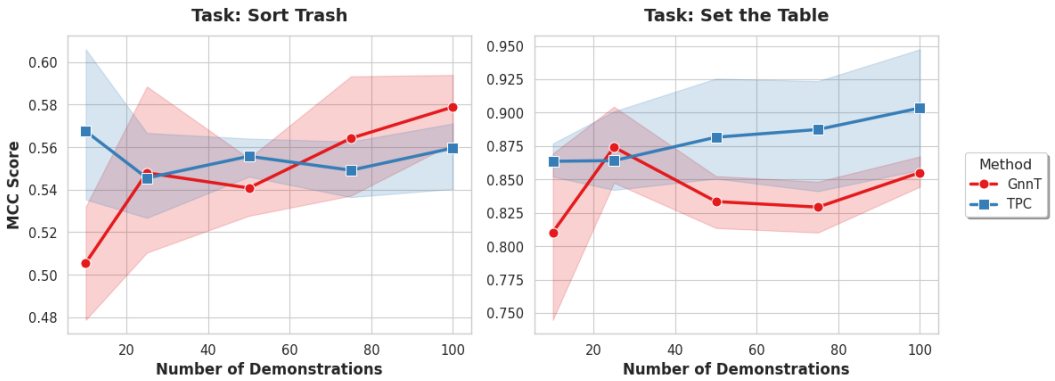


Figure 8: Evolution of the MCC score relative to the number of training demonstrations for TPC and GnnT on the Watchdog dataset. The lines and shaded regions represent the mean and standard deviation across 10 random demonstration samples.

The analysis of the MCC scores as a function of the number of demonstrations (see Figure 8) highlights several key dynamics:

First, the non-parametric TPC method demonstrates remarkable resilience in extreme few-shot regimes (fewer than 25 demonstrations), effectively avoiding the overfitting that degrades the predictive GnnT model when data is scarce. However, the scaling behavior reveals a strong dependence on the task's dynamic complexity. In the continuously evolving Sort Trash environment, GnnT benefits monotonically from increased data, ultimately surpassing TPC beyond 75 demonstrations as its sequence modeling successfully internalizes complex temporal dynamics. Conversely, in the structurally constrained Set the Table task, TPC maintains a consistent advantage across all data regimes, suggesting that direct geometric and relational alignment yields a more robust anomaly signal than future prediction for static, topological failures. Ultimately, these diverging performance trends empirically validate the necessity of our dual-architecture framework to accommodate varying levels of data availability and environmental complexity.

## D.2 Results details per task

Figure 9, 10 and 11 summarizes the threshold-independent discriminative performance of all evaluated methods. Across all tasks, both of our methods consistently outperform prior work. In particular, **TPC** achieves the best overall performance, with average AUPR reaching 0.690 (BotFails) and 0.784 (Watchdog), and MCC up to 0.666 and 0.703, respectively. **GnnT** follows closely, outperforming all learning-based baselines with AUPR of 0.612 / 0.740 and MCC of 0.529 / 0.719.

**Comparison to trajectory matching baselines.** Classical trajectory metrics (AP-RMS, Fréchet, Soft-DTW) show highly inconsistent behavior. While AP-RMS performs reasonably on structured sorting tasks (e.g., AUPR 0.70 on Sort 1), it collapses on more complex or cluttered scenarios involving object identity, spatial arrangement, or task progression (e.g., 0.15 on Waste-sorting). This highlights a key limitation: these methods ignore inter-object relations and cannot capture errors beyond geometric deviations.

**Comparison to learned representations.** Density-based (logpZO, lopO) and reconstruction-based (AE-Recon) methods provide moderate performance but remain unstable across tasks. For

