Certifying Robust Graph Classification under Orthogonal Gromov-Wasserstein Threats

Hongwei Jin*
Mathematics and Computer Science Division
Argonne National Laboratory
Lemont, IL 60439
jinh@anl.gov

Zishun Yu, Xinhua Zhang
Department of Computer Science
University of Illinois Chicago
Chicago, IL 60607
{zyu32,zhangx}@uic.edu

Abstract

Graph classifiers are vulnerable to topological attacks. Although certificates of robustness have been recently developed, their threat model only counts local and global edge perturbations, which effectively ignores important graph structures such as isomorphism. To address this issue, we propose measuring the perturbation with the orthogonal Gromov-Wasserstein discrepancy, and building its Fenchel biconjugate to facilitate convex optimization. Our key insight is drawn from the matching loss whose root connects two variables via a monotone operator, and it yields a tight outer convex approximation for resistance distance on graph nodes. When applied to graph classification by graph convolutional networks, both our certificate and attack algorithm are demonstrated effective.

1 Introduction

State-of-the-art graph classifiers such as graph convolutional networks (GCNs) have been recently revealed vulnerable to adversarial structural attacks, namely adding and removing edges [1, 2]. Due to the hardness of finding the strongest attack, an alternative strategy aims to certify robustness, *i.e.*, proving that no perturbation in the threat model can successfully attack the classifier. To this end, [3, 4] extended randomized smoothing to discrete noises with applications in, *e.g.*, community detection. Although our focus is on graph classification, certificates have been studied for node classification using, *e.g.*, convex outer adversarial polytope [5, 6]. Most relevant to our method is [7], which constructed the tightest lower bound of the margin via Fenchel biconjugation.

However, all existing threat models only count the total edges removed or added (global budget) or the change of each node's degree (local budget), while overlooking the important notion of *isomorphism*. In contrast, there has been a wealth of more refined discrepancy measures between graphs, including kernels [8, 9] and GCNs [10, 11]. Recently, the Gromov-Wasserstein discrepancy [GW, 12], which extends the Gromov-Wasserstein distance [13], has emerged as an effective transportation distance between structured data, alleviating the incomparability issue between different structures by aligning the *intra*-relational geometries. Thanks to its favorable properties such as efficiency and awareness of isomorphism, GW has been extensively applied to domain adaptation [14], word embedding [15], graph classification [16], metric alignment [17], generative modeling [18], and graph matching and node embedding [19, 20]. Therefore, it is clearly natural to adopt such a measure in the certification of graph robustness. Analogously, Wasserstein distance has already been employed in distributional robustness [21] and in attacking images [22], where perturbations can be found that better reflect the image content compared with the standard ℓ_p attacks.

Unfortunately, GW is not tractable to evaluate, differing from the standard Wasserstein distance which is a linear program. So all existing optimization techniques settle for local solutions, lacking

*Work done when the author was at the University of Illinois Chicago

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

an analyzable guarantee. To develop certificates, tractable lower bounds of GW would be essential, but the Kantorovich dual no longer exists and the existing lower bounds [13, 23] are not tight [24]. Further, underlying the standard probability coupling is the metrics within each graph, *e.g.*, shortest-path distance [13, SP distance], which requires nontrivial discrete optimization. This significantly complicates the entire certificate as the subject of perturbation is exactly the topological structure. In addition, certificates based on randomized smoothing [4, 25], which adds noise to the input, are also hindered by the significant challenge in designing the noise distribution that aligns with GW.

The goal of this work, therefore, is to construct the first robustness certificate for GW-style threat models, with applications in graph classification. This is achieved by casting the entire computation procedure of GW as a two-layer model, each of which is subsequently relaxed in a tractable fashion:

$$\text{graph structure} \xrightarrow[\text{matching loss}]{\text{resistance distance}} \text{distance btw nodes} \xrightarrow[\text{convex envelop}]{\text{orthogonal GW}} \text{discrepancy btw graphs.}$$

Firstly, we circumvent the discrete optimization in SP distance by resorting to the resistance distance [26], which, along with its scaled version known as commute time, is one of the most commonly used distances in computer science, machine learning, and beyond [27]. Compared with SP distance, it admits a *closed-form* based on matrix inversion, which is a *monotone* operator on the positive semi-definite cone. This enables our first contribution, where a convex relaxation is designed for resistance distance in Section 3 based on the matching loss [28].

Our second contribution achieves convex relaxation of the GW-style discrepancy measure. Our seed inspiration is drawn from the latest orthogonal GW discrepancy (OGW) [24] reviewed in Section 2.2—it preserves most of the desirable properties of GW while admitting a tight and efficient lower bound. We next make the key observation in Section 2.3 that the lower bound, despite not convex, admits a *closed-form* Fenchel dual, which in turn facilitates an efficient evaluation of the Fenchel biconjugate. Such a convex lower bound is applied to graph convolution networks in Section 4, where both attacking algorithm and robustness certificates are developed.

Our experiments in Section 5 verify the effectiveness of our attacker and certificate, in that a large proportion of the graphs can be proved either vulnerable or robust.

Related Work Certification or verification has been a longstanding pursuit in machine learning and cyber-physical systems. Formally, it inquires if $f(x)$ can be driven down to negative over $x \in \mathcal{X}$, where both f and $\mathcal{X}$ can be too complicated for global optimization. So a lower bound of the optimal objective is sought, which provides incomplete verification (some true properties are confirmed true). A lower bound can also be useful in branch-and-bound algorithms [6]. A natural approach is the Lagrange dual [29, 30], which minorizes the primal by the weak duality. However, it can be loose, and a provably tighter lower bound is the convex envelope or Fenchel biconjugate [7]. But they themselves are efficiently computable only under limited scenarios.

In general, the objective f is composed of several "layers", *i.e.*, $f = f_0 \circ f_1 \circ \ldots$, where f_i is simpler but no longer scalar-valued ($i \geq 1$). So it is natural to relax their graphs. For example, $\min f_0(f_1(x))$ is equivalent to $\min_{x,y} f_0(y)$ s.t. $y = f_1(x)$. So we can possibly use the convex envelop of f_0 (which is simpler than f) and employ the convex hull of $\{(x, y) : y = f_1(x)\}$. The latter is represented by the convex envelope relaxation of the ReLU activation [31–35], which has also found applications in graph neural networks [6, 36]. However, they have been so far limited to $\mathbb{R} \to \mathbb{R}$ nonlinear functions while a multivariate extension is not available yet, as the convex hull is much harder to compute.

Another commonly used approach is randomized smoothing, which adds noise to the input [25, 37–39]. Although it has also been shown effective in graph robustness [3, 4], the design of discrete noise can be quite intricate and so far no such methods exist for GW-style threat models. Other methods include semi-definite relaxation [40, 41] which is often loose, Lipschitz continuity analysis [42–44] where the local and global estimation of curvature is difficult in the discrete domain, and reformulation linearization technique [45] used by [5] for directed but not undirected graphs.

2 Gromov-Wasserstein Threats for Attacking Graph Classifiers

We consider attacking a graph classifier where the edge connectivity is perturbed. Here both the original graph $\mathcal{D}$ and the resulting graph $\mathcal{C}$ are assumed **undirected**, **unweighted**, and **connected**. Let the adjacency matrix of $\mathcal{D}$ be $A \in \{0, 1\}^{n \times n}$, where $A_{ii} = 0$, and $A_{ij} \in \{0, 1\}$ $(i \neq j)$ indicating whether an edge exists between nodes i and j. Here n is the *order* of the graph, *i.e.*, number of nodes.

Now suppose the graph's edge connectivity is perturbed, leading to a new adjacency matrix $\tilde{A} := A + X$ (corresponding to $\mathcal{C}$). Obviously, X needs to be symmetric, zero diagonal, and $X_{ij} \in \{-A_{ij}, 1 - A_{ij}\}$. If we relax it to the continuous domain such as convex relaxation, X_{ij} is naturally relaxed to $[-A_{ij}, 1 - A_{ij}]$. The **threat model** governs additional budgets on X besides the above admissible conditions. Global change budget can be written as $\mathbf{1}^\top (X \circ S)\mathbf{1} \leq \delta_g$, where $S = 1 - 2A$ and $\circ$ stands for the Hadamard product. Local budget is $(X \circ S)\mathbf{1} \leq \delta_l \mathbf{1}$ (elementwise). If we only allow adding edges, then $X_{ij} \geq 0$. All these basic constraints are linear in X, and we will collectively refer to them as $X \in \mathcal{X}$, a convex set.

Suppose we have trained a graph classifier $\mathcal{G}$ (*e.g.*, graph convolution network, GCN) which maps an adjacency matrix A to a K-dimensional logit vector $\mathcal{G}(A)$ corresponding to K classes. Assume the true class is y and $\mathcal{G}$ predicts correctly, *i.e.*, $y = \arg\max_c \mathcal{G}_c(A)$. Then the margin of classification is

$$M(A) := \min_{c \neq y} \{\mathcal{G}_y(A) - \mathcal{G}_c(A)\}. \tag{1}$$

The attacker seeks a feasible perturbation $X \in \mathcal{X}$ that drives the classification margin to negative, *i.e.*, a misclassification. $\mathcal{G}$ is called robust on this graph if $\min_{X \in \mathcal{X}} M(A + X)$ is positive, and is called vulnerable if the minimal value is negative. It is noteworthy that we employ graph classification and GCN only as an example context of applying our proposed relaxation technique. Other tasks such as node classification are readily applicable too – simply replace the M function.

Although the global and local budgets are natural and reasonable characterizations of the perturbation, they ignore an important notion on graphs: isomorphism. Exact isomorphism can be relaxed by defining a distance between two graphs, which quantify their difference under the most favorable permutation of nodes. One of the most commonly used distance is the Gromov-Wasserstein distance [13], which has been extended to Gromov-Wasserstein discrepancy [GW, 12]. So in addition to the standard local and global budgets specified by $\mathcal{X}$, it is natural to further constrain the perturbation in terms of the GW distance. To conclude, the attack and certification tasks can be formalized as $\min_{X \in \mathcal{X}} M(A + X)$ s.t. GW(new graph $A + X$, old graph A) $\leq \delta_\Omega$ for some budget $\delta_\Omega > 0$.

2.1 Definition of Gromov-Wasserstein distance

Since our threat model will only add or remove edges while the nodes remain intact, we only need to compare two graphs $\mathcal{C}$ and $\mathcal{D}$ with the same order. Although GW takes a prior distribution on nodes that represents their importance, it is often observed that a simple uniform distribution performs equally well or better [12, 16, 46], and many applications do not have such an external prior. So we can restrict GW to uniform distributions, leading to the following expression under ℓ_2 distance:

$$\mathsf{GW}(\mathcal{C}, \mathcal{D}) = \min_{P \in \mathbb{R}_+^{n \times n} : P\mathbf{1} = P^\top \mathbf{1} = n^{-1}\mathbf{1}} \sum\nolimits_{i,j,p,q} (C_{ij} - D_{pq})^2 P_{ip} P_{jq}. \tag{2}$$

Here $\mathbf{1}$ is an all-one vector, and C_{ij} is a distance measure between nodes i and j in $\mathcal{C}$, *e.g.*, SP distance. We will denote the $n \times n$ matrices as C (for $\mathcal{C}$) and D (for $\mathcal{D}$). We also follow the idea of [12], where C_{ij} does not have to be a metric and ℓ_2 can be extended to asymmetric losses. They call it GW discrepancy, and we do not take the square root of the right-hand side of (2) when defining GW. In the sequel, we will also write $\mathsf{GW}(C, D)$ instead of $\mathsf{GW}(\mathcal{C}, \mathcal{D})$ whenever it causes no confusion.

