From Pixels to Graphs: Deep Graph-Level Anomaly Detection on Dermoscopic Images

Dehn Xu[1[0009−0005−8457−8153]], Tim Katzke[1,2[0009−0000−0154−7735]], and Emmanuel Müller[1,2[0000−0002−5409−6875]]

[1] Department of Computer Science, TU Dortmund University, Germany
[2] Research Center Trustworthy Data Science and Security, University Alliance Ruhr (UA Ruhr), Germany
{dehn.xu,tim.katzke}@tu-dortmund.de

Abstract. Graph Neural Networks (GNNs) have emerged as a powerful approach for graph-based machine learning tasks. Previous work applied GNNs to image-derived graph representations for various downstream tasks such as classification or anomaly detection. These transformations include segmenting images, extracting features from segments, mapping them to nodes, and connecting them. However, to the best of our knowledge, no study has rigorously compared the effectiveness of the numerous potential image-to-graph transformation approaches for GNN-based graph-level anomaly detection (GLAD). In this study, we systematically evaluate the efficacy of multiple segmentation schemes, edge construction strategies, and node feature sets based on color, texture, and shape descriptors to produce suitable image-derived graph representations to perform graph-level anomaly detection. We conduct extensive experiments on dermoscopic images using state-of-the-art GLAD models, examining performance and efficiency in purely unsupervised, weakly supervised, and fully supervised regimes. Our findings reveal, for example, that color descriptors contribute the best standalone performance, while incorporating shape and texture features consistently enhances detection efficacy. In particular, our best unsupervised configuration using OCGTL achieves a competitive AUC-ROC score of up to 0.805 without relying on pretrained backbones like comparable image-based approaches. With the inclusion of sparse labels, the performance increases substantially to 0.872 and with full supervision to 0.914 AUC-ROC.

Keywords: Image-to-Graph Transformation · Deep Graph Anomaly Detection

1 Introduction

Anomaly detection in images plays a pivotal role across a wide range of applications, from identifying cracks or surface defects in industrial inspection [4] to spotting unusual activities in security footage [9], and detecting tumors or lesions in medical imaging [6]. State-of-the-art convolutional neural networks and vision transformers excel at these tasks when large, labeled datasets are

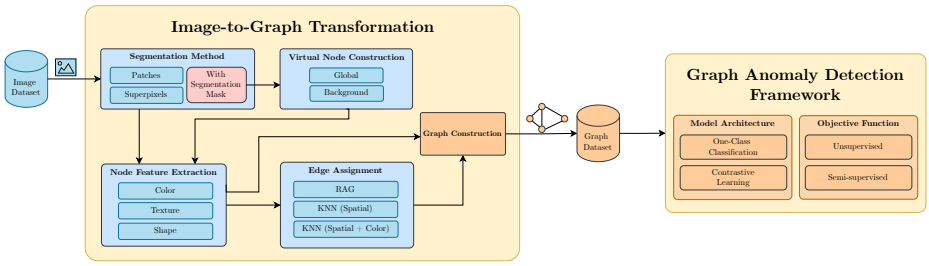


Fig. 1: Overall process of our two-step evaluation study. Images are transformed into graph representations, on which we conduct graph anomaly detection.

available. However, they treat every pixel uniformly and often require extensive pretraining. In anomaly detection, where normal examples vastly outnumber the rare, unpredictable anomalies, this pixel-level redundancy can obscure subtle deviations, inflate computational and data availability costs, and demand powerful hardware that may be impractical in resource-limited clinical or edge settings.

Graph-structured representations offer a compelling alternative by abstracting an image into a compact set of nodes and edges [8,11,28]. Nodes correspond to locally homogeneous regions, defined by superpixels, learned patches, or provided segmentation masks, and can be associated with higher-level features, abstracted from these regions' low-level pixel content. Edges capture spatial adjacency, feature similarity, or both. For anomaly detection, this compressed representation sharply reduces input dimensionality and noise, enabling models to focus on semantically meaningful units rather than millions of individual pixels. The relational inductive bias of graphs likewise confers robustness to small rotations or translations, ensuring that a tiny shift in position does not mask critical deviations without needing data augmentation. Moreover, their compactness allows training lightweight graph-anomaly detectors from scratch, eschewing massive pretrained backbones, thereby reducing both runtime and energy consumption.

In this work, we explore these potential advantages in conjunction with raw anomaly detection performance systematically and build a bridge between image-based and graph-based anomaly detection. To ground this in a high-stakes medical scenario, we focus on the HAM10000 image dataset [32]. This corpus of high-resolution skin-lesion images contains a clear majority of benign nevi as natural "normal" samples alongside multiple smaller, clinically significant classes as diverse anomalies (melanoma, basal cell carcinoma, vascular lesions, and others) and provides pixel-level segmentation masks. Crucially, node features, like shape descriptors, texture histograms, size measures, and color moments, directly align with the established ABCDE criteria [12] (Asymmetry, Border irregularity, Color variation, Diameter, Evolving) used by dermatologists. As is most common in anomaly detection, we evaluate our pipelines primarily in the unsupervised setting (training exclusively on nevi examples) while also investigating supervised variants that leverage varying degrees of labeled anomalies.

As our main contribution, we systematically analyze the effectiveness of various combinations of segmentation strategies, edge-construction methods, node-feature sets, and state-of-the-art graph anomaly detection algorithms in the aforementioned context. By isolating each component, we quantify how region compression, relational robustness, and model compactness translate into detection performance, data reduction, as well as training and inference speed. We focus on identifying bottlenecks in each step of the process, highlighting areas for improvement and future research. Our findings reveal that comparable performance to the results of other image-based studies can still be achieved, even with the reduction of the available data features via segmentation. Furthermore, this data reduction has distinct advantages in both runtime and data efficiency. To ensure reproducibility, our implementation is publicly available at `https://github.com/deX-de/Deep-GLAD-on-Dermoscopic-Images`.

2 Related Work

Researchers have applied the concept of converting images into graphs in various machine learning studies. Han et al. [14] proposed an end-to-end approach that transforms images into non-overlapping patches, extracts features using a CNN-stem, and trains on a GNN. Most other works employed a two-step approach with a separate image-to-graph transformation and downstream task. Specifically, they focused on superpixel algorithms such as SLIC [1], Quickshift [35] or Felzenszwalb [10] to first segment the image into meaningful regions, extract relevant features from these regions and assign edges between the nodes [8,11,28]. Subsequently, they utilized GNNs to learn on extracted image-derived graph representations. The study conducted by Annaby et al. [3] converted images into graphs in the context of melanoma classification. Similarly, they transformed the image into a region adjacency graph using the SLIC algorithm. However, in their downstream task, they handcrafted graph- and node-level features in the spatial and spectral domain. They then used these features in shallow machine learning models in combination with conventional image-based features.

Graph anomaly detection (GAD) seeks to uncover irregularities or nonconformity in graph-structured data at varying granularities [26]. Usually, this is done in an unsupervised or weakly supervised setting. Principal methodological paradigms encompass reconstruction-based models that learn to regenerate normal graph elements and flag high reconstruction error as anomalous [20,23]; contrastive techniques that derive normality by discriminating between augmented or heterogeneous views [18,19,21]; and one-class classification approaches that enclose normal instances within a compact decision region, treating outliers beyond its boundary as anomalies [27,39]. For GAD on image-derived graph representation, the aim is either (1) detecting unusual objects or patterns in an image (essentially node-level anomaly detection) [2,33,34,36], (2) detecting if a whole image is anomalous (graph-level anomaly detection) [40], or both [13,25,38]. While GAD on image-derived graphs is not a new phenomenon, a thorough analysis of

the specific performances of diverse candidate combinations of image-to-graph transformations with current state-of-the-art GLAD methods is missing.

