ISSN:2958-0420 Contents lists available at JJOAS-N

Jordan Journal of Applied Science - Natural Science Series
Applied Science Private University

Research Article *Vol. 20*

Can Graph Reasoning Improve Contextual Understanding in Histopathology?

Mohammad Khaleel Sallam Ma'aitah[1*], Abdulkader Helwan[2], Esraa Mahmoud Sariera[3].

[1]Electrical Engineering / Robotics and Artificial Intelligence Engineering, Faculty of Engineering and Technology, Applied Science Private University, Amman Jordan.

[2]Lebanese American University, Lebanon.

[3]Information Technology / Data Science and Artificial Intelligence, Faculty of Information Technology, Isra University, Amman Jordan.

ARTICLE INFO

Article history:

Received 15 Jul 2026

Accepted 06 Oct 2026

Published 01 Nov 2026

DOI:https://doi.org/10.35192/jjoas-n.v20i1.206

***Corresponding author:**

[1]Electrical Engineering / Robotics and Artificial Intelligence Engineering, Faculty of Engineering and Technology, Applied Science Private University, Amman Jordan.

Email: m_almaayta@asu.edu.jo

ABSTRACT

Background: Automated classification of histopathology tissue types is a fundamental task in computational pathology, yet standard deep learning architectures process image patches independently, potentially neglecting relational structure between tissue regions.

Objectives: In this work, we rigorously evaluate whether augmenting a ResNet-18 backbone with a Graph Convolutional Network (GCN) that performs message passing over a feature-space k-nearest-neighbor graph, built from patch feature similarity rather than true spatial coordinates, improves patch-level tissue classification.

Methods: In an earlier version of this evaluation, we found that near-perfect validation accuracy was in fact an artifact of label leakage: the graph was constructed using ground-truth labels at both training and evaluation time, which cannot be reproduced at inference. We correct this throughout, using a label-free graph construction, an independent external test set (CRC-VAL-HE-7K, a disjoint patient cohort), 5-fold cross-validation, and two additional models: a Graph Attention Network (GAT) variant and a non-graph attention-pooling baseline. Because cross-validation folds share overlapping training data and a common fixed test set, we additionally report a corrected resampled significance test that accounts for this non-independence, rather than relying solely on a naive paired *t*-test.

Results: Under this corrected protocol on NCT-CRC-HE-100K, none of the three graph-based variants differs significantly from the backbone-only model (95.48% ± 0.36% accuracy) on either accuracy or macro-F1 (corrected $p \geq 0.25$ throughout), though all three point estimates are at or below the backbone. On a second, independent public dataset (Kather-2016), the GCN variant shows a small but statistically significant improvement that survives the corrected test (+0.48 points accuracy, corrected p = 0.0034), while GAT and the attention-pooling baseline show no effect, an inconsistent, dataset-dependent pattern rather than a reliable general benefit. We also directly quantify the leakage artifact across all 5 folds on Kather-2016, reproducing the original protocol to show it inflates accuracy by roughly 5-6 percentage points over the corrected result, consistently across folds.

Conclusion: our results indicate that, once correctly evaluated, this form of feature-space graph-based relational reasoning provides at best a small, dataset-dependent benefit, not the large, consistent improvement suggested by the original, leaky evaluation, and we report the leakage mechanism and evaluation pitfalls in detail as a cautionary, reproducible example for related work.

Keywords: histopathology tissue classification; graph-enhanced reasoning network; graph convolutional network; graph attention network; label leakage; cross-validation.

Introduction

Histopathological examination remains the gold standard for cancer diagnosis and treatment planning, guiding critical clinical decisions from surgical resection to adjuvant therapy [1]. Pathologists assess not only cellular morphology but also the spatial arrangement of tissue structures, including glandular formation, tumor-infiltrating lymphocytes, and stromal interactions [2]. However, manual assessment is time-consuming and subject to inter-observer variability, motivating the development of automated deep learning systems for computational pathology [3], [4]. While recent advances in convolutional neural networks (CNNs) and Vision Transformers (ViTs) have improved patch-level classification accuracy, these methods typically process histopathology images as independent patches or fixed grid elements, potentially overlooking the long-range spatial dependencies and topological relationships

Published by Jordan Journal of Applied Science - Natural Science Series — This is an open access article under the CC BY 4.0 license.

that pathologists rely on for diagnostic reasoning [5].

This limitation raises a fundamental question in computational pathology: can explicit graph-based reasoning over inter-patch relationships improve model performance, even without access to true spatial or biological tissue topology? In natural scene understanding and molecular property prediction, graph neural networks (GNNs) have demonstrated substantial benefits for tasks requiring relational inference [6], [7]. However, in medical vision generally and histopathology specifically, the efficacy of graph-based reasoning remains underexplored. Standard CNNs and ViTs aggregate local or global context through fixed receptive fields or self-attention mechanisms, yet these operations occur simultaneously rather than through structured message passing over explicit relationships. For tasks like tissue type classification, where morphologically similar classes (e.g., stroma vs. smooth muscle) share local texture features but differ in context, a patch-independent approach may fail to resolve ambiguity. We hypothesize that augmenting visual encoders with a graph reasoning module allows the model to leverage inter-patch *feature-space* relationships, which are not the same as, and are not guaranteed to recover, the true spatial adjacency that a pathologist uses when visually comparing neighboring regions of a slide; we return to this distinction throughout the paper, since neither dataset used here provides the spatial coordinates that would be needed to construct a graph of genuine tissue neighbors.

To investigate this hypothesis, we utilize tissue type classification on the NCT-CRC-HE-100K dataset as a rigorous case study. This dataset presents several challenges relevant to our inquiry: nine tissue classes with varying degrees of visual similarity, potential class imbalance, and the absence of whole-slide spatial context (patches are extracted independently). It is important to clarify that while improving tissue classification performance is a valuable technical goal, our primary scientific objective is to evaluate whether graph-based reasoning provides a measurable advantage over standard backbone architectures in this demanding setting. The NCT-CRC-HE-100K dataset serves as a representative benchmark for histopathology analysis, allowing us to isolate the impact of graph reasoning on model behavior, particularly for morphologically ambiguous tissue types.

Deep Learning for Histopathology Classification

Deep learning has transformed computational pathology, with CNNs achieving pathologist-level performance on various diagnostic tasks [8], [9]. Early approaches employed patch-level classification with majority voting for slide-level prediction [10], while more recent methods leverage Multiple Instance Learning (MIL) to handle weak slide-level labels [11]. Vision Transformers have further improved performance by capturing long-range dependencies through self-attention [12], with growing adoption across medical image analysis tasks, including histopathology [13], [14]. However, these methods fundamentally treat patches as independent units or tokens in a sequence, neglecting the explicit topological structure that characterizes tissue architecture. Several studies have noted that classification errors often occur for morphologically similar tissue types, suggesting that local appearance features alone are insufficient for resolving all ambiguities [15], and that class imbalance further complicates evaluation and error analysis in medical image classification [16].

Graph Neural Networks in Medical Imaging

Graph neural networks have emerged as a powerful framework for modeling relational structure in medical data. In radiology, GNNs have been applied to model relationships between anatomical structures and to support disease diagnosis across a range of imaging tasks [17]. In computational pathology, graph-based approaches have primarily focused on cell- and patch-level graphs constructed from segmentation masks or spatial coordinates, where nodes represent tissue entities and edges encode spatial or morphological proximity [2], [5]. These methods have shown promise for survival prediction and tumor microenvironment characterization, where cell-cell interactions are biologically meaningful [18]. However, cell-graph construction typically requires accurate nuclei segmentation or whole-slide spatial coordinates, which introduces additional complexity and potential error propagation. Our approach operates at the patch level using only feature-space similarity, avoiding the need for fine-grained segmentation or spatial metadata.

