TOWARDS PROVABLY ROBUST COMBINATORIAL OPTIMIZATION SOLVERS: CERTIFICATION AND ENHANCEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Combinatorial optimization (CO) has received increasing attention from the machine learning (ML) community, often regarding accelerating the solving process or discovering heuristics beyond hand-craft. In this paper, we dive into another aspect for bridging ML to CO: certification of the solver's robustness, especially in a solver-agnostic manner. In fact, existing theoretically-grounded certification methods are mainly confined to classification problems, while the robustness of CO solvers has largely been evaluated empirically. We explore the certified robustness concept as well as robustness enhancement strategy for CO solvers. First, we introduce a general definition of certified robustness for CO solvers, and extend the randomized smoothing, a well-established robustness certification technique originally designed for categorical classifiers. Using the objective value attained by the solver as the robustness evaluation metric, we provide theoretical guarantees that bound, in 1-Wasserstein distance, the variation between the corresponding smoothed cumulative distribution functions (CDFs) under bounded perturbations to problem instances. Beyond certification, we leverage the local-instance perspective underlying smoothing and observe that some nearby instances can yield solutions that remain feasible for the given instance while attaining lower objective values on it. Based on this observation, a training-free robustness enhancement method is devised by searching for such easy instances and using their solutions as feasible alternatives for the given instance, without modifying or retraining the solver. Experiments across four CO problems and 14 solvers demonstrate the applicability of the certification framework and the effectiveness of the enhancement method. With a small inference-time sampling budget, the enhancement method reduces the objective value on 50 of 56 attacked instances (89.3%).

1 INTRODUCTION

Combinatorial optimization (CO) [31; 28], as a fundamental problem, seeks the optimal solution from a set of discrete alternatives under certain constraints. Various approaches have been developed, including exact algorithms (e.g., branch and bound [14]), approximation algorithms (e.g., greedy algorithms [18]), heuristic methods [38], and metaheuristics (e.g., evolutionary algorithms [27]). Recently, learning-based models have also emerged to automate the design of effective heuristics [19; 21]. However, recent studies have revealed that CO solvers can be highly sensitive to limited perturbations of problem instances, where small structural or numerical modifications may lead to substantial degradation in solving performance [32; 12; 25]. In particular, ROCO [25] shows that limited modifications to a problem instance can uncover nearby hard instances on which the same solver performs substantially worse, revealing its sensitivity to specific instance structures. These findings motivate studying CO solver robustness through the stability of solving performance under bounded perturbations to problem instances.

Recent work has therefore initiated the evaluation of CO solvers' robustness, establishing empirical frameworks for robustness verification. [12] proposes a sound and efficient model to assess the robustness tailored for differentiable neural networks for CO. [25] conducts empirical analyses on the robustness of combinatorial solvers, presenting a viable robustness metric suitable for a broader range of solvers. However, these efforts focus primarily on empirical robustness, a paradigm that evaluates a solver's resilience by testing its performance against a predefined set of perturbations.

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.

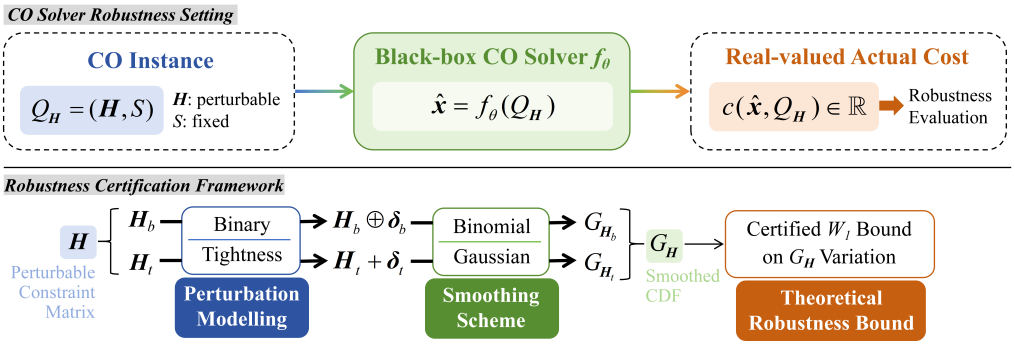


Figure 1: Overview of the CO solver robustness setting and the proposed robustness certification framework. The upper part depicts a black-box CO solver that maps an input instance to a solution, with actual cost serving as the robustness evaluation metric. The lower part depicts perturbation modeling and smoothing scheme on the perturbable constraint matrix, yielding a certified 1-Wasserstein (W_1) bound on the variation between smoothed cumulative distribution functions (CDFs) of actual cost under bounded perturbation.