2.2 Orthogonal GW and its upper and lower bounds

Unfortunately, GW relies on a nonconvex optimization, and the optimal P in (2) is intractable to find. Although polytime lower bounds have been proposed by [13], they are shown quite loose [24]. Since our aim is to certify robustness under such a measure, a tight and efficient lower bound is in demand (see below). To this end, we resort to the recently proposed orthogonal GW [OGW, 24], which does offer such a lower bound. OGW first rewrites the GW in the Koopmans-Beckmann form [47]:

$$\mathsf{GW}(C, D) = \tfrac{1}{n^2}\Big(\|C\|_F^2 + \|D\|_F^2 - 2 \max_{P \in \mathcal{E} \cap \mathcal{N}} \operatorname{tr}(CPDP^\top) \Big), \tag{3}$$

$$\text{where} \quad \mathcal{E} := \{P \in \mathbb{R}^{n \times n} : P\mathbf{1} = P^\top \mathbf{1} = \mathbf{1}\}, \quad \mathcal{N} := \mathbb{R}_+^{n \times n}, \quad \|C\|_F^2 := \sum\nolimits_{ij} C_{ij}^2. \tag{4}$$

Let Π be the set of $n \times n$ permutation matrices. The Birkhoff-von Neumann theorem asserts that the domain of coupling $(\mathcal{E} \cap \mathcal{N})$ is the convex hull of Π. In addition, Π can be characterized by

$$\Pi = \mathcal{E} \cap \mathcal{N} \cap \mathcal{O}_n, \quad \text{where} \quad \mathcal{O}_n := \{P \in \mathbb{R}^{n \times n} : P^\top P = PP^\top = I\}. \tag{5}$$

Since both $\mathcal{O}_n$ and $\mathcal{E}$ are spectral constraints, [24] proposed using these two domains only:

$$\mathsf{OGW}(C, D) := \tfrac{1}{n^2}\Big(\|C\|_F^2 + \|D\|_F^2 - 2 \max_{P \in \mathcal{E} \cap \mathcal{O}_n} \operatorname{tr}(CPDP^\top)\Big). \tag{6}$$

To summarize, the attacker tries to minimize the margin by solving the discrete problem:

$$\min_X M(A+X), \quad s.t., \quad X \in \mathcal{X},\ X_{ij} \in \{-A_{ij}, 1-A_{ij}\},\ \mathsf{OGW}(C_{A+X}, C_A) \le \delta_\Omega, \tag{7}$$

where C_{A+X} and $C_A = D$ are the base distance matrices for $A+X$ and A, respectively. Since OGW is still intractable to compute, we will next review how [24] proposed to approximate it.

Lower bound of OGW for certificates To certify the robustness, one only needs to optimize (7) with the discrete constraint relaxed into $X_{ij} \in [-A_{ij}, 1-A_{ij}]$. If the optimal objective value remains above 0, then robustness is certified. To tackle the intractability of OGW, a conservative approach (no false positive) is to relax the feasible domain by replacing OGW with an efficient *lower bound*.

Although the optimal P in (2) is still intractable to solve, OGW admits an efficient lower bound that is inspired by the quadratic assignment literature [48]. To begin with, [48] noted that $\mathcal{O}_n \cap \mathcal{E} = \left\{\frac{1}{n}\mathbf{1}\mathbf{1}^\top + VQV^\top : Q \in \mathcal{O}_{n-1}\right\}$, where V is any $n \times (n-1)$ matrix satisfying $V^\top \mathbf{1} = \mathbf{0}$ and $V^\top V = I$. So denoting $\hat{X} := V^\top X V$ and $s_X := \mathbf{1}^\top X \mathbf{1}$ for **any** compatible matrix X, we obtain

$$\max_{P \in \mathcal{O}_n \cap \mathcal{E}} \operatorname{tr}(CPDP^\top) = \tfrac{1}{n^2} s_C s_D + \mathcal{Q}(C, D), \tag{8}$$

$$\text{where} \quad \mathcal{Q}(C, D) := \max_{Q \in \mathcal{O}_{n-1}} \{\operatorname{tr}(\hat{C}Q\hat{D}Q^\top) + \operatorname{tr}(\hat{E}^\top Q)\}, \qquad E := \tfrac{2}{n} C \mathbf{1}\mathbf{1}^\top D. \tag{9}$$

So a lower bound of OGW (named OGW_{lb}) can be naturally derived by decoupling the two occurrences of Q in $\mathcal{Q}(C, D)$:

$$\mathsf{OGW}_{lb}(C, D) := \tfrac{1}{n^2}\left(\|C\|_F^2 + \|D\|_F^2 - \tfrac{2}{n^2} s_C s_D - 2\mathcal{Q}_{ub}(C, D)\right) \tag{10}$$

$$\text{where} \quad \mathcal{Q}_{ub}(C, D) := \max_{Q_1 \in \mathcal{O}_{n-1}} \operatorname{tr}(\hat{C}Q_1\hat{D}Q_1^\top) + \max_{Q_2 \in \mathcal{O}_{n-1}} \operatorname{tr}(\hat{E}^\top Q_2). \tag{11}$$

One can derive that Q_2 is optimized at $U_E V_E^\top$ where $U_E \Lambda_E V_E^\top$ is the singular value decomposition (SVD) of $\hat{E}$. This results in $\operatorname{tr}(\hat{E}^\top Q_2) = \|\hat{E}\|_*$, the trace norm, which is the sum of its singular values. An optimal Q_1 is $P_1 P_2^\top$, if $\hat{C}$ and $\hat{D}$ respectively admit eigen-decompositions $\hat{C} = P_1 \operatorname{diag}(\lambda_1) P_1^\top$ and $\hat{D} = P_2 \operatorname{diag}(\lambda_2) P_2^\top$ [49, 50]. This yields an optimal value of $\operatorname{tr}(\hat{C}Q_1\hat{D}Q_1^\top)$ as $\lambda_1^\top \lambda_2$. The overall computation costs $O(n^3)$ and the gap arising from decoupling Q_1 and Q_2 is small because in general, the matrix $\hat{E}$ is much smaller than $\hat{C}$ and $\hat{D}$ in magnitude, as noted by both [48] and [24].

Upper bound of OGW for the attacker A conservative estimate of whether the attack can be successful is obtainable by further restraining the feasible domain. This can be served by replacing OGW with its *upper bound*, which can be trivially achieved by locally optimizing Q in (9). We denote it as OGW_{ub}. *Locally* optimizing over $\mathcal{O}_{n-1}$ (*a.k.a.* Stiefel manifold) has been very well studied [51–53], and we adopt a straightforward approach of projected quasi-Newton, noting that the projection of any matrix Q on $\mathcal{O}_{n-1}$ is simply $U_Q V_Q^\top$, where the SVD of Q is $U_Q \Lambda_Q V_Q^\top$.

As is shown by [24], both OGW and OGW_{lb} are nonnegative, symmetric, and evaluates zero when $\mathcal{C}$ and $\mathcal{D}$ are isomorphic. Their square root also satisfies the triangle inequality. Experiments on graph classification and barycenter show that they well capture the structural dissimilarities between graphs. Similarly to GW, we compute OGW in practice via OGW_{ub}.

2.3 Convex lower bound of OGW

Although OGW_{lb} minorizes OGW, it is still not convex in its input C, falling short of the requirement of certification. We next derive a convex lower bound via Fenchel biconjugation. In general, biconjugation can be computationally expensive, but interestingly, our case admits efficient recipes.

Recall that in the context of perturbation, we assume C and D correspond to the new and original graphs, respectively, and our goal is to find a convex lower bound for the part of $\mathsf{OGW}(C, D)$ that

depends on C, *i.e.*, $f(C) := \|C\|_F^2 - 2\max_{P\in\mathcal{E}\cap\mathcal{O}_n} \mathrm{tr}(CPDP^\top)$. We first compute its Fenchel dual:

$$f^*(R) := \max_C \left\{\mathrm{tr}(R^\top C) - f(C)\right\} = \max_C \left\{\mathrm{tr}(R^\top C) - \|C\|_F^2 + 2\max_{P\in\mathcal{E}\cap\mathcal{O}_n} \mathrm{tr}(CPDP^\top)\right\} \quad (12)$$

$$= \max_{P\in\mathcal{E}\cap\mathcal{O}_n} \max_C \left(\mathrm{tr}(R^\top C) - \|C\|_F^2 + 2\,\mathrm{tr}(CPDP^\top)\right) = \max_{P\in\mathcal{E}\cap\mathcal{O}_n} \left\|\tfrac{1}{2}R + PDP^\top\right\|_F^2 \quad (13)$$

$$= \tfrac{1}{4}\|R\|_F^2 + \|D\|_F^2 + \max_{P\in\mathcal{E}\cap\mathcal{O}_n} \mathrm{tr}(RPDP^\top) \quad (14)$$

$$\le \tfrac{1}{4}\|R\|_F^2 + \|D\|_F^2 + \tfrac{1}{n^2}s_R s_D + \mathcal{Q}_{ub}(R, D). \quad (15)$$

As $f \ge f^{**}$ $(= (f^*)^*$, the biconjugation), $\mathsf{OGW}(C, D)$ as a function of C can be lower bounded by

$$\mathsf{OGW}(C, D) \ge \tfrac{1}{n^2}\left(\|D\|_F^2 + f^{**}(C)\right) = \tfrac{1}{n^2}\left(\|D\|_F^2 + \max_R\{\mathrm{tr}(R^\top C) - f^*(R)\}\right) \quad (16)$$

$$\ge \tfrac{1}{n^2}\max_R \left\{\mathrm{tr}(R^\top C) - \tfrac{1}{4}\|R\|_F^2 - \tfrac{1}{n^2}s_R s_D - \mathcal{Q}_{ub}(R, D))\right\} \quad (17)$$

$$=: \boxed{\Omega(C)}\,. \quad (18)$$

The evaluation of $\Omega(C)$ requires solving a convex optimization, utilizing the closed form for $\mathcal{Q}_{ub}$ in (11). In the optimization for the certificate below, we will avoid such an optimization over R by considering the dual of $\Omega(C)$, which can be easily read off from the objective of R in (17).

3 Convex Outer Approximation for Resistance Distance

So far, we have taken the distance matrix C as the source of variation. In the certification problem, however, variations originate from the perturbation X on the graph structure. So we will next relate X to C_{A+X}, and formulate a *convex* domain in X. Unfortunately, this is difficult, so we resort to the convex outer approximation of the graph of $X \mapsto C_{A+X}$. Such an approach has been commonly used in approximating the ReLU activation [31], and we extend it to a multivariate setting via a new approach of matching loss [28].

3.1 Using resistance distance as the base measure

Although the SP distance has enjoyed significant popularity, its computation requires a nontrivial discrete optimization, obstructing a convex relaxation. As such, we propose using resistance distance as the base distance because it is a bona fide metric [26], admits a closed form, and costs $O(n^3)$ to compute all-pair distance (same as SP distance). In Section 5.1 and 5.2, we will show that it not only eases computation, but also performs competitively on machine learning tasks. Let the Laplacian of the perturbed graph be $\tilde{L} := \mathrm{diag}(\tilde{A}\mathbf{1}) - \tilde{A}$. Then the resistance distance between node i and j is

$$C_{ij} = \tilde{L}^\dagger_{ii} + \tilde{L}^\dagger_{jj} - 2\tilde{L}^\dagger_{ij}\,, \quad (19)$$

where $\tilde{L}^\dagger$ is the pseudo-inverse of $\tilde{L}$. Since we only consider connected undirected graphs, it follows $\tilde{L}^\dagger = (\tilde{L} + \frac{1}{n}\mathbf{1}\mathbf{1}^\top)^{-1} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. A closely related distance measure is the commute time, which multiply the resistance distance by the volume of the graph ($\mathbf{1}^\top\tilde{A}\mathbf{1}$). Since the volume is a scalar and relates closely with the global budget, it can be incorporated with ease as discussed in Appendix A.1.