Graph-Based Methods for Histopathology

Several recent works have explored graph-based representations for histopathology analysis. Chen et al. [18] modeled whole-slide images as spatially-resolved graphs of patch features for context-aware survival prediction, and Bazargani et al. [19] introduced a multi-scale relational graph convolutional network that combines multiple-instance learning with graph-structured message passing across magnifications for slide-level grading. Graham et al. [20] instead used a dilated convolutional network (not a graph) for gland instance segmentation in colon histology, illustrating the broader family of structure-aware architectures used in computational pathology. Unlike these approaches, which construct graphs from spatial coordinates or segmentation masks derived from whole-slide images, our feature-space graph construction requires neither, enabling relational reasoning on datasets, such as the one used here, where patch coordinates and whole-slide context are unavailable. This flexibility comes with an important limitation that we return to in Section 3.11: without true spatial or biological structure to anchor them, feature-space edges only approximate meaningful tissue relationships, and, as we show, are not guaranteed to improve on a strong backbone. We employ a two-phase training protocol that separates visual feature learning from graph reasoning optimization, improving training stability compared to end-to-end approaches.

Reasoning Mechanisms in Medical AI

Beyond graph-based methods, several studies have explored explicit, multi-step reasoning mechanisms for medical image analysis. Chain-of-thought prompting, originally proposed for large language models [21], has motivated a growing body of work on step-wise reasoning in medical vision-language tasks. Neuro-symbolic approaches combine neural perception with symbolic reasoning rules, particularly for tasks with well-defined clinical guidelines [22]. These works share a common motivation: medical decision-making often involves multi-step inference rather than direct pattern recognition. Our graph reasoning module aligns with

this perspective, modeling tissue classification as a process of contextual evidence aggregation rather than independent patch labeling. Unlike iterative methods that refine predictions across multiple passes, our approach refines representations over a graph structure in a single forward pass, capturing feature-space relationships between patches.

Contributions

In this work, we rigorously evaluate a Graph-Enhanced Reasoning Network (GRN) that augments a standard visual backbone with explicit relational modeling for histopathology classification. Our contributions are fourfold. First, we formulate tissue classification as a graph reasoning problem where patches serve as nodes and feature similarity defines edges, enabling explicit modeling of inter-patch relationships without requiring spatial coordinates, and we identify and correct a label-leakage failure mode in this formulation in which ground-truth labels used to construct the impossible to reproduce at inference. Second, we introduce a two-phase training protocol that first establishes robust visual features and subsequently optimizes the graph reasoning module, improving training stability and convergence. Third, we provide a rigorous empirical evaluation, consisting of an independent external test set, 5-fold cross-validation with both naive and non-independence-corrected significance testing, and two additional model variants (GAT, non-graph attention-pooling), on two independent public datasets (NCT-CRC-HE-100K and Kather-2016), showing that once evaluated correctly, graph-based reasoning of this form provides no statistically defensible benefit on our primary benchmark and, at best, a small, dataset-dependent benefit on a second one, not graph make the module's benefit appear far larger than it is and suggested. Fourth, we report the magnitude of the leakage artifact directly, so that the mechanism and its consequences are transparent and reproducible for other work constructing patch-level graphs from feature similarity or label-based grouping. Through this study, our aim is to contribute an honest, rigorously validated data point to the broader discourse on how graph-based architectures can be tailored to improve reliability in computational pathology systems, and to caution against the specific evaluation pitfall that made our own earlier results look far more positive than they were. The overall architecture is illustrated in Figure 1.

The pipeline processes input tissue patches through three stages: (1) a ResNet-18 backbone pretrained on ImageNet extracts patch-level feature vectors; (2) a graph construction module connects patches within the same mini-batch via k-nearest neighbors in feature space, using no label information at either training or inference time; (3) a Graph Convolutional Network (GCN) performs message passing, with a residual connection to the original backbone feature, to refine each patch's representation with information from its batch-local neighbors. Training follows a two-phase protocol: Phase 1 (dashed purple path) fine-tunes the backbone with a standard classifier; Phase 2 (solid green path) freezes the backbone and trains the graph reasoning module. The reasoning process the large, consistent improvement the original, leaky evaluation involves retrieving similar patches within the current batch, aggregating neighbor features, and updating each patch's own representation with contextual evidence. Predictions are made independently for every patch (node) by a shared linear classifier applied to its refined embedding; there is no pooling step and no single per-batch or per-graph prediction.

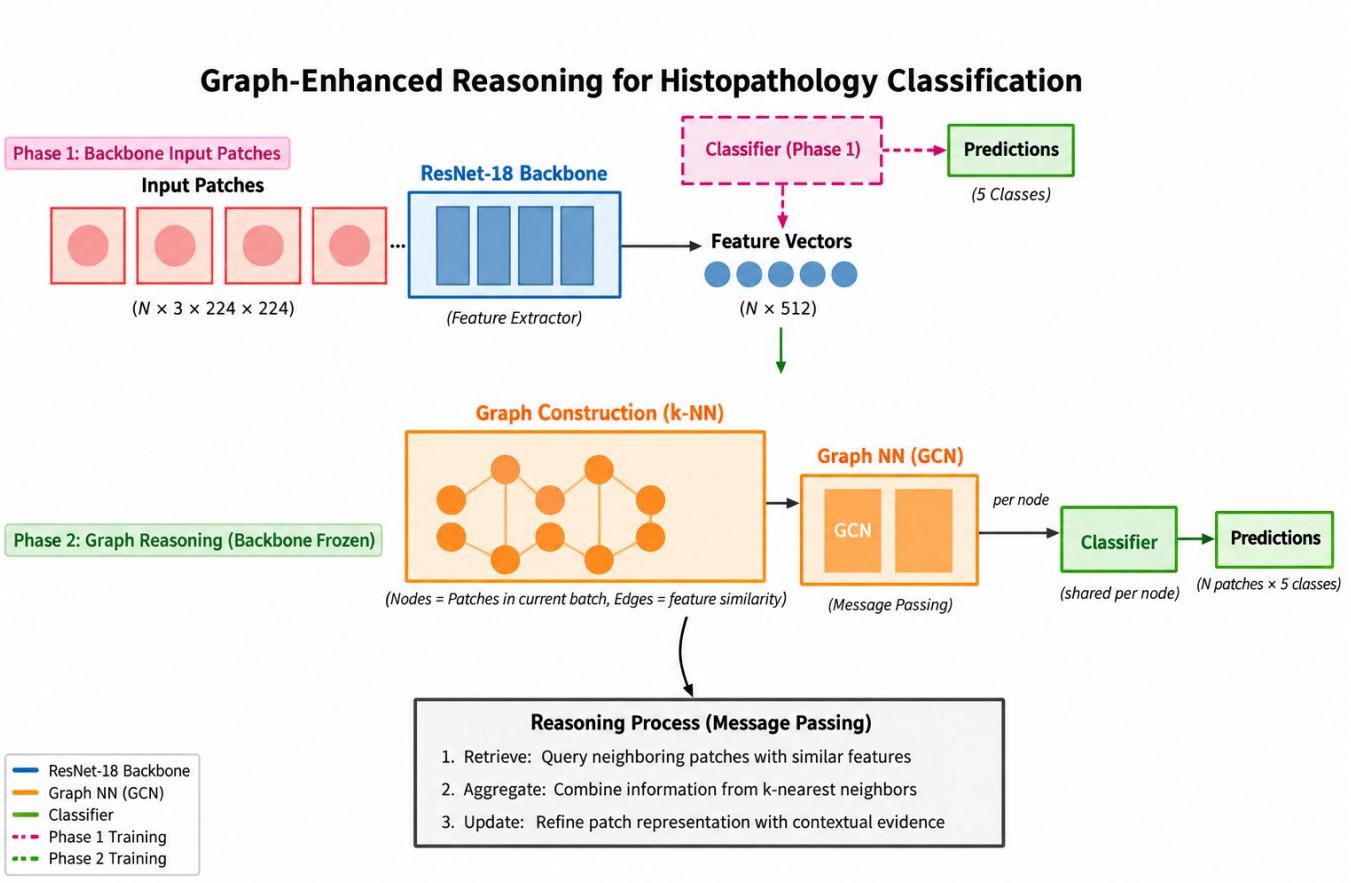


Figure 1. Architecture overview of the Graph-Enhanced Reasoning Network for histopathology classification.