3 Preliminaries and Problem Definition

A graph $\mathcal{G}$ is an ordered pair $(\mathcal{V}, \mathcal{E})$ of $n = |\mathcal{V}|$ nodes forming the *node set* $\mathcal{V} = \{v_1, \ldots, v_n\}$ that can be considered as abstract representations of entities together with the *edge set* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ that show the relationships between aforementioned entities. Two nodes $u, v \in \mathcal{V}$ are *adjacent* if $(u, v) \in \mathcal{E}$ or $(v, u) \in \mathcal{E}$. An *attributed graph*, represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$, includes an attribute set $X = \{\mathbf{x}_v \in \mathbb{R}^d \,|\, v \in \mathcal{V}\}$, where d is the node feature dimension. In principle, an image $\mathbf{I} \in \mathbb{R}^{W \times H \times C}$ of width W, height H, and color channels C, can be thought of as a $4 \cdot D$-connected graph with $\mathcal{V} = \{v_i \,|\, i \in [W \cdot H]\}$, where $\phi : \mathcal{V} \to [W] \times [H]$ is a bijective mapping from nodes to pixel coordinates, $\mathcal{E} = \{(v_i, v_j\} \,|\, v_i \neq v_j, \|\phi(v_i) - \phi(v_j)\| \leq \sqrt{D}\}$, and $X = \{\mathbf{I}_{x,y,\cdot} \,|\, (x, y) \in [W] \times [H]\}$.

The task of graph-level anomaly detection (GLAD) is to distinguish anomalous graphs from normal ones within a given graph dataset [22]. Traditionally, researchers used graph kernels to extract graph-level features. They subsequently applied general shallow anomaly detection methods (e.g., Local Outlier Factor (LOF) [5] or One-Class Support Vector Machines (OC-SVM) [31]) to detect anomalous graphs. With advances in deep learning and graph representation learning, more sophisticated GNN-based methods automatically extract relevant graph features end-to-end. These models primarily operate unsupervised, with most training data containing only the normal class. However, specific GLAD model architectures can incorporate semi-supervision through minor adjustments to the objective function.

The problem statement of our work can be formalized as follows: Given the image set $\mathcal{D}_{img} = \{(\mathbf{I}_1, y_1), \ldots, (\mathbf{I}_n, y_n)\}$, we evaluate different graph construction configurations composed of various segmentation methods, node features, and edge assignments to construct graphs $\mathcal{D}_{graph} = \{(\mathcal{G}_1, y_1), \ldots, (\mathcal{G}_n, y_n)\}$ in the context of anomaly detection. For anomaly detection, the labels $y_1, \ldots, y_n$ denote whether a sample is either normal ($y = 0$) or anomalous ($y = 1$). The evaluation of these configurations considers various state-of-the-art GLAD methods trained on the training split $\mathcal{D}^{train}_{graph} \subset \mathcal{D}_{graph}$ to predict whether unseen graphs are normal or anomalous. In an unsupervised setting, we only consider normal data during training. For (semi-)supervised learning, additional labeled anomalies are incorporated into $\mathcal{D}^{train}_{graph}$.

4 Benchmark Design

In this section, we design a two-step graph-based benchmark pipeline tailored to skin lesion analysis on the HAM10000 dataset [32]. Beginning with a clinically relevant, high-resolution dataset, we convert dermoscopic images into graphs through segmentation-based node construction, rich visual feature extraction, edge assignments, and optionally virtual nodes. This representation allows for

akiec	bcc	bkl	df	mel	nv	vasc	Total
327	514	1099	115	1113	6705	142	10015

Table 1: Image distribution of HAM10000.

effective GLAD, which we evaluate using recent state-of-the-art models. Our benchmark is designed to grasp irregular, heterogeneous patterns of dermatological anomalies and allows both unsupervised and semi-supervised detection settings.

4.1 Dataset Description

The Human Against Machine dataset (HAM10000) consists of 10,015 dermoscopic high-resolution skin lesion images, each with respective diagnoses that are benign or malignant. Correctly diagnosing skin lesions through machine learning is both clinically and economically meaningful, as it makes saving lives possible with fewer human resources. Moreover, the HAM10000 dataset includes segmentation masks, hand-drawn by a professional dermatologist, that match each skin lesion.

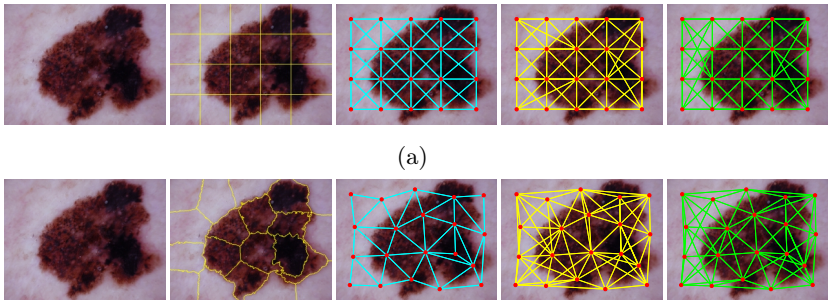
Though the availability of specific diagnosis labels facilitates supervised classification, the imbalanced nature of this dataset, as shown in Table 1, with the majority of images belonging to the nevus (nv) class, makes it well-suited for anomaly detection [6]. Additionally, labeling skin lesions demands the expertise of dermatologists, which involves considerable time and capital. Therefore, HAM10000 is particularly relevant in an unsupervised and semi-supervised context. Furthermore, the ABCDE schema [12], used to differentiate between benign and malignant skin lesions, motivates the application of meaningful node features given by their color, shape, and texture.

HAM10000 shares structural homogeneity across all classes. Unlike natural image datasets, where objects possess distinct geometric features, dermatological lesions appear as blob-like regions without a consistent structure. The irregular structure in skin lesions reduces the importance of position encoding, which is often critical in image-based classification. This characteristic aligns well with GNNs on graph-level tasks, which naturally handle nodes permutation-invariantly.

4.2 Image-to-Graph Transformation

The intuitive image-to-graph transformation, as detailed in Section 3, is generally impractical due to the sheer amount of pixels in high-resolution images.

Hence, the solution explored in our approach is to partition the image into segments of pixels with a segmentation algorithm such as Patch-based decomposition, as visualized in Figure 2a. With this method, the image is divided into a grid of non-overlapping patches $\mathcal{S} = \{S_i \mid i = 1, 2, \ldots, n\}$, where each patch S_i



(a)

(b)

Fig. 2: HAM10000 melanoma image, segmented (a) as patches and (b) with SLICO, followed by edge construction via RAG, KNN_s and KNN_sc.

is a rectangle of size $P \times P$ pixels with $\lfloor H/P \rfloor \cdot \lfloor W/P \rfloor = n$. Another technique is using a superpixel algorithm, specifically Simple Linear Iterative Clustering (SLIC) [1], to segment images, commonly in the context of graph classification on images [28]. This algorithm assigns pixel coordinates $(x, y) \in [W] \times [H]$ to segments S_i through local k-means over a five-dimensional vector space of the pixels' color and spatial features. We utilize a variant of SLIC, SLIC-zero (SLICO, see Figure 2b), which dynamically chooses the compactness parameter that controls the weight between the color and spatial distance for each superpixel.

The resulting segments $\mathcal{S} = \{S_1, \ldots, S_n\}$ can then be used to construct the nodes of the graph $\mathcal{V} = \{v_1, \ldots, v_n\}$. Subsequently, different edge construction techniques are used to connect these nodes. The Region Adjacency Graph (RAG) is built analogously to the grid graph. For any two pixel coordinates $(x, y), (x', y')$ of differing segments S_i, S_j, and connectivity D, if $\|(x, y) - (x', y')\| \leq \sqrt{D}$ we connect the nodes v_i and v_j. A connectivity of 1 and 2 leads to 4-connected and 8-connected neighborhoods, respectively. Assuming pre-computed node features from the feature extraction process, a more efficient approach is using k-nearest neighbors (KNN) on features such as spatial centroid coordinates (KNN_s) or a combination of these coordinates with mean color values (KNN_sc). For our work, all edges are set to be undirected.