In summary, letting $Z = -(\tilde{L} + \frac{1}{n}\mathbf{1}\mathbf{1}^\top)^{-1}$, C is determined through $X \to \tilde{A} \to \tilde{L} \to Z \to \tilde{L}^\dagger \to C$. As both the first two steps and the last two steps are affine, we will denote them as $\tilde{L}_X$ and C_Z respectively. Only the step of $\tilde{L} \to Z$ is nonlinear, posing the major challenge we will address next.

3.2 Convex outer approximation of matrix inversion

Let $\mathcal{S}_+$ and $\mathcal{S}_{++}$ be respectively the positive semi-definite and positive definite matrix cones sized n-by-n. We consider the set

$$\mathcal{F} := \left\{(\tilde{L}, Z) \in \mathcal{S}_+ \times (-\mathcal{S}_{++}) : Z = -(\tilde{L} + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)^{-1},\ \tilde{L}\mathbf{1} = \mathbf{0},\ \tilde{L}_{ij} \le \beta_{ij}\ \forall\, i \ne j\right\}.$$

The conditions on $\tilde{L}$ are naturally motivated from graph Laplacian, and it follows implicitly that $\tilde{L} + \frac{1}{n}\mathbf{1}\mathbf{1}^\top \in \mathcal{S}_{++}$. A trivial choice of β_{ij} is 0. If we only allow adding edges, then β_{ij} can be $-A_{ij}$.

Given a strictly convex function F, the associated matching loss is defined as [28]

$$\ell(Z, \Phi) = F^*(Z) + F(\Phi) - \operatorname{tr}(Z^\top \Phi), \tag{20}$$

and it is well known that the minimum value of ℓ is 0, attained if, and only if, $Z = \nabla F(\Phi)$. Now consider $F(\Phi) = -\log\det(\Phi)$ over $\Phi \in \mathcal{S}_{++}$. It is easy to show that F is strictly convex, $\nabla F(\Phi) = -\Phi^{-1}$, and $F^*(Z) = -n - \log\det(-Z)$ over $Z \in -\mathcal{S}_{++}$. $\nabla F^*(Z) = -Z^{-1}$. Therefore, setting $Z = -(\tilde{L} + \frac{1}{n}\mathbf{1}\mathbf{1}^\top)^{-1}$ is equivalent to enforcing

$$\ell(Z, \tilde{L} + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top) = F^*(Z) + F(\tilde{L} + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top) - \operatorname{tr}(Z^\top(\tilde{L} + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)) \le 0. \tag{21}$$

This has significantly reduced the difficulty of relaxation because almost all the nonlinearities have been subsumed by the *convex* functions F^* and F. The only remaining challenge is the bilinear term in trace, but it is much easier to relax than a general nonlinear function. In particular, by the property of Laplacian, $Z\mathbf{1} = -\mathbf{1}$ and $Z \in -\mathcal{S}_{++}$. Let $T_{ii} = -Z_{ii}$ and $T_{ij} = 2Z_{ij} - Z_{ii} - Z_{jj}$ for $i \neq j$. Then $T \ge \mathbf{0}$ elementwise. So it follows that

$$-\operatorname{tr}(Z^\top \tilde{L}) = -\sum_{i\neq j} \tfrac{1}{2}(T_{ij} - T_{ii} - T_{jj})\tilde{L}_{ij} + \sum_i T_{ii}\tilde{L}_{ii} \tag{22}$$

$$= -\tfrac{1}{2}\sum_{i\neq j} T_{ij}\tilde{L}_{ij} + \sum_i T_{ii}\sum_j \tilde{L}_{ij} \overset{(\text{by } \tilde{L}\mathbf{1}=\mathbf{0})}{=} -\tfrac{1}{2}\sum_{i\neq j} T_{ij}\tilde{L}_{ij} \ge -\tfrac{1}{2}\sum_{i\neq j}\beta_{ij}T_{ij}. \tag{23}$$

Let B be a symmetric matrix such that $\operatorname{tr}(BZ) = \frac{1}{2}\sum_{i\neq j}\beta_{ij}T_{ij} = \frac{1}{2}\sum_{i\neq j}\beta_{ij}(2Z_{ij} - Z_{ii} - Z_{jj})$. Then the constraint set $\mathcal{F}$ is enclosed by a *convex* outer approximation as

$$\mathcal{F}_{out} := \{(\tilde{L}, Z) \in \mathcal{S}_+ \times (-\mathcal{S}_{++}) :\ \tilde{L} + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top \in \mathcal{S}_{++},\ \tilde{L}\mathbf{1} = \mathbf{0},\ \tilde{L}_{ij} \le \beta_{ij}\ \forall\, i \neq j, \tag{24}$$

$$Z\mathbf{1} = -\mathbf{1},\ F^*(Z) + F(\tilde{L} + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top) + 1 - \operatorname{tr}(BZ) \le 0\}. \tag{25}$$

Overall, we can relax the budget on OGW as follows:

$$\{X : \text{perturbation } X \text{ satisfies } \mathsf{OGW}(C, C_A) \le \delta_\Omega\} \tag{26}$$

$$= \{X : \mathsf{OGW}(C, C_A) \le \delta_\Omega,\ (\tilde{L}, Z) \in \mathcal{F},\ C = C_Z,\ \tilde{L} = \tilde{L}_X\} \tag{27}$$

$$(\text{by (17)}) \subseteq \{X : \Omega(C_Z) \le \delta_\Omega,\ (\tilde{L}_X, Z) \in \mathcal{F}\} \tag{28}$$

$$(\text{by } \mathcal{F} \subseteq \mathcal{F}_{out}) \subseteq \{X : \Omega(C_Z) \le \delta_\Omega,\ (\tilde{L}_X, Z) \in \mathcal{F}_{out}\}. \tag{29}$$

Finally, we attain a convex domain. The last two subsumptions summarize our key contributions of convex relaxation in Section 2.2 and 3, and we will experimentally demonstrate that they are tight.

4 Optimization of Attacks and Robustness Certificates with Relaxed OGW

Attacker algorithm As discussed in Section 2.2, an attacker solves (7) by resorting to an upper bound of OGW, requiring $\mathsf{OGW}_{ub} \le \delta_\Omega$. Here OGW_{ub} only needs an efficient local optimization. To deal with the constraint, we resort to the Lagrange dual:

$$\max_{\lambda\ge 0}\ \min_{X\in\mathcal{X}, X_{ij}\in\{-A_{ij}, 1-A_{ij}\}} M(A+X) + \lambda(\mathsf{OGW}_{ub}(C_{A+X}, C_A) - \delta_\Omega). \tag{30}$$

Given λ, X can be optimized greedily as shown in Algorithm 2. To ease notation, we denote $\mathsf{OGW}_{ub}(C_{A+X}, C_A)$ as $\Upsilon(A+X)$. So the problem becomes a simple search for the smallest λ such that $\Upsilon(A+X) \le \delta_\Omega$, and it can be solved by binary search as in Algorithm 1. Due to the lack of strong duality, the success of the attack is checked by the sign of $M(A+X)$ at the X found by Algorithm 1, *not* by the sign of the final objective value in (30).

4.1 Certificate algorithm

When the GCN has a single hidden layer, [7] showed that $M^*(A)$ can be computed in a closed form over the domain of A_{ij} that is relaxed to $[0, 1]$ ($i \neq j$), intersected with local and global budgets.

Algorithm 1: Attacking with binary search

Input: global budget δ_g and OGW budget δ_Ω
$\lambda \leftarrow 1$, and compute X_λ via Algorithm 2.
if $\Upsilon(A + X_\lambda) \geq \delta_\Omega$ **then**
  Double λ until $\Upsilon(A + X_\lambda) \leq \delta_\Omega$
  $u \leftarrow \lambda$ (upper est.), $l \leftarrow \frac{u}{2}$ (lower est.)
else
  Halve λ until $\Upsilon(A + X_\lambda) \geq \delta_\Omega$
  $l \leftarrow \lambda$, $u \leftarrow 2l$
while $u - l > 0.01$ (or any threshold) **do**
  $\lambda \leftarrow (u + l)/2$
  If $\Upsilon(A + X_\lambda) \geq \delta_\Omega$ **then** $l \leftarrow \lambda$ **else** $u \leftarrow \lambda$
Return X_λ

Algorithm 2: Compute X_λ for a given λ

$A_0 \leftarrow A$
for $t = 0, 1, \ldots, \delta_g - 1$ **do**
  $C_t \leftarrow$ cost matrix for A_t.
  $Q_t \leftarrow$ locally optimal Q in $\mathcal{Q}(C_t, C_A)$
  for $e \in$ *all pairs of nodes* **do**
    If flipping edge e on A_t violates any local budget **then** $V_e \leftarrow \infty$ and continue
    $A' \leftarrow$ adjacency matrix flipping e on A_t
    Update C for A'
    $V_e \leftarrow M(A') + \lambda\Upsilon(A')$, where the Q in $\mathcal{Q}(C, C_A)$ is initialized by Q_t
  $A_{t+1} \leftarrow$ flip on A_t the edge $\arg\min_e V_e$
Return $X_\lambda := A_{\delta_g} - A$

As a result, $M^{**}(A)$ can also be computed efficiently through a convex optimization. Since M^{**} minorizes M, we can confirm the robustness (*i.e.*, (7) cannot be reduced to negative) if the lowest possible value of M^{**} is nonnegative over an even *larger* domain of X than that in (7). This is a sufficient but not necessary condition, and the tightness of the relaxation can be verified via the percentage of graphs that can neither be certified as robust, nor successfully attacked by Algorithm 1.

Formalizing this idea, the certification algorithm needs to solve the following problem based on (29),

$$\min_{X,Z} M^{**}(A + X), \quad s.t., \quad X \in \mathcal{X},\ \Omega(C_Z) \leq \delta_\Omega,\ (\tilde{L}_X, Z) \in \mathcal{F}_{out}. \tag{31}$$

Plugging in the definition of $\mathcal{F}_{out}$ and noting that the conditions on $\tilde{L}$ are automatically satisfied by a graph Laplacian, we can explicitize the final *convex* optimization problem as

$$\min_{X \in \mathcal{X}, Z} \quad M^{**}(A + X) \tag{32}$$

$$s.t. \quad \Omega(C_Z) \leq \delta_\Omega,\ Z \in \mathcal{Z} := \{Z \in -\mathcal{S}_{++} :\ Z\mathbf{1} = -\mathbf{1}\} \tag{33}$$

$$F^*(Z) + F(\tilde{L}_X + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top) + 1 - \operatorname{tr}(BZ) \leq 0. \tag{34}$$

Optimization algorithm Solving this optimization efficiently requires appropriately leveraging the structure in the problem. Since both M^{**} and Ω involve an inner optimization, it will be very inefficient if we nest their evaluation inside the overall procedure. Noting that both M^* and Ω^* have a closed form, we will dualize these terms and introduce two Lagrange multipliers to enforce the inequalities $\Omega(C_Z) \leq \delta_\Omega$ and (34):

$$\min_{X \in \mathcal{X}, Z \in \mathcal{Z}} \max_{\alpha \geq 0, \gamma \geq 0} \max_{U, S} \operatorname{tr}(U^\top(A + X)) - M^*(U) + \alpha \operatorname{tr}(S^\top C_Z) - \alpha\Omega^*(S) \tag{35}$$

$$- \alpha\delta_\Omega + \gamma F^*(Z) + \gamma F(\tilde{L}_X + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top) - \gamma \operatorname{tr}(BZ) + \gamma \tag{36}$$

$$\overset{\Psi := \alpha S}{\Longleftrightarrow} \min_{X \in \mathcal{X}, Z \in \mathcal{Z}} \max_{\alpha \geq 0, \gamma \geq 0} \max_{U, \Psi} \operatorname{tr}(U^\top(A + X)) - M^*(U) + \operatorname{tr}(\Psi^\top C_Z) - \alpha\Omega^*(\Psi/\alpha) \tag{37}$$

$$- \alpha\delta_\Omega + \gamma F^*(Z) + \gamma F(\tilde{L}_X + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top) - \gamma \operatorname{tr}(BZ) + \gamma. \tag{38}$$

The introduction of Ψ makes the resulting objective concave in all the max variables $(\alpha, \gamma, U, \Psi)$, noting that $\alpha\Omega^*(\Psi/\alpha)$ is the perspective function of Ω^*, hence convex. This is not the case in (35) because $\alpha\Omega^*(S)$ is not jointly convex. Finally, swapping the min and max in (37), we obtain the dual objective. Fixing the max variables, we can 1) minimize over X by projected quasi-Newton because the objective is smooth and strongly convex, and the domain $\mathcal{X}$ is simple (Appendix A.2); 2) minimize over Z, which admits a *closed-form* solution. We defer the details to Appendix A.3. Finally, we maximize over $(\alpha, \gamma, U, \Psi)$ by L-BFGS-B [54], terminating once the dual objective gets positive.