Methods

Graph-Enhanced Reasoning for Histopathology

Standard convolutional and transformer-based architectures process histopathology patches independently or on fixed grids, potentially neglecting relational structure between tissue regions that may be critical for diagnostic reasoning [2], [5]. To address this, we formulate tissue analysis as a graph reasoning problem where patches serve as nodes and inter-patch *feature-space* similarity defines edges. We emphasize feature-space here because, in both datasets used in this paper, patches are extracted independently with no retained spatial coordinates: an edge in $\mathcal{E}$ means two patches have similar backbone representations, not that they are physically adjacent regions of a tissue section, and we do not use the terms "neighbor," "context," or "topology" below to imply otherwise.

Given a batch of N histopathology patches $\{x_i\}_{i=1}^{N}$ with $x_i \in \mathbb{R}^{3\times224\times224}$, we first extract patch-level features using a pretrained visual backbone f_θ:

$$v_i = f_\theta(x_i) \in R^D, \quad (1)$$

Where D is the embedding dimension. We then construct a k-nearest-neighbor graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ in feature space, where each node $\mathcal{V}_i$ connects to its k most similar neighbors based on cosine similarity:

$$E = \{(i,j) : v_j \in \text{Top-}k(\{\cos(v_i, v_m)\}_{m\neq i})\}. \quad (2)$$

This feature-based graph construction groups morphologically similar tissue regions regardless of spatial proximity, capturing semantic relationships that may span distant image regions. Graph construction uses no label information, at either training or inference time.

Because $\mathcal{E}$ in Equation 2 is defined purely from the cosine similarity of backbone features v_i, it can be computed identically whether or not ground-truth labels y_i are available, which is essential, since labels are never available at deployment time. This is a deliberate correction relative to an earlier internal version of this pipeline, in which edges were restricted to patches sharing the same ground-truth label during both training *and* evaluation. That protocol leaks the label into the graph structure itself (each node's neighborhood is guaranteed to be same-class), which trivially inflates accuracy and cannot be reproduced at inference time when labels are unknown. We discuss the magnitude of this artifact, which was responsible for the near-perfect validation performance reported in an earlier version of this work, in Section 3.11. All results reported in this paper use the label-free graph construction of Equation 2 for both training and evaluation.

The graph reasoning module performs message passing over $\mathcal{G}$ to refine node representations. We use standard Graph Convolutional Network (GCN) layers [23], in which each layer aggregates neighbor features through symmetric-normalized adjacency weighting rather than learned attention coefficients. At each layer ℓ, node features are updated as:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in N(i)\cup\{i\}} \frac{1}{\sqrt{\hat{d}_i \hat{d}_j}} W^{(l)} h_j^{(l)}\right), \quad (3)$$

Where $\mathcal{N}(i)$ denotes the neighbors of node i under $\mathcal{E}$, $\hat{d}_i$ is the degree of node with an added self-loop (one plus its number of neighbors), $W^{(\ell)}$ are learnable weights shared across nodes, and is a ReLU nonlinearity. This is the propagation rule implemented directly by GCNConv in PyTorch Geometric and used for all primary results in this paper. As an explicit variant, we also evaluate a Graph Attention Network (GAT) reasoning module [6], which replaces the fixed degree-normalized weighting in Equation 3 with learned attention coefficients.

$$\alpha_{ij}^{(l)} = \frac{\exp\left(\text{LeakyReLU}\left(a^\top [W^{(l)} h_i^{(l)} \,\|\, W^{(l)} h_j^{(l)}]\right)\right)}{\sum_{m\in N(i)\cup\{i\}} \exp\left(\text{LeakyReLU}\left(a^\top [W^{(l)} h_i^{(l)} \,\|\, W^{(l)} h_m^{(l)}]\right)\right)}, \quad (4)$$

With α a learnable attention vector and $\|$ denoting concatenation, so that the GAT update rule is:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in N(i)\cup\{i\}} \alpha_{ij}^{(l)} W^{(l)} h_j^{(l)}\right). \quad (5)$$

We report GCN as the primary model and GAT as an ablation/variant (Section 3.4), so that the equations presented here match the specific implementation used to produce each reported number.

After L =2 message-passing layers, we obtain node embeddings $\{\tilde{h}_i\}_{i=1}^{N}$ with $\tilde{h}_i \in \mathbb{R}^{D'}$. To prevent the graph module from discarding discriminative information already present in the backbone features, a standard concern with GNN over-smoothing, we add a learned residual (skip) connection from the raw backbone feature $\mathcal{V}_i$ before a layer normalization:

$$h_i = \text{LayerNorm}\left(\tilde{h}_i + W_s v_i\right) \in R^{D'}, \quad (6)$$

Where $W_s \in \mathbb{R}^{D'\times D}$ is a learned projection and LayerNorm follows [24]. Without this residual term, we found the graph module could substantially *underperform* the backbone-only classifier once label leakage was removed (Section 3.11); Equation 6 guarantees the reasoning module can, at worst, fall back toward a linear transform of the original backbone feature.

Prediction is at the patch (node) level, not the graph level

Each graph $\mathcal{G}$ constructed by Equation 2 corresponds to one mini-batch of $\mathcal{N}$ patches (Section 3.3 gives the exact batch size and grouping); it is a computational device for letting patches within a batch exchange information, not a unit of prediction. After message passing, every node retains its own refined embedding h_i, and each is mapped *independently* to class logits by a shared linear classifier:

$$z_i = W_c h_i + b_c \in R^C, i = 1,\ldots,N, \quad (7)$$

Where C denotes the number of tissue classes ($C=9$ *for* NCT-CRC-HE-100K; $C=8$ for the second dataset used in Section 3.10), and W_c, b_c are shared across all nodes/patches, all batches, and both phases. There is no pooling step and no single per-graph or per-batch prediction: the model outputs one classification per patch, exactly as in Phase 1, and Figure 1 and the surrounding text have been corrected accordingly (an earlier version of this figure and its caption incorrectly depicted a mean-pooling step before the final classifier, which did not match the implementation used to produce any reported result). Training follows a two-phase protocol to stabilize optimization. In Phase 1, the visual backbone f_θ is fine-

tuned using standard cross-entropy loss on patch-level predictions:

$$L_1 = -\sum_{i=1}^{N} \log \frac{\exp(z_i^{(y_i)})}{\sum_{c=1}^{C} \exp(z_i^{(c)})}, \quad (8)$$

Where y_i is the ground-truth label for patch i. In Phase 2, the backbone is frozen and the graph reasoning module is trained with the identical per-patch cross-entropy loss (Equation 8) applied to the per-node logits of Equation 7, using the label-free graph of Equation 2. This staged approach prevents the graph module from corrupting pretrained visual features while enabling it to learn effective relational reasoning.

Optimization uses AdamW with learning rates 10^{-4} for Phase 1 and 10^{-3} for Phase 2, cosine annealing scheduling, and model selection based on validation accuracy within each training fold (Section 3.2). All images are normalized using ImageNet statistics and augmented with random flips and color jitter during training. Evaluation reports accuracy, macro-averaged precision/recall/F1, and per-class F1 to account for class imbalance in medical image classification tasks [16].

