GATE-AD: Graph Attention Network Encoding for Few-Shot Industrial Visual Anomaly Detection

Aggelos Psiris
Harokopio University of Athens
Athens, Greece
aggelospsiris@hua.gr

Yannis Panagakis
National and Kapodistrian University of Athens
Archimedes, Athena Research Center
Athens, Greece
yannisp@di.uoa.gr

Maria Vakalopoulou
CentraleSupelec, University Paris Saclay
Archimedes, Athena Research Center
Paris, France / Athens, Greece
maria.vakalopoulou@centralesupelec.fr

Georgios Th. Papadopoulos
Harokopio University of Athens
Archimedes, Athena Research Center
Athens, Greece
g.th.papadopoulos@hua.gr

Abstract

Few-Shot Industrial Visual Anomaly Detection (FS-IVAD) is a critical task in modern manufacturing, where automated product inspection systems must identify rare defects using only a handful of normal, defect-free training samples. Existing approaches typically model normality by assessing each patch largely through its individual appearance, e.g., through similarity to stored normal features, reconstruction from support samples, or distance from a learned normal representation. This paper introduces GATE-AD, whose key contribution is to instead model few-shot normality through local contextual relations: a patch is considered normal when it can be consistently recovered from its learned dependencies with its spatial neighbors. To realize this formulation, frozen self-supervised ViT patch tokens are organized on a spatial neighbor graph and processed by a Graph Attention Network (GAT), while representation alignment and a Scaled Cosine Error (SCE) objective enable reconstruction-based anomaly scoring. Node masking and attention dropout further regularize learning in the low-shot regime. On MVTec AD, VisA, and MPDD, GATE-AD achieves strong detection and localization performance across 1- to 8-shot settings, attaining the highest image-level AUROC in 10 of 12 settings and the highest image-level AP in 8 of 12 settings under a unified evaluation protocol, while also achieving the lowest inference latency among the compared methods. In order to facilitate reproducibility and further research, the source code of GATE-AD is available at https://github.com/gthpapadopoulos/GATE-AD.

1 Introduction

Industrial Visual Anomaly Detection (IVAD) comprises a cornerstone of modern quality assurance pipelines, enabling the automated identification of defects (such as scratches, dents, and structural failures) in manufactured products [22, 23]. In practice, production lines change frequently and defective samples are rare by nature, making the collection of large labeled datasets prohibitively expensive. This motivates the **Few-Shot Industrial Visual Anomaly Detection (FS-IVAD)** setting, where a model must detect and localize anomalies using only a small number, typically 1 to 8, of defect-free reference images per product category, with no abnormal samples available during training [6, 14].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

The core challenge in FS-IVAD is learning a compact, robust representation of *normality* from very limited data, such that deviations from it can be reliably identified at inference. This is challenging due to: a) **Data scarcity**, where normality must be captured from as few as a single image, b) **Subtle variance**, where genuine structural failures must be distinguished from natural appearance variation, and c) **Domain shift**, where the learned appearance patterns of normality must generalize across unseen product defects [14, 7, 8].

Existing approaches fall broadly into three main categories: a) *Memory bank methods* [5, 27, 7, 35] retrieve anomaly scores by comparing test patches against a stored bank of normal features, but their memory requirements and nearest-neighbor search cost generally increase with the size of the stored support representation; b) *Vision-language methods* [14, 19, 24, 4] leverage vision-text representations (e.g., CLIP-based) to align visual patches with textual descriptions of normality, but suffer from a granularity gap, since Vision-Language Models (VLMs) are trained on image-level supervision while industrial defects are fine-grained and local; c) *Reconstruction-based methods* [8, 20, 30] train a model to reconstruct normal patterns and score anomalies by residual error, but risk the identity-mapping problem and tend to overfit to the limited normal samples available. The current work argues that the key limitation shared across all FS-IVAD categories is the failure to explicitly model **local relational structure** among image patches, i.e., the contextual consistency between a patch and its spatial neighbors: in defect-free images, each patch is statistically predictable from its immediate surroundings, while industrial defects (e.g., scratches, cracks, contamination, missing parts, etc.) disrupt precisely this neighbor-to-neighbor consistency, often without modifying the global image statistics.

To address the key challenge mentioned above, a novel approach to reconstruction-based FS-IVAD is proposed, termed **GATE-AD**. The framework casts few-shot normality modeling as a masked, representation-aligned relational reconstruction problem over frozen self-supervised ViT patch tokens, extracted using DINOv2 [26]/ DINOv3 [28]. A fundamental consideration behind the proposed design is that learning a relational normality prior requires each patch representation to be inferred from its local spatial context, while preserving the rich semantic representation of the frozen backbone. To this end, the patch tokens are organized on a fixed spatial neighbor graph, and the individual components of GATE-AD serve complementary roles within this formulation: a) the GAT encoder performs feature-dependent message passing over the predefined spatial neighborhood, assigning non-uniform importance to neighboring patches and thereby modeling their local dependencies [32]; b) GraphMAE-style node masking [12] prevents trivial copying of the input features and forces the GAT encoder to infer masked patch representations from their surrounding context; c) the representation-alignment head provides the shared latent space in which the graph-conditioned prediction is compared with the corresponding frozen ViT target, thereby providing the reconstruction signal used to train the GAT encoder; and d) the Scaled Cosine Error (SCE) down-weights well-reconstructed patches and places greater emphasis on harder reconstruction errors, while attention dropout regularizes the learned neighborhood dependencies in the low-shot regime. At inference, patches that cannot be consistently reconstructed from their spatial context produce high SCE reconstruction errors and are identified as anomalous. The key contributions of GATE-AD are:

- GATE-AD is, to the best of the authors' knowledge, the first FS-IVAD framework to **reformulate normality as a relational reconstruction problem**, where a patch is considered normal if its representation can be recovered from its learned dependencies with the spatial neighborhood, rather than solely from its similarity to stored normal features.
- To realize this relational formulation, GATE-AD introduces a **masked graph reconstruction scheme** that learns normal spatial dependencies by predicting patch representations from their local neighborhood, with representation alignment providing a shared latent space between the graph-conditioned prediction and the corresponding frozen DINO target.
- Extensive evaluation on MVTec AD, VisA, and MPDD demonstrates **strong and consistent performance across** 1 **to** 8 **shot settings**, with GATE-AD achieving the highest image-level AUROC in $10/12$ settings and the highest image-level AP in $8/12$ settings under a unified evaluation protocol, while also attaining the lowest inference latency, 22.5% lower than the next fastest compared method.

The remainder of the paper is organized as follows: Section 2 outlines the relevant FS-IVAD literature; Section 3 describes the proposed GATE-AD approach; Section 4 provides experimental evaluation results and comparisons with literature methods; Section 5 concludes the paper.

2 Related Work

Visual feature embeddings and memory banks. Feature embedding-based methods rely on mapping an input image to a high-dimensional feature space, where anomalous samples correspond to low-density areas or lie far from normal ones. PatchCore [27] establishes patch-level memory-bank matching, AnomalyDINO [7] strengthens it with DINOv2 features and foreground masking, and VisionAD [33] further enriches the normal memory through augmentation and multi-layer features. DFM [34] jointly optimizes the feature extractor and matching head, while SubspaceAD [18] models normal variation through a PCA subspace. Other approaches use graph representations to enhance feature modeling or matching [35, 10, 29]. Common challenges include modeling complex intra-class variation from very few normal samples, memory-dependent inference costs for retrieval-based methods, and sensitivity to spatial misalignment.

Vision-language model reasoning and adaptation. Vision-Language Models (VLMs) support linguistic descriptions that often encapsulate high-level priors for modeling abnormalities. WinCLIP+ [14] combines state words and prompt templates with multi-scale features. FADE [21] adapts CLIP to extract multi-scale, image-patch embeddings better aligned with language. PromptAD [19] employs a dual-branch vision-language decoder for both normality and abnormality modeling. Lv et al. [24] learn a class-shared prompt generator. InCTRL [37] learns an in-context residual learning model, while AnomalyR1 [3] autonomously processes image and domain-knowledge inputs. Despite "open-vocabulary" capabilities, VLM-based approaches can suffer from the vision-language granularity gap, limited domain specificity, and sensitivity to prompt design.

Reconstruction-based frameworks and prototype refinement. Reconstruction-based approaches detect anomalies through discrepancies between query representations and their estimated normal counterparts. FastRecon [8] employs a regression scheme with distribution regularization. FastRef [30] relies on characteristic transfer from query features to prototypes and anomaly suppression through prototype alignment. One-to-Normal [20] performs a personalized one-to-normal transformation of query images. FoundAD [36] learns a non-linear projection operator on the natural image manifold. Key limitations include the risk of reconstructing both normal and abnormal samples equally well, loss of reconstruction detail (often) larger than the defects themselves, tendency to overfit to normal samples, and reduced ability to handle high intra-class variance.

3 Proposed approach

3.1 Overview

The proposed GATE-AD framework models few-shot normality through masked, representation-aligned graph reconstruction. Normal patches are expected to be predictable from their local spatial context, while anomalies violate these dependencies. A masked GAT encoder models such relations, and representation alignment provides the latent space where graph-conditioned predictions are compared with the corresponding frozen ViT features. The general architecture of GATE-AD is illustrated in Fig. 1, while its individual components are detailed in the followings.

3.2 Image patchification and feature extraction

Given an input image $I_{mg} \in \mathbb{R}^{H\times W\times 3}$, it is resized so that the smaller dimension equals β pixels. A dense local feature token $\mathbf{x}_i \in \mathbb{R}^D$, $i \in [1, N]$, is then extracted for each 16×16 / 14×14 pixel image patch (standard for DINOv3/DINOv2) using a frozen ViT backbone (DINOv3 [28]/DINOv2 [26]). Following common practices, the $\mathbf{x}_i$ representation is obtained by averaging the patch token across the last ζ transformer layers [7, 36].

3.3 Masked graph attention network encoder

Encoder type selection. A key component of reconstruction-based FS-IVAD is the encoder used to model normal appearance patterns. In GATE-AD, a GAT encoder [32] operates on the patch-level representations $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and models their local dependencies. This choice is motivated by the observation that anomalous patches may violate the consistency between local features

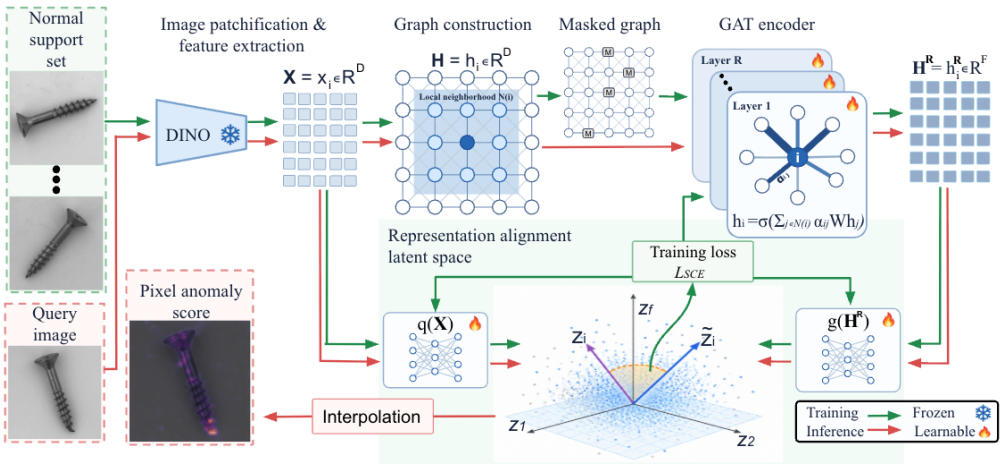

Figure 1: General architecture of the proposed masked, representation-aligned GAT encoding scheme for FS-IVAD.

and their surrounding context [1]. In contrast to standard graph convolution [17], GAT assigns feature-dependent weights to neighboring nodes and therefore performs selective local aggregation. Transformer attention [31] provides a similar adaptive mechanism over global pairwise interactions, with quadratic complexity in the number of patches, whereas GAT restricts message passing to the fixed sparse local graph, yielding linear complexity for a fixed neighborhood size and a favorable accuracy-efficiency trade-off for FS-IVAD.

Encoder network architecture. The GAT encoder takes as input the patch-level representation $\mathbf{X}$, with each token $\mathbf{x}_i \in \mathbb{R}^D$ defining a node at the corresponding location of the ViT patch grid. Graph edges are determined solely by spatial proximity, defining a local neighborhood $\mathcal{N}(i)$ of size K for each node, whose effect is examined in Appendix A.3; no feature-space neighbor search is performed. Setting $\mathbf{h}_i^0 = \mathbf{x}_i$, each GAT layer aggregates information over $\mathcal{N}(i)$ and updates the node representation. Consequently, after r layers, $\mathbf{h}_i^r$ depends on the information contained in the r-hop spatial neighborhood of node i [32, 25].

Graph attentional layer. The input to a single attention layer is the set of node features $\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^N$, $\mathbf{h}_i \in \mathbb{R}^F$ ($\mathbf{h}_i \equiv \mathbf{x}_i$ at the first layer), and the output is a new set $\mathbf{H}' = \{\mathbf{h}'_i\}_{i=1}^N$, $\mathbf{h}'_i \in \mathbb{R}^{F'}$. A shared learnable linear transformation $\mathbf{W} \in \mathbb{R}^{F' \times F}$ is applied to every node, while a shared self-attention mechanism $\alpha : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \to \mathbb{R}$ is applied to neighboring nodes:

$$e_{ij} = \alpha(\mathbf{W}\mathbf{h}_i, \mathbf{W}\mathbf{h}_j), \; j \in \mathcal{N}(i), \tag{1}$$

which indicates the importance of node j's features to node i. Following common practice, α is implemented as a single-layer feedforward network parametrized by $\mathbf{a} \in \mathbb{R}^{2F'}$ with a LeakyReLU non-linearity:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i \| \, \mathbf{W}\mathbf{h}_j]\right)\right)}{\sum_{m \in \mathcal{N}(i)} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i \| \, \mathbf{W}\mathbf{h}_m]\right)\right)}, \tag{2}$$

where $\|$ denotes vector concatenation. The normalized coefficients α_{ij} are then used to compute a non-linear combination of the neighborhood features that yields the output $\mathbf{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W}\mathbf{h}_j\right)$, where $\sigma(\cdot)$ is the ELU activation.

Representation alignment. Repeated message passing in GNNs may lead to over-smoothing, whereby node representations become increasingly similar [16]. To limit this effect and retain consistency with the original visual representation, a Representation Alignment (RA) module is introduced. In particular, the output of the last GAT layer $\mathbf{H}^R$ is aligned with the corresponding frozen ViT features $\mathbf{X}$ in a common latent space $\mathbf{Z} \in \mathbb{R}^f$ ($f \leq F$). The two representations are projected through lightweight learnable transformations:

$$\mathbf{z}_i = q(\mathbf{x}_i), \; \tilde{\mathbf{z}}_i = g(\mathbf{h}_i^R), \;\; \mathbf{z}_i, \tilde{\mathbf{z}}_i \in \mathbb{R}^f, \tag{3}$$

where $g(\cdot)$ denotes a Linear→ReLU→Linear MLP and $q(\cdot)$ a single linear layer. During training, a subset of nodes $\mathcal{M} \subseteq \{1, \ldots, N\}$ is selected for feature masking, as detailed below. The Scaled Cosine Error (SCE) [12] is then used to measure the discrepancy between the graph-conditioned representation $\tilde{\mathbf{z}}_i$ and its corresponding ViT target $\mathbf{z}_i$, supervising the reconstruction of masked nodes:

$$\mathcal{L}_{SCE} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \left(1 - \frac{\mathbf{z}_i^T \tilde{\mathbf{z}}_i}{\|\mathbf{z}_i\| \, \|\tilde{\mathbf{z}}_i\|}\right)^{\gamma}, \ \gamma \geq 1. \tag{4}$$

For $\gamma > 1$, SCE reduces the contribution of well-reconstructed nodes and emphasizes harder reconstruction cases. This is useful in the few-shot setting, where easily reconstructed normal patches may otherwise dominate the objective. The choice of SCE and γ is examined in Appendix A.8 against MSE and plain cosine error.