Remark on connected graph. As we set up above, the graphs under consideration are always connected. This is the case for all the (original) graphs in our datasets. If a perturbation makes a graph disconnected, then both Ω and OGW will be infinity. Therefore the attacker in Algorithm 1 will automatically avoid dropping an edge e that disconnects a graph, because the corresponding V_e in Algorithm 2 will be infinity. As for the certificate, since $F(\Phi) = -\log\det(\Phi)$, the term $F(\tilde{L}_X + \frac{1}{n}\mathbf{1}\mathbf{1}^\top)$ in (34) creates a barrier to ensure $\tilde{L}_X + \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is positive definite, hence connected.

Table 1: Statistics of the datasets

dataset	# graphs	# class	# features	ave. edge	min edge	max edge	avg. node	min node	max node
MUTAG	188	2	7	38	20	66	17.5	10	28
PTC-MR	344	2	18	25.0	2	142	13.0	2	64
COX2	467	2	38	86.0	68	118	41.0	32	56
BZR	405	2	56	74.0	26	120	35.0	13	57

Table 2: Graph classification accuracy

	Dataset	Graph Kernels		GW based SVM		OGW based SVM	
		GK (k=3)	WL	SP	RD	SP	RD
Vec.	BZR	78.8 ± 0.5	78.5 ± 0.6	78.7 ± 0.4	79.0 ± 3.0	78.7 ± 0.4	78.7 ± 0.4
Attr.	COX2	78.2 ± 0.4	78.2 ± 0.4	78.2 ± 0.4	78.2 ± 0.4	78.2 ± 0.4	78.2 ± 0.4
Disc.	MUTAG	66.5 ± 0.9	78.8 ± 4.8	67.5 ± 2.2	72.1 ± 4.8	76.1 ± 4.8	79.8 ± 5.6
Attr.	PTC-MR	61.3 ± 2.8	61.3 ± 2.8	56.1 ± 0.6	56.4 ± 1.3	57.9 ± 5.4	59.0 ± 3.0

5 Experimental Results

The goal of our experiments lies in two folds: i) verify that resistance distance (RD) well characterizes graph structures and performs better than or as well as SP distance in classification and barycenter problems, when applied under OGW and GW; ii) demonstrate the effectiveness of our attack and certificate algorithms on real datasets, thereby confirming the tightness of our relaxations. The code is available at [55].

Datasets. We experimented on four graph datasets whose statistics are given in Table 1 [56]. They contain a collection of molecules where the vertices represent atoms and edges are chemical bonds. The class label represents certain property of the molecules, *e.g.*, mutagenic effect on a specific bacterium (MUTAG) and carcinogenicity of compounds for male rats (PTC-MR). BZR and COX2 consist of ligands for the benzodiazepine receptor and cyclooxygenase-2 inhibitors, respectively.

5.1 Effectiveness of resistance distance in classification

To verify the effectiveness of resistance distance, we first followed [16] to study the graph classification accuracy of an SVM, whose kernel $k(\mathcal{C}, \mathcal{D})$ is defined as $\exp\left(-\gamma \cdot \mathsf{OGW}(\mathcal{C}, \mathcal{D})\right)$ and the base measure for OGW is RD or SP distance. We split the dataset into 75% / 25% for training / testing, and tuned the γ and regularizer weight in SVM by 5-fold cross validation on the training set.

The accuracy achieved by SVM based on GW and OGW is presented in Table 2. In addition, we also present graph kernels including graphlet sampling [57] and Weisfeiler-Lehman method [9] as the baselines. Clearly, taking resistance distance as the base measure produces similar or better performance than SP distance under both GW and OGW.

5.2 Effectiveness of resistance distance in barycenter

Furthermore, we also validated the resistance distance in the barycenter problem. Given a set of structured data represented as graphs $\{\mathcal{D}_i : i = 1, 2, \ldots, S\}$, it finds the Fréchet mean defined as $\arg\min_{\mathcal{C}} \sum_{i=1}^{S} \lambda_i d(\mathcal{C}, \mathcal{D}_i)$, where d can be GW and OGW, and uniform weights λ_i are endowed on all the examples as in [12]. We generated 8 cycle-like graphs with structured noises from order 15 to 25 (Figure 1), and constructed the barycenters of 20 nodes with respect to GW and OGW discrepancies under the base measure of SP and RD. Following Section 3 of [24], we took block coordinate update between C and P_i in (6), where the former has a closed form and the latter is locally optimized inside GW and OGW.

Figure 2 shows the results of constructed barycenter with different discrepancies and base measures. Clearly, both SP and RD capture the key structure property (cycle graph) with some additional structural noise under GW and OGW. It is hard to differentiate which structure is better in the context of noised samples.

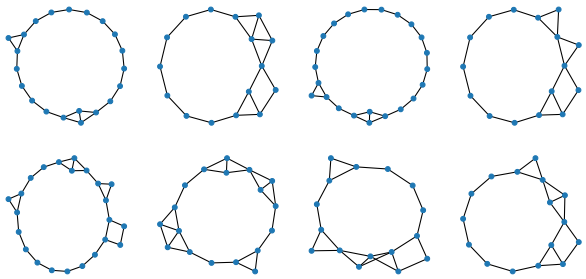
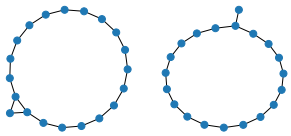
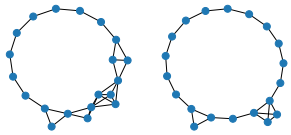


Figure 1: Synthetic graph samples



(a) GW + SP (b) GW + RD



(c) OGW + SP (d) OGW + RD

Figure 2: Barycenters

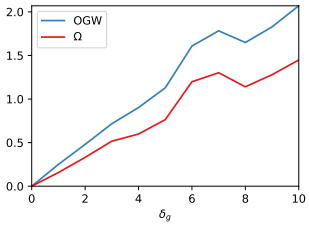
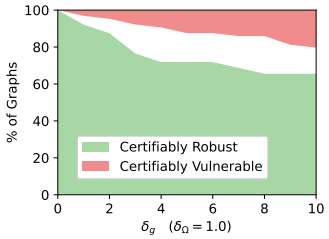
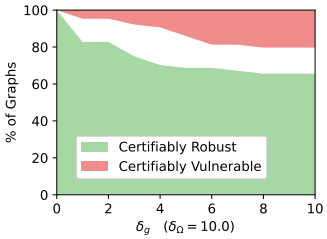


Figure 3: Gap between Ω and OGW



(a) Robust training and $\delta_\Omega = 1$



(b) Robust training and $\delta_\Omega = 10$

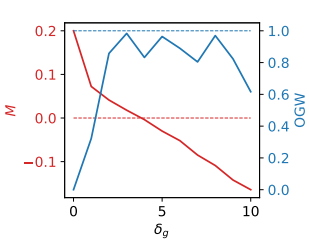
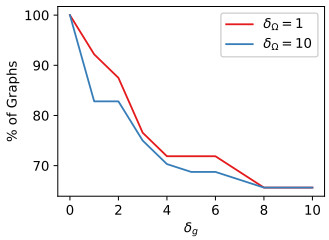
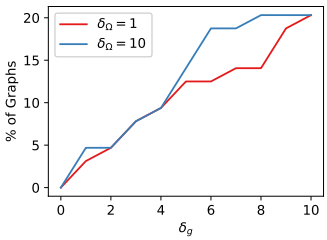


Figure 4: F and Ω in attack on graph No. 125 of MUTAG



(c) Certificate



(d) Attack

Figure 5: Fraction of robust and vulnerable graphs on MUTAG

5.3 Performance of certificates and attacks

We next measure the effectiveness of our robustness certificate and attack algorithm jointly, and the criterion is the fraction of graphs that are neither certified as robust nor vulnerable (*i.e.*, successfully attacked). Due to space limitation, we only report the result of MUTAG in the main paper, and defer the result of the other three datasets to Appendix B. To the best of our knowledge, there is no existing certification algorithm that addresses GW or OGW budgets, except generic global optimization techniques which do not scale well when nested inside the overall problem (7).

Settings. Following [7], we split each dataset into 30%, 20%, and 50% for training, validation, and testing, respectively. A GCN model was then learned using a single linear convolutional layer with 64 hidden nodes, followed by average pooling. [7] also constructed M^{**} with ReLU activation, but showed larger gap than linear activation. Since our focus here is on the tightness of certificate and attack under OGW, we would like to insulate the complication from ReLU and hence focused on linear activation. Following [5, 36], the GCN is trained with a hinge loss that promotes large margin from (1) for robustness: $\sum_{c \neq y} \max\{0, 1 + \max_A \{\mathcal{G}_c(A) - \mathcal{G}_y(A)\}\}$, where A is optimized under the budget of $\delta_l = 1$ and $\delta_g = 10$.

Results. To start with, we empirically checked the tightness of convexified OGW. We took the first graph in the MUTAG dataset, and randomly perturbed its structure (adding or deleting edges) for 20 times with the global budget δ_g varied from 1 to 10. Figure 3 shows the average values of OGW and Ω with resistance distance used as the base measure. As δ_g increases, both OGW and Ω grow, but not monotonically. This is consistent with the fact that perturbing multiple edges may just lead to small changes in GW and OGW due to isomorphism. Moreover, the gap between OGW and Ω gets wider with higher δ_g, but their relative difference remains almost intact.

Figure 5 shows the fraction of certifiably robust graphs in the lower green area, and 100 minus the fraction of certified vulnerable graphs in the upper red area. This leaves the white area showing the gap of undetermined graphs, and a narrow white area indicates the effectiveness of both the certificate and the attacker. Here, we fixed $\delta_l = 1$, which is sufficient for the global budget δ_g on MUTAG.

Sub-plots (a) and (b) present the result for $\delta_\Omega = 1$ and 10, respectively. The gap grows with δ_g and stays below 20% of the graphs. Sub-plots (c) and (d) provide a clearer comparison, between different values of δ_Ω, over the performance of certificate and attack. Increasing the value of δ_Ω enlarges the feasible domain in (33), allowing more (less) graphs to be certified vulnerable (robust).

Figure 4 shows, for graph No. 125 in MUTAG, the value of OGW and M as δ_g is increased and $\delta_\Omega = 1$. Interestingly, the increase of δ_g does not monotonically consume more budget in Ω, which is consistent with the property of Ω.