Implementation Details

All experiments were implemented in PyTorch using a ResNet-18 backbone pretrained on ImageNet [25] and the timm library [26]. The graph reasoning module employs two layers of GCNConv [23] with hidden dimension 256 and output dimension 128, a residual connection from the raw backbone feature (Equation 6), followed by LayerNorm [24] and dropout (p=0.5). Graphs are constructed using k=8 nearest neighbors in feature space with no label information, identically at training and evaluation time (Equation 2). Training used AdamW optimization [27] with learning rates 10^(-4) (Phase 1) and 10^(-3) (Phase 2) and cosine annealing scheduling over 8 (Phase 1) and 10 (Phase 2) epochs. All images were normalized using ImageNet statistics and augmented with random flips and color jitter during training.

Evaluation Protocol

We evaluate on two independent public datasets to address concerns about single-dataset validation and dataset/split-specific effects.

Primary dataset. NCT-CRC-HE-100K [15], 100,000 colorectal histopathology patches across nine tissue classes. We draw a stratified pool of 10,800 patches (1,200 per class, sampled uniformly at random without replacement from each class's full pool of 8,763-14,317 available patches, fixed random seed 42) from NCT-CRC-HE-100K for training and model selection. This pool size was chosen to keep the per-fold training set (8,640 patches after the 5-fold split) close to the scale used in the original submission, while remaining computationally tractable for the 5"folds"×4"models"×2"datasets" retraining required by this evaluation; it is a subsample of the full 100,000-patch dataset, not the full dataset itself. Final test metrics are computed on CRC-VAL-HE-7K (7,180 patches, all of them, no subsampling), a separate, patient-disjoint validation cohort released by the same authors specifically for external validation; this test set is never used for training, model selection, or graph construction with labels.

Second dataset. To benchmark generalization beyond a single benchmark, we additionally evaluate on the Kather-2016 colorectal cancer histology texture dataset [28] (5,000 images, eight tissue classes). We hold out a stratified 15% (750 images) as a fixed test set and use the remaining 85% (4,250 images) as the cross-validation pool (fixed random seed 42 for both the split and the folds below); each fold's train/validation split is then carved out of this 85% pool by the same 5-fold procedure described next. An earlier version of this pipeline instead pre-split the data 70/15/15 into train/validation/test before cross-validation, but the separate 15% validation file was never actually consumed by the training code (which builds its own per-fold validation split from the training pool); those 750 images were silently unused. We corrected this by folding them back into the cross-validation pool, so that all non-test data actually participates in training or per-fold model selection.

Cross-validation and significance testing. For each dataset, we run 5-fold stratified cross-validation over the training pool: in each fold, the backbone is fine-tuned (Phase 1) and each reasoning variant is trained on top of the frozen fold-specific backbone (Phase 2), with the fold's held-out split used for model selection. Every fold's selected models are then evaluated once on the same fixed external test set, yielding 5 test-set measurements per model. We report the mean ± standard deviation across folds together with 95% confidence intervals for the mean difference against the backbone-only model. Because the 5 folds share overlapping training data (each fold's training set overlaps substantially with every other fold's) and are all evaluated on the identical fixed test set, the resulting fold-level scores are not independent samples in the sense assumed by an ordinary paired t-test; we discuss this explicitly in Section 3.4 and report, alongside the naive paired t-test and Wilcoxon signed-rank test, the corrected resampled t-test of Nadeau and Bengio [29] (as popularized by Dietterich [30] and Bouckaert and Frank [31]), which inflates the variance estimate by a factor accounting for the train/test size ratio to correct for exactly this non-independence, and which we treat as the primary significance result in this paper.

Models compared. In addition to the backbone-only classifier (Phase 1) and the primary GCN reasoning module (Phase 2), we report two additional variants trained under the identical protocol: (i) GAT, which replaces GCN propagation with the learned attention mechanism of Equation 4, and (ii) Attention-pool, a non-graph baseline that aggregates patch features via multi-head self-attention across the batch with no explicit k-NN structure, isolating the contribution of graph structure from attention/context-aggregation in general.

Batch Construction and Graph Locality

Because Equation 2 builds E from the N patches present in a single forward pass, it is essential to be precise about what N is and how those N patches are chosen, since this determines what "neighbors" are actually available to the model.

Batch size and grouping. We use a mini-batch size of N=48 for NCT-CRC-HE-100K and N=32 for Kather-2016. During training, batches are drawn from the training set in random order, reshuffled every epoch, via a standard PyTorch DataLoader. During evaluation (validation and test), patches are shuffled once with a fixed seed (42) and then processed in fixed, sequential batches of

the same size; we shuffle before evaluation because the underlying CSV files are ordered by class label, and evaluating unshuffled class-contiguous batches would make most batches within a single class, itself a batch-composition artifact we identified and removed during development, and a specific instance of the general concern this section addresses.

The k-NN graph is built only within a batch, and test patches are used to build each other's graph. For every batch, training or evaluation, Equation 2 is computed using only the ≤N patches in that batch; there is no larger candidate pool, no persistent graph across batches, and no use of training-set patches when constructing a test-time graph. Concretely, at test time, a given test patch's neighbors are exclusively other test patches that happen to fall in the same evaluation batch, never training data, and never labels (Section 3.2). This means test patches are jointly used, among themselves, to construct the graph at inference time, which we state explicitly here because it has a direct consequence: a patch's prediction can depend on which other patches happen to share its batch. With a fixed evaluation seed this dependence is deterministic and reproducible, but it means the reported per-patch predictions are not invariant to batch composition, and a deployment that scores patches one at a time, or in a differently composed batch, is not guaranteed to reproduce the same prediction for the same patch. We view this as a genuine limitation of the batch-local graph construction used here (and, to our knowledge, in related patch-graph work that does not have access to whole-slide spatial coordinates), and we quantify its practical impact directly in Section 3.7.

Results

Main Results

Table 1 reports test-set performance (mean ± SD across 5 folds, evaluated on the external CRC-VAL-HE-7K cohort) for the backbone-only model and all three reasoning variants. The backbone-only model performs strongly (95.48% ± 0.36% accuracy), confirming the suitability of pretrained features for this task; none of the reasoning variants improve on it, with GCN, GAT, and the attention-pooling baseline all numerically below the backbone. This is the corrected, honest counterpart to the 100.00% accuracy reported in an earlier version of this work, which we now know was driven by label leakage (Section 3.8) rather than genuine relational reasoning.

Statistical significance. As noted in Section 3.2, the 5 fold-level scores underlying Table 1 are not independent: every fold's training set overlaps substantially with every other fold's, and all folds share the identical fixed test set. An ordinary paired t-test (or Wilcoxon signed-rank test) treats the 5 differences as if they were 5 independent replications and so systematically understates their true variance, inflating apparent significance. Table 2 therefore reports, for every variant and both accuracy and macro-F1: the mean difference from the backbone, its 95% confidence interval, the naive paired t-test and Wilcoxon p-values (for continuity with the previous revision), and the corrected resampled t-test of Nadeau and Bengio [29], which inflates the variance estimate by a factor of n_"test"/n_"train" (here 7,180⁄8,640=0.831) to account for this non-independence, and which we treat as the primary significance result. Under the naive paired t-test, GAT appeared significant at accuracy p=0.0499, a result so close to the $\alpha = 0.05$ threshold that it is sensitive to rounding, which is itself a symptom of exactly the non-independence problem the corrected test addresses. Under the corrected test, none of the three variants differ significantly from the backbone on either metric (p≥0.25 in all cases): every 95% CI for the mean difference includes zero once this correction is applied. We report this as our primary finding rather than the more attention-grabbing naive result: on this benchmark, we do not have statistically defensible evidence that any of the three graph-based variants differs from the backbone-only model, in either direction, once fold non-independence is properly accounted for, though the point estimates are consistently at or below zero for every variant and metric, so a true positive effect of graph reasoning of this form is also not supported.