Feature masking. To prevent direct reproduction of the input features and enforce the use of local context, node feature masking is employed following GraphMAE [12]. Specifically, at each training iteration, a fraction μ of the graph nodes is selected to form the masked set $\mathcal{M}$, and their corresponding features are replaced by a learnable mask token. The GAT is therefore required to recover the masked representations from the information propagated through their neighboring nodes, encouraging the learning of local dependencies between patches.

Graph regularization. Since FS-IVAD operates under limited normal supervision, the learned attention coefficients may over-adapt to particular neighborhood relations. To regularize the aggregation process, attention dropout is applied to the coefficients α_{ij} during training [32]. Specifically, each coefficient is independently masked with probability δ, reducing the dependence on individual edges and encouraging information to be aggregated from multiple neighboring nodes.

3.4 Inference and anomaly scoring

At inference, a query image I_{mg} is processed by the frozen backbone, yielding the patch-level representation $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$. The corresponding nodes are arranged on the same spatial graph and passed through the trained GAT encoder, producing the final representation $\mathbf{H}^R$. At inference, the anomaly score s_i of each patch is computed using the SCE reconstruction error in Eq. (4), evaluated over all nodes. For image-level scoring, the top-k patch scores are selected and averaged, where k is determined from a fixed ratio ϵ of the total number of patches. For pixel-level scoring, the patch scores $\{s_i\}_{i=1}^N$ are arranged according to their spatial locations to form a coarse anomaly map, which is Gaussian-smoothed [7] and bilinearly interpolated to the original image resolution.

4 Experimental results

4.1 Setup

Datasets. Three widely used public benchmarks for high-resolution FS-IVAD are employed: MVTec AD [2] ($3,629$ defect-free training and $1,725$ test images, 5 texture and 10 object categories), VisA [38] ($10,821$ images: $9,621$ normal and $1,200$ anomalous, 12 object categories in 3 domains), and MPDD [15] (888 defect-free training and 458 test images, 6 object categories from real-world manufacturing).

Experimental protocol and metrics. GATE-AD is evaluated under the k-shot setting, where k normal samples are provided per class for training and evaluation is performed on the corresponding test set. Image-level performance is measured using AUROC and Average Precision (AP), while pixel-level performance is measured using AUROC and the Area Under the Per-Region Overlap curve (AUPRO). Unless otherwise stated, experiments are repeated over three fixed support-set seeds, and results are reported as mean $\pm$ standard deviation.

Implementation and evaluation configurations. GATE-AD employs an input resolution of $\beta = 512$, with patch representations obtained by averaging the last $\zeta = 8$ backbone layers. The GAT encoder consists of $R = 3$ attentional layers with hidden dimension $F = 256$, while the latent dimension is set to $f = 256$. The graph topology is fixed across images, with each patch connected

to its $K = 8$ spatially adjacent neighbors on the ViT patch lattice; no feature-space nearest-neighbor search is performed. The SCE scaling factor is $\gamma = 2$, the node masking ratio is $\mu = 0.2$, and the attention dropout rate is $\delta = 0.3$. The dataset-specific backbones and pooling rules, as well as a detailed hyper-parameter analysis, are provided in Appendix A.

Two experimental configurations are used for different purposes. For the controlled cross-method comparison of Section 4.2, GATE-AD and the re-evaluated baselines use the same DINOv3 ViT-B/16 backbone, a common top-1% image-level aggregation rule, and identical support images for each dataset, shot, and support-set seed. For the component and architectural analyses of Section 4.3, GATE-AD instead uses the dataset-specific configurations selected through the hyper-parameter analysis of Appendix A; within each benchmark, all remaining settings are kept fixed across matched variants so that only the component or architectural choice under examination is changed.

The model is implemented in PyTorch Geometric and optimized with Adam at a learning rate of $3{\times}10^{-4}$ for a maximum of $2{,}000$ epochs with early stopping. A training-budget analysis is provided in Appendix A.12. Rotation-based augmentation is applied only to categories that exhibit pose variation, following AnomalyDINO [7]. All experiments were conducted on an NVIDIA RTX 6000 Ada Generation GPU.

4.2 Comparative evaluation

Baselines. GATE-AD is compared with representative FS-IVAD methods spanning three main families: a) feature-based methods, including SPADE [5], PatchCore [27], AnomalyDINO* [7], VisionAD* [33], and SubspaceAD* [18]; b) vision-language methods, including WinCLIP+ [14], PromptAD [19], IIPAD [24], and APRIL-GAN [4]; and c) reconstruction- and projection-based methods, including FastRecon [8], FoundAD* [36], and RegAD [13]. Results without * are taken from the literature, while methods marked with * are additionally re-evaluated under the unified experimental protocol defined in Section 4.1, following the few-shot sampling procedure of AnomalyDINO [7]. These methods are selected because public implementations are available and permit evaluation under the common protocol without altering their core anomaly-detection mechanism. Table 1 reports the 1/2/4/8-shot results on MVTec AD, VisA, and MPDD. Indicative 1-shot detection results are shown in Fig. 2.

Discussion. From Table 1, the following observations can be made: a) GATE-AD **matches or exceeds the state of the art in most image-level settings**, achieving the highest image AUROC in $10/12$ and AP in $8/12$ cases under the unified protocol, while remaining competitive at the pixel level. b) It shows a clear **advantage in the very low-shot regime** ($1/2$-shots), where normal evidence is most limited. c) GATE-AD maintains **stable performance across shot counts and support selections**, with the performance gap decreasing as additional normal references become available. d) The gains are particularly evident on **MPDD**, the most challenging manufacturing benchmark considered. The above observations are consistent with the proposed normality-learning scheme: masking enforces neighborhood-based reconstruction, attention dropout regularizes local dependencies, and representation alignment supervises recovery in a compact latent space rather than directly in the original DINO representation. Together, these mechanisms enable relational normality to be learned effectively from very few normal samples. Detailed object-level results, product-agnostic defect-type analysis, qualitative examples, and failure cases are provided in Appendix B, with the defect-type analysis showing stronger localization for local surface defects than for broader structural defects.

4.3 Ablation study

Component analysis. In order to isolate the contribution of the main components of GATE-AD, a 'DINO 1-NN' baseline is first defined, following the patch-level nearest-neighbor formulation of AnomalyDINO [7], where each query patch is scored according to its nearest normal DINO feature, without graph construction or representation alignment. Starting from this baseline, the following variants are considered: a) 'DINO + RA', where the same DINO features are projected to the representation-alignment latent space and scored using the SCE objective, without relational modeling; b) 'GATE-AD w/o RA', where the graph encoder is retained but RA is removed and the SCE objective is directly estimated between the final GAT representation and the input feature tokens; c) 'GATE-AD w/o masking & dropout', where both low-shot regularization mechanisms are

Table 1: FS-IVAD detection results on the MVTec AD, VisA, and MPDD benchmarks. Methods marked with * are re-evaluated under the unified experimental protocol.

Method	MVTec AD				VisA				MPDD			
	Image		Pixel		Image		Pixel		Image		Pixel	
	AUROC	AP	AUROC	AUPRO	AUROC	AP	AUROC	AUPRO	AUROC	AP	AUROC	AUPRO
1-shot												
SPADE [5] (arXiv'20)	82.9 ±2.6	91.7 ±1.2	92.0 ±0.3	85.7 ±0.7	80.7 ±5.0	82.3 ±4.3	96.2 ±0.4	85.7 ±1.1	-	-	-	-
PatchCore [27] (CVPR'22)	83.4 ±3.0	92.2 ±1.5	92.0 ±1.0	79.7 ±2.0	79.9 ±2.9	82.8 ±2.3	95.4 ±0.6	80.5 ±2.5	-	-	-	-
AnomalyDINO* [7] (WACV'25)	96.2 ±0.8	98.0 ±0.4	**97.2 ±0.0**	**94.6 ±0.1**	90.3 ±0.5	91.3 ±0.5	97.5 ±0.1	91.1 ±0.5	69.2 ±1.5	**72.5 ±1.3**	**97.3 ±0.1**	92.0 ±0.1
VisionAD* [33] (arXiv'25)	96.1 ±0.3	98.3 ±0.1	97.0 ±0.1	93.9 ±0.1	92.9 ±0.2	94.0 ±0.2	97.7 ±0.1	93.0 ±0.5	69.0 ±0.8	72.0 ±0.4	97.0 ±0.2	91.6 ±0.3
SubspaceAD* [18] (CVPR'26)	95.8 ±0.8	98.0 ±0.3	97.0 ±0.1	93.6 ±0.1	89.3 ±1.2	90.7 ±0.8	97.5 ±0.1	92.5 ±0.5	69.2 ±0.9	72.1 ±1.3	97.2 ±0.1	91.9 ±0.4
WinCLIP+ [14] (CVPR'23)	93.1 ±2.0	96.5 ±0.9	95.2 ±0.5	87.1 ±1.2	83.8 ±4.0	85.1 ±4.0	96.4 ±0.4	85.1 ±2.1	-	-	-	-
APRIL-GAN [4] (CVPRW'23)	92.0 ±0.3	95.8 ±0.2	95.1 ±0.1	90.6 ±0.2	91.2 ±0.8	93.3 ±0.8	96.0 ±0.0	90.0 ±0.1	-	-	-	-
PromptAD [19] (WACV'24)	94.6 ±1.7	97.1 ±1.0	95.9 ±0.5	87.9 ±1.0	86.9 ±2.3	88.4 ±2.6	96.7 ±0.4	85.1 ±2.5	-	-	-	-
IIPAD [24] (ICLR'25)	94.2	97.2	96.4	89.8	85.4	87.5	96.9	87.3	-	-	-	-
FoundAD* [36] (ICLR'26)	94.6 ±0.8	97.1 ±0.4	96.7 ±0.1	93.1 ±0.1	89.2 ±1.3	90.6 ±1.0	97.0 ±0.1	91.0 ±0.4	63.1 ±0.6	68.1 ±1.1	96.9 ±0.2	91.2 ±0.4
GATE-AD*	**97.1 ±0.2**	**98.3 ±0.1**	96.9 ±0.0	93.7 ±0.1	**93.5 ±0.1**	**94.3 ±0.2**	**97.7 ±0.0**	**93.2 ±0.2**	**70.7 ±1.5**	72.3 ±1.0	97.1 ±0.1	**92.0 ±0.4**
2-shot												
SPADE [5] (arXiv'20)	81.0 ±2.0	90.6 ±0.8	91.2 ±0.4	83.9 ±0.7	79.5 ±4.0	82.0 ±3.3	95.6 ±0.4	84.1 ±1.6	-	-	-	-
PatchCore [27] (CVPR'22)	86.3 ±3.3	93.8 ±1.7	93.3 ±0.6	82.3 ±1.3	81.6 ±4.0	84.8 ±3.2	96.1 ±0.5	82.6 ±2.3	-	-	-	-
AnomalyDINO* [7] (WACV'25)	96.7 ±1.2	98.2 ±0.7	**97.5 ±0.1**	**95.3 ±0.2**	91.1 ±0.5	91.8 ±0.4	97.9 ±0.1	92.2 ±0.6	73.5 ±3.0	75.4 ±2.0	97.4 ±0.1	92.6 ±0.3
VisionAD* [33] (arXiv'25)	97.3 ±0.5	98.2 ±0.2	97.3 ±0.1	94.6 ±0.1	92.5 ±0.6	93.4 ±0.5	97.8 ±0.1	93.4 ±0.5	73.5 ±2.7	76.4 ±2.6	97.3 ±0.1	92.5 ±0.1
SubspaceAD* [18] (CVPR'26)	96.0 ±0.7	98.1 ±0.3	97.1 ±0.1	93.8 ±0.1	90.4 ±1.2	91.4 ±0.8	97.6 ±0.1	92.8 ±0.4	71.2 ±2.5	73.9 ±2.4	97.4 ±0.1	92.6 ±0.4
WinCLIP+ [14] (CVPR'23)	94.4 ±1.3	97.0 ±0.7	96.0 ±0.3	88.4 ±0.9	84.6 ±2.4	85.8 ±2.7	96.8 ±0.3	86.2 ±1.4	-	-	-	-
APRIL-GAN [4] (CVPRW'23)	92.4 ±0.3	96.0 ±0.2	95.5 ±0.0	91.3 ±0.1	92.2 ±0.3	94.2 ±0.3	96.2 ±0.0	90.1 ±0.1	-	-	-	-
PromptAD [19] (WACV'24)	95.7 ±1.5	97.9 ±0.7	96.2 ±0.3	88.5 ±0.8	88.3 ±2.0	96.1 ±0.6	97.1 ±0.3	85.8 ±2.1	-	-	-	-
IIPAD [24] (ICLR'25)	95.7	97.9	96.7	90.3	86.7	88.6	97.2	87.9	-	-	-	-
RegAD [13] (ECCV'22)	85.7	-	94.6	-	-	-	-	-	63.4	-	93.2	-
FastRecon [8] (ICCV'23)	91.0	-	95.9	-	-	-	-	-	73.7	-	97.0	-
FoundAD* [36] (ICLR'26)	96.2 ±0.7	98.0 ±0.4	97.4 ±0.1	94.2 ±0.1	91.4 ±1.0	91.6 ±0.6	97.9 ±0.0	93.5 ±0.3	70.8 ±5.5	73.2 ±4.6	97.5 ±0.1	93.0 ±0.4
GATE-AD*	**97.6 ±0.2**	**98.5 ±0.1**	97.3 ±0.0	94.3 ±0.1	**93.0 ±0.2**	**93.7 ±0.3**	**98.0 ±0.0**	**93.6 ±0.2**	**80.9 ±5.4**	**81.4 ±5.0**	**97.6 ±0.1**	**93.9 ±0.4**
4-shot												
SPADE [5] (arXiv'20)	84.8 ±2.5	92.5 ±1.2	92.7 ±0.3	87.0 ±0.5	81.7 ±3.4	83.4 ±2.7	96.6 ±0.3	87.3 ±0.8	-	-	-	-
PatchCore [27] (CVPR'22)	88.8 ±2.6	94.5 ±1.5	94.3 ±0.5	84.3 ±1.6	85.3 ±2.1	87.5 ±2.1	96.8 ±0.3	84.9 ±1.4	-	-	-	-
AnomalyDINO* [7] (WACV'25)	98.0 ±0.1	98.8 ±0.1	97.8 ±0.1	**95.7 ±0.2**	92.3 ±0.4	92.9 ±0.3	98.2 ±0.0	93.0 ±0.2	76.0 ±2.8	77.9 ±2.4	97.7 ±0.1	93.3 ±0.3
VisionAD* [33] (arXiv'25)	**98.0 ±0.3**	**99.2 ±0.1**	97.5 ±0.1	94.9 ±0.1	94.4 ±0.4	**95.1 ±0.4**	98.4 ±0.0	94.6 ±0.2	76.3 ±2.6	79.2 ±3.0	97.7 ±0.2	93.3 ±0.3
SubspaceAD* [18] (CVPR'26)	96.5 ±0.3	98.5 ±0.1	97.2 ±0.1	93.9 ±0.1	92.0 ±0.8	92.4 ±0.5	97.6 ±0.0	93.0 ±0.1	70.8 ±1.6	74.7 ±2.2	97.5 ±0.1	92.8 ±0.3
WinCLIP+ [14] (CVPR'23)	95.2 ±1.3	97.3 ±0.6	96.2 ±0.3	89.0 ±0.8	87.3 ±1.8	88.8 ±1.8	97.2 ±0.2	87.6 ±0.9	-	-	-	-
APRIL-GAN [4] (CVPRW'23)	92.8 ±0.2	96.3 ±0.1	95.9 ±0.0	91.8 ±0.1	92.6 ±0.4	94.5 ±0.3	96.2 ±0.0	90.2 ±0.1	-	-	-	-
PromptAD [19] (WACV'24)	96.6 ±0.9	98.5 ±0.5	96.5 ±0.2	90.5 ±0.7	89.1 ±1.7	90.8 ±1.3	97.4 ±0.3	86.2 ±1.7	-	-	-	-
IIPAD [24] (ICLR'25)	96.1	98.1	97.0	91.2	88.3	89.6	97.4	88.3	-	-	-	-
RegAD [13] (ECCV'22)	88.2	-	95.8	-	-	-	-	-	68.3	-	93.9	-
FastRecon [8] (ICCV'23)	94.2	-	97.0	-	-	-	-	-	79.9	-	97.6	-
FoundAD* [36] (ICLR'26)	97.4 ±0.5	98.8 ±0.2	**97.9 ±0.1**	94.8 ±0.2	93.2 ±0.5	92.8 ±0.4	**98.6 ±0.0**	**95.2 ±0.1**	74.3 ±2.5	77.4 ±2.0	**97.9 ±0.2**	94.0 ±0.6
GATE-AD*	97.7 ±0.1	98.6 ±0.1	97.6 ±0.0	94.5 ±0.0	**94.5 ±0.1**	94.7 ±0.3	98.3 ±0.0	94.5 ±0.1	**79.0 ±3.0**	**82.0 ±3.8**	97.7 ±0.1	**94.1 ±0.4**
8-shot												
AnomalyDINO* [7] (WACV'25)	**98.5 ±0.1**	99.3 ±0.1	98.0 ±0.0	**96.2 ±0.1**	93.0 ±0.3	93.2 ±0.3	98.4 ±0.0	93.7 ±0.2	78.2 ±2.4	79.6 ±2.6	98.0 ±0.0	93.8 ±0.2
VisionAD* [33] (arXiv'25)	98.3 ±0.2	**99.4 ±0.1**	97.7 ±0.0	95.4 ±0.1	95.1 ±0.2	95.3 ±0.2	98.6 ±0.0	95.2 ±0.2	80.1 ±1.4	82.3 ±1.8	97.9 ±0.1	93.9 ±0.1
SubspaceAD* [18] (CVPR'26)	96.8 ±0.5	98.8 ±0.2	97.2 ±0.0	94.1 ±0.2	92.6 ±0.6	93.1 ±0.7	97.6 ±0.0	93.1 ±0.2	70.8 ±2.6	74.3 ±1.5	97.6 ±0.1	92.7 ±0.5
RegAD [13] (ECCV'22)	91.2	-	96.7	-	-	-	-	-	71.9	-	95.1	-
FastRecon [8] (ICCV'23)	95.2	-	97.3	-	-	-	-	-	82.5	-	97.9	-
FoundAD* [36] (ICLR'26)	97.9 ±0.3	99.0 ±0.0	**98.1 ±0.0**	95.3 ±0.1	94.1 ±0.1	93.5 ±0.1	**98.9 ±0.0**	**96.0 ±0.1**	78.5 ±2.4	80.5 ±1.5	**98.3 ±0.1**	**94.8 ±0.1**
GATE-AD*	97.8 ±0.1	98.7 ±0.0	97.7 ±0.0	94.7 ±0.0	**95.3 ±0.1**	**95.4 ±0.3**	98.4 ±0.0	94.6 ±0.1	**80.8 ±3.0**	**82.8 ±2.1**	97.7 ±0.1	93.7 ±0.4