We additionally compared the performance of our vanilla and robust one-layer GCN model with an MLP model with node feature only, and two other state-of-the-art models, namely MemGNN [58] and FactorGCN [59]. The results are deferred to Appendix E.

6 Conclusion

We designed a new convex lower bound for the orthogonal Gromov-Wasserstein distance based on Fenchel biconjugation. Combined with a convex relaxation of the resistance distance using matching loss, it provides a tight certificate of robustness for graph classification with GCNs.

Future work can extend the approach to certifying distributional robustness, and further leverage the closed form of resistance distance for provable optimization on graphs. It is noteworthy that our relaxation of OGW with resistance distance is orthogonal to the graph classifier itself. When the classifier is changed, one will only need to re-derive the expression of M^{**} for the risk or margin. The constraints in (30) and (32), however, remain intact, and that part is our contribution. It will also be interesting to extend GCN to multiple layers GCN, by, *e.g.*, leveraging the convex envelop discussed in Appendix D of [7].

Acknowledgments and Disclosure of Funding

This material is based upon work supported by the National Science Foundation under Grant No. 1910146.

References

[1] D. Zügner, A. Akbarnejad, and S. Günnemann. Adversarial attacks on neural networks for graph data. *In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 2847–2856. ACM, 2018.

[2] K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin. Topology attack and defense for graph neural networks: An optimization perspective. *In International Joint Conference on Artificial Intelligence (IJCAI)*. 2019.

[3] J. Jia, B. Wang, X. Cao, and N. Z. Gong. Certified robustness of community detection against adversarial structural perturbation via randomized smoothing. *In International Conference on the World Wide Web (WWW)*. 2020.

[4] A. Bojchevski, J. Klicpera, and S. Günnemann. Efficient robustness certificates for discrete data: Sparsity-aware randomized smoothing for graphs, images and more. *In International Conference on Machine Learning (ICML)*. 2020.

[5] A. Bojchevski and S. Günnemann. Certifiable robustness to graph perturbations. *In Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[6] D. Zügner and S. Günnemann. Certifiable robustness of graph convolutional networks under structure perturbations. *In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 2020.

[7] H. Jin, Z. Shi, A. Peruri, and X. Zhang. Certified robustness of graph convolution networks for graph classification under topological attacks. *In Neural Information Processing Systems (NeurIPS)*. 2020.

[8] S. V. N. Vishwanathan, N. N. Schraudolph, I. R. Kondor, and K. M. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11(40):1201–1242, 2010.

[9] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

[10] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[11] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *In Advances in Neural Information Processing Systems (NeurIPS)*. 2016.

[12] G. Peyre, M. Cuturi, and J. Solomon. Gromov-Wasserstein averaging of kernel and distance matrices. *In International Conference on Machine Learning (ICML)*. 2016.

[13] F. Memoli. Gromov-Wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11:417–487, 2011.

[14] Y. Yan, W. Li, H. Wu, H. Min, M. Tan, and Q. Wu. Semi-supervised optimal transport for heterogeneous domain adaptation. *In International Joint Conference on Artificial Intelligence (IJCAI)*. 2018.

[15] D. Alvarez-Melis and T. Jaakkola. Gromov-Wasserstein alignment of word embedding spaces. *In Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2018.

[16] T. Vayer, L. Chapel, R. Flamary, R. Tavenard, and N. Courty. Optimal transport for structured data with application on graphs. *In International Conference on Machine Learning (ICML)*. 2019.

[17] D. Ezuz, J. Solomon, V. G. Kim, and M. Ben-Chen. GWCNN: A metric alignment layer for deep shape analysis. *Comput. Graph. Forum*, 36(5):49–57, 2017.

[18] S. Cohen and D. Sejdinovic. On the Gromov-Wasserstein distance and coupled deep generative models. *In NeurIPS 2019 Workshop on Optimal Transport & Machine Learning*. 2019.

[19] H. Xu, D. Luo, H. Zha, and L. Carin. Gromov-Wasserstein learning for graph matching and node embedding. *In International Conference on Machine Learning (ICML)*. 2019.

[20] H. Xu, D. Luo, and L. Carin. Scalable Gromov-Wasserstein learning for graph partitioning and matching. *In Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[21] A. Sinha, H. Namkoong, and J. Duchi. Certifying some distributional robustness with principled adversarial training. *In International Conference on Learning Representations (ICLR)*. 2018.

[22] E. Wong, F. Schmidt, and Z. Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. *In International Conference on Machine Learning (ICML)*. 2019.

[23] S. Chowdhury and F. Mémoli. The Gromov–Wasserstein distance between networks and stable network invariants. *Information and Inference: A Journal of the IMA*, 8(4):757–787, 2019.

[24] H. Jin, Z. Yu, and X. Zhang. Orthogonal Gromov-Wasserstein discrepancy with efficient lower bound. *arXiv:2205.05838*, 2022.

[25] G.-H. Lee, Y. Yuan, S. Chang, and T. Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. *In Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[26] D. J. Klein and M. Randic. Resistance distance. *Journal of Mathematical Chemistry*, 12(81–95):1–31, 1993.

[27] U. von Luxburg, A. Radl, and M. Hein. Hitting and commute times in large random neighborhood graphs. *Journal of Machine Learning Research*, 15(52):1751–1798, 2014.

[28] P. Auer, M. Herbster, and M. K. Warmuth. Exponentially many local minima for single neurons. *In Advances in Neural Information Processing Systems 9*. 1996.

[29] R. Bunel, A. De Palma, A. Desmaison, K. Dvijotham, P. Kohli, P. H. S. Torr, and M. P. Kumar. Lagrangian decomposition for neural network verification. *In Conference on Uncertainty in Artificial Intelligence (UAI)*. 2020.

[30] S. Singla and S. Feizi. Second-order provable defenses against adversarial attacks. *In International Conference on Machine Learning (ICML)*. 2020.

[31] E. Wong and J. Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *In International Conference on Machine Learning (ICML)*. 2018.

[32] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel. Efficient neural network robustness certification with general activation functions. *In Advances in Neural Information Processing Systems (NeurIPS)*. 2018.

[33] T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, D. Boning, I. S. Dhillon, and L. Daniel. Towards fast computation of certified robustness for relu networks. *In International Conference on Machine Learning (ICML)*. 2018.

[34] K. (Dj)Dvijotham, R. Stanforth, S. Gowal, T. Mann, and P. Kohli. A dual approach to scalable verification of deep networks. *In Conference on Uncertainty in Artificial Intelligence (UAI)*. 2018.

[35] G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer. Reluplex: An efficient smtsolver for verifying deep neural networks. *In International Conference on Computer Aided Verification (CAV)*. 2017.

[36] D. Zügner and S. Günnemann. Certifiable robustness and robust training for graph convolutional networks. *In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 246–256. ACM, 2019.

[37] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. Certified adversarial robustness via randomized smoothing. *In International Conference on Machine Learning (ICML)*. 2019.

[38] B. Li, C. Chen, W. Wang, and L. Carin. Certified adversarial robustnesswith additive noise. *In Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[39] M. Lécuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. *In IEEE Symposium on Security and Privacy (S&P)*. 2019.

[40] A. Raghunathan, J. Steinhardt, and P. Liang. Certified defenses against adversarial examples. *In International Conference on Learning Representations (ICLR)*. 2018.

[41] A. Raghunathan, J. Steinhardt, and P. Liang. Semidefinite relaxations for certifying robustness to adversarial examples. *In Advances in Neural Information Processing Systems (NeurIPS)*. 2018.

[42] S. Singla and S. Feizi. Robustness certificates against adversarial examples for ReLU networks. *arXiv:1902.01235*, 2019.

[43] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. *In International Conference on Machine Learning (ICML)*. 2017.

[44] C. Anil, J. Lucas, and R. Grosse. Sorting out Lipschitz function approximation. *In International Conference on Machine Learning (ICML)*. 2019.

[45] H. D. Sherali and C. H. Tuncbilek. A reformulation-convexification approach for solving nonconvex quadratic programming problems. *Journal of Global Optimization*, 7(1):1–31, 1995.

[46] T. Vayer, R. Flamary, R. Tavenard, L. Chapel, and N. Courty. Sliced Gromov-Wasserstein. *In Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[47] T. C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *In Econometrica, Journal of the Econometric Society*. 1957.

[48] S. W. Hadley, F. Rendl, and H. Wolkowicz. A new lower bound via projection for the quadratic assignment problem. *Mathematics of Operations Research*, 17(3):727–739, 1992.

[49] F. Rendl and H. Wolkowicz. Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem. *Mathematical Programming: Series A and B*, 53(1–3):63–78, 1992.

[50] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.

[51] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

[52] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142:397–434, 2013.

[53] K. T. Arasu and M. T. Mohan. Optimization problems with orthogonal matrix constraints. *Numerical Algebra, Control & Optimization*, 8(4):413–440, 2018.

[54] R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[55] Online Supplementary. Supplementary material including code. `https://github.com/cshjin/cert_ogw`.

[56] TUDataset. Tudataset. `https://chrsmrrs.github.io/datasets/docs/home/`.

[57] N. Przulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, Jan 2007.

[58] A. H. Khasahmadi, K. Hassani, P. Moradi, L. Lee, and Q. Morris. Memory-based graph networks. *In International Conference on Learning Representations*. 2020.

[59] Y. Yang, Z. Feng, M. Song, and X. Wang. Factorizable graph convolutional networks. *Advances in Neural Information Processing Systems*, 33:20286–20296, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary Material

All code and data are available at

`https://github.com/cshjin/cert_ogw`

A Proofs and detailed algorithms

A.1 Discussion on commute time

The commute time between node i and j is π times their resistance distance, where π is the volume of the graph defined as $\pi := \mathbf{1}^\top \tilde{A}\mathbf{1}$. We can treat π as a constant, because it is equal to $\mathbf{1}^\top A\mathbf{1} + \mathbf{1}^\top X\mathbf{1}$, and we can enumerate the value of $\mathbf{1}^\top X\mathbf{1}$ for optimization over X. When we only allow adding edges, this corresponds exactly to the global budget of new edges to add. Overall, the range of $\mathbf{1}^\top X\mathbf{1}$ is narrow because the global budget is low.

A.2 Projection to $\mathcal{X}$

Let $s_{ij} = 1$ if $H_{ij} = 0$, and $s_{ij} = -1$ if $H_{ij} = 1$. Set $s_{ii} = 0$. It is easy to show that

$$\mathcal{X} = \{X \in \mathbb{R}^{n\times n} : X_{ii} = 0,\ X_{ij} + H_{ij} \in [0,1],\ \operatorname{tr}(SX) \le 2\delta_g\}. \tag{39}$$

To project an $X^{(t)} \in \mathbb{R}^{n\times n}$ to $\mathcal{X}$, we alternate between two projections:

1. Project to

$$\{X \in \mathbb{R}^{n\times n} : X_{ii} = 0,\ X_{ij} + H_{ij} \in [0,1]\}. \tag{40}$$

Denote the projection image as Z. Then Z_{ii} to 0, and

$$Z_{ij} = \text{median}(X^{(t)}_{ij}, -H_{ij}, 1 - H_{ij}). \tag{41}$$

2. Project Z to

$$\{X \in \mathbb{R}^{n\times n} : X_{ii} = 0,\ \operatorname{tr}(SX) \le 2\delta_g\}. \tag{42}$$

The resulting X^{proj} is

$$X^{\text{proj}} = \begin{cases} Z & \text{if } \operatorname{tr}(SZ) \le 2\delta_g \\ Z - \|S\|^{-2}\left(\operatorname{tr}(SZ) - 2\delta_g\right)S & \text{otherwise} \end{cases}. \tag{43}$$

We can terminate the alternating between 1 and 2 when $\left\|X^{\text{proj}} - X^{(t)}\right\|$ falls below some threshold. Usually 20 rounds will be sufficient.