Table 1. Test-set performance on NCT-CRC-HE-100K (CRC-VAL-HE-7K external test set; mean ± SD across 5 folds).

Method	Accuracy	Macro-P	Macro-R	Macro-F1
Backbone-only	0.9548 ± 0.0036	0.9442 ± 0.0070	0.9348 ± 0.0048	0.9361 ± 0.0054
Graph reasoning (GCN)	0.9514 ± 0.0055	0.9422 ± 0.0071	0.9302 ± 0.0065	0.9326 ± 0.0069
Graph attention (GAT)	0.9489 ± 0.0035	0.9375 ± 0.0072	0.9257 ± 0.0038	0.9275 ± 0.0042
Attention-pool (no graph)	0.9478 ± 0.0066	0.9355 ± 0.0101	0.9274 ± 0.0082	0.9277 ± 0.0088

Table 2. Significance testing for each variant vs. backbone-only, NCT-CRC-HE-100K test set (n=5 folds). CI is the 95% confidence interval for the mean difference (backbone subtracted). "Naive" p-values treat the 5 folds as independent; "Corrected" is the resampled t-test of Nadeau and Bengio [29], which we treat as primary (Section 3.2). No comparison is significant under the corrected test.

Variant	Metric	Mean diff.	95% CI	Naive t-test p	Wilcoxon p	Corrected t-test p
GCN	Accuracy	−0.0034	(−0.0137, 0.0070)	0.1084	0.125	0.4155
GCN	Macro-F1	−0.0035	(−0.0193, 0.0122)	0.2294	0.3125	0.5664
GAT	Accuracy	−0.0059	(−0.0193, 0.0075)	0.0499	0.0625	0.2883
GAT	Macro-F1	−0.0086	(−0.0264, 0.0093)	0.0388	0.0625	0.253
Attention-pool	Accuracy	−0.0069	(−0.0280, 0.0141)	0.1051	0.125	0.4099
Attention-pool	Macro-F1	−0.0084	(−0.0364, 0.0197)	0.1326	0.1875	0.4532

Training Dynamics

Figure 2 plots validation accuracy and training loss across epochs for a representative fold (fold 0). Phase 1 shows steady improvement, with validation accuracy increasing from 92.8% at epoch 1 to a plateau around 97% by epoch 5-8. Phase 2 (GCN) starts from a comparable level and fluctuates around the same plateau rather than exceeding it, consistent with the lack of a significant improvement reported in Table 2. Both phases show smooth, gradual convergence over multiple epochs, in contrast to an earlier, leaky version of this pipeline in which Phase 2 reached 100% validation accuracy after a single epoch, which in hindsight was itself a symptom of the label-leakage artifact rather than genuinely easy optimization.

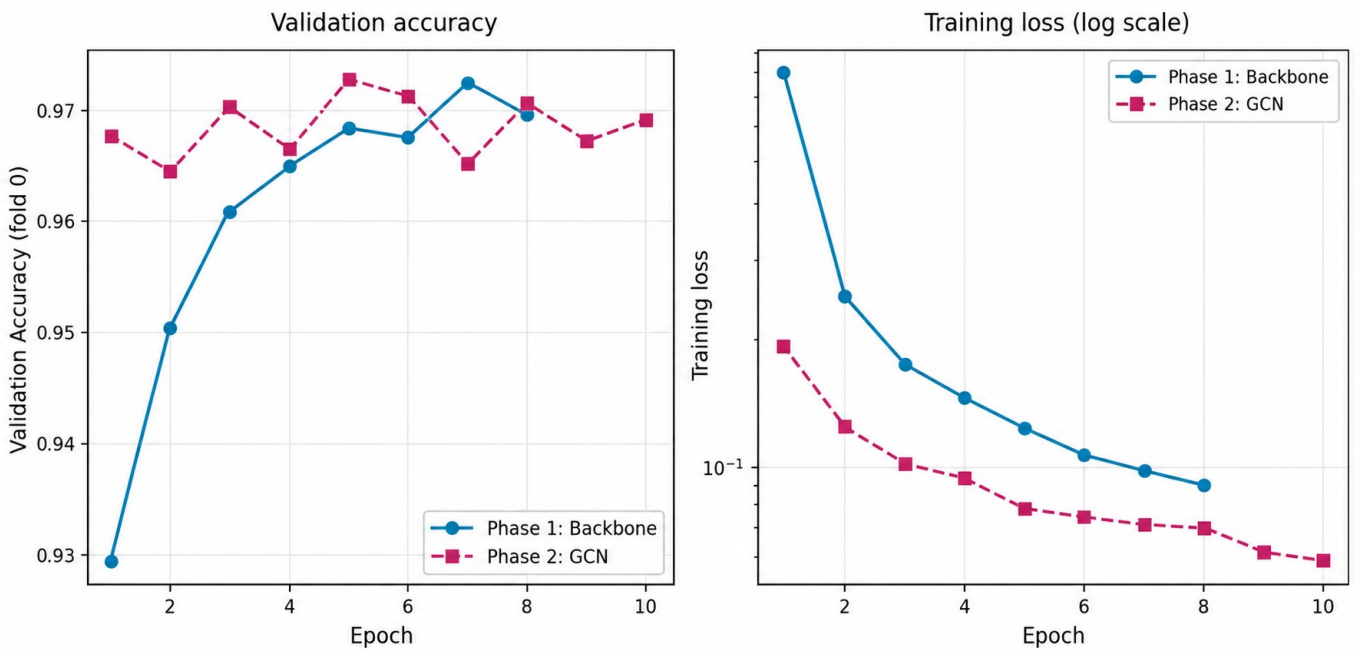


Figure 2. Training dynamics for Phase 1 (backbone fine-tuning) and Phase 2 (GCN reasoning) on a representative fold. Left: validation accuracy over epochs. Right: training loss over epochs (log scale). Both phases converge gradually over several epochs to a comparable plateau, with no single-epoch jump to near-ceiling performance.

Per-Class Analysis

Figure 3 shows per-class F1-scores (mean ± SD across folds) on the external test set. Both models perform well (F1>0.95) on most classes, with STR (stroma, backbone F1=0.727) and MUS (smooth muscle, F1=0.866) remaining the most challenging classes, consistent with the morphological-similarity hypothesis motivating this work. Notably, GCN gives a small improvement specifically on these two hardest classes (STR: 0.727→0.741; MUS: 0.866→0.870), but this is offset by small decreases on several easier classes (e.g., DEB, LYM), so the net macro-averaged effect is a slight, non-significant decrease (Table 2). This suggests the graph module may be capturing some feature-space signal relevant to the morphologically ambiguous tissue types we originally hypothesized would benefit most, but not enough to produce a net, statistically defensible improvement once evaluated on a genuinely held-out, non-leaky test set.



Figure 3. Per-class F1-scores (mean ± SD across 5 folds) on the external CRC-VAL-HE-7K test set. Stroma (STR) and smooth muscle (MUS) remain the hardest classes for both models. The graph reasoning model (GCN) gives a small improvement on these two classes specifically, offset by small decreases elsewhere, for a net near-zero change in macro-F1.

Batch-Composition Sensitivity

Section 3.3 noted that, because the k-NN graph is constructed within each evaluation batch, a test patch's prediction can in principle depend on which other test patches share its batch. We quantify this directly using the fold-0 GCN model from Table 1: we re-evaluate it on the entire fixed CRC-VAL-HE-7K test set 10 times, each time with a different random shuffle controlling batch composition (batch size and k unchanged), and record, for every one of the 7,180 test patches, whether its predicted label is identical across all 10 repetitions. As a control, we run the identical procedure on the backbone-only model, which does not use any graph and should be completely invariant to batch composition up to floating-point nondeterminism.