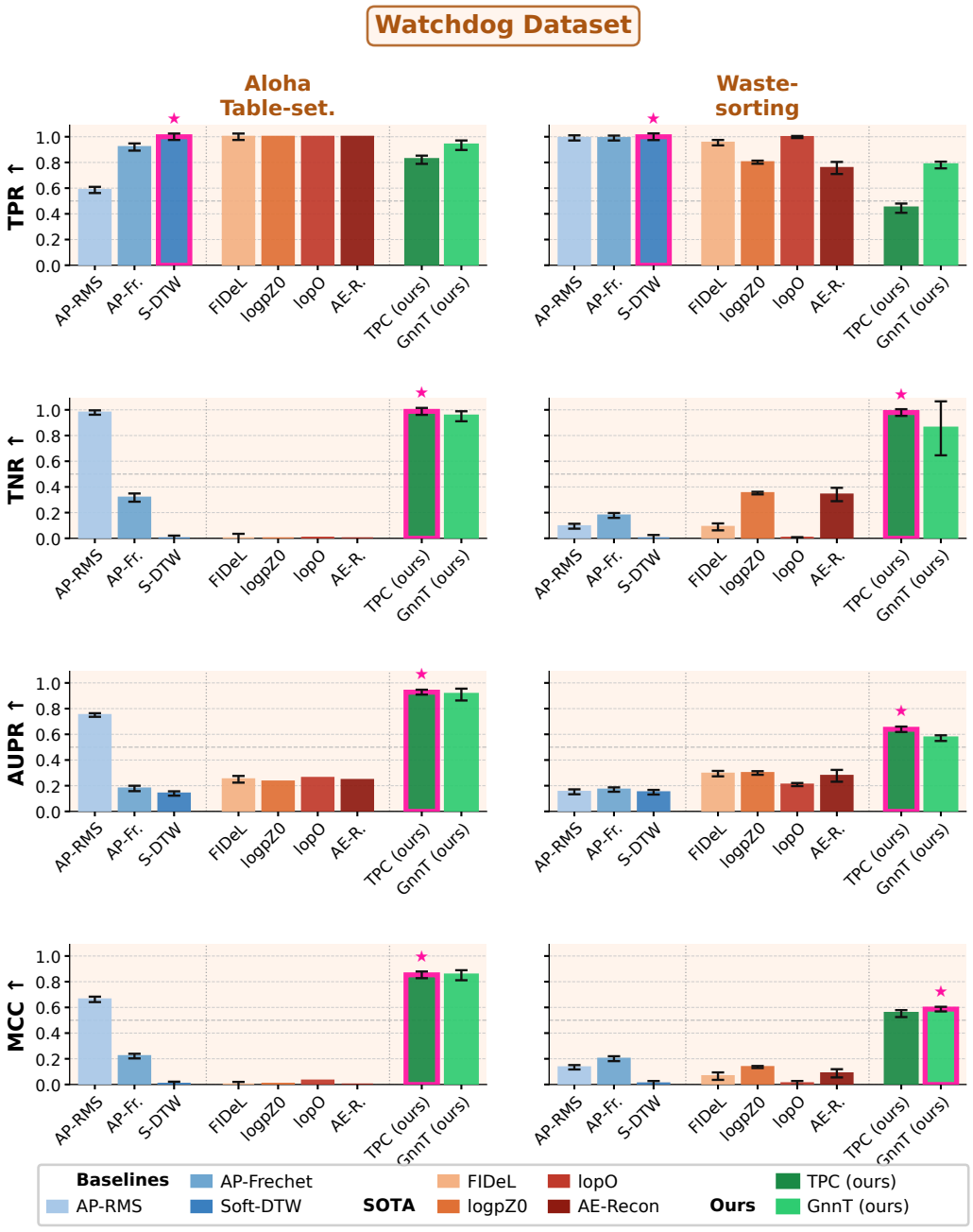


Figure 9: **threshold-independent discriminative performance - Watchdog dataset tasks.** True Positive Rate (TPR), True Negative Rate (TNR), Average Area Under the Precision-Recall Curve (AUPR) and maximum Matthews Correlation Coefficient (MCC) across Watchdog dataset tasks. Error bars denote the standard deviation across task subsets. Best results framed in purple with a star.

