SPECCERT: DETERMINISTIC JOINT CERTIFICATION AND MARGIN-EXPLICIT TRAINING FOR GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks deliver state-of-the-art accuracy in graph classification and support many real-world applications, including drug discovery, fraud detection, and malware classification. However, they remain vulnerable to adversarial perturbations on both graph structure and node features, posing serious challenges for security-critical applications. Although existing certified methods provide formal robustness guarantees through smoothing or ensemble voting, they suffer from fundamental limitations: count-based certification bounds discard confidence information, training objectives are not explicitly matched to certification margins, and topology-dependent partitioning can incur substantial overhead. In this paper, we propose SpecCert, a deterministic certification framework for graph neural networks with margin-explicit training and a confidence-weighted extension of count-based certification. Specifically, we introduce a topology-independent hash partitioning scheme that decomposes graphs into sub-views where each perturbation affects a bounded number of views. We then develop Margin-Explicit Certification Training (MECT), a differentiable surrogate objective designed to encourage larger certification-relevant margins. Finally, we derive a confidence-weighted certificate and combine it with the count-based certificate through a logically sound OR rule. SpecCert provides formal robustness guarantees under simultaneous bounded structure and node-feature perturbations. Experiments on eight benchmark datasets show that the deterministic framework provides competitive certified accuracy and substantially lower certification cost than the evaluated smoothing baselines. The confidence-weighted branch is formally valid but its empirical tightness depends on the distribution of sub-view confidences. The source code for SpecCert is available at https://anonymous.4open.science/r/SpecCert-17E0/.

1 INTRODUCTION

Graph classification predicting a class label for an entire graph is a fundamental task in graph analytics, with deployments in fraud detection (Weber et al., 2019), malware detection (Kong & Yan, 2013; Hassen & Chan, 2017; Yan et al., 2019), and healthcare (Li et al., 2017; Chen et al., 2018). Graph neural networks (Kipf & Welling, 2016; Velickovic et al., 2017; Xu et al., 2019) have established themselves as the state-of-the-art classifiers for this task, leveraging message passing aggregation to learn graph-level representations from the joint structure and feature signal. Yet graph neural networks are repeatedly shown vulnerable to adversarial perturbations (Dai et al., 2018; Zügner et al., 2018; Tang et al., 2020; Wan et al., 2021; Mu et al., 2021; Wang et al., 2022; 2023). An attacker who modifies a small number of edges or node features can drive the classifier to predict an incorrect label, impeding deployment in security and safety-critical applications where formal robustness guarantees are essential.

Two complementary lines of defense have emerged. Empirical defenses Chen et al. (2020); Zhao et al. (2021); Charpentier et al. (2023) harden a classifier through architectural modifications, robust aggregation, or adversarial training, but cannot guarantee performance under arbitrary attacks: In (Mujkanovic et al., 2023), the authors showed that essentially every published empirical defense for graph neural networks can be circumvented by adaptive attacks designed against the defense itself. Certified defenses (Bojchevski et al., 2020; Wang et al., 2020; Zhang et al., 2020; Schuchardt

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2021; 2023; Xia et al., 2024) address this gap with formal guarantees. The predicted label is provably preserved whenever the perturbation budget is bounded. Existing certified defenses for graph classification, however, suffer from five limitations. *First*, several (Bojchevski et al., 2020; Wang et al., 2020; Zhang et al., 2021b; Xia et al., 2024) certify only structure perturbations and cannot guarantee robustness against feature manipulations. *Second*, they often achieve sub-optimal robustness because the certificate is consumed at a count-based vote margin and discards the per-sample softmax confidence that the base classifier already produces. *Third*, existing deterministic methods rely on topology-dependent partitioning that incurs high computational overhead and structural rigidity. Fourth, their guarantees are probabilistically correct only with confidence $(1 - \alpha)$ for some small $\alpha > 0$. *Fifth*, the underlying training recipe is plain cross-entropy, which optimizes per-sub-view classification accuracy but is indifferent to the aggregate vote margin that the certificate actually consumes, leaving certified accuracy unnecessarily low on harder benchmarks where the count margin saturates. At the core of these limitations lies a fundamental objective mismatch: existing methods train for per-sub-view accuracy, yet certify using a discrete margin that ignores predictive confidence. This motivates a critical research question: Can we design a deterministic certification framework that fully exploits both discrete vote counts and continuous confidence, while systematically aligning the training objective with the certification constraint?

We propose SpecCert, a deterministic certification framework with deterministic guarantees against bounded structure and feature perturbations. SpecCert partitions the graph's edges and node features into disjoint groups using a topology-blind avalanche-mixer hash. Each perturbation therefore affects a bounded number of sub-views, allowing the ensemble prediction to be certified without Monte Carlo sampling. Beyond partitioning, SpecCert introduces MECT, a differentiable surrogate designed to encourage the vote margins consumed by certification, and a confidence-weighted certificate that can be combined with the count-based certificate through a sound OR rule. Following Cohen et al. (2019), we train the base classifier on hashed sub-views to align the training and inference distributions.

We evaluate SpecCert on eight benchmark datasets (e.g., MUTAG Debnath et al. (1991), PROTEINS Borgwardt et al. (2005), COLLAB Yanardag & Vishwanathan (2015)) for graph classification. Following standard practice, we report certified accuracy, the fraction of test graphs that are correctly classified, and admit a certified perturbation size no smaller than a given radius and compare against four certified defense baselines: the deterministic GNNCert (Xia et al., 2024) and the three Monte Carlo smoothing methods of (Bojchevski et al., 2020), (Wang et al., 2020), and (Zhang et al., 2021b). SpecCert provides competitive certified accuracy across the evaluated benchmarks and certifies the test sets substantially faster than the evaluated Monte Carlo baselines. The largest empirical differences appear on the more difficult datasets and at larger perturbation budgets; the individual contributions of partitioning, MECT, and the confidence-weighted branch are analyzed separately below.

Our major contributions are: (i) We propose SpecCert, a deterministic certification framework for graph classification that combines topology-blind hash partitioning with margin-explicit training and certificates for structure–feature perturbations. **(ii)** We derive robustness guarantees for count-based certification, its confidence-weighted extension, and the joint structure–feature threat model. The OR construction is pointwise no weaker than either individually sound branch. **(iii)** We evaluate SpecCert against representative certified defenses and analyze the robustness–efficiency trade-off, including the sensitivity of the confidence-weighted branch and MECT.

2 Background and Related Work

Adversarial attacks on graph classification. A growing body of work (Dai et al., 2018; Zügner et al., 2018; Tang et al., 2020; Chen et al., 2021; Mu et al., 2021; Zhang et al., 2021a; Wan et al., 2021; Zhang et al., 2023; Wang et al., 2022; 2023) demonstrates that graph classifiers are vulnerable to adversarial perturbations of either graph structure or node features. Attackers can induce misclassification by injecting or deleting a small number of edges, modifying node features, or executing both jointly. For instance, Wang et al. (Wang et al., 2022) leveraged bandit feedback to design black-box structure attacks with provable attack performance guarantees. Such vulnerabilities expose graph classifiers deployed in security and safety-critical applications, such as fraud detection and malware classification, to severe risk, motivating the development of defenses with formal robustness guarantees.

Empirical defenses against adversarial attacks on graphs. A complementary line of work develops empirical defenses (Chen et al., 2020; Zhao et al., 2021; Charpentier et al., 2023) that aim to harden a graph classifier against perturbed inputs through architectural modifications, robust aggregation, or adversarial training. While these defenses often improve robustness against the specific attacks they were designed for, they cannot guarantee their performance under arbitrary adversarial perturbations. Consequently, they perpetuate a cat-and-mouse dynamic between attackers and defenders. Mujkanovic et al. (2023) systematically demonstrated that essentially every published empirical defense for graph neural networks can be circumvented by adaptive attacks designed against the defense itself, leaving formal robustness guarantees as the only durable response.

Provable defenses against adversarial attacks on graphs. To provide formal robustness guarantees, certified defenses Bojchevski et al. (2020); Jin et al. (2020); Wang et al. (2020); Zhang et al. (2020); Scholten et al. (2022); Schuchardt et al. (2023); Xia et al. (2024) ensure the predicted label is provably preserved whenever the number of perturbed edges or feature-perturbed nodes is bounded. The dominant paradigm generalizes randomized smoothing (Lecuyer et al., 2019; Cohen et al., 2019; Levine & Feizi, 2019) from the image domain to graphs by aggregating predictions over many noisy or sub-sampled versions of a testing graph through a majority vote, with several recent extensions (Bojchevski et al., 2020; Wang et al., 2020; Zhang et al., 2021b; Scholten et al., 2022; Schuchardt et al., 2023; Xia et al., 2024) that exploit message passing structure, input-space symmetries, or hash-based partitioning to tighten the resulting bound. However, a fundamental structural gap remains across these paradigms: many existing deterministic methods rely primarily on discrete vote margins and do not explicitly exploit continuous per-sample predictive confidence, and they pair this with a cross-entropy training objective indifferent to the very margin being certified. This training–certification objective mismatch, compounded by probabilistic guarantees, feature-coverage gaps, and high inference costs in smoothing-based methods, fundamentally limits certified accuracy on harder benchmarks. To bridge this structural gap, we propose SpecCert: a deterministic certified defense that covers structure and feature perturbations jointly, consumes the confidence dimension that count-based certificates discard, and systematically unifies the training and inference objectives through a margin-explicit training scheme.

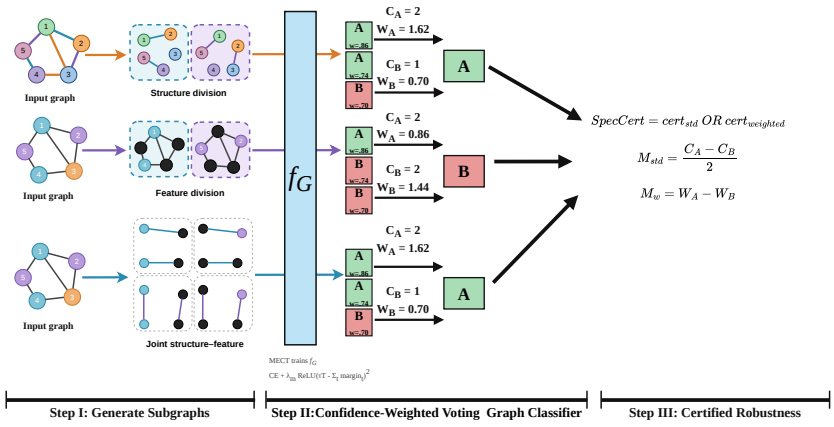


Figure 1: Overview of SpecCert. The black node features mean special values, e.g., zero.