A.3 Closed-form Solution for Z in (37)

We copy (37) for convenience:

$$\max_{\alpha\ge 0,\gamma\ge 0}\ \max_{U,\Psi}\ \min_{X\in\mathcal{X},Z\in\mathcal{Z}}\ \operatorname{tr}(U^\top(A+X)) - M^*(U) + \operatorname{tr}(\Psi^\top C_Z) - \alpha\Omega^*(\Psi/\alpha) \tag{44}$$

$$- \alpha\delta_\Omega + \gamma F^*(Z) + \gamma F(\tilde{L}_X + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top) - \gamma\operatorname{tr}(BZ) + \gamma \tag{45}$$

Given $(\alpha, \gamma, U, \Psi)$, the optimization over Z is

$$\min_{Z: Z\prec\mathbf{0}, Z\mathbf{1}=-\mathbf{1}} \gamma^{-1}\operatorname{tr}(\Psi^\top C_Z) + F^*(Z) - \operatorname{tr}(BZ). \tag{46}$$

It is easy to see that C_Z is linear in Z (not only affine):

$$C_Z = -Z_{ii} - Z_{jj} + 2Z_{ij}. \tag{47}$$

So we can define a linear operator $\mathcal{A}$ as $\mathcal{A}(Z) = C_Z$, and then $\operatorname{tr}(\Psi^\top C_Z) = \operatorname{tr}(\mathcal{A}^*(\Phi)Z)$, where $\mathcal{A}^*$ is a adjoint of $\mathcal{A}$ and can be derived as follows for a symmetric Φ:

$$\mathcal{A}^*(\Psi) = \begin{cases} 2\Psi_{ij} & \text{if } i \neq j \\ -2\sum_j \Psi_{ij} & \text{if } i = j \end{cases}. \tag{48}$$

So clearly $\mathcal{A}^*(\Psi)\mathbf{1} = \mathbf{0}$. By the definition of B in Section 3.2, we also derive $B\mathbf{1} = \mathbf{0}$.

We next use Lagrangian multiplier μ to enforce $Z\mathbf{1} = -\mathbf{1}$. To ensure symmetry, we equivalently enforce $Z\mathbf{1} + Z^\top\mathbf{1} + 2\mathbf{1} = \mathbf{0}$. Letting $J = B - \gamma^{-1}\mathcal{A}^*(\Psi)$, the Lagrangian of (46) becomes

$$\max_{\mu} \min_{Z \prec \mathbf{0}} \gamma^{-1} \operatorname{tr}(\Psi\mathcal{A}(Z)) + F^*(Z) - \operatorname{tr}(BZ) - \mu^\top(Z\mathbf{1} + Z^\top\mathbf{1} + 2\mathbf{1}) \tag{49}$$

$$= -\min_{\mu} \max_{Z \prec \mathbf{0}} \left\{\operatorname{tr}((J + \mu\mathbf{1}^\top + \mathbf{1}\mu^\top)Z) - F^*(Z) + 2\mathbf{1}^\top\mu\right\} \tag{50}$$

$$= -\min_{\mu} \left\{F(J + \mu\mathbf{1}^\top + \mathbf{1}\mu^\top) + 2\mathbf{1}^\top\mu\right\}. \tag{51}$$

The optimal Z is

$$J + \mu\mathbf{1}^\top + \mathbf{1}\mu^\top = \nabla F^*(Z) = -Z^{-1} \qquad \Rightarrow \qquad Z = -(J + \mu\mathbf{1}^\top + \mathbf{1}\mu^\top)^{-1}. \tag{52}$$

To solve μ, notice $J\mathbf{1} = \mathbf{0}$. Taking the derivative of (51) with respect to μ, we get

$$-2(J + \mu\mathbf{1}^\top + \mathbf{1}\mu^\top)^{-1}\mathbf{1} + 2\mathbf{1} = \mathbf{0} \qquad \Rightarrow \qquad \mu = \tfrac{1}{2n}\mathbf{1}. \tag{53}$$

This implies that the optimal Z is

$$Z^\star = -(J + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)^{-1} = -(B - \gamma^{-1}\mathcal{A}^*(\Psi) + \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)^{-1}. \tag{54}$$

B More Experimental Results

B.1 Results of certificate and attack from other datasets than MUTAG

As part of Section 5.3, the certificate and attack results for BZR, COX2 and PTC-MR are shown in Figure 6 to 8. They corroborate and reinforce the conclusions in Section 5.3. Due to the variance of characteristics in different datasets, we fixed $\delta_l = 5, 5, 1$ for the datasets BZR, COX2 and PTC-MR, respectively.

B.2 Computational complexity

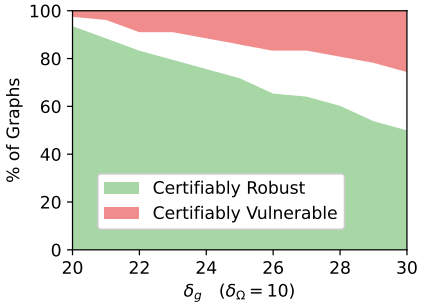
We measured the wall-clock time of certificate from the MUTAG dataset. Figure 9a shows the runtime of each iteration for graphs with different orders. As described in Section 4.1 and Appendix A.2 and A.3, the per-iteration cost of the optimization is $O(n^3)$. Figure 9b further shows the total time taken to find a certificate, *i.e.*, a positive value from our dual objective (44), noting that we can early-stop once a positive value is reached. Overall, the cost is mild.

We implemented the algorithm in Python with wrapped L-BFGS-B algorithm from Scipy, and ran the experiments on a machine with Intel CPU i9-9900X.

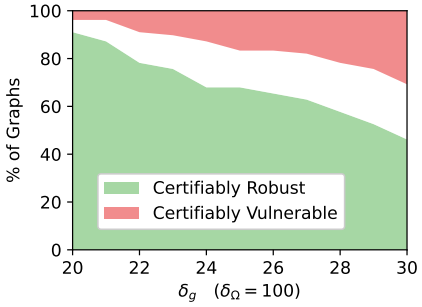
C Reachable graphs given specific budgets

Without the constraints of local and global budget, checking all the reachable graphs essentially finds all possible perturbations under the budget δ_Ω on Ω, which is (NP) hard. Alternatively, we examined the Ω value on the real dataset MUTAG by extracting all the graphs with 12 nodes, and then presented their pairwise Ω distance (first line in each cell) and δ_g (second line) in Figure 10.

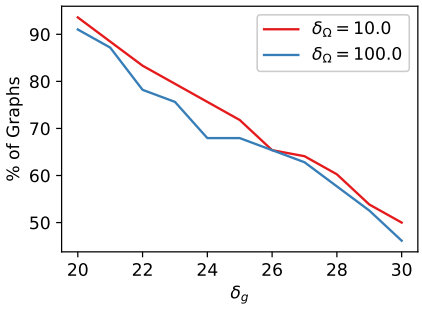
To better visualize the result, Figure 11 and 12 set the budget δ_Ω to 0.5 and 1 respectively, and a darker shade represents a higher value of Ω. A cell is marked with two numbers (red for Ω and black for #perturbed-edge) computed from a pair of reachable graphs, if its Ω value falls below the δ_Ω budget. In Figure 11, we observe that in the first row, the columns 5, 10, 11, 12, 13 exhibit high values of #perturbed-edge, but their Ω value is 0. In these cases the pair of graphs are isomorphic, although their topology differs a lot. We also see a block of four isomorphic graphs in the bottom-right corner.



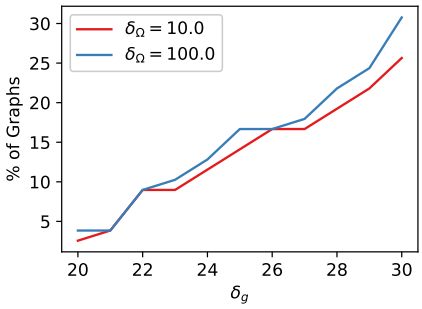
(a) $\delta_\Omega = 10$



(b) $\delta_\Omega = 100$

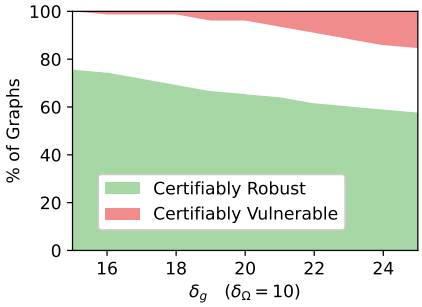


(c) Certificates

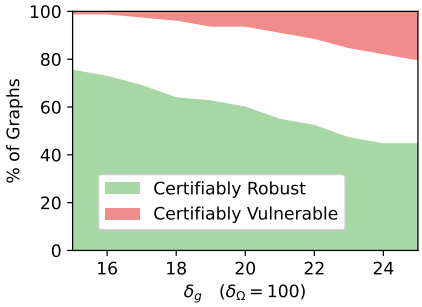


(d) Attacks

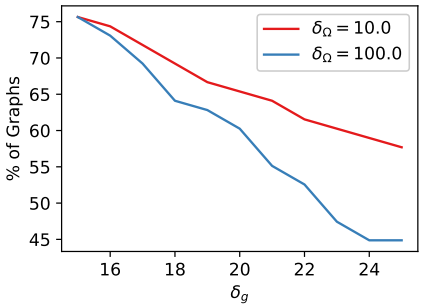
Figure 6: Certificate and attack on BZR ($\delta_l = 5$)



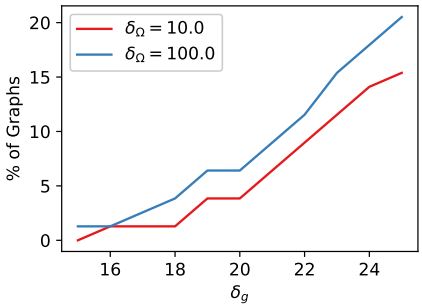
(a) $\delta_\Omega = 10$



(b) $\delta_\Omega = 100$

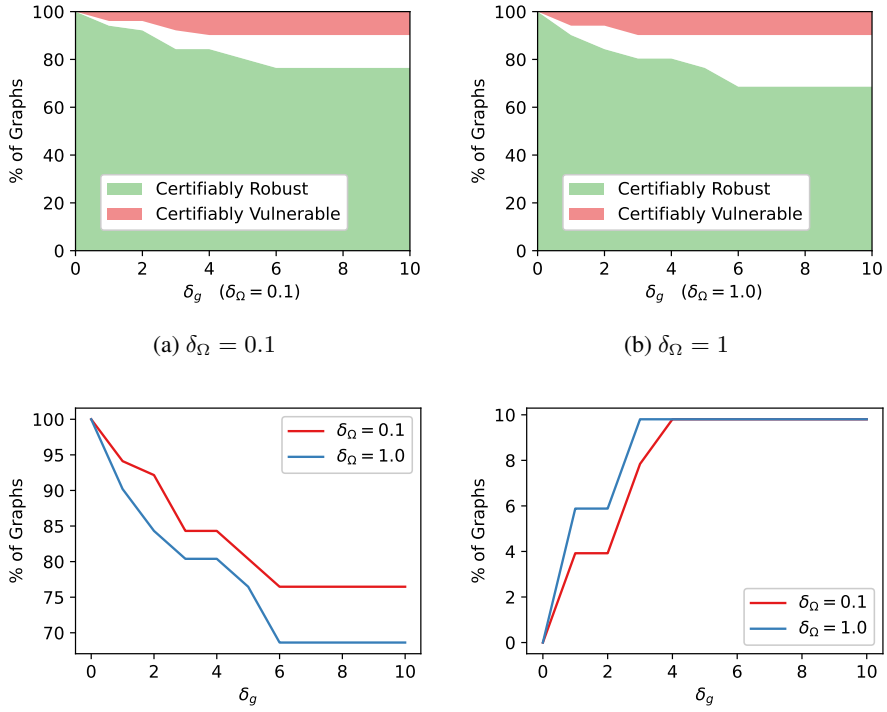


(c) Certificates



(d) Attacks

Figure 7: Certificate and attack on COX2 ($\delta_l = 5$)