Descriptive features are extracted from the image segments to enable comprehensive learning of visual patterns. These features transform the raw pixel values of segmented regions into a concise set of numerical attributes. We categorize relevant visual information into three types, which are evaluated both independently and in combination: color, texture, and shape. For color-based node features, we assign each node the mean color, standard deviation, and skewness of the pixel intensities within its segments in the RGB, HSV, and CIELAB color spaces. Texture features are derived from the Local Binary Pattern (LBP) [24] with $P = 8, R = 1$ and the Gray-level Co-occurrence Matrix (GLCM) [16] on contrast, dissimilarity, energy, correlation, and homogeneity

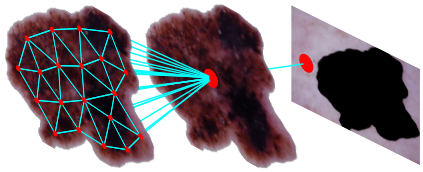


Fig. 3: Virtual nodes used on HAM10000. Segmented nodes are each connected to the lesion node, which in turn is connected to the skin node.

at angles $\{0°, 45°, 90°, 135°\}$. Moreover, we extracted a total of 38 translation-, scaling-, and rotation-invariant moments proposed by [11] as shape features.

Since the HAM10000 dataset includes bitmasks, isolation of skin lesions is feasible. This process is clinically relevant due to its simplified ABCDE schema application. However, it also results in the loss of information from the surrounding skin tissue. To address this limitation, we introduce two virtual nodes: v_g and v_b. Node v_g represents the isolated skin lesion, while node v_b represents the surrounding skin tissue. Both virtual nodes possess an identical number of node features as those derived from the image segmentation method. As illustrated in Figure 3, node v_g is connected to all other nodes, serving as a global communication hub. In contrast, node v_b is linked only to v_g.

4.3 Graph Anomaly Detection Methods

Next, we evaluated the three best-performing state-of-the-art GLAD models from the most recent graph-level anomaly detection benchmark [37], namely SIGNET [19], CVTGAD [18], and OCGTL [27] on the obtained graph representations. SIGNET is a contrastive learning-based approach that optimizes bottleneck subgraphs by enhancing shared structural information across graph views while suppressing irrelevant details. Similarly, CVTGAD extends contrastive learning to both node and graph levels using a simplified transformer with cross-view attention. Finally, OCGTL is an ensemble of $K+1$ feature extractors with a two-part objective function. Each normal graph embedding is mapped to a minimal hypersphere through the one-class classification objective $\mathcal{L}_{\text{OCC}}(\mathcal{G}) = \sum_{k=1}^{K} \|\text{GIN}_k(\mathcal{G}) - c\|$, while the contrastive loss $\mathcal{L}_{\text{GTL}}(\mathcal{G})$ between embeddings of GIN_0 and $\{\text{GIN}_k\}_{k=1}^{K}$ ensures relevance with diversity.

Motivated by [29], we similarly extend the one-class classification objective of OCGTL to enable a semi-supervised approach to subsequently evaluate the impact of (weak) supervision. Given the original loss function in [27]:

$$\mathcal{L}_{\text{OCGTL}} = \mathbb{E}_{\mathcal{G}}[\mathcal{L}_{\text{OCC}}(\mathcal{G}) + \mathcal{L}_{\text{GTL}}(\mathcal{G})],$$

where $\mathbb{E}_{\mathcal{G}}[\cdot]$ is the expectation over the distribution of $\mathcal{G}$, we modify it such that training maximizes the distance between labeled anomalies and the center:

$$\mathcal{L}_{\text{Semi-OCGTL}} = \begin{cases} \mathbb{E}_{\mathcal{G}}[\sum_{k=1}^{K} \|\text{GIN}_k(\mathcal{G}) - c\|^{-1} + \mathcal{L}_{\text{GTL}}(\mathcal{G})], & y = -1 \\ \mathbb{E}_{\mathcal{G}}[\mathcal{L}_{\text{OCC}}(\mathcal{G}) + \mathcal{L}_{\text{GTL}}(\mathcal{G})], & \text{otherwise} \end{cases}.$$

Here, $y \in \{-1, 0, 1\}$ corresponds to labeled anomalies, unlabeled samples, and labeled normal samples, respectively. We apply semi-supervision exclusively to OCGTL, as extending the other models requires modifications to their architecture, e.g., adding a separate classification head, implementing one-class classification, or introducing a reconstruction-based loss term.

5 Experiments

To rigorously evaluate the various combinations of image-to-graph transformations and graph-based anomaly detection methods introduced in Section 4, we first construct two types of graphs on every lesion image, namely patch-based graphs, where each image is divided into a non-overlapping 4×5 grid of 20 patches, as well as superpixel graphs, with SLICO segmentation using $n = 20$. Each strategy is evaluated with and without virtual nodes and the provided ground truth segmentation masks.

For both segmentation strategies, we evaluate edge construction via Region Adjacency Graphs (connectivity = 2) and k-nearest-neighbor graphs based on spatial adjacency with and without color similarity ($k = 6$). Node features span the categories from Section 4.2, from basic color statistics to advanced texture and shape descriptors (see Appendix A for more details on the specifics of the image-to-graph transformations). All anomaly-detection models introduced in Section 4.3 are evaluated, each with a lightweight backbone of two GIN layers and hidden dimension = 16, to mitigate the well-known issue of over-smoothing [30] on these small graphs. Details regarding specific hyperparameters are provided in Appendix B.

5.1 Experimental Setup

We employ five-fold class-stratified cross-validation on the official HAM10000 training set with a fixed random seed. In each split, four folds ($\approx 80\%$) form the training set, and the remaining fold ($\approx 20\%$) is held out for testing. We frame anomaly detection as one-vs-rest: nevus is "normal", with other lesion types (melanoma, basal cell carcinoma, etc.) being "anomalous".

For the unsupervised experiments, we trained each model exclusively on nevus samples from the training folds, ignoring the anomalous samples. Evaluation then proceeded on the entire test fold. Motivated by literature indicating that even a small fraction of labeled anomalies can substantially improve performance [15], we also explored two supervised regimes applying the semi-supervised adaptation of OCGTL under identical conditions. For weak supervision, we retained random samples of anomalies (in addition to all nevus examples) from the training folds, s.t. the respective training folds comprised of 5% labeled anomalies. For full supervision, we included every labeled anomaly from the training folds alongside the nevus samples. Contrary to classification, we still do not differentiate between anomalous classes.

	Features	Patch			SLICO		
		RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}
Mask ✗ VN ✗	RGB_{avg}	70.9±2.4	71.8±2.2	66.5±2.3	66.8±3.4	64.4±2.0	64.9±3.7
	Color	**74.2±2.0**	72.5±1.8	70.9±2.0	73.0±1.8	71.8±2.0	72.3±1.4
	Texture	68.3±2.4	69.2±1.6	66.9±1.9	68.2±1.8	67.6±2.2	67.5±1.6
	Shape	—	—	—	59.6±1.2	55.1±1.4	55.9±1.7
	All	73.8±3.0	**75.0±2.3**	70.9±1.5	72.4±1.6	71.4±1.4	71.2±2.1
Mask ✓ VN ✓	RGB_{avg}	68.7±2.4	67.7±3.0	67.7±2.0	72.8±2.2	71.7±2.5	70.2±2.9
	Color	69.5±1.3	69.3±1.6	70.2±1.5	**80.6±1.6**	**77.5±1.9**	**78.6±1.7**
	Texture	59.4±1.6	59.7±3.0	61.3±2.0	66.5±2.0	63.6±2.1	64.6±1.6
	Shape	—	—	—	59.2±1.9	59.7±1.4	60.4±1.5
	All	69.5±1.8	71.3±2.3	**71.3±1.7**	76.9±1.7	75.4±1.8	76.6±1.3

Table 2: Mean AUC-ROC and SD values in % per image-to-graph transformation pipeline across all GLAD models, averaged over all splits. Best performance per column in bold, best performance per row underlined. For the features, 'All' corresponds to the complete feature set.

All models are trained for 20 epochs with Adam (learning rate = 0.001) and a batch size of 128. To counter the curse of dimensionality, we apply PCA to each feature set (except for RGB_{avg}) prior to training, retaining 95% of the variance; this reduces computational cost while preserving the informative signal.

5.2 Performance Comparison

In the following, we analyze how image-to-graph transformation pipelines, GLAD models, and supervision levels influence the image-level AUC-ROC performance (reported in %). Analogous plots and tables for AUC-PR, along with the exact per-configuration results, are provided in Appendix D. For patch-based segmentation, shape features are excluded, as they are not informative.