removed; d–e) variants where attention dropout or feature masking is independently discarded; and f) the complete 'GATE-AD' configuration. All variants follow the dataset-specific ablation protocol defined in Section 4.1, with the remaining configuration kept fixed within each benchmark. Table 2 reports I-AUROC and P-AUROC across all datasets and shot settings.

Discussion. From Table 2, the following key observations can be made: a) Frozen DINO features already provide a **strong normality representation**, as demonstrated by the competitive DINO 1-NN baseline. However, introducing RA directly on the same features does not consistently improve nearest-neighbor matching, with DINO + RA performing below DINO 1-NN in several settings. Hence, **representation alignment alone is not sufficient to explain the gains of GATE-AD**. b) The complete GATE-AD formulation clearly improves over the non-graph variants, with the largest gains observed in the 1-shot regime, where normal evidence is most limited. This supports the central premise of the proposed approach: rather than characterizing a patch only through its similarity to stored normal features, normality is learned through its **consistency with the surrounding spatial**

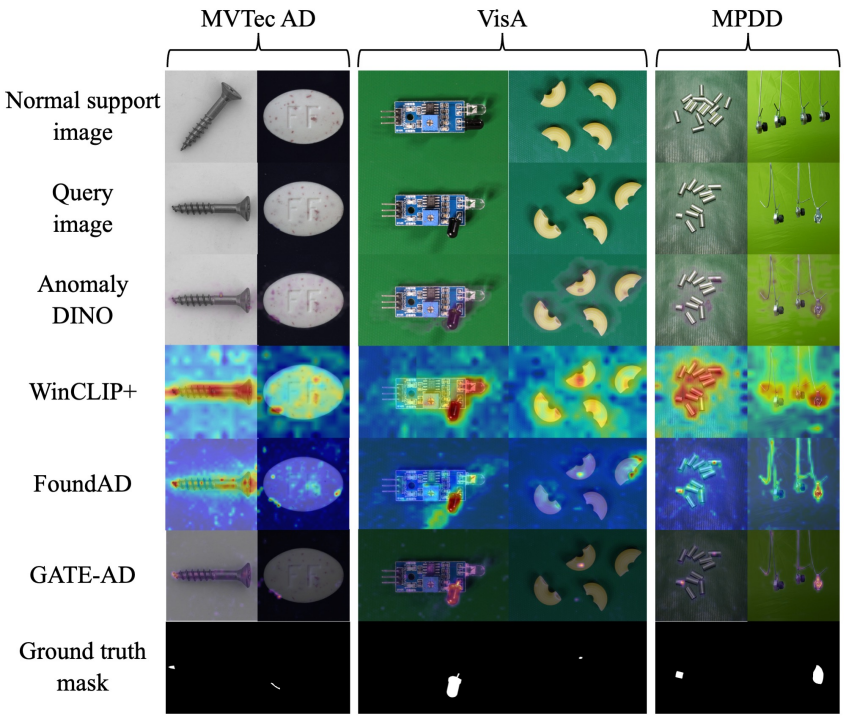

Figure 2: Indicative 1-shot defect detection results. GATE-AD localizes more accurately and robustly defects, across multiple pose, placement, and abnormality type settings.

Table 2: Component ablation of GATE-AD on MVTec AD, VisA, and MPDD (%, mean±std).

Variant	Components				MVTec AD		VisA		MPDD	
	GAT	RA	Mask.	Drop.	I-AUROC	P-AUROC	I-AUROC	P-AUROC	I-AUROC	P-AUROC
1-shot										
DINO 1-NN	–	–	–	–	96.2 ±0.7	95.9 ±0.1	87.4 ±1.1	97.1 ±0.1	78.0 ±2.8	96.9 ±0.1
DINO + RA	–	✓	–	–	96.0 ±0.1	96.1 ±0.0	88.7 ±0.3	97.0 ±0.1	79.3 ±4.2	97.2 ±0.3
GATE-AD w/o RA	✓	–	✓	✓	94.1 ±0.5	95.1 ±0.1	80.5 ±0.3	96.0 ±0.2	55.7 ±1.1	94.4 ±0.4
GATE-AD w/o mask. & drop.	✓	✓	–	–	96.0 ±0.5	95.7 ±0.1	91.1 ±0.9	95.5 ±0.4	81.2 ±0.7	95.9 ±0.3
GATE-AD w/o dropout	✓	✓	✓	–	96.3 ±0.5	96.0 ±0.1	91.9 ±0.3	96.9 ±0.4	82.5 ±0.4	96.3 ±0.3
GATE-AD w/o masking	✓	✓	–	✓	96.8 ±0.5	95.9 ±0.0	91.6 ±0.9	97.2 ±0.4	82.4 ±0.7	96.4 ±0.3
GATE-AD	✓	✓	✓	✓	**97.7 ±0.2**	**96.3 ±0.1**	**93.5 ±0.2**	**97.7 ±0.1**	**84.2 ±0.9**	**97.5 ±0.1**
2-shot										
DINO 1-NN	–	–	–	–	96.7 ±0.9	96.5 ±0.1	89.4 ±1.1	97.5 ±0.1	81.1 ±1.4	97.1 ±0.1
DINO + RA	–	✓	–	–	96.6 ±0.0	96.6 ±0.1	90.2 ±0.2	97.9 ±0.0	81.1 ±2.2	97.0 ±0.2
GATE-AD w/o RA	✓	–	✓	✓	94.8 ±0.2	95.7 ±0.1	82.5 ±1.4	96.4 ±0.2	58.2 ±1.0	94.6 ±0.5
GATE-AD w/o mask. & drop.	✓	✓	–	–	95.3 ±1.0	96.1 ±0.2	91.1 ±0.1	95.7 ±0.2	82.2 ±0.6	96.2 ±0.3
GATE-AD w/o dropout	✓	✓	✓	–	97.0 ±0.9	96.6 ±0.4	92.4 ±0.4	96.6 ±0.5	83.5 ±0.8	96.7 ±0.5
GATE-AD w/o masking	✓	✓	–	✓	96.7 ±1.1	96.4 ±0.3	92.2 ±0.1	96.4 ±0.2	83.4 ±0.6	96.4 ±0.5
GATE-AD	✓	✓	✓	✓	**98.0 ±0.2**	**96.8 ±0.1**	**92.8 ±0.3**	**97.9 ±0.1**	**84.3 ±0.5**	**97.5 ±0.1**
4-shot										
DINO 1-NN	–	–	–	–	97.6 ±0.8	**96.9 ±0.1**	91.7 ±1.1	97.9 ±0.0	83.6 ±1.7	97.3 ±0.2
DINO + RA	–	✓	–	–	97.6 ±0.0	96.2 ±0.0	91.0 ±0.3	97.2 ±0.0	83.4 ±3.2	97.5 ±0.2
GATE-AD w/o RA	✓	–	✓	✓	95.6 ±0.4	96.2 ±0.2	87.9 ±1.3	96.8 ±0.3	58.6 ±3.2	95.1 ±0.4
GATE-AD w/o mask. & drop.	✓	✓	–	–	96.9 ±0.4	96.5 ±0.1	94.2 ±0.2	97.1 ±0.2	83.6 ±0.3	96.9 ±0.1
GATE-AD w/o dropout	✓	✓	✓	–	97.4 ±0.4	96.6 ±0.1	94.2 ±0.4	97.2 ±0.1	83.9 ±0.4	97.2 ±0.2
GATE-AD w/o masking	✓	✓	–	✓	97.3 ±0.4	96.7 ±0.1	**94.4 ±0.1**	98.0 ±0.2	83.8 ±0.3	97.2 ±0.2
GATE-AD	✓	✓	✓	✓	**97.9 ±0.2**	**96.9 ±0.1**	**94.4 ±0.2**	**98.3 ±0.1**	**84.1 ±0.9**	**97.6 ±0.1**
8-shot										
DINO 1-NN	–	–	–	–	**98.3 ±0.4**	**97.2 ±0.0**	93.0 ±0.2	98.2 ±0.0	86.0 ±1.9	97.6 ±0.1
DINO + RA	–	✓	–	–	97.7 ±0.0	**97.2 ±0.0**	92.7 ±0.3	98.2 ±0.0	85.5 ±3.1	97.6 ±0.2
GATE-AD w/o RA	✓	–	✓	✓	95.8 ±0.6	95.5 ±0.2	90.5 ±0.3	97.1 ±0.3	64.5 ±1.3	95.5 ±0.3
GATE-AD w/o mask. & drop.	✓	✓	–	–	97.3 ±0.4	96.7 ±0.3	94.5 ±0.3	97.0 ±0.1	87.3 ±0.4	97.3 ±0.2
GATE-AD w/o dropout	✓	✓	✓	–	97.5 ±0.4	96.8 ±0.2	95.1 ±0.4	97.5 ±0.2	87.5 ±0.4	97.6 ±0.2
GATE-AD w/o masking	✓	✓	–	✓	97.7 ±0.5	96.8 ±0.1	95.1 ±0.2	98.2 ±0.1	87.6 ±0.4	97.7 ±0.2
GATE-AD	✓	✓	✓	✓	98.2 ±0.1	97.0 ±0.1	**95.2 ±0.2**	**98.3 ±0.1**	**87.8 ±0.5**	**97.8 ±0.0**

Table 3: Architectural design analysis on the MVTec AD, VisA, and MPDD benchmarks (%, mean±std). Best results are shown in bold.

		MVTec AD		VisA		MPDD	
Shot	**Variant**	**I-AUROC**	**P-AUROC**	**I-AUROC**	**P-AUROC**	**I-AUROC**	**P-AUROC**
1	GATE-AD (GAT)	**97.7** ±0.2	**96.3** ±0.1	**93.5** ±0.2	**97.7** ±0.1	**84.3** ±1.1	**97.4** ±0.2
	GATE-AD (GCN)	97.3 ±0.2	95.9 ±0.2	92.7 ±0.4	97.4 ±0.1	82.1 ±0.9	95.7 ±0.1
	Transformer Encoder (TE)	97.1 ±0.2	95.7 ±0.1	93.4 ±0.2	97.5 ±0.1	82.4 ±0.9	96.3 ±0.1
	GATE-AD AE	93.2 ±1.0	95.3 ±0.1	80.8 ±0.5	96.2 ±0.1	59.2 ±2.5	94.2 ±0.3
	MAE	66.6 ±5.0	62.0 ±10.1	55.0 ±1.6	71.2 ±0.6	45.0 ±2.7	87.2 ±6.2
2	GATE-AD (GAT)	**98.0** ±0.2	**96.8** ±0.1	**92.8** ±0.3	**97.9** ±0.1	**84.5** ±0.7	**97.4** ±0.2
	GATE-AD (GCN)	97.6 ±0.2	96.4 ±0.2	**92.8** ±0.5	97.6 ±0.1	82.9 ±0.8	96.5 ±0.1
	Transformer Encoder (TE)	97.1 ±1.0	96.5 ±0.2	92.1 ±1.0	97.5 ±0.2	83.4 ±1.0	97.0 ±0.1
	GATE-AD AE	94.2 ±0.3	95.1 ±0.1	82.3 ±1.2	96.5 ±0.2	57.8 ±1.1	94.4 ±0.3
	MAE	66.2 ±3.2	64.4 ±5.0	62.2 ±0.7	71.4 ±0.5	49.4 ±3.9	86.3 ±4.7
4	GATE-AD (GAT)	**97.9** ±0.2	**96.9** ±0.1	**94.4** ±0.2	**98.3** ±0.1	**84.5** ±0.7	**97.5** ±0.2
	GATE-AD (GCN)	97.7 ±0.2	96.7 ±0.1	94.0 ±0.2	98.1 ±0.1	83.3 ±0.8	97.3 ±0.1
	Transformer Encoder (TE)	97.4 ±0.6	96.8 ±0.2	94.1 ±0.6	98.2 ±0.1	83.6 ±0.4	**97.5** ±0.1
	GATE-AD AE	94.9 ±0.4	95.7 ±0.1	87.8 ±0.3	97.0 ±0.1	59.8 ±3.4	95.0 ±0.2
	MAE	70.2 ±0.8	66.2 ±3.6	64.3 ±0.7	72.5 ±0.6	44.5 ±4.7	84.9 ±3.3
8	GATE-AD (GAT)	**98.2** ±0.1	**97.0** ±0.1	**95.2** ±0.2	**98.3** ±0.1	**87.3** ±0.8	**97.8** ±0.1
	GATE-AD (GCN)	98.1 ±0.2	96.9 ±0.2	95.1 ±0.2	**98.3** ±0.1	87.2 ±0.7	97.7 ±0.1
	Transformer Encoder (TE)	97.7 ±0.3	96.9 ±0.2	94.5 ±0.5	**98.3** ±0.1	87.1 ±0.4	97.7 ±0.1
	GATE-AD AE	95.9 ±0.5	95.6 ±0.1	89.7 ±0.4	97.2 ±0.2	65.2 ±0.9	95.4 ±0.2
	MAE	72.2 ±0.1	66.2 ±0.1	62.3 ±0.1	70.8 ±1.2	46.4 ±2.0	77.3 ±10.7

context. c) The substantial degradation observed when RA is removed from the graph-based formulation further shows that **relational modeling and representation alignment are complementary**. In particular, the GAT estimates a context-dependent representation from the local neighborhood, while RA defines the latent space in which this prediction is compared with the corresponding DINO target. d) Feature masking and attention dropout provide further improvements, particularly in the $1/2$-shot regime. Masking **enforces neighborhood-based reconstruction** instead of direct feature copying, whereas attention dropout **regularizes the learned local dependencies** and reduces over-reliance on individual edges; removing both generally leads to a larger performance reduction. Overall, these results indicate that the effectiveness of GATE-AD does not originate from any individual component in isolation, but from their joint formulation for learning **relational normality from very few normal samples**. A query-conditioned representation analysis in Appendix B.4 further illustrates how the GAT forms a context-conditioned prediction and how its disagreement with the projected DINO target localizes anomalous regions.