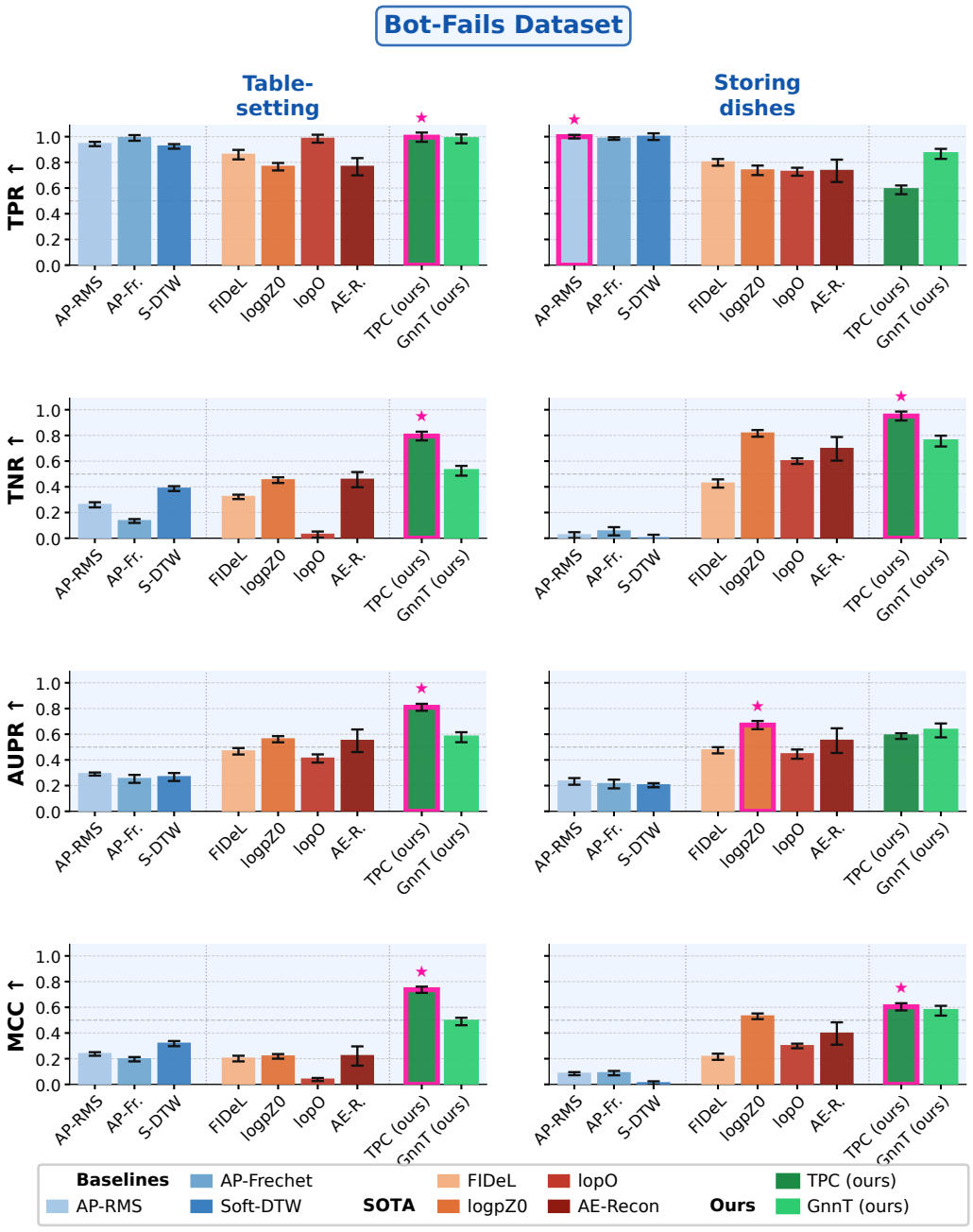


Figure 10: **threshold-independent discriminative performance - Bot-Fails dataset tasks (Table-setting and Storing dishes).** True Positive Rate (TPR), True Negative Rate (TNR), Average Area Under the Precision-Recall Curve (AUPR) and maximum Matthews Correlation Coefficient (MCC). Error bars denote the standard deviation across task subsets. Best results framed in purple with a star.