Table 2 summarizes performance by graph-construction pipeline, averaged over all GLAD models and data splits. In this table, combinations with Mask but without VN are omitted, as their performance was generally similar to or worse than the corresponding VN variant; detailed plots including these variants are provided in Appendix D. For SLICO, using both VN and Mask yields better performance, whereas for Patch, the configuration without either performs best. RAG is (bar one exception) the strongest edge-construction strategy with SLICO, with no clear winner for Patch-based segmentation. Node feature choice has a pronounced effect: Color alone is the strongest single feature set (appearing in all three top average configurations) and performs comparably to variants that additionally include Texture and Shape. However, the benefit of adding Texture and Shape depends strongly on supervision and the GLAD model. The complete

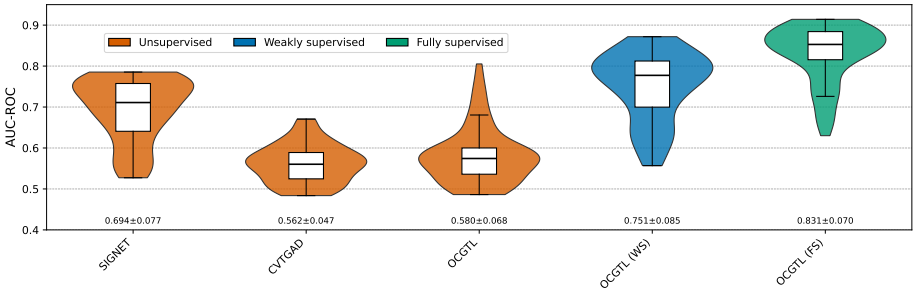
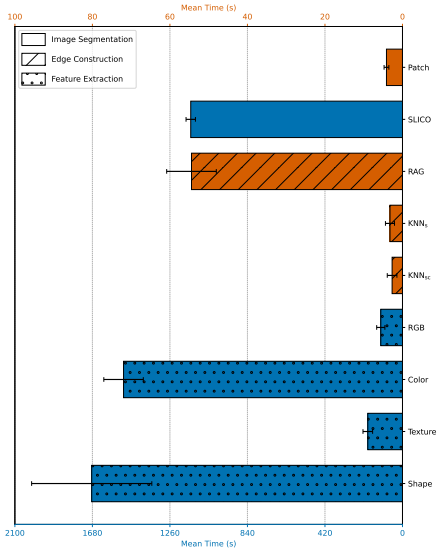


Fig. 4: AUC-ROC performance distribution including median, lower quartile, and upper quartile per GLAD model, as well as mean and SD, across all image-to-graph transformations.

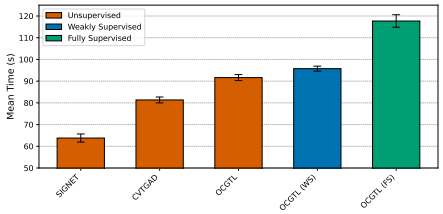
feature set substantially improves the best unsupervised method (SIGNET) and all supervised OCGTL variants in the absence of virtual nodes and segmentation masks.

Performance differences across GLAD models and supervision regimes are substantial. Figure 4 shows the distribution of AUC-ROC values across all image-to-graph pipelines and data splits. In the unsupervised setting, SIGNET achieves the strongest average performance, while the overall best result is obtained with OCGTL. With Mask and VN, SIGNET reaches up to 77.4 ± 1.4 and OCGTL 80.5 ± 1.9; without Mask and VN, the respective values are 72.0 ± 1.2 and 68.1 ± 2.2. In comparison, CVTGAD attains up to 66.8 ± 5.1 with Mask and VN, and 66.4 ± 3.8 without either. Notably, with the complete node feature set, SIGNET is relatively insensitive to the segmentation and edge-construction choices. Under weak supervision with only 5% anomalies in the training set, OCGTL improves to 87.2 ± 0.6, and in the fully supervised setting it reaches 91.4 ± 0.2. In general, the supervised variants benefit less from VN and Mask but more from the complete feature set.

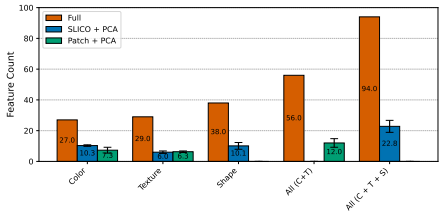
To contextualize these findings, a recent study [6] reports image-level AUC-ROC scores for a variety of image-based anomaly-detection methods employing ImageNet-pretrained feature extractor backbones on the ISIC2018 dataset [7] (with HAM10000 constituting its training set, and all images originating from the same source). The results are obtained under unified evaluation protocols and averaged over three independent runs per method. The authors adopt the same one-vs-all setup, using nevus as the normal class and the other six classes as anomalies, with training conducted on 6,705 normal images and testing on 1,512 images (909 normal, 603 anomalous). Here, the best reconstruction-based method configuration achieves an AUC-ROC of 79.2 ± 0.6, while the best self-supervised configuration reaches 80.7 ± 0.5. This places our best unsupervised GLAD configuration with Mask and VN (OCGTL with RAG+SLICO+Color, 80.5 ± 1.9) within the range of the strongest image-based baselines on a closely comparable task definition.



(a) Image-to-graph runtimes.



(b) GLAD model runtimes.



(c) Feature count reduction.

Fig. 5: (a) Total runtime (scaled 1:21) for each graph transformation on the dataset. (b) Average model runtimes per dataset split (train & inference) (c) Feature counts for feature sets and when reduced via PCA, across all runs.

5.3 Efficiency Analysis

In addition to anomaly detection performance, the runtime of both models and pre-processing steps is crucial for practical applicability. To contextualize these values, Appendix C provides details on the execution environment.

Figure 5a shows the average time required for each step of the image-to-graph transformation pipeline when converting the whole HAM10000 dataset to graphs. While the division into patches takes virtually no time, the segmentation via SLICO averages approx. 19 minutes. Edge creation, performed after computing spatial and color node features, is negligible for KNN and under one minute for RAG. Node feature extraction varies more substantially: average RGB and texture features require only 2–3 minutes, whereas general color and shape features take on average 25 and 28 minutes, respectively.

Strictly converting each image into a graph with an average of 20 nodes and up to 94 features (depending on the feature set) drastically reduces the feature dimensionality per image. Figure 5c provides an overview of the original number of features per feature set and the average reduced size after applying PCA to respective node segmentations included in the experiments. Notably, the performance reported in Section 5.2 was achieved with a maximum of just over 20 features per node. This compression corresponds to a feature dimensionality per graph representation of roughly 400 on average, compared to the original $600 \times 450 = 270{,}000$ pixels per image.

Beyond reducing memory requirements, the image-to-graph conversion also constrains model complexity and runtime. Hence, runtimes on these graph representations are modest: the average total runtime per dataset split (train + test) across all GLAD models ranges between 1 and approx. 2 minutes (Figure 5b).

5.4 Limitations

Due to the focus on HAM10000, our study is constrained by its focus on a single medical domain, where benign intra-class lesion variability can closely resemble genuine anomalies, and image artifacts (such as calibration markers) remain unmodeled. The competitiveness of these approaches on higher resolution images of the same domain, for example, the data set of the ISIC Challenge 2024 [17], also remains to be explored. Moreover, we have yet to directly compare our two-step graph-based pipeline with pixel-level anomaly detectors or fully end-to-end image-to-graph methods under a unified experimental setup, which would clarify their exact relative runtime efficiency and detection performance. Finally, we did not systematically explore alternative hyperparameter configurations, opting instead for sensible default values.

6 Conclusion and Outlook

In this work, we systematically evaluated the impact of graph-structured representations on image anomaly detection using the HAM10000 dermatoscopic image dataset. Our findings demonstrate that simple region-based graph abstractions, with drastically reduced feature dimensionality, not only significantly reduce runtime and overall dependence on pretrained models but can also achieve performance competitive to image-based models in specific domains. While highlighting the potential of graph-based approaches, several avenues remain to investigate further application potentials and enhance performance.

Image datasets with more heterogeneous and higher-resolution content could benefit from more sophisticated node segmentation and feature extraction approaches. This approach may involve applying off-the-shelf, pretrained segmentation networks coupled with lightweight, pretrained deep feature extractors on each node region. Furthermore, we see promise in exploring more sophisticated edge construction methods, along with the inclusion of edge features that capture richer relationships between nodes. For instance, incorporating geometric properties, such as distances or angles between nodes or higher-order relationships, could provide additional context that enhances anomaly detection performance.