3 METHODOLOGY

We introduce SpecCert, a deterministic certification framework for graph neural networks under bounded adversarial perturbations of structure and features. The key insight is that partitioning an input graph into T sub-views such that each perturbation is confined to one sub-view yields a provably robust ensemble vote, with guarantees degrading only linearly in the number of corrupted sub-views. This insight yields three components: (1) a topology-blind avalanche mixer partitioner with uniform statistics by design (Sec. 3.3); (2) a MECT that optimizes the continuous surrogate of the vote margin consumed by the certificate (Sec. 3.4); and (3) a confidence-weighted extension of the discrete majority-vote certificate, combined through a sound OR rule (Sec. 3.5). Guarantees are stated in Sec. 3.6 and proved in Appendix A.

3.1 Graph Classification Setup

Let $G = (V, E, \mathbf{X})$ denote an attributed, undirected graph with vertex set $V = \{v_1, \dots, v_n\}$, edge set $E \subseteq \binom{V}{2}$, and feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ whose i-th row $\mathbf{X}_{v_i}$ is the d-dimensional feature vector attached to node v_i. We use $\mathcal{Y} = \{1, 2, \dots, C\}$ for the set of class labels and write $y_G \in \mathcal{Y}$ for the ground-truth label of G. A base graph classifier is a function $f_\theta : \mathcal{G} \to \Delta^{C-1}$ mapping graphs to softmax vectors on the $(C-1)$-simplex, parameterised by weights θ and trained on a dataset $\mathcal{D}_{\text{tr}} = \{(G^{(i)}, y^{(i)})\}_i$. The objective of a certified defense is to produce, for every test graph G, a predicted label $\hat{y}$ together with a certified perturbation budget $r^\star(G) \in \mathbb{Z}_{\geq 0}$ such that every graph G' obtained from G by at most $r^\star(G)$ edge insertions/deletions (or, respectively, node-feature perturbations) is guaranteed to receive the same label as G.

3.2 Threat Model

We adopt the strongest evasion-time adversary consistent with a white-box certified-defense setting. The attacker has full knowledge of the base classifier f_θ and of the partitioning procedure. Given a test graph G, the attacker constructs an arbitrary perturbed graph G^{p} subject to $\|E \oplus E^{\text{p}}\|_0 \leq r_s$ (at most r_s edges inserted and/or deleted) and $|\{v \in V : \mathbf{X}^{\text{p}}_v \neq \mathbf{X}_v\}| \leq r_f$ (at most r_f nodes with arbitrarily modified features), where $\oplus$ denotes symmetric difference and $\|\cdot\|_0$ counts non-zero entries. The defender's goal is to return an $(\hat{y}, r_s^\star, r_f^\star)$ triple such that the prediction on G^{p} is provably $\hat{y}$ whenever $r_s \leq r_s^\star$ and $r_f \leq r_f^\star$, with a deterministic robustness guarantee, without Monte Carlo sampling of any smoothing distribution.

3.3 Topology-Blind Graph Partitioning

Let $\text{ID} : V \to \mathbb{Z}_{\geq 0}$ assign a stable identifier to every node, and for $e = \{u, v\} \in E$ let $\ell_e = \min(\text{ID}(u), \text{ID}(v))$ and $h_e = \max(\text{ID}(u), \text{ID}(v))$ so that the edge is storage-order invariant. Given $T \in \mathbb{N}$ and any deterministic hash $\mathcal{H} : \{0,1\}^\star \to \mathbb{Z}_{2^b}$ with full avalanche, SpecCert places e in group

$$g_e \;=\; \mathcal{H}\big(\,\ell_e \,\|\, h_e\,\big) \bmod T, \tag{1}$$

where $\|$ denotes a lossless byte encoding of (ℓ_e, h_e). Eq. 1 depends on node identifiers alone. Therefore, the hash is topology-blind so inserting an edge $e \notin E$ or deleting $e \in E$ modifies exactly one $E_t = \{e \in E : g_e = t\}$, yielding sub-views $G_t = (V, E_t, \mathbf{X})$ and the bound $M^{\text{p}} \leq r_s$, where $M^{\text{p}} = \sum_t \mathbb{I}[G_t \neq G_t^{\text{p}}]$ counts corrupted sub-views under an r_s-edge perturbation.

The certificate depends only on hash uniformity; any avalanche-complete hash provides statistically equivalent certificates. We use an avalanche integer mixer with lower per-edge cost than cryptographic hashes; empirical equivalence across hash families is verified in Fig. 5. Under sum graph pooling with sum neighbor aggregation, the forward pass reads only three fields of a sub-view: the node count, the symmetrized edge-index tensor $\mathbf{I}_t \in \mathbb{Z}^{2 \times 2|E_t|}$, and the feature matrix, so SpecCert realizes each G_t as a shallow alias of the parent graph with its edge-index tensor overwritten by the restriction to E_t, bit-exact for the forward pass while eliminating the per-sub-view graph-copy overhead.

The hashing framework extends to node-feature perturbations: with T_f feature groups, node v is assigned to $\mathcal{H}(\text{ID}(v)) \bmod T_f$ and its row $\mathbf{X}_v$ is retained in the t-th sub-view only if it falls in group t, other rows being replaced by a sentinel $\mathbf{x}_\varnothing$. Joint structure–feature division is the Cartesian product of the two partitions, yielding $T_s T_f$ sub-views $G_{s,u} = (V, E_s, \mathbf{X}^{(u)})$ arranged on a grid in which each edge perturbation corrupts one row and each feature perturbation corrupts one column (Theorem 3.4).

3.4 Margin-Explicit Certification Training

Plain cross-entropy training on sub-views optimises the per-sub-view loss $\mathcal{L}_{\text{CE}}(\theta) = \mathbb{E}_{(G,y),\,t}[-\log f_\theta(G_t)[y]\,]$, where $f_\theta(G_t)[y]$ denotes the softmax probability assigned to class y on sub-view t. But this is not the quantity consumed by the certificate. The certificate consumes the integer count-based vote margin

$$M(G) = \Big\lfloor \tfrac{1}{2}\big(N_{\hat{y}}(G) - \max_{c \neq \hat{y}}\big(N_c(G) - \mathbb{I}[\hat{y} < c]\big)\big)\Big\rfloor,\; N_c(G) = \textstyle\sum_{t=0}^{T-1} \mathbb{I}\big[\arg\max f_\theta(G_t) = c\big] \tag{2}$$

where $\hat{y} = \arg\max_c N_c(G)$ is the ensemble prediction, $\mathbb{I}[\cdot]$ is the indicator function, and the $\mathbb{I}[\hat{y} < c]$ term encodes a deterministic tie-break in favor of the smaller-indexed class. The margin in Eq. 2 is what determines the certified radius, and cross-entropy training is indifferent to it: a configuration with near perfect per-sub-view accuracy and $N_{\hat{y}} - N_{\text{runner-up}} = 2$ certifies out to radius one, while a configuration with weaker per sub-view accuracy but $N_{\hat{y}} - N_{\text{runner-up}} = 20$ certifies out to radius ten. The certificate rewards the latter; cross-entropy alone rewards neither specifically. This objective mismatch is the motivation for MECT.

The integer margin in Eq. 2 is non-differentiable in θ owing to its $\arg\max$. We therefore introduce the differentiable *expected vote margin* below by replacing each hard vote $\mathbb{I}[\arg\max f_\theta(G_t) = c]$ with its softmax-continuous counterpart $f_\theta(G_t)[c]$:

$$\mathcal{M}_{\text{exp}}(G, y; \theta) \;=\; \sum_{t=0}^{T-1} \Big(f_\theta(G_t)[y] \;-\; \max_{c \neq y} f_\theta(G_t)[c] \Big) \tag{3}$$

The expected margin in Eq. 3 is bounded above by T and attains this bound when every sub-view places its full softmax mass on the true class. It is differentiable in θ, making gradient-based optimization of a certification-oriented surrogate possible. However, because the discrete count margin depends on argmax transitions, increasing $\mathcal{M}_{\text{exp}}$ does not guarantee a monotonic increase in $M(G)$ for every parameter update. MECT should therefore be interpreted as a differentiable surrogate for the certification-relevant margin rather than an exact optimization of the discrete certificate. MECT exploits the last property by penalizing the squared shortfall of $\mathcal{M}_{\text{exp}}$ below a target τT, where $\tau \in (0, 1]$ is a hyperparameter we call the *margin fraction*:

$$\mathcal{L}_{\text{MECT}}(\theta) \;=\; \frac{1}{T^2}\, \mathbb{E}_{(G,y)\sim\mathcal{D}_{\text{tr}}} \Big[\big(\tau T - \mathcal{M}_{\text{exp}}(G, y; \theta) \big)_+^2 \Big] \tag{4}$$

with $(x)_+ = \max(x, 0)$ the positive-part operator and the T^{-2} normalization in Eq. 4 chosen so that $\mathcal{L}_{\text{MECT}}$ remains on a comparable scale to $\mathcal{L}_{\text{CE}}$ across choices of T. The full SpecCert training objective comprises cross-entropy with an optional consistency regulariser that shrinks the per-class softmax variance across sub-views of the same parent graph, together with $\mathcal{L}_{\text{MECT}}$:

$$\mathcal{L}_{\text{SpecCert}}(\theta) \;=\; \mathcal{L}_{\text{CE}}(\theta) \;+\; \lambda_c \cdot \mathcal{L}_{\text{cons}}(\theta) \;+\; \lambda_m \cdot \mathcal{L}_{\text{MECT}}(\theta) \tag{5}$$

where $\mathcal{L}_{\text{cons}}(\theta) = \mathbb{E}_{(G,y)}[\, C^{-1} \sum_{c=1}^{C} \text{Var}_{t\in[T]}\big(f_\theta(G_t)[c]\big)\,]$ denotes the average across classes of the per-class softmax variance over the T sub-views of a parent graph, and $\lambda_c, \lambda_m \geq 0$ are non-negative weights in Eq. 5. A standard label-smoothing regulariser is retained for the cross-entropy term to prevent the base classifier softmax from saturating at zero/one, without which the confidences κ_t introduced below would collapse to unity and the weighted certificate would degenerate.