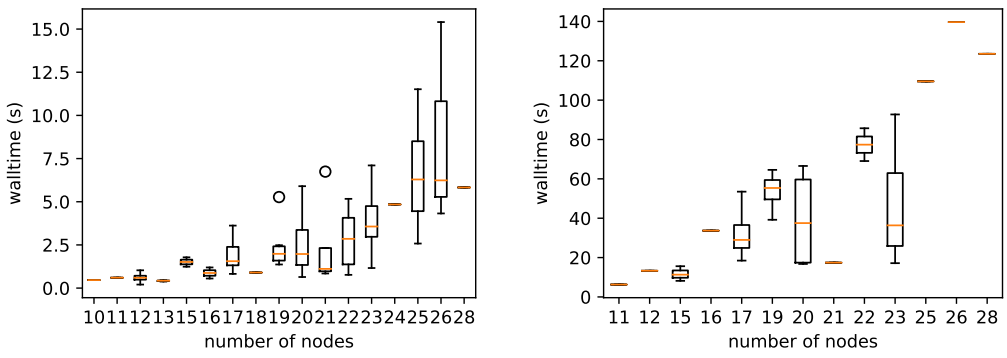
(a) $\delta_\Omega = 0.1$

(b) $\delta_\Omega = 1$

(c) Certificates

(d) Attacks

Figure 8: Certificate and attack on PTC-MR ($\delta_l = 1$)



(a) Time (in seconds) per iteration

(b) Time (in seconds) for entire certificate

Figure 9: Running time for certification (MUTAG)

Comparing Figure 12 with Figure 11, clearly more pairs of graphs become reachable thanks to the increase in δ_Ω.

Similarly, Figure 13 and 14 set the threshold of δ_g to 4 and 8 respectively, and a darker shade represents a higher value of #perturbed-edge. A cell is marked with two numbers (red for #perturbed-edge and black for Ω) computed from a pair of reachable graphs, if its #perturbed-edge falls below the δ_g budget.