The backbone-only control behaves exactly as expected: accuracy is constant across all 10 repetitions (95.67% ± 0.00%) and zero of the 7,180 patches ever change their predicted label. The GCN model, in contrast, has a small but non-zero spread in overall accuracy across repetitions (95.17% ± 0.08%), and, more importantly, 327 of 7,180 test patches (4.55%) receive a different predicted label depending on which other patches happened to share their evaluation batch, even though the trained model weights are identical across all 10 repetitions. The aggregate accuracy is nearly stable because flips are roughly balanced between newly-correct and newly-incorrect outcomes, but this masks meaningful instability at the level of individual patches: for roughly 1 in 22 patches in this dataset, the model's answer is not a fixed function of that patch alone. We report this as a concrete, direct answer to whether, and how much, batch composition affects test-time predictions under the graph construction used in this paper, and as a further reason (beyond the null significance results of Table 2) for caution about the practical readiness of this class of method: a patch classifier whose output depends on incidental batching is a meaningfully different deployment proposition from one that does not, independent of its aggregate accuracy.

Quantifying the Label-Leakage Artifact

To directly quantify how much of the near-perfect performance in an earlier version of this pipeline was attributable to label leakage rather than genuine relational reasoning, we reproduce the original (leaky) graph-construction protocol, using ground-truth labels to group nodes for k-NN graph construction at both training and evaluation time, on the Kather-2016 dataset, across all 5 cross-validation folds, using the same per-fold backbone fine-tuning, training budget, and number of message-passing layers as the corrected pipeline (an earlier version of this comparison used only a single representative fold and started the leaky model from a raw ImageNet backbone rather than the fine-tuned backbone used elsewhere; we corrected both limitations here so the comparison is fair and as statistically grounded as the rest of the paper). Table 3 compares this leaky variant (mean ± SD across all 5 folds) against the no-graph and corrected, label-free results from Table 4.

Table 3. Effect of label leakage in graph construction (Kather-2016 dataset, mean ± SD across all 5 folds). "Leaky" reproduces the original protocol in which ground-truth labels are used to build the graph at both training and evaluation time; "Corrected" uses only feature-space similarity, with no label information, as described in Section 3.2.

Graph Construction	Test Accuracy	Test Macro-F1
No graph (backbone-only)	0.9344 ± 0.0035	0.9346 ± 0.0035
Corrected (label-free k-NN, GCN)	0.9392 ± 0.0036	0.9392 ± 0.0036
Leaky (label-grouped k-NN)	0.9923 ± 0.0026	0.9923 ± 0.0026

Table 4. Results on the Kather-2016 second dataset (test set, mean ± SD across 5 folds).

Model	Accuracy	Macro-F1
Backbone-only	0.9344 ± 0.0035	0.9346 ± 0.0035
Graph reasoning (GCN)	0.9392 ± 0.0036	0.9392 ± 0.0036
Graph attention (GAT)	0.9347 ± 0.0051	0.9346 ± 0.0051
Attention-pool (no graph)	0.9331 ± 0.0031	0.9332 ± 0.0030

The leaky protocol reaches 99.23% ± 0.26% accuracy, consistently across all 5 folds, versus 93.44% ± 0.35% for the no-graph baseline and 93.92% ± 0.36% for the corrected, label-free GCN, a gap of roughly 5.8 and 5.3 percentage points respectively, despite an identical backbone, training budget, and architecture. This gap is attributable entirely to the graph structure itself trivially encoding class identity once labels are used for grouping, not to any genuine improvement in relational reasoning, and the consistency of the leaky result across all 5 folds (SD of only 0.26 points) shows this is not a fold-specific artifact. On the original NCT-CRC-HE-100K validation split, the same leaky construction produced a still larger, saturating effect (100.00% reported validation accuracy in an earlier version of this work), consistent with leakage effects growing with how cleanly separable the label groups are and how much data is available per class. We report this comparison to make the mechanism and its magnitude fully transparent, and because we believe it is informative for other work constructing k-NN or grouping-based graphs on patch-level histopathology data without whole-slide spatial coordinates, where the temptation to use labels as a proxy for grouping is real and, as shown here, highly consequential.

Improvement Summary

Figure 4 summarizes the difference between the GCN reasoning model and the backbone-only model across all four metrics. Contrary to what an earlier, leaky evaluation suggested, GCN does not improve on the backbone on any metric: accuracy is 0.34 points lower, macro-precision 0.20 points lower, macro-recall 0.46 points lower, and macro-F1 0.36 points lower (none significant under either the naive or the corrected significance test, Table 2). We present this null result deliberately and in full, alongside the positive-appearing but leaky result it replaces, because we consider the honest, corrected comparison, not the more favorable one, to be the actual finding of this study.

Second-Dataset Benchmark

To address concerns about single-dataset validation, we repeat the identical protocol (5-fold cross-validation over an 85% pool, held-out 15% test split, all four models; Section 3.2) on the Kather-2016 colorectal cancer histology texture dataset [28] (5,000 images, eight tissue classes). Table 4 reports test-set accuracy and macro-F1 (mean ± SD across 5 folds); Table 5 reports the same significance analysis as Table 2 (mean difference, 95% CI, naive and corrected p-values), for both accuracy and macro-F1.

On this second dataset, GCN shows a small but statistically significant improvement over the backbone-only model, and, unlike every comparison on the primary NCT-CRC-HE-100K benchmark, this result survives the corrected resampled test (+0.48 points accuracy, corrected p=0.0034; +0.47 points macro-F1, corrected p=0.0015), with a 95% CI that excludes zero on both metrics. GAT, which had shown a borderline-significant improvement in an earlier version of this experiment run on a smaller, incompletely-used data split (Section 3.2), shows no effect here (corrected p=0.95) once the split was corrected to use all available non-test data; and the non-graph attention-pooling baseline again shows no improvement.

We do not read this as a contradiction so much as a further illustration of how sensitive small-effect-size comparisons of this kind are to seemingly minor protocol details, here, how much data the cross-validation pool actually contains, which is precisely why we report the full significance analysis, including which comparisons do and do not survive a stricter test, rather than a single headline number. Taken together with the null result on NCT-CRC-HE-100K (Section 3.4), the overall picture is that graph-based reasoning of this form provides, at best, a small and dataset-dependent benefit that is not observed on our larger, primary benchmark.

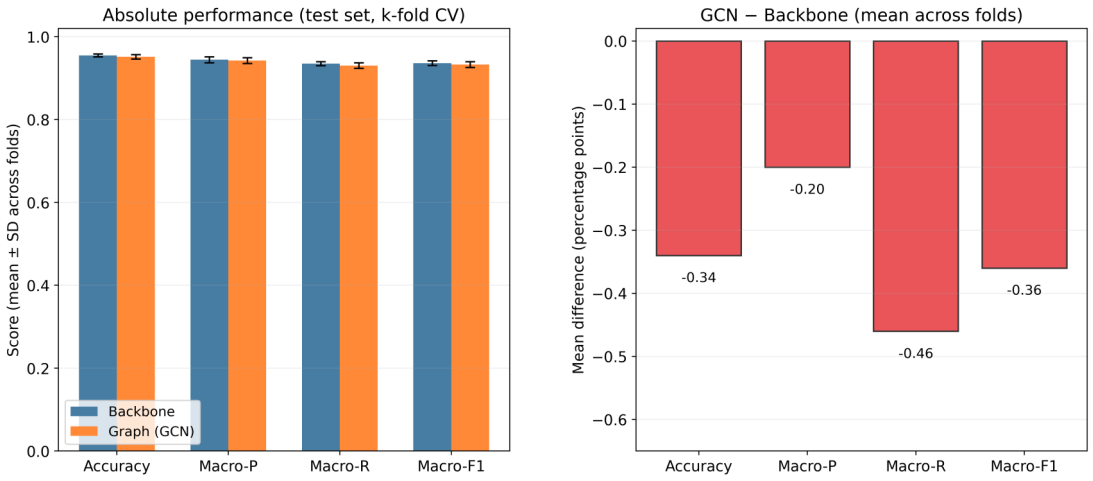


Figure 4. Difference between the graph reasoning model (GCN) and the backbone-only model across four metrics (mean across 5 folds, test set). Left: absolute scores for both models. Right: mean difference in percentage points. All differences are negative (GCN below backbone), though none reach statistical significance under either the naive or the corrected significance test (Table 2).