3.5 Confidence-Weighted Dual-Margin Certification

The count-based margin $M(G)$ of Eq. 2 treats every sub-view as a unit-weight voter. The softmax output also provides a confidence value that can be used to form an additional, independently sound certificate. For a clean or perturbed graph G with sub-views $\{G_t\}$, let $\mathbf{p}_t = f_\theta(G_t) \in \Delta^{C-1}$, $q_t = \arg\max_c \mathbf{p}_t[c]$, and $\kappa_t = \mathbf{p}_t[q_t] \in [1/C, 1]$. Define the confidence-weighted tally

$$W_c(G) = \sum_{t=0}^{T-1} \kappa_t\, \mathbb{I}[q_t = c], \qquad c \in \mathcal{Y}, \tag{6}$$

and the confidence-weighted prediction $\tilde{y}(G) = \arg\max_c W_c(G)$ with weighted margin $w(G) = W_{\tilde{y}}(G) - \max_{c\neq\tilde{y}} W_c(G)$. If $\kappa_{(1)} \geq \cdots \geq \kappa_{(N_{\tilde{y}})}$ are the confidences of the sub-views voting for $\tilde{y}$, let $S_r = \sum_{i=1}^{r} \kappa_{(i)}$ for $0 \leq r \leq N_{\tilde{y}}$, with $S_0 = 0$.

A corruption of one sub-view can reduce the weighted gap by at most $1 + \kappa_t$ when that sub-view originally voted for $\tilde{y}$. Consequently, a sufficient condition for preserving the weighted prediction under at most r corrupted sub-views is $w(G) > S_r + r$. This condition is valid without a calibration assumption: calibration affects the tightness of the bound, not its soundness. Overconfident incorrect sub-views or underconfident correct sub-views can nevertheless make the weighted certificate conservative or uninformative in practice.

We combine the two individually sound predictor–certificate pairs using a logical OR:

$$\begin{aligned}\mathrm{cert}_{\mathrm{SpecCert}}(G, r) = &\big[\hat{y}(G) = y_G \ \wedge \ M(G) \geq r\big] \\ &\vee \ \big[\tilde{y}(G) = y_G \ \wedge \ R_{\mathrm{wt}}(G) \geq r\big],\end{aligned} \tag{7}$$

where $R_{\mathrm{wt}}(G) = \max\{r \in \{0, \ldots, N_{\tilde{y}}\} : w(G) > S_r + r\}$. The first disjunct uses the count-based predictor $\hat{y}$ and count certificate M; the second uses the distinct confidence-weighted predictor $\tilde{y}$ and weighted certificate R_{wt}. Each branch certifies all perturbations of size at most r whenever its certificate is at least r. The ground-truth label appears in Eq. 7 only because certified accuracy is an evaluation metric; it is not used to select a prediction at inference time. Lemma 3.3 establishes soundness of the OR construction, and the resulting certified accuracy is never lower than that of either branch.

When multiple training seeds are available, averaging pre-softmax logits sub-view-wise preserves soundness because all models read the same perturbed sub-view. The ensemble should be fixed before evaluating the test set; test labels must not be used to select among seeds or ensembles.

3.6 Provable Robustness Guarantees

We now state the certificates that SpecCert consumes. Theorem 3.1 is the discrete count-based bound; Theorem 3.2 is the continuous confidence-weighted bound. Lemma 3.3 promotes their individual soundness into the joint soundness of the dual-OR rule in Eq. 7, and Theorem 3.4 handles the joint structure–feature threat model. All proofs are deferred to Appendix A.

Theorem 3.1 (Standard count-based certificate)**.** *Let G be a test graph with sub-views $\{G_t\}$ induced by structure division, $\hat{y} = \arg\max_c N_c(G)$, and let G^{p} be any graph obtained from G by at most r_s edge insertions or deletions. Writing $M^{\mathrm{p}} = \sum_t \mathbb{I}[G_t \neq G_t^{\mathrm{p}}]$ for the number of corrupted sub-views, the ensemble prediction is preserved, i.e. $\arg\max_c N_c(G^{\mathrm{p}}) = \hat{y}$, whenever $M^{\mathrm{p}} \leq M(G)$.*

The topology-blind hash of Eq. 1 gives $M^{\mathrm{p}} \leq r_s$. See Appendix A.1 for details.

Theorem 3.2 (Confidence-weighted certificate)**.** *Adopt the notation of Theorem 3.1, let $\tilde{y} = \arg\max_c W_c(G)$, and let $\kappa_{(1)} \geq \kappa_{(2)} \geq \cdots \geq \kappa_{(N_{\tilde{y}})}$ denote the confidences of the $N_{\tilde{y}}$ sub-views that voted for $\tilde{y}$, sorted in non-increasing order. Writing $S_r = \sum_{i=1}^{r} \kappa_{(i)}$ for $r \leq N_{\tilde{y}}$, with $S_0 = 0$, the confidence-weighted prediction on G^{p} equals $\tilde{y}$ whenever $M^{\mathrm{p}} \leq R_{\mathrm{wt}}(G)$.*

A complete proof is given in Appendix A.2.

Lemma 3.3 (Soundness of the dual certificate)**.** *Let $(g_1, \mathcal{C}_1)$ and $(g_2, \mathcal{C}_2)$ be two (predictor, certificate) pairs, each individually sound for the same threat model in the sense that $g_i(G^{\mathrm{p}}) = g_i(G)$ whenever the perturbation satisfies $\mathcal{C}_i(G) \geq r$. Then the dual certificate in Eq. 7 is sound: whenever it evaluates to one, on every admissible perturbation of size at most r, at least one of g_1, g_2 outputs the correct label y_G.*

Moreover, the certified accuracy under the dual certificate is at least that of either certificate alone. The full proof is deferred to Appendix A.3.

Theorem 3.4 (Joint structure–feature certificate)**.** *Under structure–feature division with T_s structure groups and T_f feature groups, let G^{p} be obtained from G by at most r_s edge perturbations and at most r_f node-feature perturbations. Substituting this bound into Theorem 3.1 yields the joint certified region $\mathcal{R}_{\mathrm{joint}}(G) = \{(r_s, r_f) \in \mathbb{Z}^2_{\geq 0} : r_s T_f + r_f T_s \leq M(G)\}$.*

Every edge perturbation corrupts exactly one grid row of T_f sub-views, and every feature perturbation corrupts exactly one grid column of T_s sub-views; hence $M^{\mathrm{p}} \leq r_s T_f + r_f T_s$. A detailed proof is provided in Appendix A.4.

Relationship between the two certificates. Theorem 3.1 is tight for the discrete majority-vote rule when only vote counts are available. The confidence-weighted branch uses additional information and can, in principle, yield a different certificate, but its practical tightness depends on the sub-view confidence distribution. The OR construction guarantees that the reported certified accuracy is no worse than either branch.

4 EVALUATION

4.1 EXPERIMENTAL SETUP

Datasets: We use eight standard benchmark datasets for graph classification: DBLP (Pan et al., 2013), DD (Dobson & Doig, 2003), ENZYMES (Hu et al., 2020), MUTAG (Debnath et al., 1991), NCI1 (Wale et al., 2006), PROTEINS (Borgwardt et al., 2005), REDDIT-B (Yanardag & Vishwanathan, 2015), and COLLAB (Yanardag & Vishwanathan, 2015). Table 2 in Appendix B reports per-dataset statistics. Following standard practice for certified-defense evaluations on TU benchmarks, we sample two-thirds of the graphs in each dataset for training and use the remaining third for testing.

Base graph classifier: We adopt GIN (Xu et al., 2019) as the base graph classifier unless stated otherwise, using its publicly available implementation. Training employs the Adam optimiser with a learning rate of 0.001, a batch size of 32, and runs for $1{,}000$ epochs.

Compared methods. We compare SpecCert against four certified defense baselines: GNNCert Xia et al. (2024), Zhang et al. Zhang et al. (2021b), Wang et al. Wang et al. (2020), and Bojchevski et al. Bojchevski et al. (2020). GNNCert Xia et al. (2024) is a deterministic certified defense using hash-based partitioning and majority-vote certification. The remaining baselines are Monte Carlo smoothing methods. Zhang et al. Zhang et al. (2021b) extends randomised ablation Levine & Feizi (2019) to graphs by creating $S_z = 1{,}000$ subsampled graphs per test instance (edge retention fraction $\tau = 10\%$) and taking a majority vote. Wang et al. Wang et al. (2020) and Bojchevski et al. Bojchevski et al. (2020) both construct $S_w = S_b = 1{,}000$ noisy graphs per test instance and aggregate via majority vote, differing only in their perturbation strategy: Wang et al. flip each node-pair connection with probability $\beta = 0.3$, while Bojchevski et al. independently insert or delete each edge with probabilities $p^+ = p^- = 0.3$. All three Monte Carlo baselines compute certified perturbation sizes with confidence parameter $\alpha = 0.001$, so their guarantees hold with probability $(1 - \alpha)$ only. All five methods share the same GIN base classifier and identical train/test splits for a fair comparison.

Evaluation metric: Following previous studies Zhang et al. (2021b); Wang et al. (2020); Bojchevski et al. (2020); Xia et al. (2024), we use certified accuracy as the evaluation metric. Given a perturbation size r and a testing dataset $\mathcal{D}_{\text{test}}$, certified accuracy is defined as the fraction of testing graphs that (i) are correctly classified by the ensemble, and (ii) have a certified perturbation size no smaller than r (or, in the joint setting, at most r_s edges and r_f node-features).

Parameter setting: SpecCert exposes the same partitioning hyperparameters as the GNNCert Xia et al. (2024): a hash function $\mathcal{H}$, a structure-group count T_s, and a feature-group count T_f. Unless stated otherwise we use $\mathcal{H} =$ MurmurHash3 (Sec. 3.3) and $T_s = T_f = 30$. The impact of all three parameters, together with the impact of the MECT weight λ_m and the dual-certification mechanism, is ablated in Appendix D.

4.2 EXPERIMENTAL RESULTS

SpecCert provides competitive certified robustness while maintaining deterministic guarantees. Figure 2 compares the certified accuracy of SpecCert against all four baselines on the eight benchmark datasets. The three Monte Carlo baselines collapse to zero certified accuracy at $r \approx 6$–12 on every dataset, because their certified radius decays geometrically under a $(1 - \alpha)$ confidence requirement. GNNCert (Xia et al., 2024) provides a guarantee out to $r \approx 14$, but exhibits a sharp drop on harder datasets, where the count-based vote margin saturates. SpecCert maintains competitive certified accuracy, particularly at larger perturbation budgets on several of the harder datasets. The deterministic partitioner, MECT ablation, and confidence-weighted decomposition are evaluated separately in Appendix D.