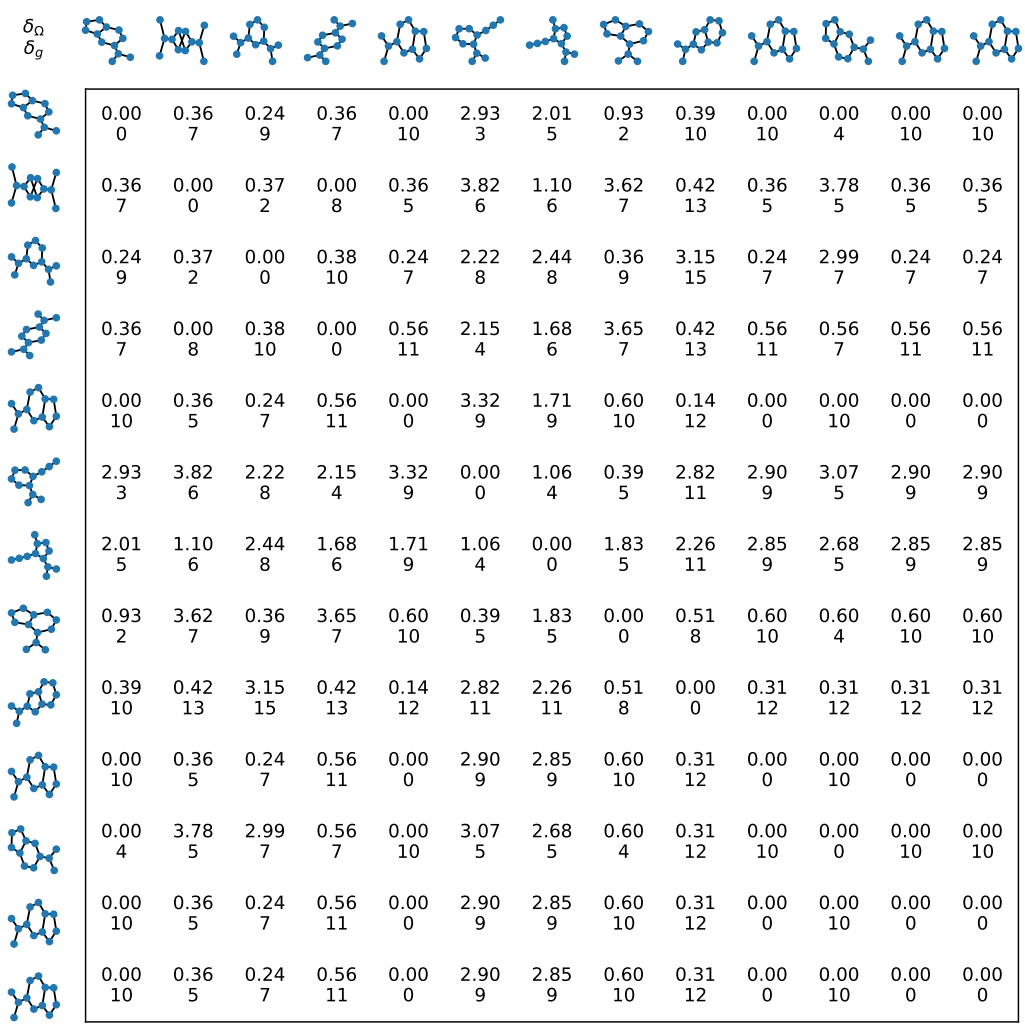


Figure 10: Pairwise Ω distance and δ_g

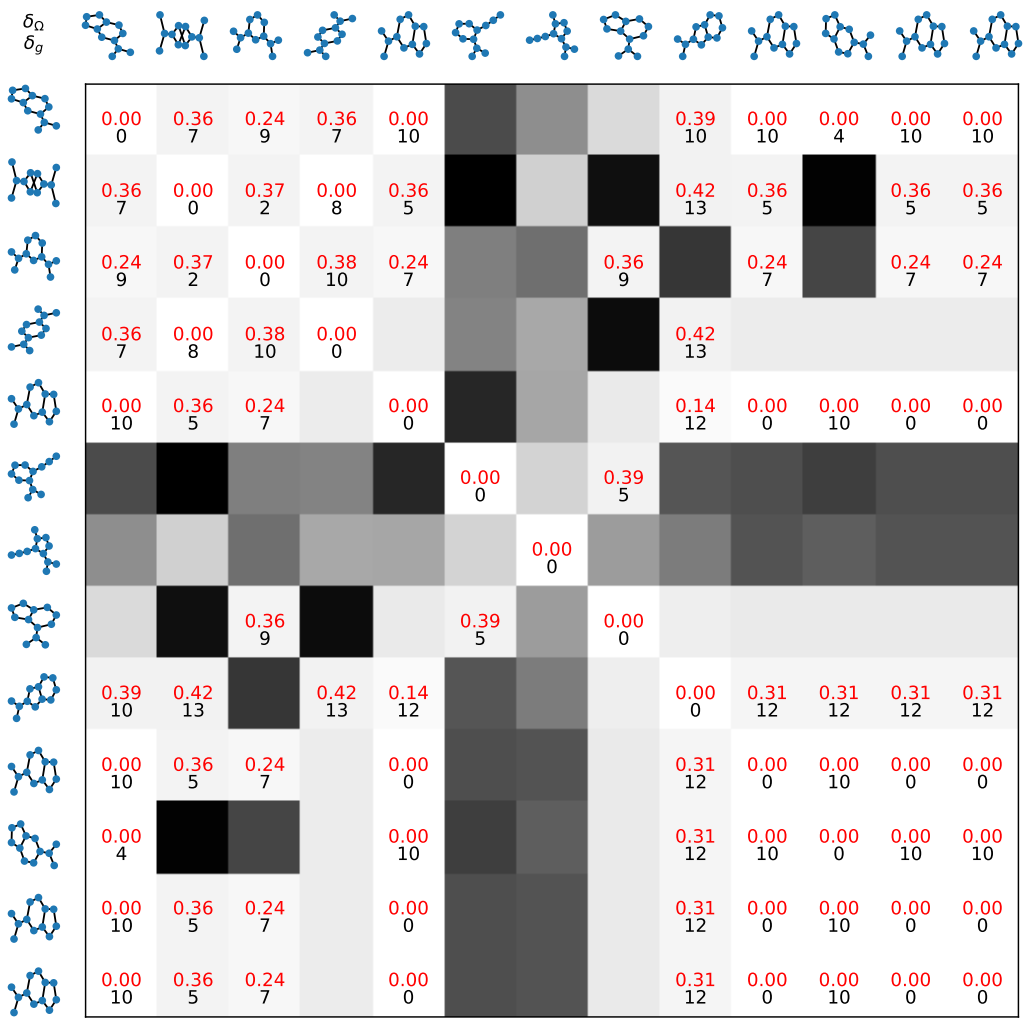


Figure 11: Reachable graphs given $\delta_\Omega = 0.5$

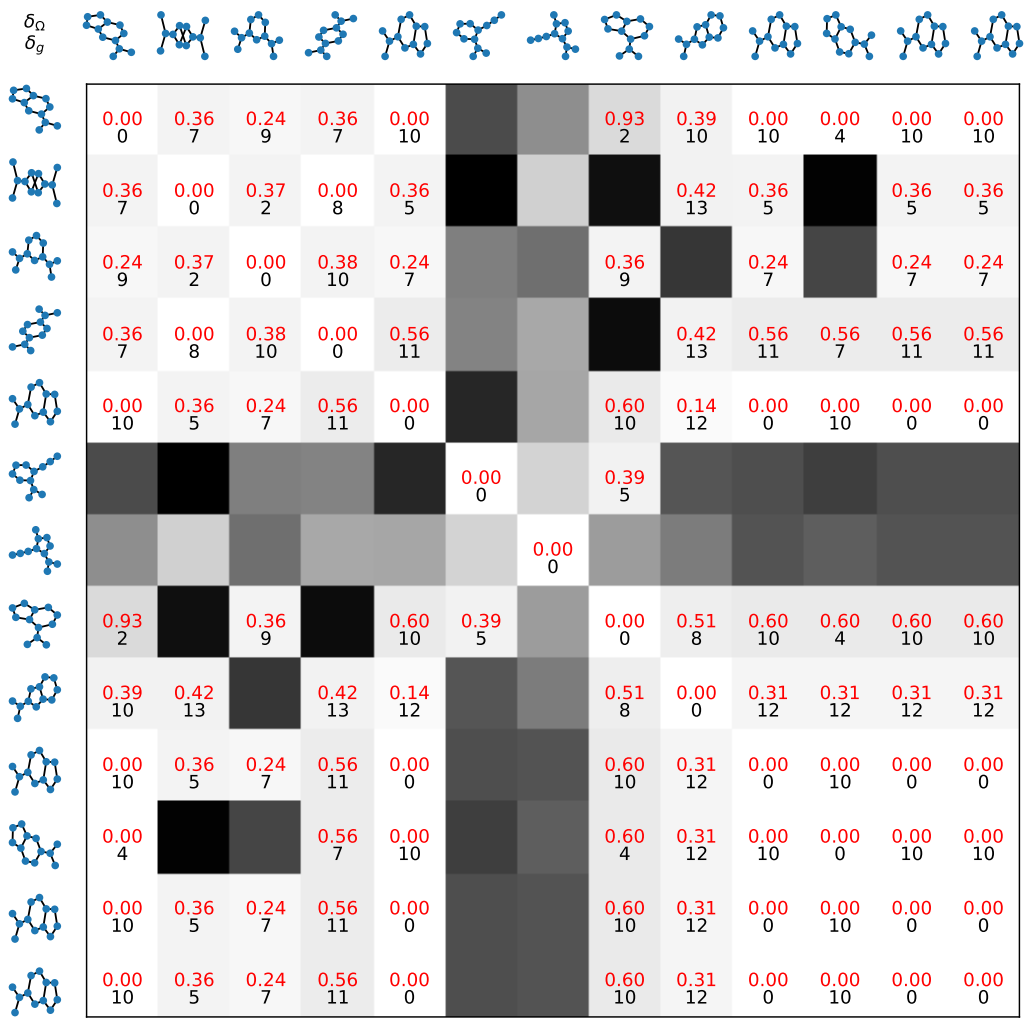


Figure 12: Reachable graphs given $\delta_\Omega = 1$

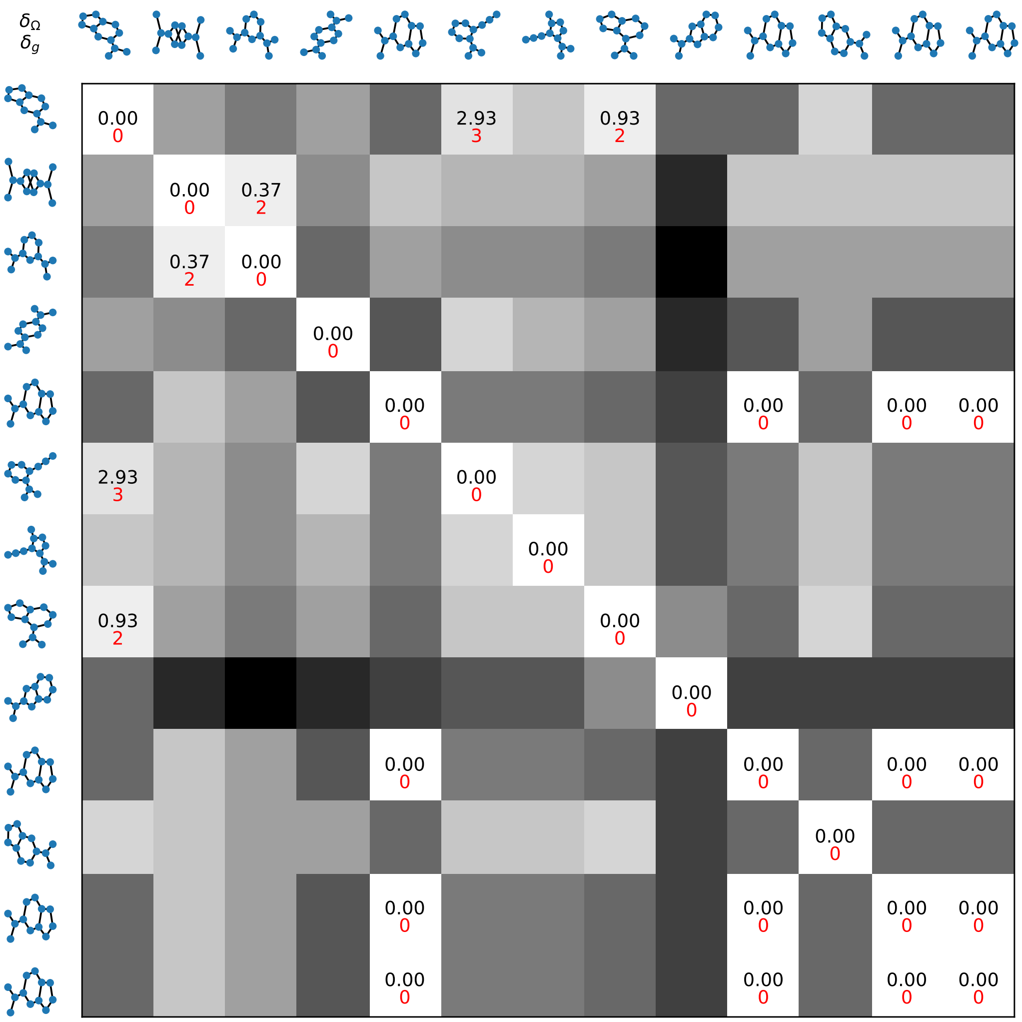


Figure 13: Reachable graphs given $\delta_g = 4$

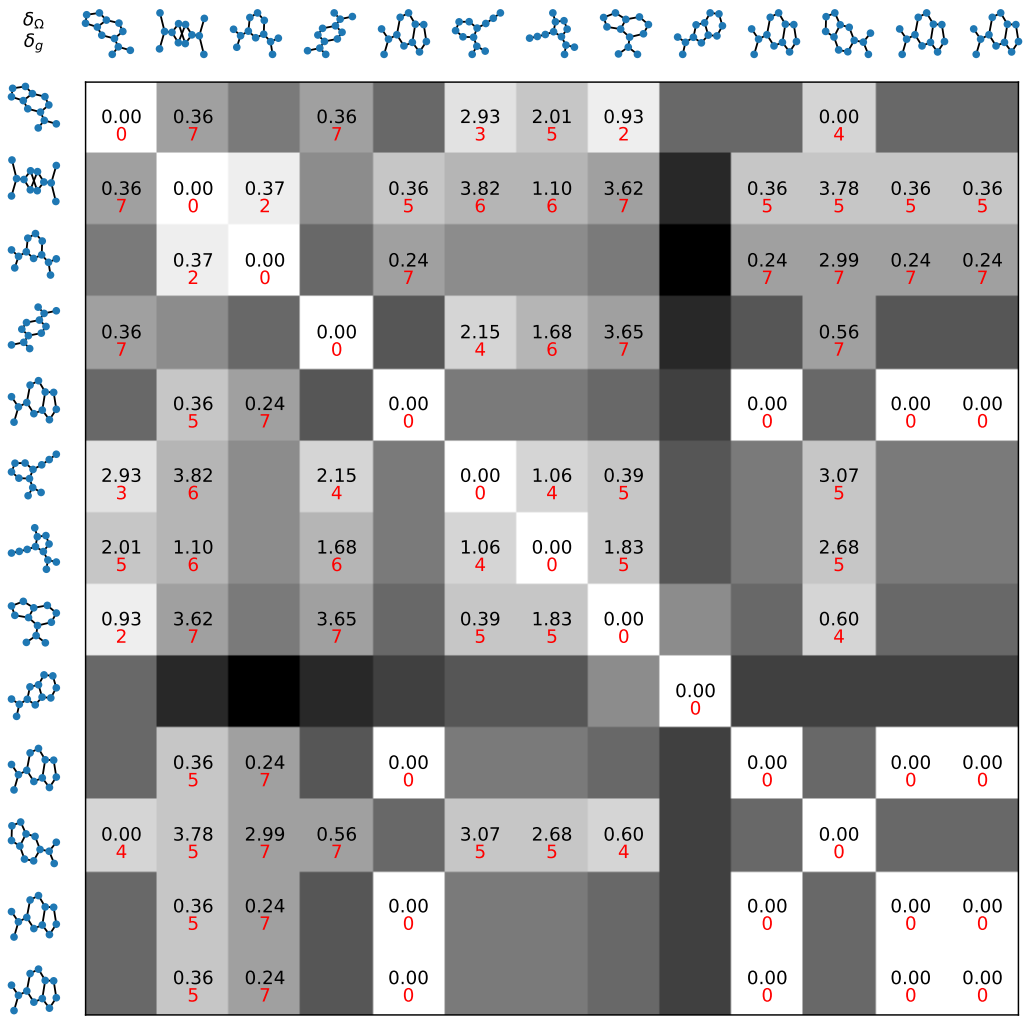
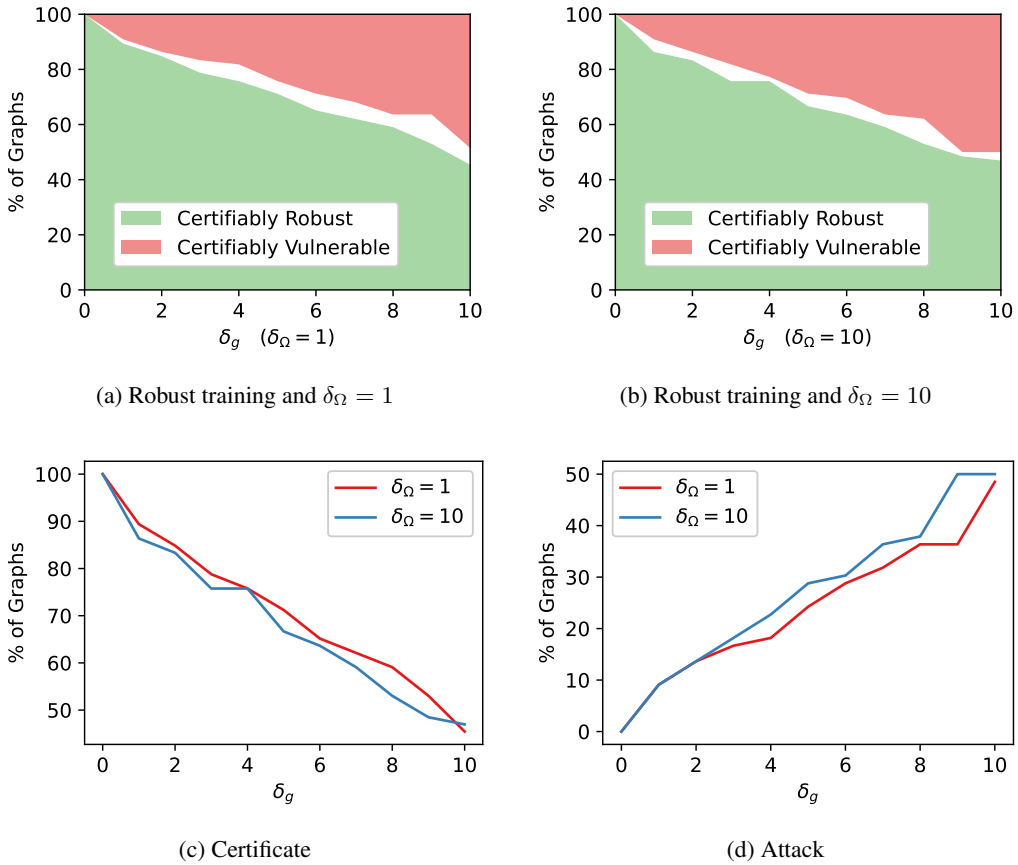


Figure 14: Reachable graphs given $\delta_g = 8$

D Plots for Better Turned Hyperparameter



(a) Robust training and $\delta_\Omega = 1$

(b) Robust training and $\delta_\Omega = 10$

(c) Certificate

(d) Attack

Figure 15: Fraction of robust and vulnerable graphs on MUTAG (with tuned hyperparameter)

E Comparison of Classification Performance

Table 3: Performance of Classification

Dataset	Vanilla-GCN	Robust-GCN	MLP	MemGNN	FactorGCN
BZR	81.8	80.3	79.9	84.7	82.4
COX2	79.9	78.6	78.2	79.0	81.9
MUTAG	69.5	67.4	65.0	77.8	82.6
PTC_MR	57.8	57.8	57.3	59.8	54.6

We compared the performance of our vanilla and robust one-layer GCN model with a MLP model with node feature only, and two other models, namely MemGNN [58] and FactorGCN [59]. To be consistent with our setting, we split the training, validation and test sets into 30, 20, and 50% respectively. All the other hyperparameters followed the standard setting from the papers. Table 3 reports the average accuracy on the test set with 5 runs, where most times the robust model sacrifices only a slight amount of accuracy compared with our vanilla model.