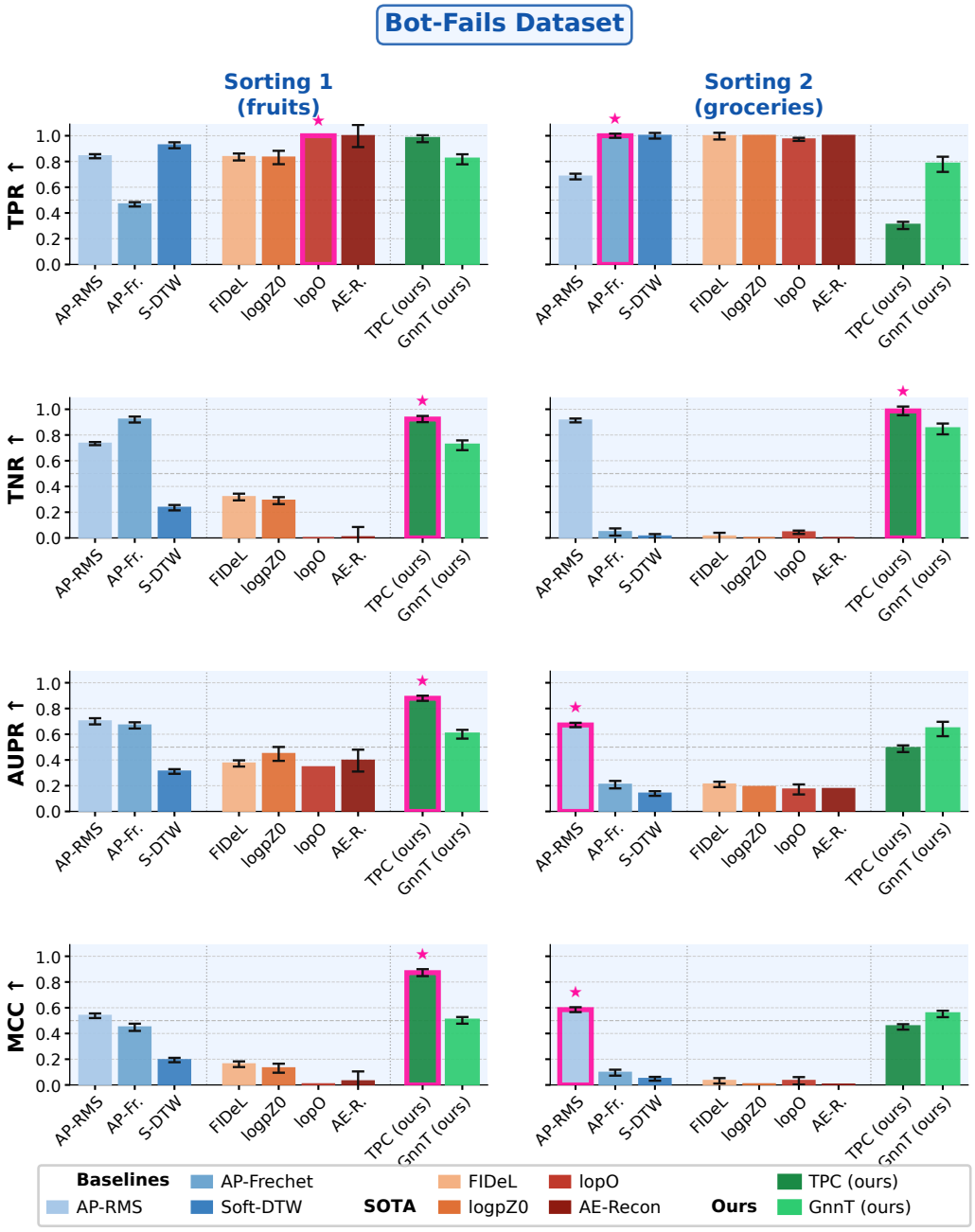


Figure 11: **threshold-independent discriminative performance - Bot-Fails dataset tasks (Sorting tasks).** True Positive Rate (TPR), True Negative Rate (TNR), Average Area Under the Precision-Recall Curve (AUPR) and maximum Matthews Correlation Coefficient (MCC). Error bars denote the standard deviation across task subsets. Best results framed in purple with a star.

instance, logpZO reaches AUPR $0.67$ on *Dish Tidy Up* but drops below $0.20$ on *Sort 2*. Similarly, FIDeL improves robustness to visual noise but struggles with relational failures (AUPR $0.21$ on Sort 2). These results confirm that implicit representations fail to capture structured interactions and task semantics.

**Effect of relational modeling.** Both TPC and GnnT explicitly encode object interactions, leading to substantial gains in MCC (up to $+0.4$ over baselines). TPC is particularly effective in low-data regimes, where its non-parametric formulation avoids overfitting and yields strong performance across all tasks. In contrast, GnnT benefits from larger datasets and excels in dynamic scenarios (e.g., Sort tasks), where temporal prediction captures delayed effects and interaction patterns that static metrics miss.

**Failure modes and insights.** We observe that GnnT underperforms TPC on simpler or low-variability tasks (e.g., Table-setting), suggesting that predictive models require sufficient data to fully exploit temporal structure. Conversely, TPC degrades in highly dynamic settings (e.g., moving conveyor), where alignment-based metrics struggle with temporal variability.