On MUTAG, SpecCert achieves lower test-time certification cost than all evaluated baselines. Table 1 compares per-method training and testing wall-clock costs on MUTAG. SpecCert certifies the entire test set in 1.5 seconds, 5 times faster than GNNCert Xia et al. (2024) and substantially faster than the evaluated Monte Carlo baselines Bojchevski et al. (2020); Zhang et al. (2021b); Wang et al. (2020). The test-time gap over the principal baseline is the direct consequence of the avalanche-mixer partitioner of Sec. 3.3, which replaces the cryptographic hashing pipeline of Xia et al. Xia et al. (2024) by a vectorized integer mixer with comparable uniformity and substantially lower per-edge cost. Multi-dataset costs (Table 3 in Appendix D) reproduce the same pattern on PROTEINS, ENZYMES,

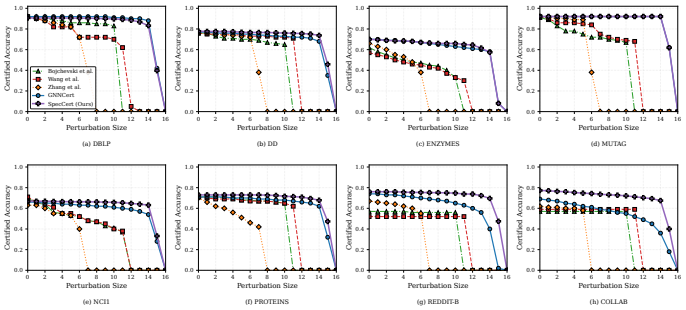


Figure 2: Comparing the certified accuracy of our SpecCert with existing certified defenses.

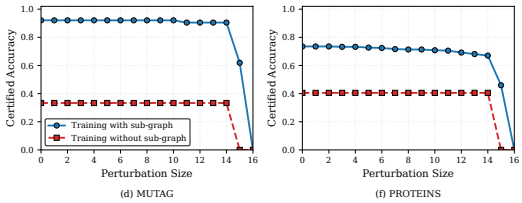


Figure 3: Comparing the certified accuracy of our SpecCert when the base graph classifier is trained with/without sub-graphs.

Table 1: Comparing the computation costs of SpecCert and existing defenses on MUTAG.

Method	**Training (s)**	**Testing (s)**
Bojchevski et al. (2020)	1455	38
Wang et al. (2020)	1344	48
Zhang et al. (2021b)	1136	25
GNNCert (2024)	773	8
SpecCert (Ours)	1402	1.5

and NCI1. The higher training cost results from the optional multi-seed ensemble, while certification requires only a single pass after training.

Training the base classifier on hashed sub-graphs is essential, not optional. Figure 3 compares the certified-accuracy curves of SpecCert when the base classifier is trained on hashed sub-graphs versus on full training graphs. On MUTAG and PROTEINS, the certified accuracy at $r = 0$ increases from 0.333 to 0.921 and from 0.405 to 0.735, respectively, with the gap maintained at every radius. This improvement is caused by a distributional mismatch: a base classifier trained on full graphs learns patterns spanning the entire edge set, which the hash partition fragments at test time. Sub-graph training aligns the training and inference distributions. SpecCert and GNNCert Xia et al. (2024) share this distributional-alignment property, but SpecCert further combines it with MECT, enforcing both correct prediction and margin-based voting rather than relying only on cross-entropy.

Increasing the number of structure groups T_s and feature groups T_f generally extends the observed certified-radius range, with diminishing returns past $T = 30$ in the evaluated settings. Figures 6 and 7 in Appendix D sweep T_s and T_f over $\{10, 30, 50\}$ across the benchmark suite. The qualitative behavior is the same in both cases: $T = 10$ produces an early cliff in certified accuracy at small r, while $T = 30$ and $T = 50$ maintain the curve out to nearly the maximum radius. The reason is that the certified radius scales linearly with the vote margin (Theorem 3.1), and the symmetric treatment of structure and feature partitions in Theorem 3.4 means the same mechanism controls both cases. The flattening between $T = 30$ and $T = 50$ confirms that the $T_s = T_f = 30$ default sits at the elbow of the cost–robustness trade-off.

Figure 5 in Appendix D shows that MurmurHash3, MD5, SHA-1, and SHA-256 produce nearly identical certified accuracy, showing similar certified-accuracy behavior across the evaluated hash

families. SpecCert uses MurmurHash3 because its lower per-edge cost improves efficiency without changing certification behavior.

Replacing GIN with GCN or GAT (Figure 8, Appendix D) leaves certified accuracy essentially unchanged, suggesting robustness across the evaluated backbones because the guarantee depends on the partitioning scheme and certification objective rather than on a particular backbone.

SpecCert provides a valid sufficient joint structure–feature certified region. Unlike baselines limited primarily to structure perturbations, SpecCert covers simultaneous structure and node-feature budgets under the region in Theorem 3.4; the region is conservative in general because overlapping row–column corruptions may be counted more than once. Figure 9 in Appendix D visualises the resulting certified regions.

The dual-certification mechanism is a sound optional extension. Figure 10 decomposes the two disjuncts. The OR curve is, by construction, the upper envelope of the count-based and confidence-weighted certified sets and is therefore never worse than either branch. In our ablation study, the weighted branch is generally more conservative than the count-based branch, so the OR curve largely coincides with the standard curve. We therefore do not attribute the main empirical gains to the weighted branch; its value is the additional formally valid certificate and the possibility of improvement under more favorable confidence distributions. The current aggregate ablation does not separately report the subset of graphs for which $\tilde{y} \neq \hat{y}$; consequently, we do not attribute the observed certified-accuracy behavior specifically to predictor disagreement.

MECT provides a margin-explicit training signal. Figure 11 sweeps $\lambda_m \in \{0, 0.5, 1, 2\}$ on four benchmarks, where $\lambda_m = 0$ reduces to plain cross-entropy training. MECT gives the clearest gains on the harder PROTEINS benchmark, while making little difference on smaller benchmarks where cross-entropy already produces a large vote margin. See Appendix D for the full ablation.

The advantage of deterministic certification is most visible under matched compute budgets. Existing Monte Carlo baselines Bojchevski et al. (2020); Wang et al. (2020); Zhang et al. (2021b) compute their certificates from 1,000 noisy graphs per test instance, while SpecCert and GNNCert Xia et al. (2024) both produce their certificates from only 30 sub-views. Under a matched-compute budget where the Monte Carlo baselines are downsampled to 30 noisy graphs per test instance, Figure 4 in Appendix C shows that all three Monte Carlo baselines collapse to zero certified accuracy at $r \approx 4$, while both deterministic methods remain unaffected. These results show that the certified-accuracy advantage of SpecCert over the evaluated Monte Carlo baselines persists under matched compute and is not explained solely by the larger original sampling budget.

5 Conclusion

We propose SpecCert, a deterministic certification framework that combines topology-blind partitioning with a margin-explicit training surrogate and certificates for bounded structure and feature perturbations. The partitioner provides a low-overhead deterministic decomposition, while MECT is designed to encourage the margins consumed by certification. We also derive a confidence-weighted certificate and show that its OR combination with the count-based certificate is sound and never weaker at the level of certified accuracy. The current experiments show that the weighted branch can be conservative, motivating future work on confidence modeling and calibration. Other future directions include richer attacks, such as node or subgraph injection, on larger real-world graphs.

Limitations. SpecCert is evaluated on standard benchmarks and bounded edge/feature attacks; larger real-world graphs and node/subgraph injection remain future work. The weighted certificate is formally sound without calibration, but its tightness depends on the quality and distribution of the softmax confidences. The current ablations do not show an empirical improvement from the weighted branch, so we present it as an optional extension rather than a universally superior certificate. Finally, MECT is a differentiable surrogate and does not guarantee a monotonic increase in the discrete vote margin after every optimization step. We do not separately isolate the effects of label smoothing and the consistency regularizer on clean accuracy and certified accuracy; their interaction with MECT and confidence tightness remains an empirical direction for future work.

Reproducibility Statement. All datasets used in this work are public graph-classification benchmarks. The released implementation contains the data-loading and preprocessing routines, the topology-blind structure and feature partitioners, the GIN-based models, the MECT training objective, the standard and confidence-weighted certification routines, and the scripts used to generate the reported tables and figures. The default experiments use the two-thirds/one-third train/test split described in Appendix B, Adam with learning rate 0.001, batch size 32, $1{,}000$ training epochs, label smoothing, and $T_s = T_f = 30$ unless otherwise stated. The released source also records the evaluated ablations, dataset-specific settings, and computational configurations. Experiments were conducted with Python 3.10.12, PyTorch 2.0.1, CUDA 11.8, NumPy 1.26.4, SciPy 1.15.3, and scikit-learn 1.7.2 on the hardware configuration reported in Appendix B. The public code repository associated with this submission contains the generated CSV, JSON, and NPY artifacts used for the reported plots and tables. In particular, for reproducibility, the complete pointwise certified-accuracy values underlying Fig. 2 are provided in the released result files. The figures can therefore be regenerated without retraining when the corresponding result artifacts are available; full end-to-end regeneration additionally requires the public datasets, the released implementation, and the stated software environment. The theoretical certificates are deterministic conditional on the stated partitioning and threat model, whereas training results can vary with random initialization and hardware-level numerical differences. We distinguish these sources of variation from the deterministic certification guarantee.

Ethics Statement. This work uses public graph-classification benchmarks and does not collect, process, or release personal, medical, or otherwise sensitive individual-level data. The motivating applications include security-sensitive settings such as fraud and malware detection, but the experiments are limited to robustness certification and do not provide instructions for compromising deployed systems. The released code is intended for defensive research and should be used in accordance with the licenses and terms of the underlying datasets and software dependencies. The authors are responsible for the claims, analyses, and artifacts presented in this paper.

AI Use Statement. Generative AI tools were used to assist with language editing, organization, and consistency checks during manuscript preparation. They were not used to generate experimental results, select reported numerical values, establish theorems, or replace verification of the proofs and implementation. The authors reviewed and verified the final mathematical statements, experimental descriptions, references, and reported results, and take full responsibility for the contents of this paper.

REFERENCES

Aleksandar Bojchevski, Johannes Gasteiger, and Stephan Günnemann. Efficient robustness certificates for discrete data: Sparsity-aware randomized smoothing for graphs, images and more. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1003–1013. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/bojchevski20a.html.

Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.

Bertrand Charpentier, Simon Geisler, Lukas Gosch, Stephan Günnemann, Daniel Sturm, and Daniel Zügner. Adversarial training for graph neural networks: Pitfalls, solutions, and new directions. *Advances in Neural Information Processing Systems 36*, 2023. URL https://api.semanticscholar.org/CorpusID:286293196.

Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23 6:1241–1250, 2018. URL https://api.semanticscholar.org/CorpusID:46792472.

Jinyin Chen, Huiling Xu, Jinhuan Wang, Qi Xuan, and Xuhong Zhang. Adversarial detection on graph structured data. *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, 2020. URL https://api.semanticscholar.org/CorpusID:226238740.

Jinyin Chen, Dunjie Zhang, Zhaoyan Ming, Kejie Huang, Wenrong Jiang, and Chen Cui. Graphattacker: A general multi-task graph attack framework. *IEEE Transactions on Network Science and Engineering*, 9: 577–595, 2021. URL https://api.semanticscholar.org/CorpusID:231632159.

Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, 2019. URL https://arxiv.org/abs/1902.02918.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, L. Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:46968933.

Asim Kumar Debnath, Rosa L. Lopez de Compadre, Gargi Debnath, Alan J. Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34 2:786–97, 1991. URL https://api.semanticscholar.org/CorpusID:19990980.

Paul D. Dobson and Andrew J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330 4:771–83, 2003. URL https://api.semanticscholar.org/CorpusID:5719990.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *nature*, 585(7825):357–362, 2020.

Mehadi Hassen and Philip K. Chan. Scalable function call graph-based malware classification. *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, 2017. URL https://api.semanticscholar.org/CorpusID:7375363.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/fb60d411a5c5b72b2e7d3527cfc84fd0-Paper.pdf.

Hongwei Jin, Zhan Shi, Venkata Jaya Shankar Ashish Peruri, and Xinhua Zhang. Certified robustness of graph convolution networks for graph classification under topological attacks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8463–8474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/609a199881ca4ba9c95688235cd6ac5c-Paper.pdf.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv e-prints*, art. arXiv:1609.02907, September 2016. doi: 10.48550/arXiv.1609.02907.

Deguang Kong and Guanhua Yan. Discriminant malware distance learning on structural information for automated malware classification. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pp. 1357–1365, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450321747. doi: 10.1145/2487575.2488219. URL https://doi.org/10.1145/2487575.2488219.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672, 2019. doi: 10.1109/SP.2019.00044.

Alexander Levine and Soheil Feizi. Robustness certificates for sparse adversarial attacks by randomized ablation. *CoRR*, abs/1911.09272, 2019. URL http://arxiv.org/abs/1911.09272.

Junying Li, Deng Cai, and Xiaofei He. Learning graph-level representation for drug discovery. *CoRR*, abs/1709.03741, 2017. URL http://arxiv.org/abs/1709.03741.

Jiaming Mu, Binghui Wang, Qi Li, Kun Sun, Mingwei Xu, and Zhuotao Liu. A hard label black-box adversarial attack against graph neural networks. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021. URL https://api.semanticscholar.org/CorpusID:237267072.

Felix Mujkanovic, Simon Geisler, Stephan Gunnemann, and Aleksandar Bojchevski. Are defenses for graph neural networks robust? *ArXiv*, abs/2301.13694, 2023. URL https://api.semanticscholar.org/CorpusID:256416081.

John R. Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming. *2008 IEEE Hot Chips 20 Symposium (HCS)*, pp. 40–53, 2008. URL https://api.semanticscholar.org/CorpusID:7917593.

Shirui Pan, Xingquan Zhu, Chengqi Zhang, and Philip S. Yu. Graph stream classification using labeled and unlabeled graphs. *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 398–409, 2013. URL https://api.semanticscholar.org/CorpusID:16960812.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Yan Scholten, Jan Schuchardt, Simon Geisler, Aleksandar Bojchevski, and Stephan Günnemann. Randomized message-interception smoothing: Gray-box certificates for graph neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=t0VbBTw-o8.

Jan Schuchardt, Aleksandar Bojchevski, Johannes Gasteiger, and Stephan Günnemann. Collective robustness certificates: Exploiting interdependence in graph neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=ULQdiUTHe3y.

Jan Schuchardt, Aleksandar Bojchevski, Johannes Gasteiger, and Stephan Günnemann. Collective robustness certificates: Exploiting interdependence in graph neural networks. *ArXiv*, abs/2302.02829, 2023. URL https://api.semanticscholar.org/CorpusID:235613304.

Haoteng Tang, Guixiang Ma, Yurong Chen, Lei Guo, Wei Wang, Bo Zeng, and Liang Zhan. Adversarial attack on hierarchical graph pooling neural networks. *arXiv preprint arXiv:2005.11560*, 2020.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *ArXiv*, abs/1710.10903, 2017. URL https://api.semanticscholar.org/CorpusID:3292002.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Nikil Wale, Ian A. Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14:347–375, 2006. URL https://api.semanticscholar.org/CorpusID:2596211.

Xingchen Wan, Henry Kenlay, Robin Ru, Arno Blaas, Michael A Osborne, and Xiaowen Dong. Adversarial attacks on graph classifiers via bayesian optimisation. *Advances in Neural Information Processing Systems*, 34:6983–6996, 2021.

Binghui Wang, Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of graph neural networks against adversarial structural perturbation. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2020. URL https://api.semanticscholar.org/CorpusID:221293271.

Binghui Wang, Youqin Li, and Pan Zhou. Bandits for structure perturbation-based black-box attacks to graph neural networks with theoretical guarantees. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13369–13377, 2022. URL https://api.semanticscholar.org/CorpusID:248571655.

Binghui Wang, Meng Pang, and Yun Dong. Turning strengths into weaknesses: A certified robustness inspired attack framework against graph neural networks. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16394–16403, 2023. URL https://api.semanticscholar.org/CorpusID:257496251.

Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I. Weidele, Claudio Bellei, Tom Robinson, and Charles E. Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *ArXiv*, abs/1908.02591, 2019. URL https://api.semanticscholar.org/CorpusID:199472611.

Zaishuo Xia, Han Yang, Binghui Wang, and Jinyuan Jia. Gnncert: Deterministic certification of graph neural networks against adversarial perturbations. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 49215–49232, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/d69747e04b73cf719292c22cfcea45ec-Paper-Conference.pdf.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Jiaqi Yan, Guanhua Yan, and Dong Jin. Classifying malware represented as control flow graphs using deep graph convolutional neural network. *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 52–63, 2019. URL https://api.semanticscholar.org/CorpusID:201621232.

Pinar Yanardag and S. V. N. Vishwanathan. Deep graph kernels. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015. URL https://api.semanticscholar.org/CorpusID:207227372.

He Zhang, Bang Wu, Xiangwen Yang, Chuan Zhou, Shuo Wang, Xingliang Yuan, and Shirui Pan. Projective ranking: A transferable evasion attack method on graph neural networks. In Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (eds.), *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pp. 3617–3621. ACM, 2021a. doi: 10.1145/3459637.3482161. URL https://doi.org/10.1145/3459637.3482161.

He Zhang, Xingliang Yuan, Chuan Zhou, and Shirui Pan. Projective ranking-based gnn evasion attacks. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8402–8416, 2023. doi: 10.1109/TKDE.2022.3219209.

Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, 2020. URL https://api.semanticscholar.org/CorpusID:219955744.

Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. In *Proceedings of the 26th ACM symposium on access control models and technologies*, pp. 15–26, 2021b.

Xin Zhao, Zeru Zhang, Zijie Zhang, Lingfei Wu, Jiayin Jin, Yang Zhou, Ruoming Jin, Dejing Dou, and Da Yan. Expressive 1-lipschitz neural networks for robust multiple graph learning against adversarial attacks. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12719–12735. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zhao21e.html.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018. URL https://api.semanticscholar.org/CorpusID:29169801.

A Theoretical Guarantees and Proofs

This appendix collects the proofs of Theorems 3.1, 3.2 and 3.4 and of Lemma 3.3 stated in Sec. 3.6. We retain the notation introduced there and continue to number every displayed equation so that it may be cited by later parts of the proof.

A.1 Proof of Theorem 3.1 (Standard count-based certificate)

We provide the formal proof for Theorem 3.1 by first establishing that adversarial perturbations are geometrically confined by the topology-blind partitioner, and then showing that this confinement is sufficient to preserve the ensemble argmax. Let G be a test graph and G^{p} a perturbed graph obtained from G by at most r_s edge insertions or deletions, and denote by $\{G_t\}_{t=0}^{T-1}$ and $\{G_t^{\mathrm{p}}\}_{t=0}^{T-1}$ the structure sub-views of G and G^{p} induced by Eq. 1.

Under Eq. 1, each edge e has a deterministic group assignment g_e, so inserting an edge $e' \notin E$ adds it only to $E_{g_{e'}}$, and deleting an edge $e \in E$ removes it only from E_{g_e}. Any single edge perturbation, therefore, alters the edge set of exactly one sub-view, leaving the remaining $T-1$ sub-views bit-identical to their clean counterparts. Iterating over at most r_s independent perturbations gives $M^{\mathrm{p}} = \sum_{t=0}^{T-1} \mathbb{I}[G_t \neq G_t^{\mathrm{p}}] \leq r_s$, which is the first claim of Theorem 3.1.