Acknowledgments. This work has been partly supported by the Research Center Trustworthy Data Science and Security (https://rc-trust.ai), one of the Research Alliance centers within the UA Ruhr (https://uaruhr.de).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(11), 2274–2282 (2012). https://doi.org/10.1109/TPAMI.2012.120
2. Acharya, M., Roy, A., Koneripalli, K., Jha, S., Kanan, C., Divakaran, A.: Detecting out-of-context objects using graph contextual reasoning network. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence. pp. 629–635 (2022). https://doi.org/10.24963/ijcai.2022/89
3. Annaby, M.H., Elwer, A.M., Rushdi, M.A., Rasmy, M.E.M.: Melanoma Detection Using Spatial and Spectral Analysis on Superpixel Graphs. Journal of Digital Imaging **34**(1), 162–181 (2021). https://doi.org/10.1007/s10278-020-00401-6
4. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9584–9592 (2019). https://doi.org/10.1109/CVPR.2019.00982
5. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying density-based local outliers. ACM SIGMOD Record **29**(2), 93–104 (2000). https://doi.org/10.1145/335191.335388
6. Cai, Y., Zhang, W., Chen, H., Cheng, K.T.: MedIAnomaly: A comparative study of anomaly detection in medical images. Medical Image Analysis **102**, 103500 (2025). https://doi.org/10.1016/j.media.2025.103500
7. Codella, N., Rotemberg, V., Tschandl, P., Celebi, M.E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M., Kittler, H., Halpern, A.: Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). arXiv preprint arXiv:1902.03368 (2018), https://arxiv.org/abs/1902.03368
8. Cosma, R.A., Knobel, L., Van Der Linden, P., Knigge, D.M., Bekkers, E.J.: Geometric Superpixel Representations for Efficient Image Classification with Graph Neural Networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 109–118 (2023). https://doi.org/10.1109/ICCVW60793.2023.00018
9. Duong, H.T., Le, V.T., Hoang, V.T.: Deep Learning-Based Anomaly Detection in Video Surveillance: A Survey. Sensors **23**(11), 5024 (2023). https://doi.org/10.3390/s23115024
10. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision **59**(2), 167–181 (2004). https://doi.org/10.1023/B:VISI.0000022288.19776.77
11. Fey, M.: Convolutional Neural Networks auf Graphrepräsentationen von Bildern. Master's thesis, Technische Universität Dortmund (2017)
12. Friedman, R.J., Rigel, D.S., Kopf, A.W.: Early detection of malignant melanoma: the role of physician examination and self-examination of the skin. CA: A Cancer Journal for Clinicians **35**(3), 130–151 (1985). https://doi.org/10.3322/canjclin.35.3.130
13. Gu, Z., Zhu, B., Zhu, G., Chen, Y., Tang, M., Wang, J.: Univad: A training-free unified model for few-shot visual anomaly detection. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15194–15203 (2025)
14. Han, K., Wang, Y., Guo, J., Tang, Y., Wu, E.: Vision GNN: An Image is Worth Graph of Nodes. In: Advances in Neural Information Processing Systems 35 (2022)

15. Han, S., Hu, X., Huang, H., Jiang, M., Zhao, Y.: ADBench: Anomaly Detection Benchmark. In: Advances in Neural Information Processing Systems 35 (2022)
16. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics **SMC-3**(6), 610–621 (1973). https://doi.org/10.1109/TSMC.1973.4309314
17. Kurtansky, N.R., D'Alessandro, B.M., Gillis, M.C., Betz-Stablein, B., Cerminara, S.E., Garcia, R., Girundi, M.A., Goessinger, E.V., Gottfrois, P., Guitera, P., et al.: The slice-3d dataset: 400,000 skin lesion image crops extracted from 3d tbp for skin cancer detection. Scientific Data **11**(1), 884 (2024). https://doi.org/10.1038/s41597-024-03743-w
18. Li, J., Xing, Q., Wang, Q., Chang, Y.: CVTGAD: Simplified Transformer with Cross-View Attention for Unsupervised Graph-Level Anomaly Detection. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 185–200. Springer (2023). https://doi.org/10.1007/978-3-031-43412-9_11
19. Liu, Y., Ding, K., Lu, Q., Li, F., Zhang, L.Y., Pan, S.: Towards self-interpretable graph-level anomaly detection. Advances in Neural Information Processing Systems **36**, 8975–8987 (2023)
20. Luo, X., Wu, J., Yang, J., Xue, S., Peng, H., Zhou, C., Chen, H., Li, Z., Sheng, Q.Z.: Deep graph level anomaly detection with contrastive learning. Scientific Reports **12**(1), 19867 (2022). https://doi.org/10.1038/s41598-022-22086-3
21. Ma, R., Pang, G., Chen, L., van den Hengel, A.: Deep graph-level anomaly detection by glocal knowledge distillation. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. pp. 704–714 (2022). https://doi.org/10.1145/3488560.3498473
22. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q.Z., Xiong, H., Akoglu, L.: A Comprehensive Survey on Graph Anomaly Detection With Deep Learning. IEEE Transactions on Knowledge and Data Engineering **35**(12), 12012–12038 (2023). https://doi.org/10.1109/TKDE.2021.3118815
23. Niu, C., Pang, G., Chen, L.: Graph-level anomaly detection via hierarchical memory networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 201–218. Springer (2023). https://doi.org/10.1007/978-3-031-43412-9_12
24. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(7), 971–987 (2002). https://doi.org/10.1109/TPAMI.2002.1017623
25. Peng, Y., Lin, X., Ma, N., Du, J., Liu, C., Liu, C., Chen, Q.: Sam-lad: Segment anything model meets zero-shot logic anomaly detection. Knowledge-Based Systems **314**, 113176 (2025). https://doi.org/10.1016/J.KNOSYS.2025.113176
26. Qiao, H., Tong, H., An, B., King, I., Aggarwal, C., Pang, G.: Deep graph anomaly detection: A survey and new perspectives. IEEE Transactions on Knowledge and Data Engineering (2025). https://doi.org/10.1109/TKDE.2025.3581578
27. Qiu, C., Kloft, M., Mandt, S., Rudolph, M.: Raising the Bar in Graph-level Anomaly Detection. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence. pp. 2196–2203 (2022). https://doi.org/10.24963/IJCAI.2022/305
28. Rodrigues, J., Carbonera, J.: Graph convolutional networks for image classification: Comparing approaches for building graphs from images. In: Proceedings of the 26th International Conference on Enter-prise Information Systems. pp. 437–446 (2024). https://doi.org/10.5220/0012263200003690

29. Ruff, L., Vandermeulen, R.A., Görnitz, N., Binder, A., Müller, E., Müller, K., Kloft, M.: Deep semi-supervised anomaly detection. In: 8th International Conference on Learning Representations (2020), https://openreview.net/forum?id=HkgH0TEYwH
30. Rusch, T.K., Bronstein, M.M., Mishra, S.: A Survey on Oversmoothing in Graph Neural Networks (2023), https://arxiv.org/abs/2303.10993
31. Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C.: Support Vector Method for Novelty Detection. In: Advances in Neural Information Processing Systems 12. vol. 12 (1999)
32. Tschandl, P., Rosendahl, C., Kittler, H.: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific Data **5**(1), 180161 (2018). https://doi.org/10.1038/sdata.2018.161
33. Tu, B., Wang, Z., Ouyang, H., Yang, X., Li, J., Plaza, A.: Hyperspectral anomaly detection using the spectral–spatial graph. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–14 (2022). https://doi.org/10.1109/TGRS.2022.3217329
34. Tu, B., Yang, X., He, B., Chen, Y., Li, J., Plaza, A.: Anomaly detection in hyperspectral images using adaptive graph frequency location. IEEE Transactions on Neural Networks and Learning Systems (2024). https://doi.org/10.1109/TNNLS.2024.3449573
35. Vedaldi, A., Soatto, S.: Quick Shift and Kernel Methods for Mode Seeking. In: 10th European Conference on Computer Vision. Lecture Notes in Computer Science, vol. 5305, pp. 705–718. Springer (2008). https://doi.org/10.1007/978-3-540-88693-8_52
36. Wang, N., Shi, Y., Li, H., Zhang, G., Li, S., Liu, X.: Multi-prior graph autoencoder with ranking-based band selection for hyperspectral anomaly detection. Remote Sensing **15**(18), 4430 (2023). https://doi.org/10.3390/rs15184430
37. Wang, Y., Liu, Y., Shen, X., Li, C., Miao, R., Ding, K., Wang, Y., Pan, S., Wang, X.: Unifying unsupervised graph-level anomaly detection and out-of-distribution detection: A benchmark. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=g90RNzs8wX
38. Xie, G., Wang, J., Liu, J., Jin, Y., Zheng, F.: Pushing the limits of fewshot anomaly detection in industry vision: Graphcore. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=xzmqxHdZAwO
39. Zhao, L., Akoglu, L.: On using classification datasets to evaluate graph outlier detection: Peculiar observations and new insights. Big Data **11**(3), 151–180 (2023). https://doi.org/10.1089/BIG.2021.0069
40. Zoghlami, F., Bazazian, D., Masala, G., Gianni, M., Khan, A.: ViGLAD: Vision Graph Neural Networks for Logical Anomaly Detection. IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3502514