Architectural design analysis. Following the component analysis, the main architectural choices of GATE-AD are further examined along two complementary directions. First, the choice of relational encoder is investigated by replacing the GAT layers with either conventional Graph Convolutional Network (GCN) layers or global Transformer Encoder (TE) blocks, while retaining the remaining formulation unchanged. Second, the reconstruction formulation is examined through: a) 'GATE-AD AE', where RA is replaced by an MLP graph decoder and the SCE objective is estimated between the decoder output and the input feature representation; and b) 'MAE', which follows a conventional masked-autoencoding formulation [11] and directly reconstructs the input feature tokens. All variants follow the same dataset-specific ablation protocol as the component analysis, with only the architectural choice under examination being modified. Minor numerical variations with respect to the reference GATE-AD results may arise from independent experimental repetitions. Table 3 reports I-AUROC and P-AUROC across all datasets and shot settings.

Discussion. From Table 3, the following key observations can be made: a) GAT, GCN, and TE achieve relatively close performance on MVTec AD and VisA, where the considered relational encoders operate close to the performance ceiling. Hence, the results do not indicate that **GAT alone drives the performance gains**; instead, they further support the importance of relational modeling within the proposed formulation. b) GAT shows its clearest advantage in the more challenging 1-shot MPDD setting. Unlike the global TE with $O(N^2d)$ pairwise attention, GAT operates over the fixed $K = 8$ local graph with $O(KNd)$ complexity, i.e., linear in the number of patches for fixed K. Together with Appendix C, where TE has substantially more parameters and FLOPs and requires approximately an order of magnitude longer per-class training, this supports GAT as the

Table 4: Same-GPU inference efficiency on MVTec AD under the 1-shot setting.

Method	Latency (ms/image)	Throughput (images/s)	Peak memory (MiB)
AnomalyDINO [7]	233.58 ± 119.72	4.28	2197.7
FoundAD [36]	253.90 ± 9.98	3.94	1319.8
FastRecon [8]	152.32 ± 36.18	6.57	**434.9**
WinCLIP+ [14]	391.82 ± 96.43	2.55	646.7
APRIL-GAN [4]	625.28 ± 248.79	1.60	2799.9
GATE-AD	$\mathbf{118.02 \pm 9.60}$	**8.47**	1157.7

more favorable accuracy-efficiency choice, while the comparable accuracy of the relational encoders on MVTec AD and VisA further supports the underlying relational formulation. c) Regarding the reconstruction formulation, replacing RA with an explicit graph decoder (GATE-AD AE) leads to a substantial reduction in image-level detection performance, particularly on VisA and MPDD, indicating that **representation-aligned latent recovery is more effective than direct feature reconstruction** in the considered FS-IVAD setting. d) The considerably lower performance of MAE further shows that conventional masked reconstruction of the input feature tokens is not sufficient for effective normality modeling from very few normal samples. Overall, these results support the adopted architectural formulation, where **sparse relational encoding is combined with latent representation alignment** rather than conventional autoencoding reconstruction.

4.4 Time performance

Reported performance and discussion. Table 4 reports the inference efficiency of GATE-AD and representative FS-IVAD methods under the 1-shot setting on MVTec AD. All methods are evaluated on the same NVIDIA RTX 6000 GPU with batch size 1 and their official configurations; latency, throughput, and peak memory are averaged over three repetitions after 25 warm-up images per class. GATE-AD achieves the **lowest inference latency** (118.02 ± 9.60 ms/image) and the **highest throughput** (8.47 images/s), corresponding to approximately 22.5% lower latency and 28.9% higher throughput than the next-fastest method, FastRecon. Its peak memory usage is 1157.7 MiB, remaining below AnomalyDINO, FoundAD, and APRIL-GAN. This efficiency is consistent with the sparse GAT design, whose computational cost after the frozen backbone scales linearly with the number of patches for a fixed neighborhood size.

5 Conclusion

This paper introduced GATE-AD, a Few-Shot Industrial Visual Anomaly Detection (FS-IVAD) framework that models normality through masked, representation-aligned relational reconstruction over frozen ViT patch tokens. Rather than assessing patches independently, GATE-AD reconstructs each patch from its local spatial context using a GAT encoder, with representation alignment, SCE-based scoring, masking, and attention dropout supporting robust learning in the low-shot regime. Extensive evaluation on MVTec AD, VisA, and MPDD demonstrates strong detection and localization performance across 1- to 8-shot settings, achieving the highest image-level AUROC in $10/12$ settings and AP in $8/12$ settings under the unified evaluation protocol, while also attaining the lowest inference latency among the compared methods. Ablation studies further show that the gains arise from the joint formulation of relational modeling, representation alignment, and low-shot regularization. Future work will extend the current local formulation toward multi-scale or global relational reasoning for broader structural anomalies and continual adaptation.

Acknowledgments and Disclosure of Funding

Funding. This work has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101189557, project TORNADO (foundaTion mOdels for Robots that haNdle smAll, soft and Deformable Objects).

The views and opinions expressed in this paper are those of the authors only and do not necessarily reflect those of the European Union or the European Commission.

Competing interests. The authors declare no competing interests.

References

[1] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. Pni: Industrial anomaly detection using position and neighborhood information. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6350–6360. IEEE, 2023.

[2] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The MVTec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. *IJCV*, 129(4):1038–1059, 2021.

[3] Yuhao Chao, Jie Liu, Jie Tang, and Gangshan Wu. Anomalyr1: A GRPO-based end-to-end MLLM for industrial anomaly detection. *arXiv preprint arXiv:2504.11914*, 2025.

[4] Xuhai Chen, Yue Han, and Jiangning Zhang. APRIL-GAN: A zero-/few-shot anomaly classification and segmentation method for CVPR 2023 VAND workshop challenge tracks 1&2. *arXiv preprint arXiv:2305.17382*, 2023.

[5] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020.

[6] Yajie Cui, Zhaoxiang Liu, and Shiguo Lian. A survey on unsupervised anomaly detection algorithms for industrial images. *IEEE Access*, 11:55297–55315, 2023. doi: 10.1109/ACCESS.2023.3282993.

[7] Simon Damm, Mike Laszkiewicz, Johannes Lederer, and Asja Fischer. Anomalydino: Boosting patch-based few-shot anomaly detection with DINOv2. In *IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, pages 1319–1329. IEEE, 2025.

[8] Zheng Fang, Xiaoyang Wang, Haocheng Li, Jiejie Liu, Qiugui Hu, and Jimin Xiao. Fastrecon: Few-shot industrial anomaly detection via fast feature reconstruction. In *ICCV*, pages 17481–17490, 2023.

[9] Tryambak Gangopadhyay, Sungmin Hong, Sujoy Roy, Yash Shah, and Lin Lee Cheong. Benchmarking framework for anomaly localization: Towards real-world deployment of automated visual inspection. *Journal of Manufacturing Systems*, 69:64–75, 2023.

[10] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Univad: A training-free unified model for few-shot visual anomaly detection. In *CVPR*, pages 15194–15203, 2025.

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022.

[12] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pages 594–604, 2022.

[13] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In *ECCV*, pages 303–319. Springer, 2022.

[14] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation. In *CVPR*, pages 19606–19616, 2023.

[15] Stepan Jezek, Martin Jonak, Radim Burget, Pavel Dvorak, and Milos Skotak. Deep learning-based defect detection of metal parts: Evaluating current methods in complex conditions. In *13th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, pages 66–71. IEEE, 2021.

[16] Shima Khoshraftar and Aijun An. A survey on graph representation learning methods. *ACM Transactions on Intelligent Systems and Technology*, 15(1):1–55, 2024.

[17] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[18] Camile Lendering, Erkut Akdag, and Egor Bondarau. Subspacead: Training-free few-shot anomaly detection via subspace modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28557–28566, 2026.

[19] Yiting Li, Adam Goodge, Fayao Liu, and Chuan-Sheng Foo. Promptad: Zero-shot anomaly detection using text prompts. In *IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, pages 1093–1102, 2024.

[20] Yiyue Li, Shaoting Zhang, Kang Li, and Qicheng Lao. One-to-normal: Anomaly personalization for few-shot anomaly detection. *NeurIPS*, 37:78371–78393, 2024.

[21] Yuanwei Li, Elizaveta Ivanova, and Martins Bruveris. FADE: Few-shot/zero-shot anomaly detection engine using large vision-language model. *BMVC*, 2024.

[22] Zhuo Li, Yuhao Yan, Xiangheng Wang, Yifei Ge, and Lin Meng. A survey of deep learning for industrial visual anomaly detection. *Artif. Intell. Rev.*, 58(9):279, 2025.

[23] Jiaqi Liu, Guoyang Xie, Jinbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep industrial image anomaly detection: A survey. *Mach. Intell. Res.*, 21(1):104–135, 2024.

[24] Wenxi Lv, Qinliang Su, and Wenchao Xu. One-for-all few-shot anomaly detection via instance-induced prompt learning. In *ICLR*, 2025. URL `https://openreview.net/forum?id=Zzs3JwknAY`.

[25] Giannis Nikolentzos, George Dasoulas, and Michalis Vazirgiannis. k-hop graph neural networks. *Neural Networks*, 130:195–205, 2020.

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[27] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *CVPR*, pages 14318–14328, 2022.

[28] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[29] Fenfang Tao, Guo-Sen Xie, Fang Zhao, and Xiangbo Shu. Kernel-aware graph prompt learning for few-shot anomaly detection. In *AAAI*, number 7, pages 7347–7355, 2025.

[30] Long Tian, Yufei Li, Yuyang Dai, Wenchao Chen, Xiyang Liu, and Bo Chen. Fastref: Fast prototype refinement for few-shot industrial anomaly detection. *arXiv preprint arXiv:2506.21398*, 2025.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[33] Qishan Wang, Jia Guo, Shuyong Gao, Haofen Wang, Li Xiong, Junjie Hu, Hanqi Guo, and Wenqiang Zhang. Search is all you need for few-shot anomaly detection. *arXiv preprint arXiv:2504.11895*, 2025.

[34] Sheng Wu, Yimi Wang, Xudong Liu, Yuguang Yang, Runqi Wang, Guodong Guo, David Doermann, and Baochang Zhang. DFM: Differentiable feature matching for anomaly detection. In *CVPR*, pages 15224–15233, 2025.

[35] Guoyang Xie, Jingbao Wang, Jiaqi Liu, Feng Zheng, and Yaochu Jin. Pushing the limits of fewshot anomaly detection in industry vision: GraphCore. In *ICLR*, 2023.

[36] Guangyao Zhai, Yue Zhou, Xinyan Deng, Lars Heckler, Nassir Navab, and Benjamin Busam. Foundation visual encoders are secretly few-shot anomaly detectors. *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

[37] Jiawen Zhu and Guansong Pang. Toward generalist anomaly detection via in-context residual learning with few-shot sample prompts. In *CVPR*, pages 17826–17836, 2024.

[38] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *ECCV*, pages 392–408. Springer, 2022.

A Hyper-parameter tuning

This section provides a detailed analysis of the staged hyper-parameter tuning conducted to determine suitable and robust configurations for GATE-AD. A sequential tuning strategy is followed, where each hyper-parameter is examined while the remaining settings are kept fixed. Specifically, the analysis considers: a) input image resolution, b) backbone network, c) number of last transformer layers, d) graph neighborhood size, e) number of attentional layers, f) cardinality of the GAT layers, g) cardinality of the latent representation, h) objective function, i) masking ratio, j) dropout rate, and k) anomalous patches ratio. Most analyses are conducted on MVTec AD and VisA, while the backbone and graph neighborhood size are additionally examined on MPDD. A separate training-budget analysis is also conducted on MVTec AD under the 1-shot setting. The resulting dataset-specific configurations are used for the component and architectural analyses of Section 4.3; the controlled cross-method comparison follows the unified protocol defined in Section 4.1.

In the dataset-specific configurations, the backbone is DINOv2 ViT-L/14 on MVTec AD (with $\beta = 488$), DINOv3 ViT-B/16 on VisA, and DINOv3 ViT-L/16 on MPDD (see Appendix A.2). The image-level score uses $\epsilon = 0.025$ on MVTec AD and $\epsilon = 0.01$ on VisA, following the top-k convention of FoundAD [36]. On MPDD, max pooling is used, following RegAD [13], GraphCore [35], and FastRecon [8]. Pixel-level metrics are computed directly from the anomaly maps and therefore do not depend on the pooling rule. Since the benchmarks provide no separate validation split, the dataset-specific settings are selected on the benchmark test sets. For this reason, all cross-method comparisons follow the unified protocol.

A.1 Input image resolution

Fig. 4 illustrates how the input image resolution β (Section 3.2) affects the performance of GATE-AD across the MVTec AD and VisA benchmarks. From the presented results it can be seen that resolutions of 512 and 768 pixels exhibit comparable performance. On the contrary, a resolution of 256 pixels results in noticeably lower I-AUROC and PRO scores across all shot counts. Additionally, while 768 pixels occasionally introduce slightly higher PRO scores in certain settings, the improvement over 512 pixels is marginal. To this end, a resolution of 512 pixels is selected for the DINOv3 configurations, providing a balance between detection accuracy and computational cost.

A.2 Backbone

Fig. 5 illustrates how the backbone network (Section 3.2) affects the performance of GATE-AD across the MVTec AD, VisA, and MPDD benchmarks. From the presented results it can be seen that no single backbone or distillation size is proven to be universally optimal across all datasets. For MVTec AD, DINOv2 ViT-L/14 generally yields the best performance. For VisA, DINOv3 ViT-B/16 is proven to be the most effective. For MPDD, DINOv3 ViT-L/16 exhibits the highest accuracy.