We now show that $M^{\mathrm{p}} \leq M(G)$ is sufficient to preserve the tie-broken argmax of the count tallies. Defining $\delta_c = N_c^{\mathrm{p}} - N_c$ for each $c \in \mathcal{Y}$, only the M^{p} corrupted sub-views can cause their class indicator to flip; the remaining $T - M^{\mathrm{p}}$ sub-views contribute identically to N_c and N_c^{p}, and each corrupted sub-view can move at most one unit of vote mass between any pair of classes. Hence $|\delta_c| \leq M^{\mathrm{p}}$ for every c, giving the two-sided bound

$$N_c - M^{\mathrm{p}} \;\leq\; N_c^{\mathrm{p}} \;\leq\; N_c + M^{\mathrm{p}}, \qquad \forall c \in \mathcal{Y} \tag{8}$$

Recall that the deterministic tie-break prefers $\hat{y}$ to any competitor c whenever either $N_{\hat{y}} > N_c$, or $N_{\hat{y}} = N_c$ and $\hat{y} < c$, which is equivalent to the integer inequality $N_{\hat{y}} \geq N_c - \mathbb{I}[\hat{y} < c] + 1$ holding for every $c \in \mathcal{Y} \setminus \{\hat{y}\}$. Applying the two-sided bound in Eq. 8,

$$N_{\hat{y}}^{\mathrm{p}} - N_c^{\mathrm{p}} \;\geq\; \big(N_{\hat{y}} - M^{\mathrm{p}}\big) - \big(N_c + M^{\mathrm{p}}\big) \;=\; \big(N_{\hat{y}} - N_c\big) - 2M^{\mathrm{p}} \tag{9}$$

For the tie-break inequality to hold on G^{p}, we require $N_{\hat{y}}^{\mathrm{p}} - N_c^{\mathrm{p}} \geq 1 - \mathbb{I}[\hat{y} < c]$, which combined with Eq. 9 yields the sufficient condition $2M^{\mathrm{p}} \leq N_{\hat{y}} - N_c - 1 + \mathbb{I}[\hat{y} < c]$. Taking the minimum over $c \neq \hat{y}$, the largest M^{p} for which this holds universally is

$$M_{\max}^{\mathrm{p}} \;=\; \left\lfloor \tfrac{1}{2}\left(N_{\hat{y}} \;-\; \max_{c \neq \hat{y}} \big(N_c - \mathbb{I}[\hat{y} < c] \big) \;-\; 1 \;+\; \varepsilon \right) \right\rfloor \tag{10}$$

where $\varepsilon \in \{0, 1\}$ accounts for the parity of $N_{\hat{y}} - \max_{c \neq \hat{y}}(N_c - \mathbb{I}[\hat{y} < c])$. A straightforward parity case analysis reduces Eq. 10 to the form $M_{\max}^{\mathrm{p}} = M(G)$ used in the definition of $M(G)$ in Eq. 2. Hence for any $M^{\mathrm{p}} \leq M(G)$, the tie-break inequality holds on G^{p} for every competing class, implying $\arg\max_c N_c^{\mathrm{p}} = \hat{y}$. This completes the proof of Theorem 3.1.

A.2 Proof of Theorem 3.2 (Confidence-weighted certificate)

We detail the formal proof for the confidence-weighted certification bound introduced in Theorem 3.2 by constructing, for every corrupted-sub-view budget M^{p}, the worst-case adversarial reallocation of probability mass and showing that even under this adversary the weighted prediction on G^{p} equals $\tilde{y}$ whenever $M^{\mathrm{p}} \leq R_{\mathrm{wt}}(G)$.

Fix any competing class $c \neq \tilde{y}$. For each sub-view t, let

$$\Delta_t(c) \;=\; \kappa_t^{\mathrm{p}}\, \mathbb{I}[q_t^{\mathrm{p}} = c] \;-\; \kappa_t\, \mathbb{I}[q_t = c] \tag{11}$$

denote the change in the contribution of sub-view t to the weighted tally W_c under the perturbation, so that the per-sub-view contribution to the gap $W_{\tilde{y}} - W_c$ changes by $\Delta_t(\tilde{y}) - \Delta_t(c)$. Because $\kappa_t^{\mathrm{p}}, \kappa_t \in [1/C, 1]$, Eq. 11 vanishes for every uncorrupted sub-view, since $(\kappa_t^{\mathrm{p}}, q_t^{\mathrm{p}}) = (\kappa_t, q_t)$ there.

For a corrupted sub-view t, the adversary freely chooses q_t^{p} and can push κ_t^{p} to any value in $[1/C, 1]$; the most hurtful choice for $W_{\tilde{y}} - W_c$ depends on the original vote q_t, and a case analysis gives

$$\min_{\kappa_t^{\mathrm{p}},\, q_t^{\mathrm{p}}} \Big[\, \Delta_t(\tilde{y}) - \Delta_t(c) \,\Big] \;=\; \begin{cases} -(1+\kappa_t), & q_t = \tilde{y}, \\ -(1-\kappa_t), & q_t = c, \\ -1, & q_t \notin \{\tilde{y}, c\}, \end{cases} \tag{12}$$

with the minima attained by the choice $q_t^{\mathrm{p}} = c$ and $\kappa_t^{\mathrm{p}} = 1$ in all three branches of Eq. 12. The first branch is strictly the most damaging: a corrupted sub-view that originally voted for $\tilde{y}$ with high confidence κ_t costs the gap $1 + \kappa_t$, whereas the second and third branches each cost at most 1.

Since the adversary has M^{p} corruptions at its disposal, an optimal adversary spends them on the sub-views that originally voted for $\tilde{y}$, prioritising the highest-confidence ones first; if $M^{\mathrm{p}} > N_{\tilde{y}}$, additional corruptions fall into the weaker second or third branch and cost the gap at most one each. Under the non-trivial regime $M^{\mathrm{p}} \le N_{\tilde{y}}$, the total worst-case loss to $W_{\tilde{y}} - W_c$ is therefore $\sum_{i=1}^{M^{\mathrm{p}}} (1 + \kappa_{(i)}) = M^{\mathrm{p}} + S_{M^{\mathrm{p}}}$, where $\kappa_{(i)}$ and S_r are as in the statement of Theorem 3.2. Since c is either the runner-up with gap $W_{\tilde{y}} - W_c = w(G)$ or a non-runner-up with gap at least $w(G)$, combining with Eq. 12 yields

$$W_{\tilde{y}}^{\mathrm{p}} - W_c^{\mathrm{p}} \;\ge\; w(G) \;-\; M^{\mathrm{p}} \;-\; S_{M^{\mathrm{p}}} \tag{13}$$

By the definition of $R_{\mathrm{wt}}(G)$ in the main text (Sec. 3.5), we have $w(G) > S_{R_{\mathrm{wt}}(G)} + R_{\mathrm{wt}}(G)$. Since $M^{\mathrm{p}} \le R_{\mathrm{wt}}(G)$ and the function $r \mapsto S_r + r$ is monotone non-decreasing, $S_{M^{\mathrm{p}}} + M^{\mathrm{p}} \le S_{R_{\mathrm{wt}}(G)} + R_{\mathrm{wt}}(G) < w(G)$. Substituting this strict inequality into Eq. 13 gives $W_{\tilde{y}}^{\mathrm{p}} - W_c^{\mathrm{p}} > 0$ for every competing class $c \neq \tilde{y}$, so $\arg\max_c W_c^{\mathrm{p}} = \tilde{y}$. This completes the proof of Theorem 3.2.

A.3 Proof of Lemma 3.3 (Soundness of the dual certificate)

We prove the two claims of Lemma 3.3 in turn. For the first claim, suppose the dual certificate in Eq. 7 evaluates to one at radius r. By definition of the logical-OR, at least one of the two disjuncts is one: either (i) $g_1(G) = y_G$ and $\mathcal{C}_1(G) \ge r$, or (ii) $g_2(G) = y_G$ and $\mathcal{C}_2(G) \ge r$. Let G^{p} be any perturbation of G with budget at most r. In case (i), the individual soundness of $(g_1, \mathcal{C}_1)$ gives $g_1(G^{\mathrm{p}}) = g_1(G) = y_G$, so g_1 remains correct on G^{p}; in case (ii), the symmetric argument for $(g_2, \mathcal{C}_2)$ gives $g_2(G^{\mathrm{p}}) = y_G$. In either case at least one of $g_1(G^{\mathrm{p}}), g_2(G^{\mathrm{p}})$ equals y_G, which establishes joint soundness of the dual certificate.

For the second claim, observe that the set of graphs certified correct by the dual certificate at radius r is the union of the sets of graphs certified correct by $\mathcal{C}_1$ alone and by $\mathcal{C}_2$ alone:

$$\begin{aligned} \big\{G : \mathrm{cert}_{\vee}(G, r) = 1\big\} \;=\; &\big\{G : \mathcal{C}_1(G) \ge r,\; g_1(G) = y_G\big\} \;\cup \\ &\big\{G : \mathcal{C}_2(G) \ge r,\; g_2(G) = y_G\big\} \end{aligned} \tag{14}$$

Because a union is pointwise at least each of its constituents, the certified accuracy under the dual certificate at every radius r evaluated by Eq. 14 is lower-bounded by the maximum of the certified accuracies of $\mathcal{C}_1$ and $\mathcal{C}_2$, and hence in particular by that of either alone. This completes the proof of Lemma 3.3.

A.4 Proof of Theorem 3.4 (Joint structure–feature certificate)

Under structure–feature division the sub-views are indexed by pairs $(s, u) \in \{0, \dots, T_s - 1\} \times \{0, \dots, T_f - 1\}$, with $G_{s,u} = (V, E_s, \mathbf{X}^{(u)})$. The confinement argument in the first step of the proof of Theorem 3.1 applies independently to the edge partition and the feature partition: each single edge perturbation modifies exactly one edge group $E_{s^\star}$ and therefore alters exactly the T_f sub-views $\{G_{s^\star,u}\}_{u=0}^{T_f-1}$, while each single node-feature perturbation modifies exactly one feature group $\mathbf{X}^{(u^\star)}$ and therefore alters exactly the T_s sub-views $\{G_{s,u^\star}\}_{s=0}^{T_s-1}$. With r_s edge perturbations and r_f feature perturbations in total, a union bound on the set of corrupted sub-views yields

$$M^{\mathrm{p}} \;=\; \sum_{(s,u)} \mathbb{I}\big[G_{s,u} \neq G_{s,u}^{\mathrm{p}}\big] \;\le\; r_s\, T_f \;+\; r_f\, T_s. \tag{15}$$

Applying Theorem 3.1 to the perturbed ensemble, the count-based prediction is preserved whenever $M^{\mathrm{P}} \leq M(G)$, which by Eq. 15 is implied by the sufficient condition $r_s T_f + r_f T_s \leq M(G)$. This defines the joint certified region $\mathcal{R}_{\mathrm{joint}}(G)$ stated in Theorem 3.4. This completes the proof of Theorem 3.4.

Remark. The union bound in Eq. 15 is generally conservative. If an edge perturbation's group coincides with the grid column affected by a feature perturbation, the same sub-view can be counted twice, so the actual number of corrupted sub-views can be smaller. We use the union bound because it is deterministic, attack-independent, and valid for every admissible perturbation; tightening it would require additional information about the realized perturbation pattern.