A Image-to-Graph Transformation Details

We applied two segmentation methods to each 450×600 HAM10000 image: dividing the image patch-based into a 4×5 grid (20 patches) and SLICO superpixel segmentation with $n = 20$ segments. We then extracted edges via two strategies: a region adjacency graph (RAG) with connectivity $= 2$ and a k-nearest-neighbor graph with $k = 6$ (either based strictly on spatial or spatial and color similarity). Given two centroid coordinates $(x_i, y_i), (x_j, y_j) \in [W] \times [H]$, and their mean RGB values $(r_i, g_i, b_i), (r_j, g_j, b_j) \in \{0, \ldots, 255\}^3$, the spatial distance is calculated as:

$$d_{spatial} = \|(x_i, y_i) - (x_j, y_j)\|,$$

and the spatial-color distance as:

$$\begin{aligned} d_{spatial-color} &= \sqrt{\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2} + \frac{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}{3}} \\ &= \sqrt{\sum_{k \in \{x,y\}} \left(\frac{k_i - k_j}{\sqrt{2}}\right)^2 + \sum_{k \in \{r,g,b\}} \left(\frac{k_i - k_j}{\sqrt{3}}\right)^2} \\ &= \left\| \left(\frac{x_i}{\sqrt{2}}, \frac{y_i}{\sqrt{2}}, \frac{r_i}{\sqrt{3}}, \frac{g_i}{\sqrt{3}}, \frac{b_i}{\sqrt{3}}\right) - \left(\frac{x_j}{\sqrt{2}}, \frac{y_j}{\sqrt{2}}, \frac{r_j}{\sqrt{3}}, \frac{g_j}{\sqrt{3}}, \frac{b_j}{\sqrt{3}}\right) \right\| \end{aligned}$$

The full dataset yielded roughly the same average node and edge counts, as seen in Table 3.

	Mask	$\lvert\mathcal{V}\rvert_{\text{avg}}$	$\lvert\mathcal{V}\rvert_{\min}$	$\lvert\mathcal{V}\rvert_{\max}$	$\lvert\mathcal{E}\rvert_{\text{avg}}^{\text{rag}}$	$\lvert\mathcal{E}\rvert_{\text{avg}}^{\text{knn}_s}$	$\lvert\mathcal{E}\rvert_{\text{avg}}^{\text{knn}_{sc}}$
Patch	✗	20.00	20	20	86.00	144.00	139.75
	✓	20.00	20	20	86.00	144.00	140.37
SLICO	✗	19.97	16	20	86.00	143.66	142.02
	✓	19.99	17	21	88.24	142.22	140.49

Table 3: Graph Metrics for the transformed HAM10000 dataset.

B GLAD Model Parameterization

All models were trained using the same learning rate, number of epochs, batch size, and optimizer settings as detailed in Section 5.1. We uniformly set the hidden dimensionality to 16 and employed two GIN message-passing layers across all architectures. For OCGTL, we additionally evaluated two semi-supervised variants: one in which anomalies comprised 5% of the overall training set and another that included all available anomalous samples from the training split.

Model-specific configurations were chosen to highlight each method's default configuration biases. CVTGAD employs a global mean-pooling readout and maintains a 16-dimensional embedding in its feature-view encoder, while its structure-view encoder uses a 32-dimensional hidden space. SIGNET, by contrast, uses its default sum-pooling aggregation. Finally, OCGTL stabilizes its one-class objective via an ensemble of one reference feature extractor and five additional feature extractors.

C Execution Environment

All experiments were performed on Ubuntu 22.04.3 LTS running on an Intel® Xeon® W9-3495X processor (48 cores, 96 threads; 3.4 GHz base, 4.5 GHz turbo) and a single NVIDIA RTX™ 6000 GPU (Ada; 48 GB GDDR6 ECC; CUDA 12.0). The timings for image-to-graph transformations and GLAD models were obtained using single-core execution and single GPU utilization to ensure a fair comparison.

D Additional Performance Comparison

To complement the main results, Figures 6 and 7 provide an overview of the AUC-ROC performance across different image-to-graph transformation pipelines for the different GLAD models and supervision regimes, respectively. These figures visualize the relative impact of segmentation method, feature set, edge construction, and the use of segmentation masks and virtual nodes. In addition to the AUC-ROC analysis in the main paper, Table 4 and Figure 8 provide the corresponding AUC-PR versions of their counterparts in Section 5.2. Moreover, Tables 5 and 6 present the complete AUC-ROC results, while Tables 7 and 8 list the corresponding AUC-PR values for each individual experiment.

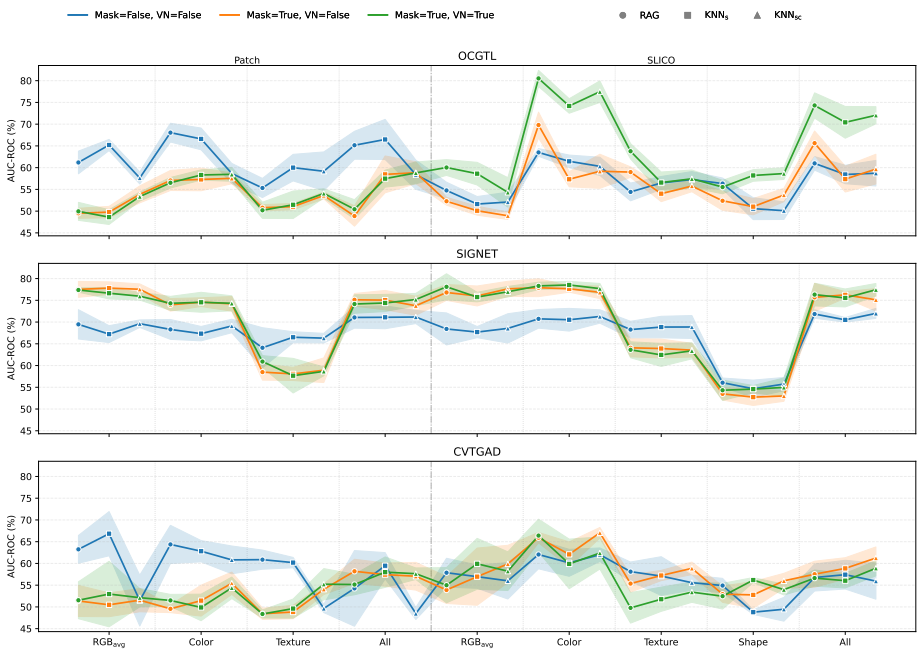


Fig. 6: AUC-ROC across image-to-graph pipelines for the three unsupervised GLAD models (OCGTL, SIGNET, CVTGAD). Each row shows one model with an identical x-layout (segmentation via Patch/SLICO, grouped by features). Lines encode the use of segmentation masks (Mask) and virtual nodes (VN); markers encode edge construction (RAG, $\mathrm{KNN_s}$, $\mathrm{KNN_{sc}}$); shaded bands indicate $\pm 1\sigma$ over splits.