A.3 Graph neighborhood size

Fig. 3 illustrates how the graph neighborhood size $K = |\mathcal{N}(i)|$ (Section 3.3) affects the performance of GATE-AD across the MVTec AD, VisA, and MPDD benchmarks. From the presented results it can be seen that GATE-AD remains generally robust across the considered neighborhood sizes. In particular, $K = 8$ consistently achieves the highest AUPRO across all datasets and shot settings, while also yielding the highest I-AUROC across all shot settings on the more challenging MPDD

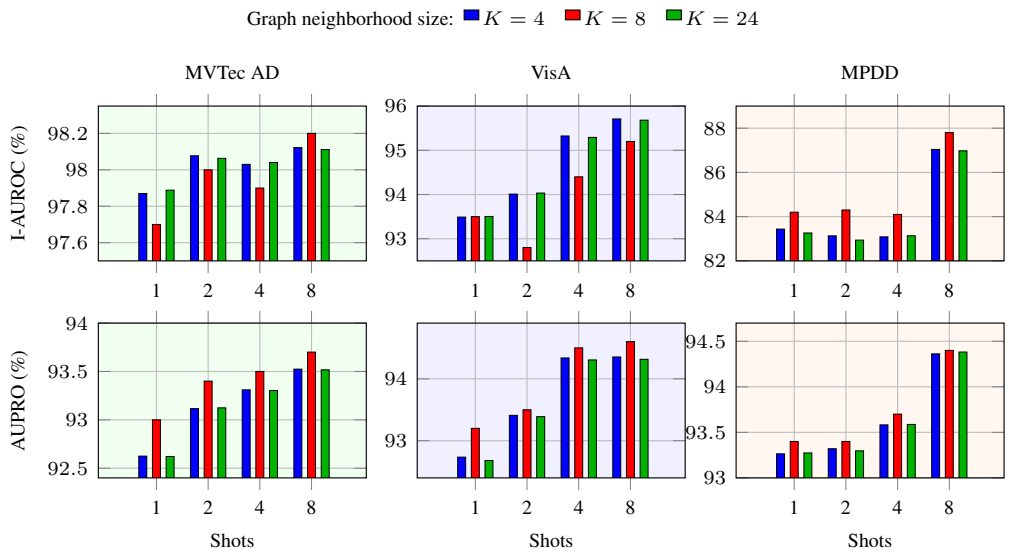

Figure 3: Effect of the graph neighborhood size on the MVTec AD, VisA, and MPDD benchmarks.

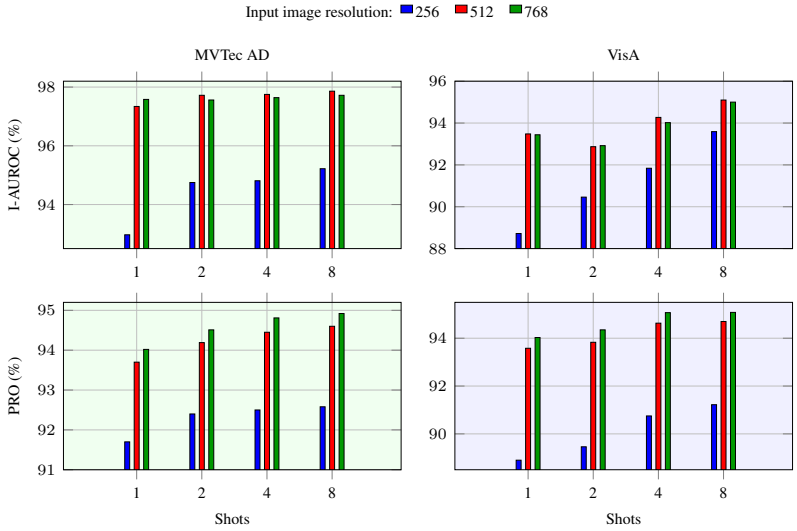

Figure 4: Effect of the input image resolution on the MVTec AD and VisA benchmarks.

benchmark. On the contrary, $K = 4$ excludes diagonal neighbors and therefore restricts the available local context, while increasing the neighborhood to $K = 24$ does not provide a consistent performance improvement and approximately triples the number of graph edges and message-passing operations with respect to $K = 8$. To this end, $K = 8$ is selected as the default graph connectivity.

A.4 Number of last transformer layers

Fig. 6 illustrates how the number of the last transformer layers ζ (Section 3.2) affects the performance of GATE-AD across the MVTec AD and VisA benchmarks. From the presented results it can be seen that increasing the number of layers generally improves Image AUROC, until it stabilizes or slightly declines when including too many layers. In particular, averaging patch tokens from the last $\zeta = 8$ layer leads to the highest overall performance.

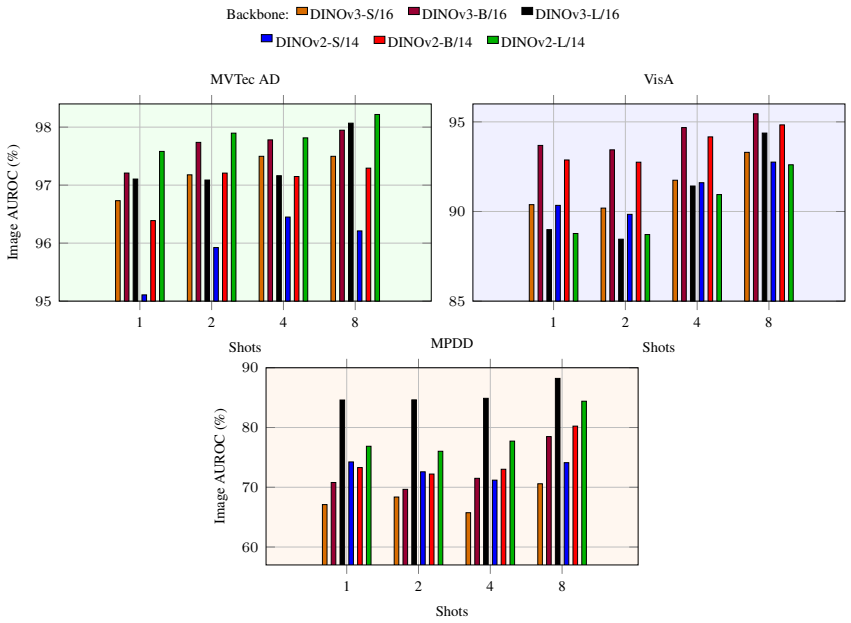

Figure 5: Effect of the backbone on the MVTec AD, VisA, and MPDD benchmarks.

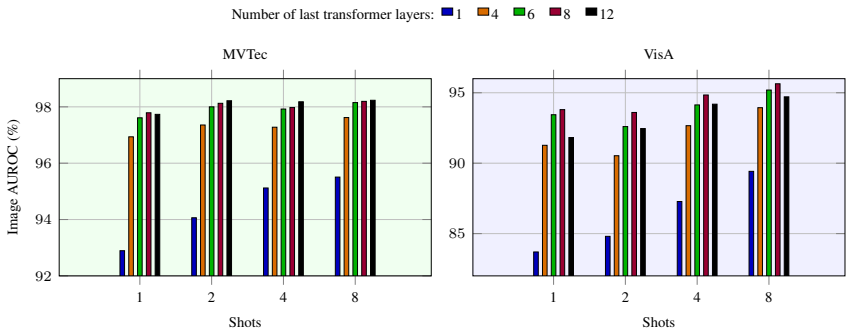

Figure 6: Effect of the number of last transformer layers on the MVTec and VisA benchmarks.

A.5 Number of attentional layers

Fig. 7 illustrates how the number of the number of attentional layers R (Section 3.3) affects the performance of GATE-AD across the MVTec AD and VisA benchmarks. From the presented results it can be seen that setting $R = 2$ or $R = 3$ results in similar performance, though using 3 layers is slightly better in terms of achieving the highest I-AUROC and PRO scores. Further increasing the network depth ($R = 4$) does not necessarily yield better results and, in some cases, leads to slight performance decrease. Based on the above observations, $R = 3$ is used to ensure robust encoding of local relations.

A.6 Cardinality of GAT layers

Fig. 8 illustrates how the cardinality of the GAT layers F (Section 3.3) affects the performance of GATE-AD across the MVTec AD and VisA benchmarks. From the presented results it can be seen that $F = 256$ is sufficient for reaching high recognition performance. Further increasing the cardinality (beyond 256) results into negligible or inconsistent improvements in Image AUROC and PRO scores. $F = 256$ is selected, since it maintains high detection accuracy, while ensuring low inference latency and reduced computational cost.

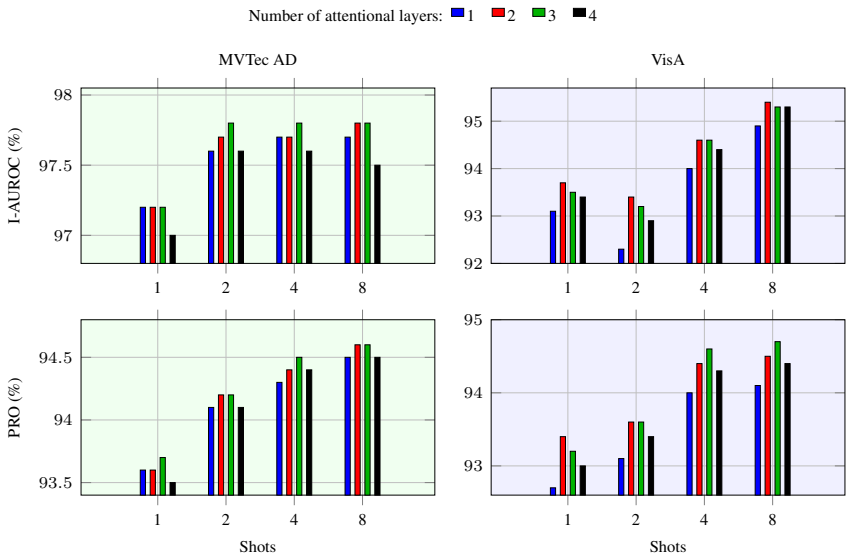

Figure 7: Effect of the number of attentional layers on the MVTec AD and VisA benchmarks.

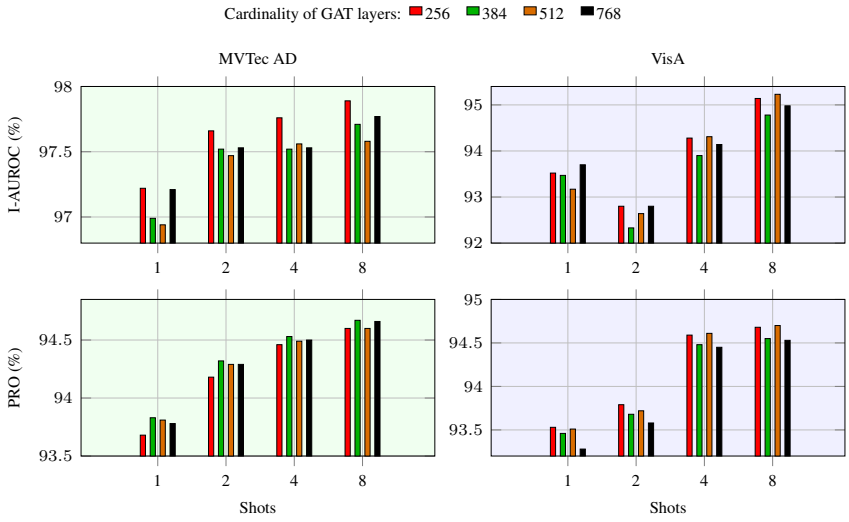

Figure 8: Effect of the cardinality of the GAT layers on the MVTec AD and VisA benchmarks.

A.7 Cardinality of latent representation

Fig. 9 illustrates how the cardinality of the latent representation f (Section 3.3) affects the performance of GATE-AD across the MVTec AD and VisA benchmarks. From the presented results it can be seen that setting $f = 256$ or higher leads to negligible variance in image AUROC. To this end, f is selected equal to 256 for achieving a low computational cost, without sacrificing recognition performance.

A.8 Objective function

Fig. 10 illustrates how the objective function (Section 3.3) affects the performance of GATE-AD across the MVTec AD and VisA benchmarks. From the presented results it can be seen that SCE is clearly advantageous over both MSE and cosine error. Concerning the SCE function, setting the scaling factor $\gamma = 2$ leads to the most consistent results.

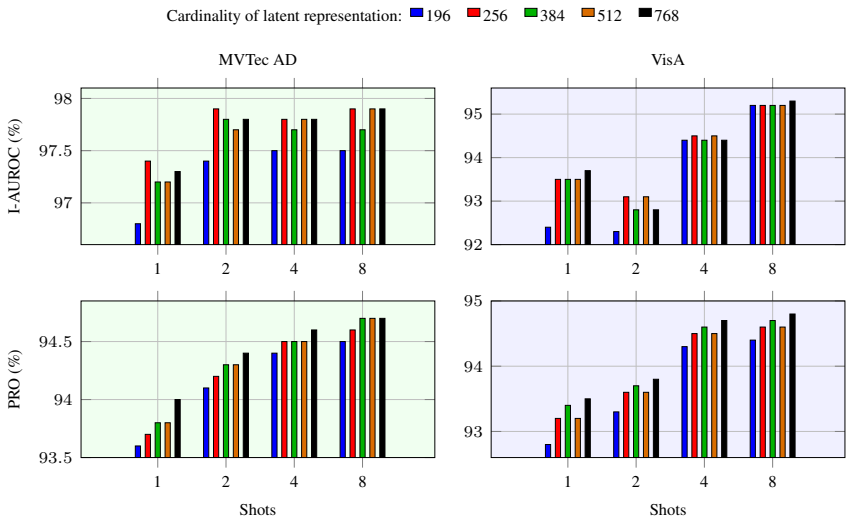

Figure 9: Effect of the cardinality of the latent representation on the MVTec AD and VisA benchmarks.

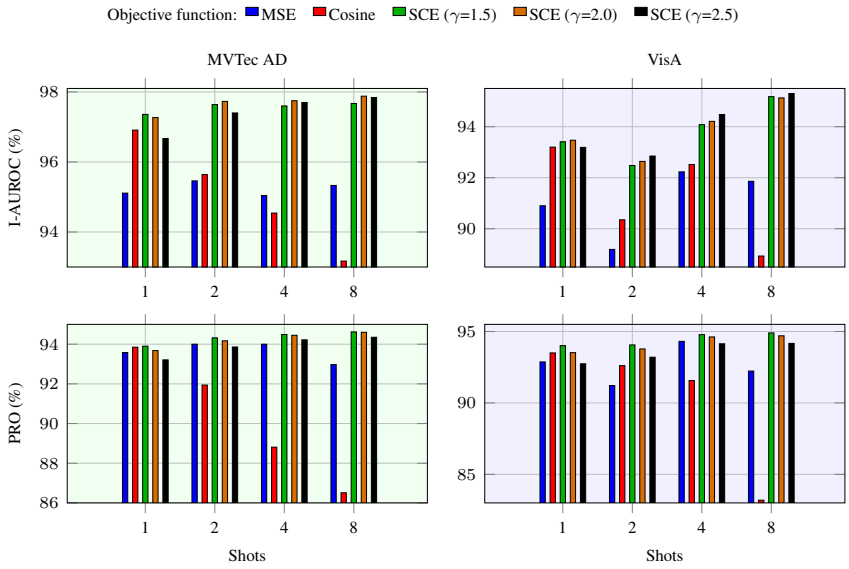

Figure 10: Effect of the objective function on the MVTec AD and VisA benchmarks.

A.9 Masking ratio

Fig. 11 illustrates how the masking ratio μ (Section 3.3) affects the performance of GATE-AD across the MVTec AD and VisA benchmarks. From the presented results it can be seen that $\mu \in [0.1, 0.6]$ results into negligible performance variations, where selecting $\mu = 0.2$ leads to the best overall results.

A.10 Dropout rate

Fig. 12 illustrates how the dropout rate δ (Section 3.3) affects the performance of GATE-AD across the MVTec AD and VisA benchmarks. From the presented results it can be seen that $\delta \in [0.1, 0.4]$ results in the highest detection performance, where $\delta = 0.3$ being the optimal value.