A.5 Relationship Between the Count-Based and Weighted Certificates

No sound certificate can guarantee a strictly larger radius from the count vector $(N_c)_{c\in\mathcal{Y}}$ alone: the bound in Theorem 3.1 is tight for the discrete majority-vote rule. The weighted certificate uses additional confidence information and is independently sound under the bounded-confidence model in Theorem 3.2; however, non-uniform confidences do not by themselves guarantee a larger empirical radius. The dual construction of Eq. 7 guarantees that the certified set is never smaller than either branch. This distinction separates the formal guarantee from the empirical tightness of the weighted bound.

B Details of Datasets

Table 2: Statistics of datasets.

Datasets	**#Training graphs**	**#Testing graphs**	**#Classes**
DBLP (Pan et al., 2013)	12,971	6,485	2
DD (Dobson & Doig, 2003)	785	393	2
ENZYMES (Hu et al., 2020)	400	200	6
MUTAG (Debnath et al., 1991)	125	63	2
NCI1 (Wale et al., 2006)	2,740	1,370	2
PROTEINS (Borgwardt et al., 2005)	742	371	2
REDDIT-B (Yanardag & Vishwanathan, 2015)	1,333	667	2
COLLAB (Yanardag & Vishwanathan, 2015)	3,333	1,667	3

Table 2 reports the per-dataset training-set size, testing-set size, and number of classes used in our experiments. The split sizes follow the two-thirds/one-third convention adopted in the certified-defense literature for TU benchmarks. DBLP (Pan et al., 2013) is the largest benchmark with 12,971 graphs; MUTAG (Debnath et al., 1991) is the smallest with 125. ENZYMES(Hu et al., 2020) is the only multi-class dataset with $C = 6$; COLLAB(Yanardag & Vishwanathan, 2015) is multi-class with $C = 3$; the remaining six benchmarks are binary.

Hardware Configuration. All experiments were conducted on a high-performance computing cluster. The server was equipped with a dual-socket Intel Xeon Gold 5317 processor running at 3.00 GHz with 48 CPU cores and 755 GB of system memory. For GPU-accelerated computation, we used 8 NVIDIA RTX A6000 GPUs, each with 48 GB of GDDR6 memory, running driver version 535.274.02.

Software Environment. Our implementation was developed in Python 3.10.12 on Ubuntu 22.04.5 LTS. We leveraged PyTorch 2.0.1 Paszke et al. (2019) as our primary deep learning framework, compiled with CUDA 11.8 Nickolls et al. (2008) support for GPU acceleration. Numerical computations were performed using NumPy 1.26.4 Harris et al. (2020) and SciPy 1.15.3 Virtanen et al. (2020). For machine learning utilities including preprocessing, evaluation metrics, and data splitting, we employed scikit-learn 1.7.2 Pedregosa et al. (2011). Additional utilities included `tqdm` 4.67.1 for progress monitoring, `psutil` 7.1.3 for system resource tracking, and `gdown` 5.2.0 for dataset retrieval.

C Comparing SpecCert with Four Baselines When the Number of Noisy Graphs Is Set to 30

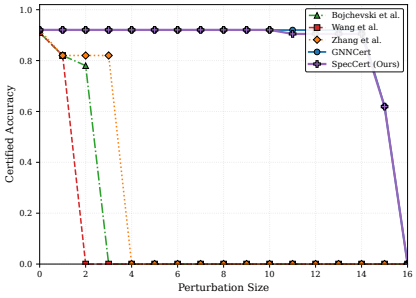


Figure 4: Compute matched certification on MUTAG. The three Monte Carlo baselines are downsampled to 30 noisy graphs per test instance, matching SpecCert's 30 sub-views. The Monte Carlo curves collapse to zero certified accuracy at $r \approx 4$, while SpecCert's deterministic curve is unchanged from Figure 2.

Existing Monte Carlo certified defenses Bojchevski et al. (2020); Wang et al. (2020); Zhang et al. (2021b) achieve their reported certified-accuracy curves by aggregating predictions over 1,000 noisy graphs per test instance, while SpecCert produces its certificate from only 30 sub-views. We re-run the three Monte Carlo baselines on MUTAG with $S_z = S_w = S_b = 30$ noisy graphs per test instance, matching SpecCert's budget exactly, and report the certified-accuracy curves in Figure 4. The results lead to two conclusions. *First*, when compute is held constant, all three Monte Carlo baselines lose roughly 2 perturbation-radius units relative to their original 1,000-sample reports, while the deterministic certificate of SpecCert is completely unaffected. SpecCert and GNNCert Xia et al. (2024) perform comparably under this matched-compute setting by design, as the experiment holds the partition budget fixed at 30 sub-views for both methods. Any differences visible in Figure 2 should therefore be interpreted together with the MECT and training-distribution ablations, rather than attributed solely to a larger partition budget. These results show that the certified-accuracy advantage of SpecCert over the evaluated Monte Carlo baselines persists under matched compute and is not explained solely by the larger original sampling budget.

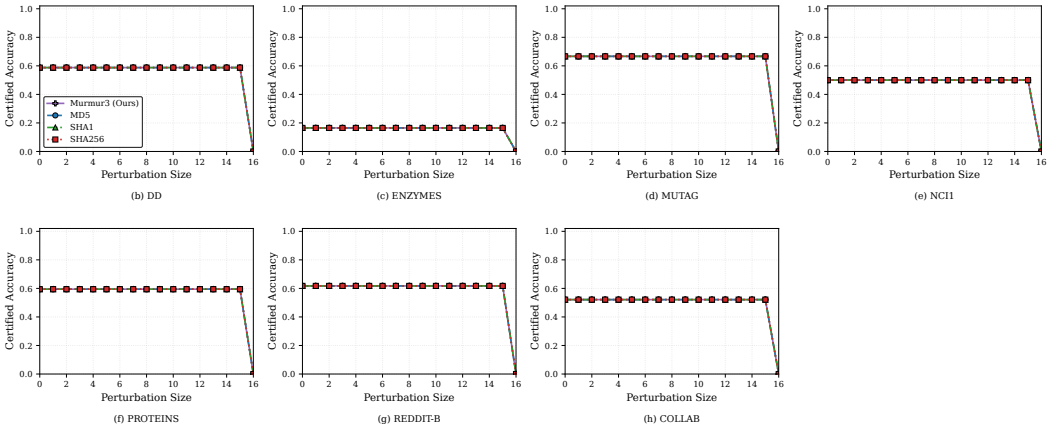


Figure 5: Impact of the hash function $\mathcal{H}$. The four hash families produce statistically equivalent certified-accuracy curves; the SpecCert default of MurmurHash3 is selected for its lower per-edge cost.

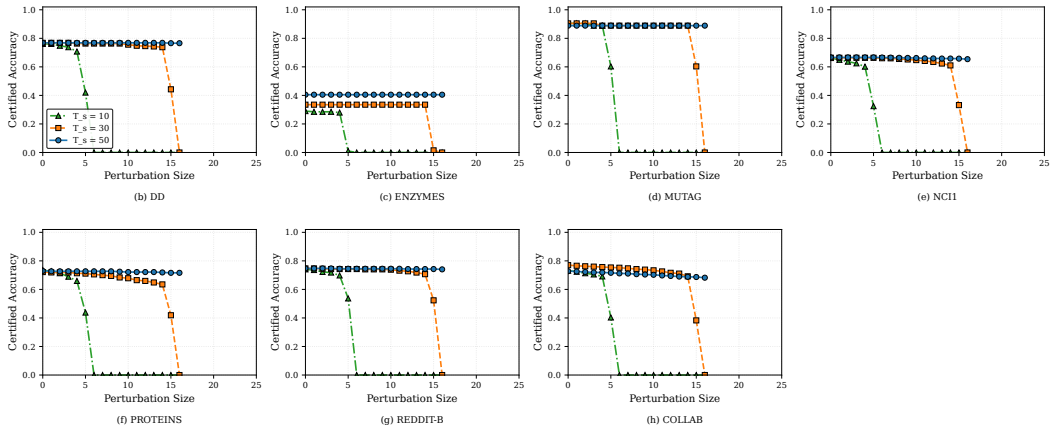


Figure 6: Impact of the number of groups T_s on SpecCert for structure division.

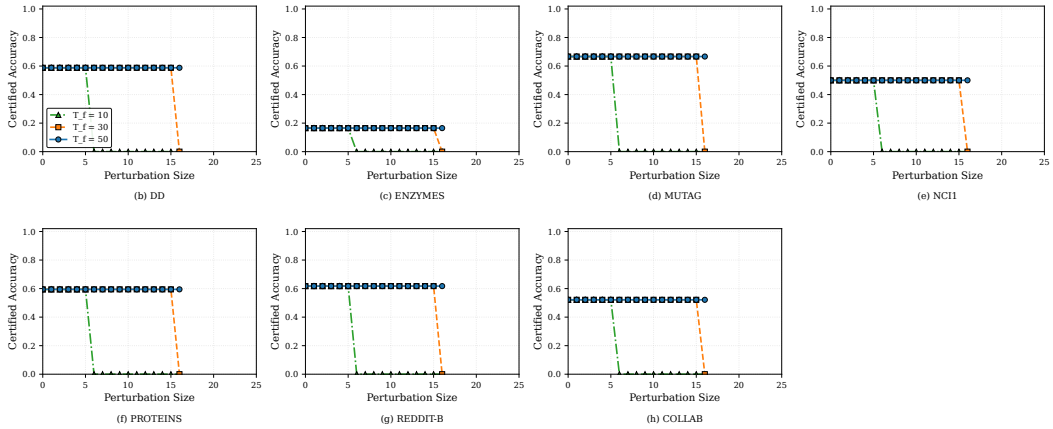


Figure 7: Impact of the number of groups T_f on SpecCert for feature division.

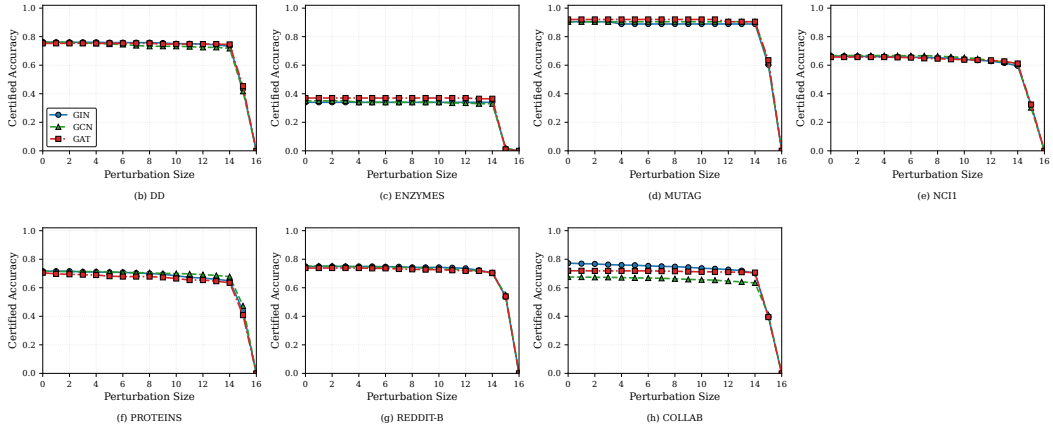


Figure 8: Impact of the base GNN architecture. GIN, GCN, and GAT produce nearly identical certified-accuracy curves under the SpecCert framework.