Fig. 7: AUC-ROC across image-to-graph pipelines for OCGTL under three supervision regimes (Unsupervised, Weakly Supervised, Fully Supervised). The visualization mirrors Fig. 6: rows correspond to supervision levels; lines encode the use of segmentation masks (Mask) and virtual nodes (VN); markers encode edge construction; shaded bands show $\pm 1\sigma$.

	Features	PATCH			SLICO		
		RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}
Mask ✗ VN ✗	RGB_{avg}	52.1±2.7	52.3±2.2	47.7±2.5	48.5±3.6	46.7±2.7	46.5±3.3
	Color	58.1±2.2	54.6±1.9	53.7±2.2	56.2±2.2	54.3±2.8	55.7±1.8
	Texture	50.9±2.6	51.5±1.9	50.2±2.3	52.1±2.0	51.6±3.0	51.0±1.8
	Shape	—	—	—	41.7±1.3	38.0±1.1	38.7±1.7
	All	**59.0±2.3**	**59.3±2.4**	56.6±2.0	56.9±1.7	55.9±1.8	56.1±2.0
Mask ✓ VN ✓	RGB_{avg}	53.6±2.2	53.2±2.3	52.2±2.6	59.1±2.4	56.5±3.0	54.2±3.2
	Color	55.6±1.7	55.3±2.2	55.6±1.5	**66.9±2.5**	**63.1±2.3**	**64.1±2.4**
	Texture	42.7±2.1	42.5±2.9	43.6±2.1	51.0±2.3	46.7±2.2	47.3±1.8
	Shape	—	—	—	41.3±2.0	41.5±1.5	41.9±1.7
	All	55.7±1.9	56.9±2.7	**57.3±1.9**	63.6±1.8	61.2±2.3	62.0±1.8

Table 4: Mean AUC-PR and SD values in % per image-to-graph transformation pipeline across all GLAD models, averaged over all splits. Best performance per column in bold, best performance per row underlined. For the features, 'All' corresponds to the complete feature set.

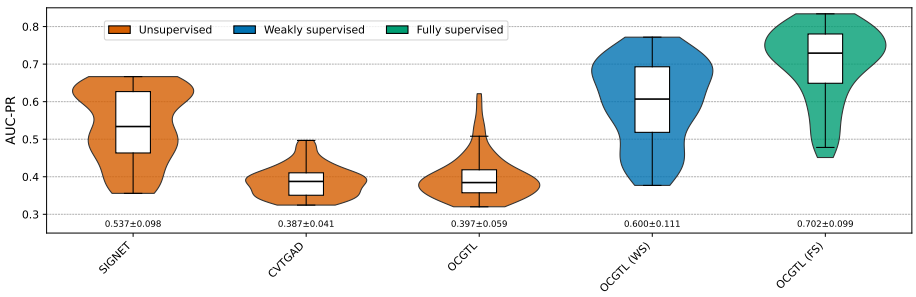


Fig. 8: AUC-PR performance distribution including median, lower quartile, and upper quartile per GLAD model, as well as mean and SD, across all image-to-graph transformations.

	Features	SIGNET						CVTGAD						OCGTL					
		Patch			SLICO			Patch			SLICO			Patch			SLICO		
		RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}
Mask ✗ VN ✗	RGB_{avg}	69.5±3.4	67.2±1.9	69.6±0.9	68.4±3.7	67.7±1.3	68.5±3.4	63.2±3.2	**66.8±5.1**	51.5±5.9	57.9±3.3	57.0±2.7	56.0±4.2	61.2±2.6	65.2±1.3	57.6±1.4	54.8±1.9	51.6±1.7	52.1±2.3
	Color	68.3±2.3	67.3±1.7	69.1±1.6	70.7±2.1	70.5±2.6	71.3±1.5	**64.4±4.4**	62.8±2.4	**60.8±3.2**	62.1±3.3	**60.1±3.1**	62.0±1.6	**68.1±2.2**	**66.6±2.5**	58.7±2.3	63.5±0.9	61.5±1.4	60.3±2.1
	Texture	64.1±4.6	66.5±1.3	66.3±1.1	68.3±1.9	68.9±2.5	68.9±2.6	60.9±2.2	60.2±1.2	49.6±1.0	58.1±2.2	57.1±4.5	55.6±1.7	55.3±2.2	60.0±3.1	**59.2±4.5**	54.4±2.0	56.5±1.5	57.3±1.8
	Shape	—	—	—	56.1±1.1	54.7±2.0	55.7±1.5	—	—	—	54.9±1.7	48.8±0.5	49.5±2.7	—	—	—	56.3±1.3	50.6±2.5	50.1±2.0
	All	71.1±2.5	71.1±2.6	71.2±1.5	71.8±1.0	70.5±0.5	72.0±1.2	54.3±8.7	59.5±3.0	48.5±1.4	56.8±3.1	57.4±3.3	55.9±4.2	65.1±3.2	66.5±4.6	58.4±3.4	61.0±1.6	58.4±2.1	58.7±2.9
Mask ✓ VN ✓	RGB_{avg}	**77.4±0.5**	**76.6±1.3**	**75.9±0.9**	78.1±3.0	75.7±1.1	77.0±1.2	51.5±4.2	53.0±7.5	52.1±2.1	55.0±4.1	59.9±5.9	58.2±4.4	49.9±2.1	48.6±1.7	53.3±1.0	60.0±1.8	58.6±2.6	54.3±3.4
	Color	74.3±1.6	74.5±2.3	74.3±1.7	**78.3±0.8**	**78.5±0.8**	**77.7±1.2**	51.5±1.4	49.9±3.1	54.4±2.5	**66.4±3.8**	59.9±5.7	**62.4±3.7**	56.5±1.2	58.3±1.2	58.5±1.6	**80.5±1.9**	**74.2±1.7**	**77.5±2.5**
	Texture	60.9±1.4	57.7±4.0	58.7±1.0	63.6±1.8	62.4±2.7	63.4±1.8	48.4±0.9	49.6±2.2	55.2±3.6	49.8±3.5	51.8±2.8	53.4±2.3	50.2±1.9	51.5±3.2	54.1±1.1	63.8±1.8	56.6±2.4	57.4±2.0
	Shape	—	—	—	54.3±2.3	54.6±0.8	55.0±2.2	—	—	—	52.5±3.0	56.2±0.6	54.0±1.2	—	—	—	55.5±1.5	58.2±1.3	58.6±1.3
	All	74.1±2.1	74.4±2.0	75.2±1.4	76.3±2.6	75.5±2.1	77.4±1.4	55.2±2.4	58.0±3.5	57.6±1.3	56.6±1.9	56.0±1.7	58.9±1.8	50.4±2.2	57.5±2.0	58.8±2.5	74.3±2.9	70.4±3.7	72.0±2.0

Table 5: Unsupervised mean ROC-AUC and SD values in % on normal class "nv" by GLAD method, averaged over all splits.