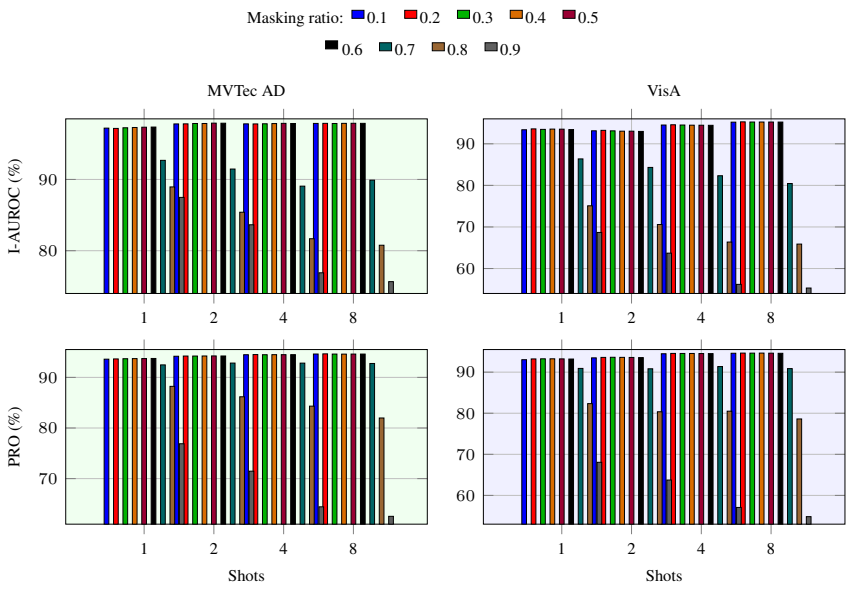

Figure 11: Effect of the masking ratio on the MVTec and VisA benchmarks.

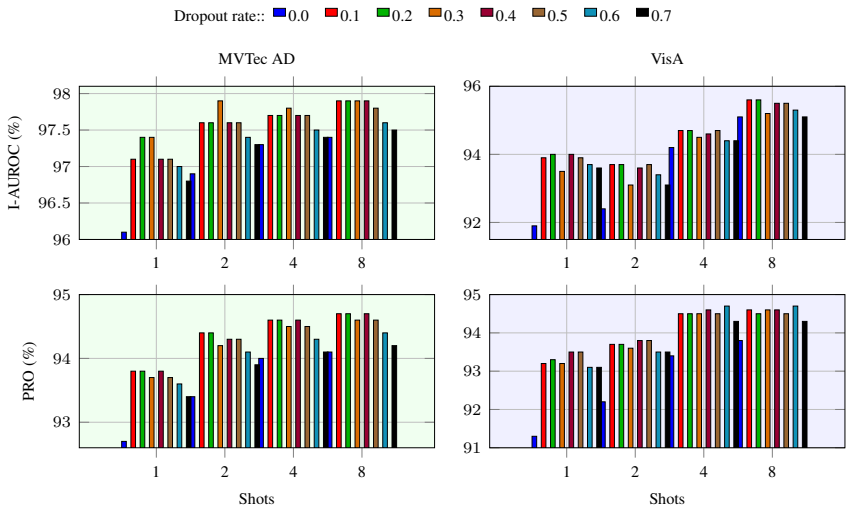

Figure 12: Effect of the dropout rate ablation on the MVTec and VisA benchmarks.

A.11 Anomalous patches ratio

Fig. 13 illustrates how the anomalous patches ratio ϵ (Section 3.4) affects the image-level detection performance of GATE-AD on MVTec AD and VisA. A ratio of $\epsilon = 0.025$ on MVTec AD and $\epsilon = 0.01$ on VisA provides the best overall performance, while deviations from these values result in only moderate performance variations. These dataset-specific values are therefore retained for the within-method analyses of Section 4.3; the comparative evaluation follows the unified protocol defined in Section 4.1.

A.12 Training budget and early stopping

The maximum training budget of 2,000 epochs is employed as an early-stopped upper optimization bound rather than as a fixed training duration. Despite the limited number of normal support images in FS-IVAD, the unit of supervision in GATE-AD is the patch node rather than the image. Hence, even a single support image provides a large number of patch-level training instances. Moreover, at every

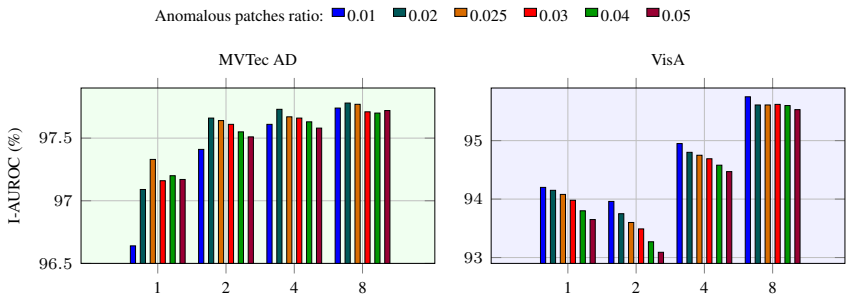

Figure 13: Effect of the anomalous patches ratio on the MVTec AD and VisA benchmarks.

epoch, a newly sampled subset of nodes is masked, such that successive epochs correspond to different reconstruction problems rather than repeated optimization over an identical input configuration. This stochastic corruption acts as an additional regularization mechanism against memorization of the available support images. In practice, early stopping terminates optimization well before the maximum budget in many cases, with an average training duration of approximately $1{,}569$ epochs across all datasets, shot settings, categories, and support-set seeds. To further examine the effect of the training duration, a fixed-budget ablation is conducted on MVTec AD under the 1-shot setting, which represents the regime most susceptible to overfitting. For this experiment, early stopping is disabled and the optimization budget is varied from 100 to $2{,}000$ epochs. As shown in Table 5, detection and localization performance progressively improve as the training budget increases and largely stabilize after approximately $1{,}000$ epochs. Increasing the budget to $2{,}000$ epochs introduces only minor fluctuations, with no indication of performance collapse. These results show that competitive performance is reached well before the maximum optimization budget and support the use of $2{,}000$ epochs as an early-stopped upper bound.

Table 5: Effect of the fixed training budget on MVTec AD under the 1-shot setting, with early stopping disabled (%).

Epochs	I-AUROC	I-AP	P-AUROC	AUPRO
100	96.21	97.77	95.04	91.54
200	97.03	98.26	95.48	92.12
500	97.25	98.48	95.88	92.55
1,000	97.40	98.54	96.11	92.70
1,500	**97.65**	**98.81**	**96.27**	**93.00**
2,000	97.33	98.57	96.18	92.75

B Additional experimental analysis

This section provides a detailed characterization of GATE-AD, complementary to the controlled comparative evaluation of Section 4.2. Unless otherwise stated, the supplementary GATE-AD analyses use the dataset-specific configurations identified through the hyper-parameter analysis of Appendix A. Object-level quantitative results, product-agnostic defect-type analysis, qualitative results, query-conditioned representation analysis, and representative failure cases are examined.

B.1 Quantitative evaluation results

Tables 6-8 illustrate the object-level detection performance of GATE-AD and AnomalyDINO [7] across the MVTec AD, VisA, and MPDD benchmarks. In order to ensure a fair and consistent comparison, both methods utilize the standardized sample selection process. Key insights on a per dataset basis are provided in the following.

MVTec AD (Table 6):

- GATE-AD achieves high detection scores even in the 1-shot setting, accomplishing an 100.0% image AUROC for classes ‘Bottle’ and ‘Leather’.

Table 6: FS-IVAD object-level detection results for GATE-AD and AnomalyDINO on the MVTec AD benchmark.

Object	1-shot		2-shot		4-shot		8-shot	
	GATE-AD	AnomalyDINO	GATE-AD	AnomalyDINO	GATE-AD	AnomalyDINO	GATE-AD	AnomalyDINO
Image AUROC (%)								
Bottle	**100.0** ±0.0	99.7 ±0.2	99.8 ±0.1	**99.9** ±0.1	**100.0** ±0.0	99.9 ±0.1	**100.0** ±0.0	**100.0** ±0.0
Cable	**94.7** ±0.1	92.7 ±0.8	**95.5** ±0.1	92.4 ±1.1	**94.3** ±0.1	93.8 ±0.9	95.0 ±0.4	**95.2** ±0.3
Capsule	**93.1** ±0.1	90.2 ±5.5	**94.3** ±0.7	89.2 ±7.9	95.0 ±0.9	**95.8** ±0.5	**95.8** ±0.8	95.6 ±0.5
Carpet	99.8 ±0.0	**100.0** ±0.0	99.9 ±0.0	**100.0** ±0.0	99.8 ±0.0	**100.0** ±0.0	99.9 ±0.0	**100.0** ±0.0
Grid	**99.6** ±0.0	99.1 ±0.2	**100.0** ±0.0	99.2 ±0.4	**100.0** ±0.0	99.5 ±0.3	**100.0** ±0.0	99.5 ±0.1
Hazelnut	**97.7** ±0.9	97.5 ±2.6	98.9 ±0.4	**99.6** ±0.5	99.6 ±0.3	**99.8** ±0.1	99.9 ±0.1	**100.0** ±0.0
Leather	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0
Metal nut	99.4 ±0.3	**99.9** ±0.1	99.2 ±0.3	**100.0** ±0.0	99.7 ±0.1	**100.0** ±0.0	99.7 ±0.0	**100.0** ±0.0
Pill	**96.3** ±1.0	93.7 ±0.9	**97.3** ±0.6	95.4 ±0.7	**96.5** ±0.5	96.0 ±0.2	**97.7** ±0.3	97.2 ±0.2
Screw	**94.6** ±0.2	93.2 ±0.3	**95.1** ±0.2	93.5 ±0.8	**94.5** ±0.0	92.7 ±2.3	**94.5** ±0.2	93.5 ±1.1
Tile	99.5 ±0.1	**100.0** ±0.0	99.7 ±0.0	**100.0** ±0.0	99.9 ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0
Toothbrush	**98.1** ±0.4	97.4 ±0.5	97.5 ±0.0	**98.1** ±1.0	96.4 ±0.0	**97.5** ±0.6	96.7 ±0.4	**97.7** ±1.6
Transistor	**93.0** ±0.4	90.9 ±1.2	**93.1** ±0.7	89.4 ±4.6	**93.3** ±0.4	93.2 ±2.2	94.6 ±0.1	**96.2** ±1.3
Wood	**99.8** ±0.0	98.0 ±0.2	**100.0** ±0.0	98.0 ±0.1	**100.0** ±0.0	97.9 ±0.2	**100.0** ±0.0	98.3 ±0.4
Zipper	**99.8** ±0.2	97.4 ±0.9	**99.9** ±0.1	98.9 ±0.4	**99.8** ±0.1	99.0 ±0.4	**99.9** ±0.1	99.6 ±0.1
Mean	**97.7** ±0.2	96.6 ±0.4	**98.0** ±0.2	96.9 ±0.7	**97.9** ±0.2	97.7 ±0.2	**98.2** ±0.1	**98.2** ±0.2
Pixel PRO (%)								
Bottle	95.7 ±0.1	**95.9** ±0.3	95.7 ±0.1	**96.3** ±0.0	95.9 ±0.1	**96.7** ±0.1	96.0 ±0.1	**96.5** ±0.4
Cable	**89.8** ±0.4	89.4 ±0.4	**90.4** ±0.1	89.5 ±0.4	90.1 ±0.1	**90.4** ±0.3	**90.5** ±0.1	**90.5** ±0.2
Capsule	96.6 ±0.1	**97.1** ±0.3	96.7 ±0.1	**97.3** ±0.6	97.0 ±0.1	**97.9** ±0.1	97.1 ±0.1	**97.9** ±0.1
Carpet	96.7 ±0.0	**97.8** ±0.0	96.8 ±0.0	**97.9** ±0.0	96.7 ±0.1	**97.8** ±0.0	96.7 ±0.1	**97.8** ±0.0
Grid	96.5 ±0.2	**97.2** ±0.1	96.6 ±0.2	**97.2** ±0.1	96.6 ±0.1	**97.2** ±0.1	96.6 ±0.1	**97.2** ±0.0
Hazelnut	96.7 ±0.1	**97.4** ±0.4	97.1 ±0.0	**98.0** ±0.3	97.3 ±0.1	**98.0** ±0.1	97.3 ±0.1	**98.1** ±0.1
Leather	**98.0** ±0.2	97.9 ±0.1	**97.9** ±0.2	97.8 ±0.0	**97.9** ±0.1	97.6 ±0.1	**97.8** ±0.2	97.6 ±0.1
Metal nut	92.2 ±0.1	**94.2** ±0.0	92.7 ±0.2	**94.6** ±0.2	93.4 ±0.0	**95.3** ±0.1	93.8 ±0.2	**95.4** ±0.3
Pill	96.6 ±0.1	**97.3** ±0.1	96.9 ±0.1	**97.5** ±0.1	96.7 ±0.1	**97.6** ±0.1	96.9 ±0.1	**97.7** ±0.1
Screw	**96.1** ±0.1	93.4 ±0.4	**96.6** ±0.1	94.3 ±0.4	**96.6** ±0.1	94.5 ±0.9	**96.8** ±0.1	95.2 ±0.5
Tile	**89.7** ±0.4	88.0 ±0.2	**90.1** ±0.5	87.6 ±0.4	**90.0** ±0.4	87.1 ±0.4	**89.9** ±0.4	86.7 ±0.2
Toothbrush	**96.0** ±0.0	94.0 ±0.8	**95.9** ±0.1	94.7 ±0.3	**95.5** ±0.0	94.7 ±0.3	**95.7** ±0.0	95.1 ±0.8
Transistor	**68.2** ±0.3	67.3 ±2.1	**70.5** ±0.1	68.4 ±3.1	**70.9** ±0.1	70.6 ±1.4	71.4 ±0.1	**75.3** ±1.9
Wood	93.4 ±0.1	**94.7** ±0.1	93.6 ±0.0	**94.6** ±0.0	93.4 ±0.0	**94.6** ±0.1	93.9 ±0.1	**94.4** ±0.3
Zipper	**93.2** ±0.1	89.2 ±1.2	**94.2** ±0.4	90.2 ±0.4	**94.1** ±0.3	91.2 ±0.3	**94.8** ±0.3	91.1 ±0.4
Mean	**93.0** ±0.2	92.7 ±0.1	**93.4** ±0.1	93.1 ±0.2	**93.5** ±0.1	93.4 ±0.1	93.7 ±0.1	**93.8** ±0.1

- GATE-AD performs robustly for challenging classes, like 'Cable' (94.7% image AUROC in the 1-shot setting) and 'Capsule' (93.1% image AUROC in the 1-shot case), significantly outperforming AnomalyDINO.
- GATE-AD maintains high localization accuracy in most cases, exhibiting up to 98.0% pixel PRO for class 'Leather' in the 1-shot setting.

VisA (Table 7):

- GATE-AD exhibits high performance for most classes across all settings, e.g., 'Capsules' (99.3% image AUROC in the 1-shot case) and 'Chewing gum' (98.8% image AUROC in the 1-shot case), clearly surpassing the respective AnomalyDINO metrics.
- For complex electronic components (i.e., printed circuit boards), GATE-AD maintains high detection accuracy: 'PCB1' (98.0% image AUROC in the 1-shot setting), 'PCB2' (92.0% image AUROC in the 1-shot case), and 'PCB4' (88.3% image AUROC in the 1-shot setting).
- For food and organic shapes, GATE-AD exhibits increased localization performance, e.g., 'Capsules' (99.3% image AUROC in the 1-shot setting) and 'Chewing gum'(98.8% image AUROC in the 1-shot case).

MPDD (Table 8):

Table 7: FS-IVAD object-level detection results for GATE-AD and AnomalyDINO on the VisA benchmark.