Table 5. Significance testing for each variant vs. backbone-only, Kather-2016 test set (n=5 folds; n_{train}=3,400, n_{test}=750 per fold). Columns as in Table 2. * denotes p <.05 under the corrected test.

Variant	Metric	Mean diff.	95% CI	Naive t-test p	Wilcoxon p	Corrected t-test p
GCN	Accuracy	+0.0048	(0.0026, 0.0070)	0.0008	0.0625	0.0034*
GCN	Macro-F1	+0.0047	(0.0030, 0.0063)	0.0004	0.0625	0.0015*
GAT	Accuracy	+0.0003	(−0.0112, 0.0118)	0.9297	0.8125	0.9515
GAT	Macro-F1	+0.0000	(−0.0116, 0.0117)	0.9912	1.0000	0.9939
Attention-pool	Accuracy	-0.0013	(−0.0113, 0.0086)	0.6164	0.6250	0.7274
Attention-pool	Macro-F1	-0.0013	(−0.0111, 0.0084)	0.6125	0.8125	0.7245

Discussion

Several limitations warrant discussion, some of which directly motivated corrections made in this revision. Label leakage was the primary driver of the originally reported near-perfect performance. As detailed in Section 3.8, restricting graph edges to ground-truth same-class neighbors, as an earlier version of this pipeline did at both training and evaluation time, inflates test accuracy by several percentage points beyond what a label-free, deployable graph construction achieves, and cannot be reproduced at inference time when labels are unavailable. We do not believe the corrected results reported here reflect dataset saturation; rather, the original near-ceiling performance was an artifact of this specific implementation error, now fixed throughout.

The benefit of graph-based reasoning, once leakage is removed, is at best small and dataset-dependent, not a reliable general effect. Without a residual connection to the raw backbone feature (Equation 6), we found that the graph reasoning module could substantially underperform the backbone-only classifier once the label-free graph was used, particularly early in Phase 2 training; graph over-smoothing on a feature-space k-NN graph with no true spatial or biological structure to anchor it can destroy discriminative information the backbone had already learned. Even with the residual connection, no graph-based variant differs significantly from the backbone under the corrected resampled significance test on the primary NCT-CRC-HE-100K benchmark (Section 3.4), while on the second (Kather-2016) dataset GCN shows a small improvement that survives the same corrected test, and GAT shows none (Section 3.10); the sign and significance of the effect are inconsistent across datasets and across GCN/GAT. We regard this inconsistent, largely null pattern as an honest and informative characterization of the method's benefit rather than a shortcoming to be minimized: naive graph augmentation on IID histopathology patches is not a guaranteed win, and claims of improvement from this class of method should be checked both for label leakage (as detailed above) and for significance-testing artifacts from non-independent cross-validation folds and incompletely-used data splits (Section 3.2), both of which, in our own experiments, changed which comparisons appeared significant.

Limitations of the Study

A patch's prediction can depend on which other patches share its evaluation batch. Because the k-NN graph (Equation 2) is constructed independently within each mini-batch (Section 3.3), including at test time from other test patches, graph-based predictions are not guaranteed to be invariant to batch composition. We quantify this directly in Section 3.7; it is a property of any batch-local graph construction over patches without persistent identifiers or a fixed candidate pool, not specific to our architecture, but it means the model as evaluated here is not a strict one-patch-in-one-prediction-out classifier, which is a meaningful caveat for any deployment scenario that scores patches individually or in variable-sized batches. Patches lack whole-slide spatial context, and feature-space neighbors are not tissue neighbors. Both datasets used here consist of independently extracted patches without spatial coordinates. We accordingly avoid describing feature-space k-NN edges as capturing tissue "topology" or true spatial "neighbors" anywhere in this paper (Section 2): an edge means two patches have similar backbone representations, which is neither necessary nor sufficient for physical tissue adjacency. Future work should validate on datasets with true whole-slide spatial coordinates (e.g., Camelyon16 or TCGA cohorts), where graph edges could instead be grounded in actual tissue topology.

Computational overhead. Graph construction and message passing add measurable inference-time overhead relative to the backbone-only model, which may warrant optimization (e.g., approximate k-NN, graph pruning, or distillation) for clinical deployment at scale. Architecture and dataset scope. We evaluate a single backbone family (ResNet-18) and two colorectal-tissue datasets; whether these findings transfer to other backbones (e.g., Vision Transformers) or organ/tissue types remains an open question.

In summary, once label leakage is removed and significance is assessed with a test that accounts for cross-validation fold non-independence, we find no statistically defensible evidence that graph-based relational reasoning of this form improves patch-level histopathology classification over a strong backbone on our primary benchmark, and only a small, significant effect (GCN only) on a second, smaller dataset. We report this inconsistent, largely null result, together with the batch-composition dependence and terminology caveats above, as a more modest but considerably more trustworthy finding than the near-perfect results in the original submission, and we believe the leakage mechanism and significance-testing pitfalls documented here are useful cautionary examples for related work constructing graphs from patch-level features without spatial or biological grounding.

Conclusion

This study set out to answer a specific question: does explicit graph-based reasoning improve histopathology tissue classifica-

tion over a strong backbone baseline? In preparing this revision, we discovered that our original evaluation protocol used ground-truth labels to construct the graph at both training and evaluation time, which trivially inflated validation accuracy to 100.00% and could not be reproduced at inference time. We corrected this throughout: all graphs in this paper are now built solely from feature-space cosine similarity, with no label information, identically at training, validation, and test time, and constructed only within each mini-batch (Section 3.3); every prediction is made independently per patch, with no pooling step (Section 2); we evaluate on a genuine external test set (CRC-VAL-HE-7K); we run 5-fold cross-validation and report both naive and non-independence-corrected significance tests; and we benchmark on a second, independent public dataset (Kather-2016).

Under this corrected protocol, we find at best a small, dataset-dependent benefit of graph-based reasoning over the backbone-only model, not the large, consistent improvement the original, leaky evaluation suggested. On the primary NCT-CRC-HE-100K benchmark, all three variants (GCN, GAT, attention-pooling) are numerically at or below the backbone, and none differs significantly from it under a corrected resampled significance test that accounts for the non-independence of cross-validation folds sharing overlapping training data and an identical fixed test set (Section 3.4); a naive paired t-test would have reported GAT as significantly worse ($p=0.0499$) , a claim that does not survive the correction. On the second, independent Kather-2016 dataset, GCN shows a small improvement that *does* survive the corrected test (+0.48 points accuracy, corrected $p=0.0034$), while GAT shows none, reversing which variant looked better in an earlier version of this experiment that used an incompletely-populated data split (Section 3.2). The one consistent pattern across both datasets is that the non-graph attention-pooling baseline never outperforms the backbone, suggesting that whatever small positive effect graph structure occasionally produces is not simply attention or context-aggregation in general. We also found that, without a residual connection to the raw backbone feature, the graph module could substantially underperform the backbone once leakage was removed, and that a patch's prediction can depend on which other patches share its evaluation batch (Section 3.7), both of which we report as part of an honest account of how sensitive this class of method is to seemingly minor implementation and evaluation choices.