	Features	OCGTL (Weakly Supervised)						OCGTL (Fully Supervised)					
		Patch			SLICO			Patch			SLICO		
		RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}
Mask ✗ VN ✗	RGB_{avg}	77.7±1.7	77.1±2.1	71.5±2.5	70.0±7.0	64.3±2.9	67.1±7.1	83.1±0.9	82.4±0.7	82.2±0.9	83.2±1.1	81.6±1.3	80.9±1.6
	Color	81.6±0.6	77.7±2.0	77.7±1.9	79.9±2.2	79.0±1.8	80.2±1.2	88.9±0.7	88.1±0.6	88.1±1.0	88.7±0.7	87.7±0.9	87.9±0.7
	Texture	77.4±1.9	75.7±1.8	75.9±1.5	75.5±2.3	73.0±2.0	73.3±1.2	83.9±0.8	83.3±0.8	83.5±1.3	84.6±0.7	82.8±0.7	82.6±0.7
	Shape	—	—	—	61.0±1.6	55.7±1.1	56.9±1.4	—	—	—	69.9±0.4	65.9±1.1	67.4±0.7
	All	**87.2±0.6**	**87.0±0.8**	**85.9±0.8**	82.5±1.9	81.3±0.8	79.9±1.5	**91.4±0.2**	**90.7±0.4**	**90.7±0.4**	90.1±0.5	89.3±0.2	89.6±0.5
Mask ✓ VN ✓	RGB_{avg}	78.7±4.1	74.7±3.4	71.8±4.8	82.3±1.3	78.8±2.4	74.8±5.1	85.9±0.9	85.6±0.9	85.3±1.1	88.4±0.6	85.3±0.6	86.8±0.4
	Color	80.3±1.6	79.3±0.8	79.2±0.9	**86.8±1.0**	85.1±0.8	**85.0±0.8**	85.0±0.9	84.7±0.6	84.8±0.7	**90.7±0.3**	**89.7±0.3**	**90.1±0.2**
	Texture	63.6±2.7	65.1±4.3	64.3±2.9	73.8±2.2	69.7±1.6	70.8±1.2	73.9±1.1	74.5±1.4	74.5±1.3	81.6±0.6	77.3±1.1	78.2±0.7
	Shape	—	—	—	62.6±1.8	60.4±3.2	61.9±1.4	—	—	—	71.1±1.0	69.3±1.1	72.6±1.3
	All	81.7±1.4	81.2±3.0	79.2±2.3	86.6±1.0	**85.6±1.1**	84.9±1.2	85.9±0.8	85.4±0.9	85.5±0.8	90.6±0.2	89.5±0.5	89.9±0.4

Table 6: Weakly and fully supervised (5% and 33.1% labeled anomalies) mean ROC-AUC and SD values in % on normal class "nv" by GLAD method, averaged over all splits.

	Features	SIGNET						CVTGAD						OCGTL					
		Patch			SLICO			Patch			SLICO			Patch			SLICO		
		RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}
Mask ✗ VN ✗	RGB_{avg}	49.7±3.5	47.7±0.9	49.5±1.6	48.8±4.0	49.4±3.3	49.1±2.9	44.6±3.4	**47.1±3.8**	34.7±4.6	39.0±2.8	40.0±2.8	38.8±4.7	42.6±2.8	45.9±1.8	37.7±0.9	37.7±1.7	35.7±1.7	36.4±1.4
	Color	48.8±3.1	47.2±1.8	50.0±2.3	52.1±2.8	52.1±3.9	53.9±2.4	**48.3±3.4**	45.4±2.6	**43.3±3.5**	43.6±3.1	42.3±3.6	**44.8±2.4**	**47.4±2.7**	44.3±1.7	38.7±0.8	43.1±1.1	40.5±0.9	39.5±1.6
	Texture	46.2±4.7	49.0±1.3	48.5±1.8	51.9±2.2	53.4±3.9	52.0±3.8	40.4±1.7	40.4±1.2	34.6±0.9	37.9±2.1	40.2±4.8	37.9±1.5	36.8±2.2	40.7±3.0	40.0±3.5	37.2±2.3	37.9±1.8	38.2±1.0
	Shape	—	—	—	38.1±1.5	37.1±1.2	37.4±2.2	—	—	—	36.2±1.4	32.6±0.4	32.7±1.6	—	—	—	39.0±1.0	34.5±2.0	34.7±1.7
	All	53.2±2.9	52.9±3.7	53.5±2.8	54.7±1.2	53.3±0.5	54.9±1.1	37.2±4.3	40.6±1.3	34.0±1.4	37.1±2.2	39.8±3.3	39.2±3.6	45.3±2.2	**46.0±4.4**	39.8±3.3	42.5±2.1	40.1±2.0	40.2±2.4
Mask ✓ VN ✓	RGB_{avg}	**65.7±1.7**	**65.5±2.1**	**64.7±2.1**	65.4±3.1	63.0±2.8	63.0±2.5	33.7±2.1	38.0±4.1	34.8±2.3	39.6±4.3	42.9±5.3	39.4±3.4	32.9±1.2	32.0±0.6	34.6±1.3	41.5±1.7	41.9±3.1	37.4±3.1
	Color	63.2±2.8	62.4±3.2	61.5±2.3	**66.2±2.7**	**66.5±2.2**	**66.4±1.4**	34.9±1.8	33.4±3.3	36.2±2.0	**48.4±4.1**	**43.6±3.9**	42.0±4.3	36.4±1.0	38.4±1.7	38.4±0.9	**62.1±4.1**	**53.3±3.1**	**58.7±4.3**
	Texture	43.0±1.9	39.7±3.6	40.6±2.0	45.2±2.1	44.0±3.2	45.3±2.5	32.7±1.3	33.1±1.9	37.5±2.5	35.1±2.5	35.0±2.0	35.3±1.5	33.0±1.4	34.9±2.2	35.7±0.5	45.0±2.6	38.9±2.4	39.7±1.3
	Shape	—	—	—	37.1±2.2	36.9±0.6	37.3±1.9	—	—	—	35.1±2.4	37.4±0.7	35.3±1.8	—	—	—	38.6±1.7	40.9±1.7	40.8±1.6
	All	61.0±2.5	59.7±3.3	60.6±1.9	64.2±2.6	62.0±3.5	64.6±2.9	38.6±2.9	41.0±3.7	39.4±1.7	40.3±1.2	38.0±2.0	39.0±1.4	33.3±1.7	41.0±2.6	**42.8±3.4**	54.6±3.9	50.8±4.0	51.8±2.4

Table 7: Unsupervised mean PR-AUC and SD values in % on normal class "nv" by GLAD method, averaged over all splits.

	Features	OCGTL (Weakly Supervised)						OCGTL (Fully Supervised)					
		Patch			SLICO			Patch			SLICO		
		RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}	RAG	KNN_s	KNN_{sc}
Mask ✗ VN ✗	RGB_{avg}	56.1±2.8	55.9±2.9	51.8±3.4	49.4±6.9	44.7±2.8	45.8±5.2	67.5±1.2	65.0±1.5	64.9±1.8	67.6±2.3	63.5±2.6	62.5±2.1
	Color	66.9±1.0	59.5±2.4	59.5±2.7	64.1±3.4	60.7±3.9	63.6±1.8	78.9±0.9	76.9±1.1	77.0±1.7	78.0±0.7	76.1±1.6	76.6±0.8
	Texture	62.1±2.3	59.6±2.0	59.2±1.8	61.3±2.0	57.1±3.2	57.6±1.3	68.9±2.0	67.9±2.1	68.6±2.1	72.0±1.6	69.2±1.3	69.4±1.4
	Shape	—	—	—	43.3±1.9	38.0±0.8	38.9±1.8	—	—	—	51.9±0.7	47.8±1.0	49.7±1.1
	All	**75.7±1.2**	**75.3±1.8**	**73.8±1.8**	69.3±2.2	67.2±2.4	66.7±2.5	**83.4±1.0**	**81.6±1.0**	**81.7±0.8**	80.7±0.7	79.0±0.8	79.6±0.4
Mask ✓ VN ✓	RGB_{avg}	63.5±4.1	58.3±3.5	55.3±4.6	69.7±1.8	62.9±2.2	56.8±6.3	72.4±2.1	72.3±1.4	71.8±2.6	79.2±1.0	71.7±1.4	74.5±0.6
	Color	69.3±1.5	68.5±1.3	67.9±1.3	75.3±1.2	72.3±1.9	72.7±1.4	73.9±1.4	73.5±1.3	74.0±1.2	**82.4±0.5**	79.6±0.6	**80.9±0.7**
	Texture	46.9±3.7	46.6±4.3	45.7±2.9	58.8±3.1	52.7±1.6	53.2±1.6	57.9±2.1	58.1±2.6	58.4±2.5	70.8±1.2	62.9±2.0	63.0±2.2
	Shape	—	—	—	43.4±2.2	42.2±2.8	42.8±1.1	—	—	—	52.4±1.7	49.9±1.4	53.5±2.0
	All	71.0±1.2	69.7±2.7	69.2±1.5	**77.2±1.0**	**75.0±1.2**	**73.7±1.7**	74.4±1.1	73.2±1.4	74.3±0.9	81.9±0.4	**80.1±0.8**	80.8±0.8

Table 8: Weakly and fully supervised (5% and 33.1% labeled anomalies) mean PR-AUC and SD values in % on normal class "nv" by GLAD method, averaged over all splits.