Object	1-shot		2-shot		4-shot		8-shot	
	GATE-AD	AnomalyDINO	GATE-AD	AnomalyDINO	GATE-AD	AnomalyDINO	GATE-AD	AnomalyDINO
Image AUROC (%)								
Candle	**97.8** ±0.4	87.9 ±0.3	**97.6** ±0.3	89.4 ±3.0	**97.5** ±0.3	91.3 ±2.9	**97.7** ±0.2	93.5 ±1.2
Capsules	**99.3** ±0.3	98.4 ±0.5	**99.5** ±0.2	98.9 ±0.1	**99.6** ±0.2	99.2 ±0.1	**99.6** ±0.2	99.2 ±0.1
Cashew	**95.5** ±0.4	86.1 ±3.6	**96.1** ±0.2	89.4 ±3.8	**98.7** ±0.2	94.5 ±0.7	**98.6** ±0.2	95.3 ±0.6
Chewing gum	**98.8** ±0.2	98.0 ±0.4	**99.1** ±0.1	98.6 ±0.4	**99.3** ±0.1	98.8 ±0.2	**99.4** ±0.1	98.8 ±0.2
Fryum	**97.4** ±0.4	94.8 ±0.5	**98.4** ±0.1	96.5 ±0.2	**98.7** ±0.2	97.0 ±0.1	**98.6** ±0.2	97.6 ±0.4
Macaroni1	**96.0** ±0.4	87.5 ±1.1	**95.4** ±0.5	87.5 ±0.9	**95.9** ±0.3	89.5 ±1.4	**96.9** ±0.3	90.1 ±1.7
Macaroni2	**75.9** ±1.4	62.2 ±4.3	**73.3** ±1.2	66.9 ±1.9	**72.6** ±1.0	70.0 ±1.7	**75.7** ±0.7	74.9 ±0.4
PCB1	**98.0** ±0.3	91.5 ±2.0	**97.9** ±0.4	91.2 ±2.7	**97.6** ±0.4	94.0 ±2.1	**97.6** ±0.3	95.5 ±0.5
PCB2	**92.0** ±0.5	84.8 ±1.2	**91.5** ±0.6	88.1 ±2.5	**91.4** ±0.7	91.1 ±1.7	92.1 ±0.3	**92.6** ±0.3
PCB3	**85.7** ±0.3	84.9 ±3.3	**91.1** ±0.2	89.4 ±3.8	94.0 ±0.2	**94.3** ±0.4	93.6 ±0.4	**95.6** ±0.2
PCB4	**88.3** ±0.8	79.9 ±13.7	78.2 ±2.0	**87.4** ±11.3	91.6 ±1.0	**96.2** ±2.6	96.2 ±0.3	**98.0** ±0.3
Pipe fryum	**96.9** ±0.6	92.7 ±2.7	**95.9** ±0.7	93.3 ±1.2	**96.1** ±0.5	94.6 ±1.9	**96.7** ±0.5	95.0 ±1.6
Mean	**93.5** ±0.2	87.4 ±1.2	**92.8** ±0.3	89.7 ±1.3	**94.4** ±0.2	92.6 ±0.9	**95.2** ±0.2	93.8 ±0.3
Pixel PRO (%)								
Candle	**97.0** ±0.2	96.8 ±0.4	**97.1** ±0.2	97.0 ±0.2	**97.3** ±0.2	97.2 ±0.1	**97.4** ±0.2	97.3 ±0.2
Capsules	**97.0** ±0.3	95.1 ±0.7	**97.1** ±0.3	95.5 ±0.2	**97.4** ±0.3	96.3 ±0.4	**97.7** ±0.3	96.7 ±0.2
Cashew	**96.4** ±0.2	96.1 ±0.9	**96.8** ±0.3	96.7 ±0.7	96.9 ±0.4	**97.4** ±0.5	96.9 ±0.3	**97.3** ±0.2
Chewing gum	**93.0** ±1.0	92.0 ±1.0	**93.2** ±0.9	92.9 ±0.3	**93.7** ±0.7	93.0 ±0.1	**93.9** ±0.5	93.1 ±0.3
Fryum	92.8 ±0.3	**93.2** ±0.2	93.8 ±0.4	**93.9** ±0.3	94.0 ±0.3	**94.5** ±0.4	93.9 ±0.3	**94.9** ±0.3
Macaroni1	97.1 ±0.3	**97.5** ±0.3	97.3 ±0.3	**97.9** ±0.3	97.6 ±0.3	**98.3** ±0.2	97.9 ±0.2	**98.6** ±0.2
Macaroni2	**96.1** ±0.3	92.0 ±0.7	**96.1** ±0.3	93.0 ±0.4	**96.4** ±0.3	93.9 ±0.8	**96.8** ±0.3	95.0 ±0.6
PCB1	**93.5** ±0.5	92.6 ±0.2	**94.2** ±0.4	92.5 ±0.5	**94.4** ±0.3	93.3 ±0.5	**94.2** ±0.3	93.9 ±0.2
PCB2	88.8 ±0.5	**89.9** ±0.2	88.9 ±0.4	**90.7** ±0.3	89.5 ±0.3	**91.4** ±0.2	89.6 ±0.3	**92.0** ±0.1
PCB3	84.8 ±0.2	**88.5** ±1.3	86.9 ±0.2	**90.8** ±0.5	89.8 ±0.3	**91.7** ±0.4	88.7 ±0.3	**93.1** ±0.3
PCB4	**84.1** ±0.5	78.5 ±6.8	**82.9** ±0.7	82.0 ±6.1	**88.9** ±0.5	84.1 ±1.5	**90.0** ±0.6	87.9 ±2.3
Pipe fryum	97.7 ±0.1	**98.0** ±0.0	97.8 ±0.1	**97.9** ±0.2	**97.9** ±0.1	97.8 ±0.1	**97.8** ±0.1	97.6 ±0.1
Mean	**93.2** ±0.3	92.5 ±0.5	**93.5** ±0.3	93.4 ±0.6	**94.5** ±0.2	94.1 ±0.1	94.6 ±0.3	**94.8** ±0.2

Table 8: FS-IVAD object-level detection results for GATE-AD and AnomalyDINO on the MPDD benchmark.

Object	1-shot		2-shot		4-shot		8-shot	
	GATE-AD	AnomalyDINO	GATE-AD	AnomalyDINO	GATE-AD	AnomalyDINO	GATE-AD	AnomalyDINO
Image AUROC (%)								
Bracket black	**83.7** ±3.0	51.0 ±6.5	**83.4** ±2.0	59.4 ±8.0	**80.9** ±2.0	64.2 ±8.1	**85.7** ±0.6	69.5 ±2.7
Bracket brown	49.7 ±1.9	**52.0** ±4.6	51.7 ±1.3	**56.8** ±4.4	56.1 ±2.1	**58.6** ±1.2	57.0 ±2.0	**61.3** ±2.8
Bracket white	**80.8** ±3.3	70.8 ±2.3	**81.1** ±2.2	72.0 ±2.8	**81.0** ±1.9	65.7 ±16.7	**93.3** ±1.7	72.5 ±20.1
Connector	**93.8** ±1.0	80.6 ±4.1	**91.8** ±0.8	83.1 ±3.8	**89.4** ±1.0	83.7 ±1.7	**93.6** ±1.3	87.9 ±3.1
Metal plate	99.3 ±0.6	**100.0** ±0.0	99.7 ±0.3	**100.0** ±0.0	99.8 ±0.3	**100.0** ±0.0	99.5 ±0.5	**100.0** ±0.0
Tubes	**97.6** ±0.8	89.2 ±1.4	**98.0** ±0.6	88.8 ±1.5	**97.6** ±0.7	91.6 ±1.9	**97.5** ±0.7	91.6 ±2.1
Mean	**84.2** ±0.9	73.9 ±1.5	**84.3** ±0.5	76.7 ±2.5	**84.1** ±0.9	77.3 ±2.4	**87.8** ±0.5	80.5 ±3.0
Pixel PRO (%)								
Bracket black	**97.4** ±0.5	89.1 ±0.6	**97.2** ±0.5	90.3 ±0.2	**97.3** ±0.3	91.5 ±1.7	**98.3** ±0.0	93.4 ±0.9
Bracket brown	**85.8** ±0.3	80.6 ±1.2	**85.7** ±0.3	83.7 ±1.7	**86.6** ±0.3	84.9 ±1.7	**86.9** ±0.6	85.3 ±2.6
Bracket white	**95.4** ±0.2	95.4 ±0.7	95.2 ±0.3	**96.3** ±1.6	95.4 ±0.3	**96.3** ±1.6	**97.6** ±0.2	95.5 ±2.0
Connector	**90.2** ±0.5	84.4 ±2.4	**89.9** ±0.5	86.2 ±2.8	**90.2** ±0.4	88.0 ±2.8	**91.2** ±0.3	89.0 ±0.7
Metal plate	**92.8** ±0.2	89.0 ±0.3	**93.4** ±0.2	89.4 ±0.9	**93.8** ±0.1	89.7 ±0.6	**93.6** ±0.1	90.5 ±0.8
Tubes	**98.7** ±0.2	86.1 ±1.4	**98.7** ±0.2	87.6 ±1.4	**98.6** ±0.2	88.5 ±1.1	**98.6** ±0.2	88.8 ±0.9
Mean	**93.4** ±0.2	87.4 ±0.6	**93.4** ±0.1	88.9 ±0.9	**93.7** ±0.1	89.8 ±1.1	**94.4** ±0.0	90.4 ±0.5

- GATE-AD achieves high detection scores even in the 1-shot setting, accomplishing an 99.3% image AUROC for class 'Metal plate' and a corresponding 97.6% for category 'Tubes'.

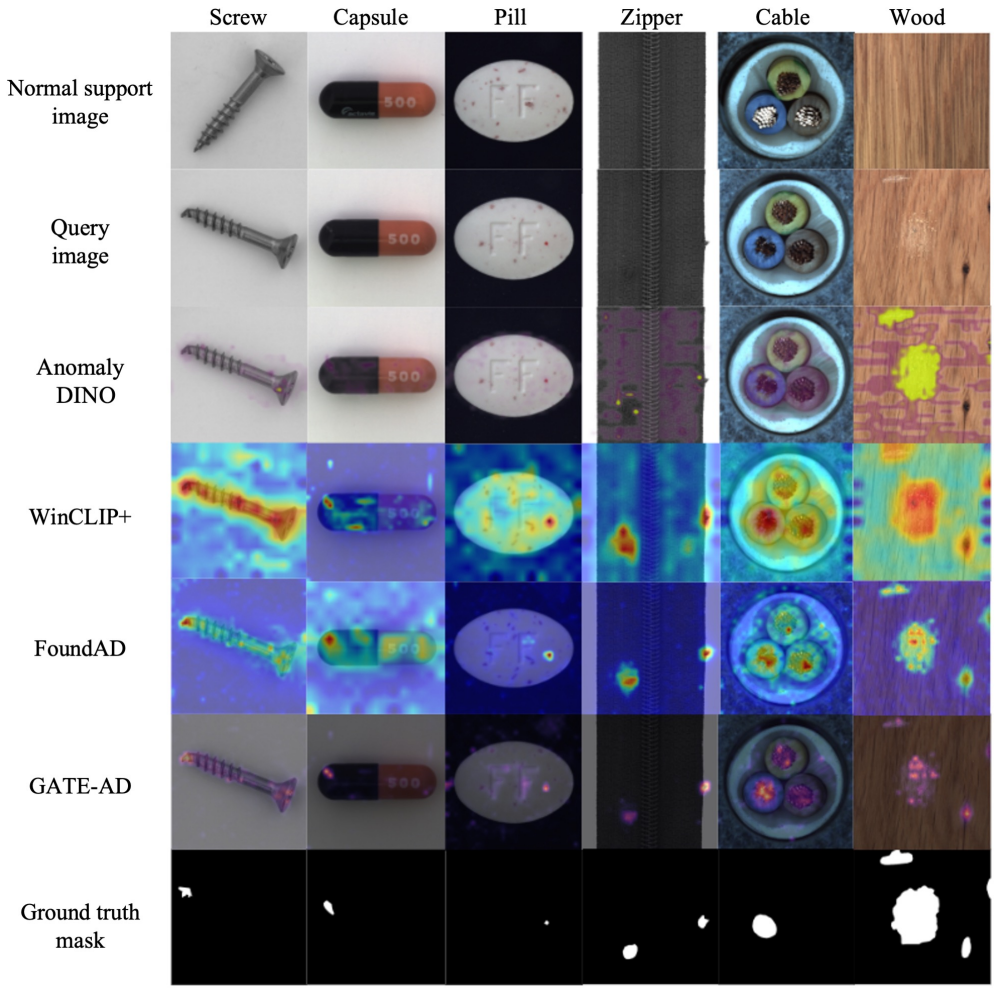

Figure 14: Indicative 1-shot defect detection results on the MVTec AD benchmark.

- GATE-AD demonstrates consistent localization, achieving 98.7% pixel PRO in the 1-shot setting for category 'Tubes' and a corresponding 97.4% for class 'Bracket black'.
- Especially for the localization task, GATE-AD demonstrates higher pixel PRO rates for the vast majority of classes across all settings.

B.2 Product-agnostic defect-type analysis

To further examine the performance of GATE-AD across different defect types, a stratified evaluation is conducted on MVTec AD using the product-agnostic defect annotations of Gangopadhyay et al. [9], grouping anomalous test images into surface (local, e.g., scratches and dents), structural (broader-scale, e.g., deformation, holes, damage, and missing parts), contamination, and combined defects. Since each subgroup contains only anomalous images, image-level AUROC is not meaningful; therefore, pixel-level P-AUROC and AUPRO are reported in Table 9 across the 1-, 2-, 4-, and 8-shot settings. GATE-AD performs strongly across all considered groups, with its clearest performance on local surface defects; at 8-shot, surface defects achieve 98.3% P-AUROC and 96.0% AUPRO, compared with 96.8% and 91.1%, respectively, for broader-scale structural defects, with the difference being most evident in AUPRO and remaining consistent across the considered shot settings. This behavior is consistent with the design of GATE-AD, which operates on a patch-level graph over a fixed local neighborhood and detects anomalies through local reconstruction inconsistency, making the formulation particularly well suited to localized defects. Nevertheless, strong localization performance is also maintained for structural, contamination, and combined defects, while extending

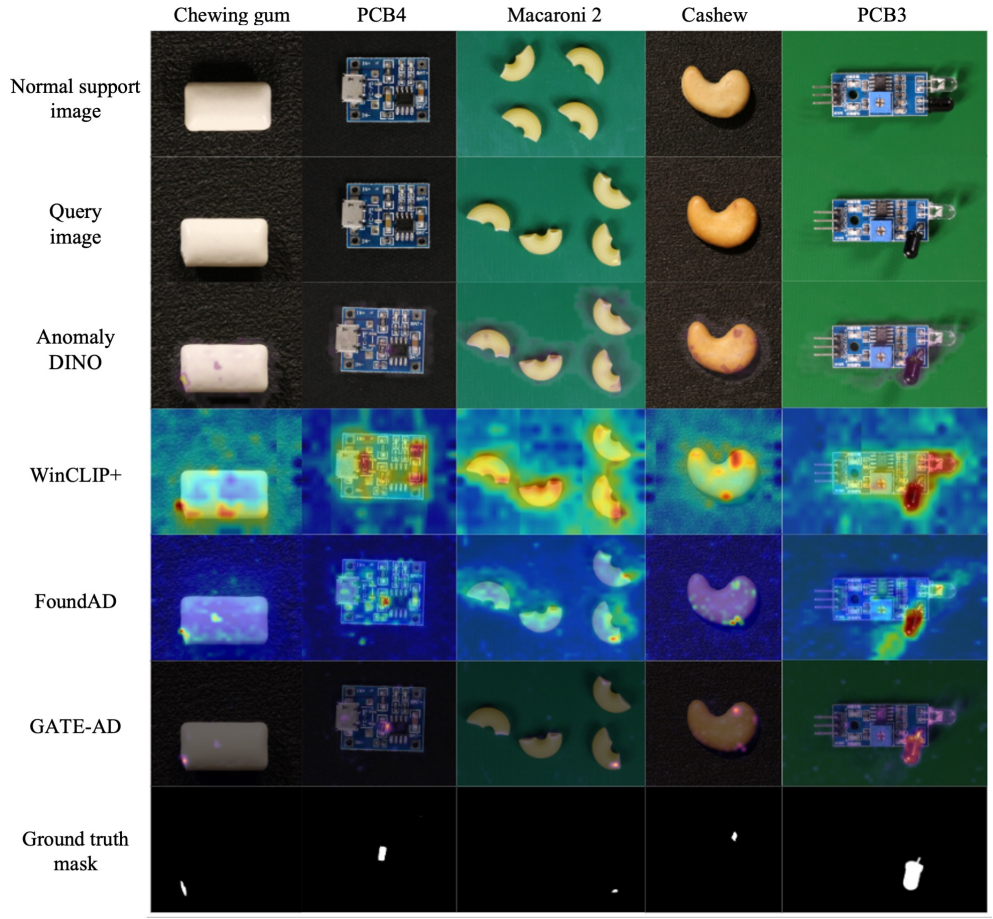

Figure 15: Indicative 1-shot defect detection results on the VisA benchmark.

the current formulation toward multi-scale or global graph reasoning constitutes a promising direction for broader structural anomalies.