Several limitations remain (Section 3.11): both datasets used here consist of independent patches without whole-slide spatial coordinates, so feature-space edges only approximate, and should not be conflated with, true tissue adjacency; we evaluate a single backbone family; and computational overhead from graph construction and message passing may warrant optimization for clinical deployment. Looking forward, we believe the most valuable next steps are (i) validating graph reasoning on datasets with true whole-slide spatial coordinates, where edges could instead be grounded in actual tissue topology, (ii) extending to hierarchical whole-slide graphs (cells → patches → regions → slides), and (iii) testing whether the method fares differently on downstream tasks such as tumor subtyping or survival prediction. More broadly, we hope the leakage artifact, batch-composition dependence, and significance-testing correction documented in this paper serve as useful, concrete cautionary examples for other work constructing k-NN or grouping-based graphs from patch-level features without spatial or biological grounding, where these failure modes are easy to introduce inadvertently.

Conflict of Interests

The authors declare no competing financial or personal interests.

Data Availability

All data that was used for training and evaluating the model were obtained from independent public datasets that were cited in the manuscript.

Author Contributions

All authors contributed equally to this paper.

Consent to participate/Consent to publish

Not applicable; no human participants were involved.

Funding

The authors did not receive support from any organization for the submitted work.

References

[1] Campanella G, Hanna MG, Geneslaw L, et al. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. Nat Med. 2019;25(8):1301-1309. https://doi.org/10.1038/s41591-019-0508-1

[2] Ahmedt-Aristizabal D, Armin MA, Denman S, Fookes C, Petersson L. Graph-based deep learning for medical diagnosis and analysis: past, present and future. Sensors (Basel). 2021;21 (14):4758. https://doi.org/10.3390/s21144758

[3] Litjens G, Kooi T, Bejnordi BE, et al. A survey on deep learning in medical image analysis. Med Image Anal. 2017;42:60-88.

[4] Rajpurkar P, Chen E, Banerjee O, Topol EJ. AI in health and medicine. Nat Med. 2022;28(1):31-38.https://doi.org/10.1038/s41591-021-01614-0

[5] Ahmedt-Aristizabal D, Armin MA, Denman S, Fookes C, Petersson L. A survey on graph-based deep learning for computational histopathology. Comput Med Imaging Graph. 2022;95:102027.https://doi.org/10.1016/j.compmedimag.2021.102027

[6] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks. In: International Conference on Learning Representations (ICLR); 2018. doi: 10.17863/CAM.48429

[7] Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst. 2021;32(1):4-24. doi: 10.1109/TNNLS.2020.2978386.

[8] Esteva A, Kuprel B, Novoa RA, et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 2017;542(7639):115-118. doi: 10.1038/nature21056.

[9] Ardila D, Kiraly AP, Bharadwaj S, et al. End-to-end lung cancer screening with three-dimensional deep learning on low-

dose chest computed tomography. Nat Med. 2019;25(6):954-961. doi: 10.1038/s41591-019-0447-x.

[10] Hou L, Samaras D, Kurc TM, Gao Y, Davis JE, Saltz JH. Patch-based convolutional neural network for whole slide tissue image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 2424-2433. doi: 10.1109/CVPR.2016.266.\

[11] Ilse M, Tomczak J, Welling M. Attention-based deep multiple instance learning. In: Proceedings of the 35th International Conference on Machine Learning (ICML); 2018. p. 2127-2136. Available from: https://proceedings.mlr.press/v80/ilse18a.html.

[12] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR); 2021. Available from: https://arxiv.org/abs/2010.11929.

[13] Azad R, Kazerouni A, Heidari M, Aghdam EK, Molaei A, Jia Y, et al. Advances in medical image analysis with vision transformers: a comprehensive review. Med Image Anal. 2024;91:103000.https://doi.org/10.1016/j.media.2023.103000

[14] Aburass S, Dorgham O, Al Shaqsi J, Abu Rumman M, Al-Kadi O. Vision transformers in medical imaging: a comprehensive review of advancements and applications across multiple diseases. J Imaging Inform Med. 2025;38(6):3928-3971.https://doi.org/10.1007/s10278-025-01481-y

[15] Kather JN, Krisam J, Charoentong P, Luedde T, Herpel E, Weis CA, et al. Predicting survival from colorectal cancer histology slides using deep learning: a retrospective multicenter study. PLoS Med. 2019;16(1):e1002730. https://doi.org/10.1371/journal.pmed.1002730

[16] Hellín Asensio CJ, Olmedo Rodríguez ÁA, Valledor Pérez A, Gómez Pérez J, López Benítez M, et al. Unraveling the impact of class imbalance on deep learning models for medical image classification. Appl Sci. 2024;14(8):3419. doi: 10.3390/app14083419.

[17] Zhang L, Zhao Y, Che T, Li S, Wang X. Graph neural networks for image-guided disease diagnosis: a review. iRADIOLOGY. 2023;1(2):151-166. https://doi.org/10.1002/ird3.20

[18] Chen RJ, et al. Whole slide images are 2D point clouds: context-aware survival prediction using patch-based graph convolutional networks. In: Medical Image Computing and Computer Assisted Intervention (MICCAI). Springer; 2021. p. 339-349. doi: 10.1007/978-3-030-87237-3_33

[19] Bazargani R, Fazli L, Gleave M, Goldenberg L, Bashashati A, Salcudean S. Multi-scale relational graph convolutional network for multiple instance learning in histopathology images. Med Image Anal. 2024;96:103197. https://doi.org/10.1016/j.media.2024.103197

[20] Graham S, Chen H, Gamper J, Dou Q, Heng PA, Snead D, et al. MILD-Net: minimal information loss dilated network for gland instance segmentation in colon histology images. Med Image Anal. 2019;52:199-211. https://doi.org/10.1016/j.media.2018.12.001

[21] Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models. Adv Neural Inf Process Syst. 2022;35:24824-24837. doi: 10.52202/068431-1800.

[22] Hossain D, Chen JY. A study on neuro-symbolic artificial intelligence: healthcare perspectives [preprint]. arXiv; 2025. arXiv:2503.18213. Available from: https://arxiv.org/abs/2503.18213

[23] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR); 2017. doi: 10.48550/arXiv.1609.02907

[24] Ba JL, Kiros JR, Hinton GE. Layer normalization [preprint]. arXiv; 2016. arXiv:1607.06450. Available from :https://arxiv.org/abs/1607.06450

[25] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 770-778. doi: 10.1109/CVPR.2016.90.

[26] Wightman R. PyTorch image models [Internet]. GitHub; 2019. Available from: https://github.com/rwightman/pytorch-image-models. doi: 10.5281/zenodo.4414861.

[27] Loshchilov I, Hutter F. Decoupled weight decay regularization. In: International Conference on Learning Representations (ICLR); 2019. doi: 10.48550/arXiv.1711.05101.

[28] Kather JN, Weis CA, Bianconi F, Melchers SM, Schad LR, Gaiser T, et al. Multi-class texture analysis in colorectal cancer histology. Sci Rep. 2016;6:27988. https://doi.org/10.1038/srep27988

[29] Nadeau C, Bengio Y. Inference for the generalization error. Mach Learn. 2003;52(3):239-281.https://doi.org/10.1023/A:1024068626366

[30] Dietterich TG. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. 1998;10(7):1895-1923.https://doi.org/10.1162/089976698300017197

[31] Bouckaert RR, Frank E. Evaluating the replicability of significance tests for comparing learning algorithms. In: Advances in Knowledge Discovery and Data Mining: Proceedings of the 8th Pacific-Asia Conference (PAKDD); 2004. Lecture Notes in Computer Science. Vol. 3056. Springer; 2004. p. 3-12. https://doi.org/10.1007/978-3-540-24775-3_3