D Other Experimental Results

This appendix collects the additional experimental observations, organized into sub-experiments: (i) per-dataset wall-clock costs under default settings, (ii) the effect of the number of structure groups T_s, (iii) the effect of the number of feature groups T_f, (iv) the impact of the hash function family, (v) the impact of the base GNN architecture, (vi) joint structure–feature certification heatmaps, (vii) the dual-certification decomposition, and (viii) the MECT margin-weight sweep.

Table 3: Comparing the computation costs of SpecCert and existing defenses on multiple datasets.

Method	PROTEINS		ENZYMES		NCI1	
	Train (s)	Test (s)	Train (s)	Test (s)	Train (s)	Test (s)
Bojchevski et al. (2020)	2351	421	1930	157	1811	920
Wang et al. (2020)	2132	788	1760	2629	1703	4488
Zhang et al. (2021b)	1575	196	1427	95	1357	525
GNNCert (2024)	829	13	760	11	815	22
SpecCert (Ours)	1655	13.2	1707	5.3	1731	54.2

Per-dataset wall-clock costs. Table 3 reports the training and testing wall-clock costs of SpecCert and the four certified-defense baselines on PROTEINS, ENZYMES, and NCI1 under default settings. The deterministic baseline of GNNCert (Xia et al., 2024) and SpecCert produce comparable testing costs of around 5–54 seconds, both of which are between an order of magnitude and two orders of magnitude faster than every Monte Carlo baseline. Notably, SpecCert achieves a faster testing cost than GNNCert on ENZYMES (5.3 s vs. 11 s), while GNNCert retains a modest testing-cost advantage on PROTEINS (13 s vs. 13.2 s) and NCI1 (22 s vs. 54.2 s). The training cost of SpecCert is moderately higher because SpecCert trains a five-seed ensemble (Sec. 3.5) rather than a single model. This is the source of the logit-level tightening that contributes to Figures 2 and 10.

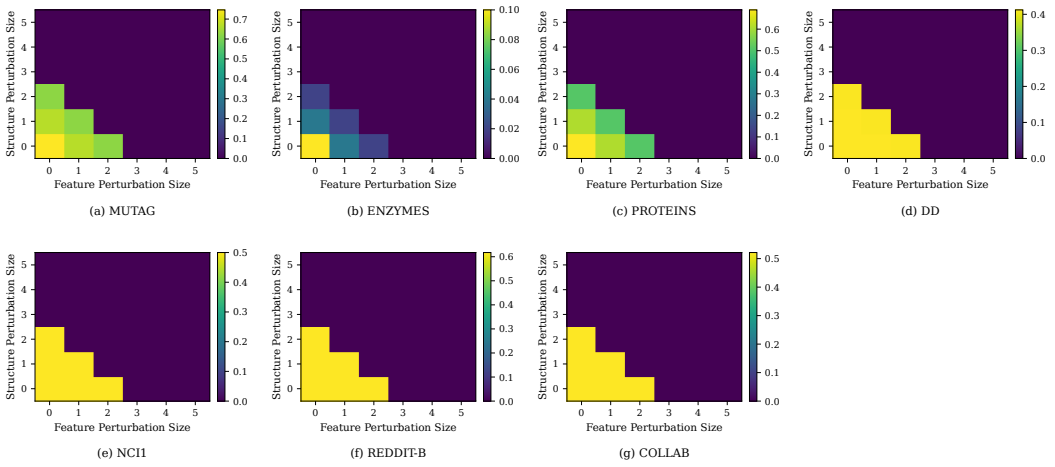


Figure 9: Joint structure–feature certified-accuracy heatmaps. Each panel reports the certified accuracy of SpecCert at every $(r_s, r_f) \in \{0, \ldots, 5\} \times \{0, \ldots, 5\}$ on one of seven benchmarks. The triangular boundary of the certified region is the locus $r_s T_f + r_f T_s = M(G)$ from Theorem 3.4.

Joint structure–feature certification. Figure 9 visualises the joint certified region $\mathcal{R}_{\text{joint}}(G)$ of Theorem 3.4 as a heatmap over the two-dimensional perturbation space (r_s, r_f) for the seven benchmarks where joint certification is meaningful. The clean-accuracy peaks at $(r_s, r_f) = (0, 0)$ are 0.746 on MUTAG, 0.689 on PROTEINS, 0.617 on REDDIT-B, 0.521 on COLLAB, 0.500 on NCI1, 0.412 on DD, and 0.100 on ENZYMES, and every panel shows a non-trivial joint certified region extending several units in either direction. The triangular geometry of each heatmap directly reflects the constraint $r_s T_f + r_f T_s \leq M(G)$: along either axis, the certificate degrades smoothly, while along the diagonal, both budgets are simultaneously consumed. The Monte Carlo baselines

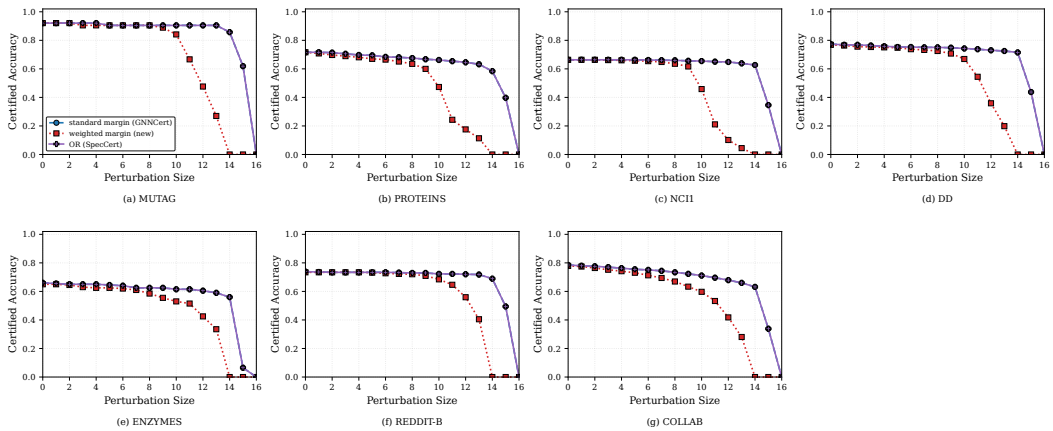


Figure 10: Decomposition of the SpecCert dual certificate. The standard count-based certificate, the confidence-weighted certificate, and their OR-combination are reported separately on seven datasets. The figure reports the two individual branches and their OR-combination. In our ablation study, the confidence-weighted branch is generally conservative and the OR curve largely follows the count-based branch.

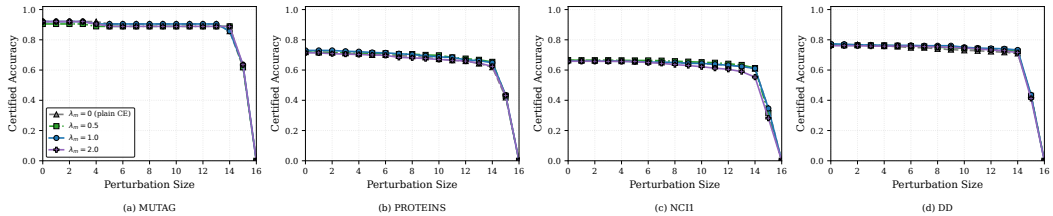


Figure 11: Impact of the MECT margin weight λ_m. The setting $\lambda_m = 0$ is plain cross-entropy training; $\lambda_m = 1$ is the SpecCert default. The benefit of MECT is largest on the harder PROTEINS benchmark and smallest on the smaller MUTAG benchmark, consistent with the interpretation that MECT is a margin-saturation mechanism that activates when the cross-entropy signal alone is insufficient to drive the vote margin to the certifiable range.

provide certificates only against structure perturbations. The heatmaps visualize the deterministic joint region induced by the sufficient condition in Theorem 3.4; the region is controlled by the count margin $M(G)$, which MECT is designed to encourage but does not optimize exactly.

Dual-certification decomposition. Figure 10 decomposes the two disjuncts of Eq. 7 on seven benchmarks. The standard count-based certificate (blue) is the primary certificate used by the implementation. The confidence-weighted certificate (red) is an independently sound alternative, while the dual-OR certificate (purple), sound by Lemma 3.3, is the upper envelope of the two certified sets and is therefore never worse than either disjunct alone. In our ablation study, the weighted branch is generally more conservative than the count-based branch, so the OR curve largely coincides with the standard curve. We consequently do not attribute the main empirical gains to the weighted branch; its role is as a formally valid optional extension whose tightness may improve under different confidence distributions.

Sensitivity to the MECT margin weight λ_m. Figure 11 sweeps $\lambda_m \in \{0, 0.5, 1, 2\}$ on four benchmarks. The setting $\lambda_m = 0$ recovers plain cross-entropy training, and $\lambda_m = 1$ is the SpecCert default. On PROTEINS, $\lambda_m = 1$ improves certified accuracy at every r by one to three points over plain cross-entropy. Pushing further to $\lambda_m = 2$ starts to hurt slightly, as the margin penalty begins to interfere with per-sub-view discrimination. On MUTAG, NCI1, and DD, all four settings perform similarly. This is expected: on these smaller benchmarks, cross-entropy alone already drives the vote margin high enough, leaving little room for MECT to help further. This behavior is consistent with the design of MECT as a margin-saturation mechanism: it activates precisely where cross-entropy training leaves headroom and remains quietly equivalent elsewhere. Across the evaluated benchmarks

and values of λ_m in our sweep, the MECT ablation is stable, with the largest gains on PROTEINS and little change on the smaller datasets. Plain cross-entropy training is recovered exactly at the $\lambda_m = 0$ corner of the SpecCert hyperparameter space. These results support $\lambda_m = 1$ as a practical default for the evaluated settings, while broader hyperparameter sensitivity remains future work.