Table 9: Defect-type localization results on MVTec AD (P-AUROC / AUPRO, %, mean±std). "Mean" denotes the unweighted average over the four defect groups.

Defect group	**1-shot**	**2-shot**	**4-shot**	**8-shot**
Surface (local)	97.9±0.0 / 95.0±0.0	98.1±0.0 / 95.5±0.0	98.2±0.0 / 95.8±0.0	98.3±0.0 / 96.0±0.1
Structural (broader-scale)	96.4±0.1 / 90.5±0.1	96.7±0.0 / 90.9±0.1	96.8±0.0 / 91.0±0.1	96.8±0.0 / 91.2±0.1
Contamination	97.3±0.1 / 94.3±0.0	97.6±0.1 / 94.7±0.1	97.8±0.1 / 94.8±0.1	97.8±0.1 / 94.8±0.1
Combined	97.5±0.0 / 91.2±0.2	97.6±0.0 / 91.5±0.1	97.5±0.0 / 91.7±0.1	97.6±0.0 / 91.9±0.1
Mean	**97.3 / 92.7**	**97.5 / 93.2**	**97.6 / 93.3**	**97.6 / 93.5**

B.3 Qualitative evaluation results

Figs. 14-16 illustrate indicative 1-shot defect detection results from the application of GATE-AD, AnomalyDINO [7], WinCLIP+ [14], and FoundAD [36] across the MVTec AD, VisA, and MPDD benchmarks. In order to ensure a fair comparison, all baseline methods are evaluated using their default (open source) configuration, while the same single normal support image is employed. Overall, the presented visualizations reinforce the above quantitative analysis results (Section B.1), demonstrating that GATE-AD localizes defects more accurately and robustly than other baseline

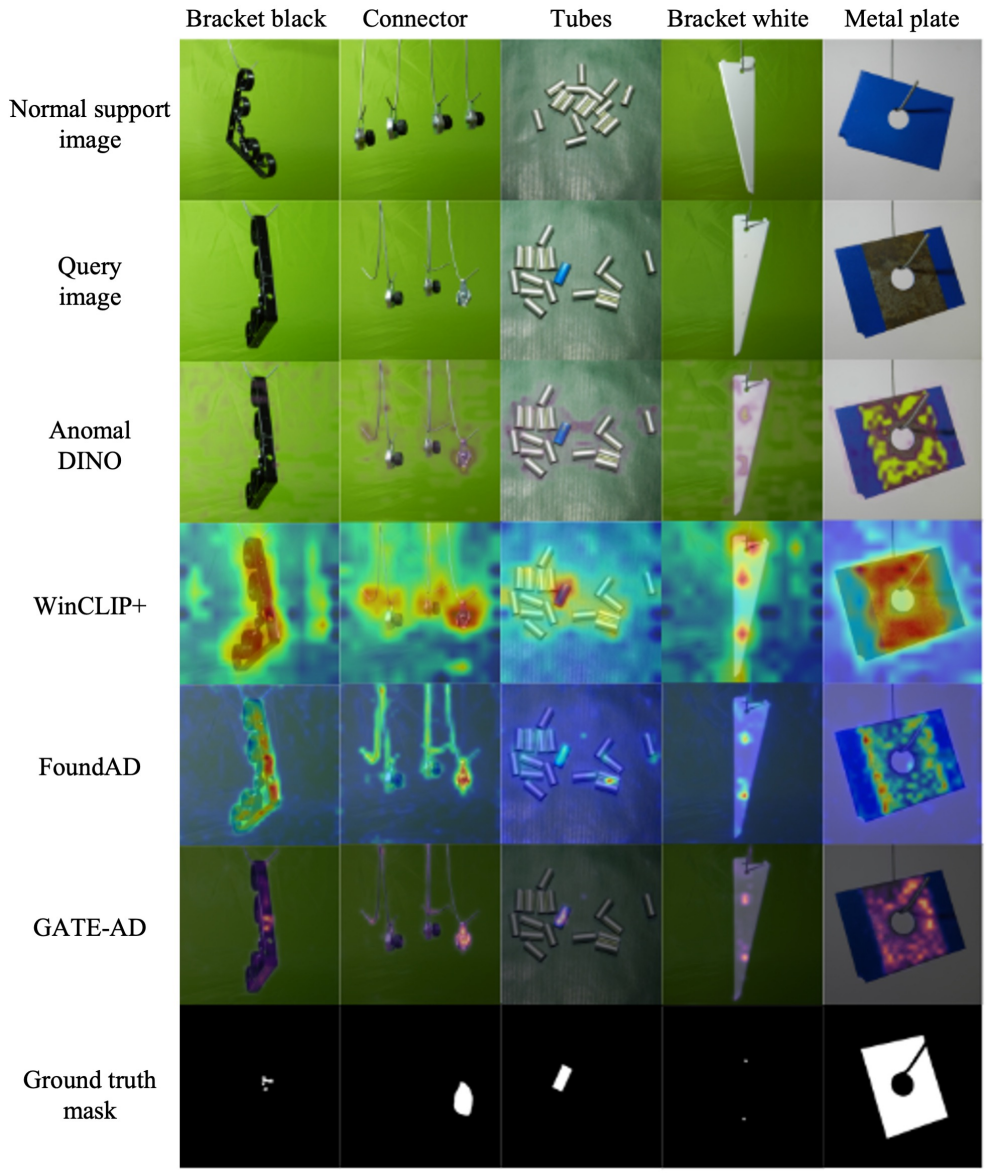

Figure 16: Indicative 1-shot defect detection results on the MPDD benchmark.

methods, across multiple pose, placement, and abnormality type settings. Key insights on a per dataset basis are provided in the following.

MVTec AD (Fig. 14):

- GATE-AD demonstrates superior localization for both structural defects (e.g., in 'Screw' and 'Cable') and surface anomalies (e.g., in 'Pill' and 'Wood').
- While baseline methods (e.g., WinCLIP+) often produce diffuse or noisy heatmaps that cover large, non-defective areas, GATE-AD produces focused, high-intensity responses that align closely with the ground truth masks.
- GATE-AD effectively distinguishes between natural texture variations and genuine defects in categories such as 'Wood'.

VisA (Fig. 15):

- GATE-AD excels in scenes containing multiple objects or complex electronic layouts, such as ‘PCB4’ and ‘PCB3’.
- For organic or irregular shapes (e.g., ‘Macaroni 2’ and ‘Cashew’), GATE-AD successfully detects fine-grained anomalies that other reconstruction-based methods (e.g., FoundAD) may fail to distinguish.
- GATE-AD-generated anomaly maps exhibit minimal activation in background regions, in sharp contradistinction to most baseline methods.

MPDD (Fig. 16):

- GATE-AD demonstrates significant robustness in the case of varied poses and lighting conditions (e.g., ‘Bracket black’ and ‘Tubes’ classes).
- GATE-AD is also shown to accurately identify missing components or surface failures (e.g., ‘Bracket black’ and ‘Connector’ categories).
- Even when the query image has a different orientation than the single support image, GATE-AD detects local defect occurrences (e.g., ‘Metal plate’ class).

B.4 Query-conditioned representation analysis

To further examine how relational reconstruction contributes to anomaly localization, Fig. 17 traces the representation of a query patch selected from either a normal region or a ground-truth defect through the different stages of GATE-AD. At each stage, the heatmap illustrates the attention between the selected query representation and all other patches in the image. For the normal query, the representation remains consistent with its surrounding context through the GAT layers, and the projected prediction closely agrees with the corresponding target, resulting in a low reconstruction error. In contrast, the defective query is initially distinguishable in the DINO representation, while successive GAT layers progressively form a smoother, context-conditioned representation from the surrounding neighborhood. The prediction projection therefore reflects the locally expected representation, whereas the target projection preserves and further separates the observed defect from normal regions. Their disagreement produces a strong localized reconstruction error at the anomalous patch. This analysis illustrates the complementary roles of the two components: the GAT reconstructs the expected patch representation from its local relational context, while representation alignment provides a more discriminative space in which the context-conditioned prediction is compared with the observed target.

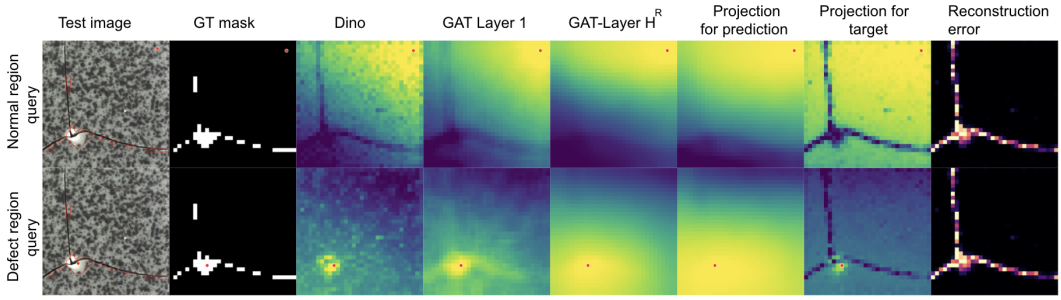

Figure 17: Query-conditioned representation analysis of GATE-AD. The red point denotes a query patch selected from a normal region (top) or from the ground-truth defect (bottom). The heatmaps show query-to-patch similarity in the original DINO representation, after the first and final GAT layers, and after the prediction and target projections, followed by the corresponding reconstruction error. For the normal query, the contextual prediction remains consistent with the observed target, producing a low reconstruction error. For the defective query, graph message passing progressively forms a smoother, context-conditioned prediction from the surrounding spatial context, while the target projection retains and enhances the distinction of the observed defect. Their disagreement produces the localized reconstruction error used by GATE-AD.

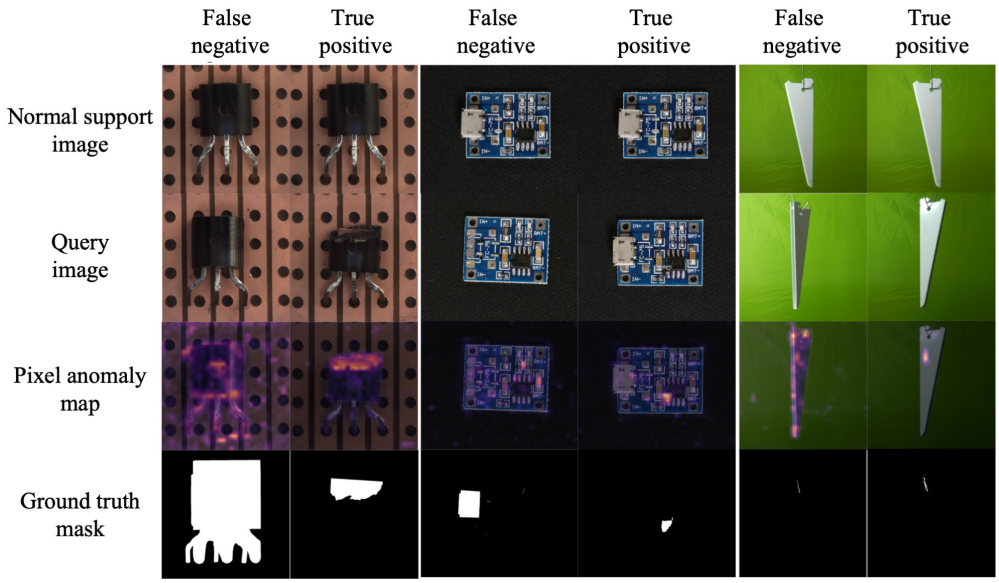

Figure 18: Indicative examples of defect detection failure cases across the MVTec AD, VisA, and MPDD benchmarks.

B.5 Failure cases

Fig. 18 illustrates indicative scenarios where GATE-AD may fail to detect or accurately localize anomalies across the MVTec AD, VisA, and MPDD benchmarks. Key insights extracted from the provided results are as follows:

- Primary source of failure comprises when anomalies arise from global or semantic changes, rather than local surface defects. This includes instances of rotated or misplaced components.
- If a query image contains a pose variation that was not present in the available (small) normal support set, GATE-AD may struggle to distinguish this natural variation from a genuine defect.
- Since GATE-AD is grounded on local-level reconstruction of patch features, it may be sensitive to distinct pixel-level mismatches, but less effective in capturing high-level structural anomalies that do not significantly disrupt local feature consistency.

C Computational analysis

This section complements the architectural analysis of Section 4.3 by examining the computational requirements of the considered relational encoders and reconstruction formulations. End-to-end inference efficiency against existing FS-IVAD methods is reported separately in Table 4. Table 10 reports the trainable parameter count, operator-counted FLOPs, and training time of GATE-AD and the matched architectural variants. The reported training time corresponds to the optimization of each object-class model after frozen-backbone feature extraction and is summarized across the evaluated shot settings. GAT and GCN exhibit comparable computational requirements, whereas the global Transformer Encoder (TE) substantially increases the parameter count, FLOPs, and training time. In particular, TE requires approximately an order of magnitude longer training than GAT across all three datasets. Regarding the reconstruction formulation, GATE-AD is also computationally lighter than the explicit autoencoding alternatives, particularly MAE. Together with the accuracy results of Table 3, these results support the adopted GAT-based formulation as providing a favorable trade-off between accuracy and efficiency.

Table 10: Computational analysis of the architectural variants.

Method	Complexity		Training time (s/class)		
	Params (M)	FLOPs (G)	MVTec AD	VisA	MPDD
GATE-AD (GAT)	**0.99**	**177.48**	**6.7** $\pm$**0.9**	**7.8** $\pm$**0.9**	4.8 $\pm$0.6
GATE-AD (GCN)	1.32	178.15	9.0 $\pm$3.5	10.4 $\pm$3.4	**4.4** $\pm$**0.3**
Transformer Encoder (TE)	22.84	232.37	62.9 $\pm$16.1	83.1 $\pm$23.4	46.6 $\pm$10.0
GATE-AD AE	4.14	182.04	16.1 $\pm$6.5	19.0 $\pm$6.7	18.6 $\pm$1.3
MAE	28.36	232.64	52.4 $\pm$11.4	64.5 $\pm$6.8	75.4 $\pm$1.5

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction section focus on the actual contribution and scope of the paper, which relates to the introduction of a new method, termed GATE-AD, for Few-Shot Industrial Visual Anomaly Detection (FS-IVAD).

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Limitations of the current approach are in principle experimentally/empirically assessed, by reporting specific fail cases of the proposed method and discussing their possible root causes (Appendix B.5).

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: The paper does not include theoretical results.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: Towards the direction of method reproducibility, the method's technical description (Section 3) and hyper-parameters (Section 4.1) are described in detail, while extensive ablation studies regarding hyper-parameter tuning (Appendix A and the method's source code (https://github.com/gthpapadopoulos/GATE-AD) are also provided.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The source code of the developed methods as well as detailed execution instructions are provided (https://github.com/gthpapadopoulos/GATE-AD).

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes].

Justification: The paper reports all required implementation details (Section 4.1).

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: The reported experiments are repeated three times with fixed seeds, and the estimated mean and standard deviation are provided (Section 4.2, and Appendix B).

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: The utilized compute resources are described in detail (Section 4.1).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No].

    Justification: The paper discusses the results of foundational research on industrial anomaly detection and the authors cannot reasonably estimate any negative societal impact.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [No].

    Justification: The developed model does not have a high risk for misuse or dual-use.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: All public code and datasets utilized in this study are appropriately credited (Section 4).

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: Details of the method's code are appropriately provided (supplemental material).

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.