While practical, this observational approach offers guarantees only for known perturbation types. Its efficacy against unknown perturbation strategies may be limited, and it lacks rigorous theoretical bounds. To establish provable robustness guarantees independent of specific perturbation strategies, we turn to certified robustness [39; 22; 10], which mathematically ensures the stability of model outputs when inputs vary within a fixed range, confining output variation within a computable bound.

Common methods of certified robustness include those based on the Lipschitz constant [15; 41], bound propagation [45; 37], randomized smoothing [10; 22], etc. Among them, randomized smoothing (RS) [22; 10] is a promising approach to achieve certified robustness of large-scale neural networks. Given a base function and an input, RS constructs a smoothed function that is certifiably robust under perturbations within a region defined by the input and the smoothing distribution. RS has been widely applied to various models, e.g., image classification [43] and object detection in vision [9], which motivates us to develop RS-based certified robustness for CO. However, a direct application of the RS framework faces a fundamental barrier, which stems from the fact that the RS-based approach is inherently tied to the input type and output type of the model. The standard and most prevalent formulation of RS certifies robustness for categorical outputs (e.g., image labels) from continuous inputs. In contrast, the inputs to CO solvers are typically structured problem instances, and their outputs are inherently non-categorical, rendering the canonical RS framework inapplicable.

To bridge this gap, we propose a novel certification framework that adapts randomized smoothing to CO solvers, as illustrated in Figure 1. We first establish the CO solver robustness setting by using actual cost (i.e., the objective value attained by the solver) as a solver-agnostic measure of solving performance, and model perturbations to problem instances for robustness analysis. Based on this formulation, we define certified robustness of a CO solver in terms of the allowable variation in actual cost when perturbations to the instance are confined within a prescribed range, and derive corresponding guarantees using smoothing schemes tailored to the perturbation models. The resulting framework treats the solver as a black box, requiring no access to its internal architecture or optimization mechanism, and therefore applies to both conventional and learning-based solvers. Beyond certification, we further leverage the local-instance perspective underlying smoothing to develop a training-free robustness enhancement method that searches for nearby easy instances and uses their solutions as feasible alternatives for the reference instance, improving solving performance at inference time without modifying or retraining the solver. **The contributions of the paper are:**

1) Conceptual advances: We identify the lack of certified robustness guarantees for CO solvers, as existing approaches rely on empirical evaluation and fail to give provable guarantees. We go beyond these prior empirical approaches by making it possible to certify the robustness.

2) Technical novelty: We introduce a robustness certification definition for CO solvers and establish theoretical guarantees under bounded input perturbations using tailored smoothing schemes. We further propose a robustness enhancement method that leverages easy instances to mitigate performance degradation, which requires no costive adversarial training as used in ROCO [25].

3) Practical effectiveness: Experiments across four CO problems and 14 solvers demonstrate the applicability of the certification framework and the effectiveness of the enhancement method. With a small inference-time sampling budget, the enhancement method reduces the objective value on 50 of 56 attacked instances (89.3%).

2 Preliminaries and Related Work

We provide detailed related work in Appendix A and here briefly discuss the most related ones: ROCO [25] which establishes an empirical benchmark for solvers including both learning-based and traditional ones. In contrast, we provide an orthogonal, and more systematic and theoretical framework. Our work also goes beyond the saturated literature on robustness for classifiers [10].

2.1 Randomized Smoothing

The original Randomized Smoothing (RS) [10] method is tailored to certify the robustness of classifiers under the ℓ_2 norm by transforming a base classifier f into a smoothed one g. Given input x, g outputs the most likely prediction of f over a Gaussian distribution:

$$g(x) = \arg\max_{c \in \mathcal{Y}} P(f(x+\varepsilon) = c), \tag{1}$$

where $\varepsilon \sim \mathcal{N}\left(0, \sigma^2 I\right)$ is an isotropic Gaussian noise. The certified robustness of the smoothed classifier g is measured by the certified radius, which is the maximum distance that the input can be perturbed without changing $g(x+\delta) = c_A$. The certified radius is calculated as follows:

$$\|\delta\|_2 < \frac{\sigma}{2}\left(\Phi^{-1}\left(\underline{p_A}\right) - \Phi^{-1}\left(\overline{p_B}\right)\right), \tag{2}$$

where the most probable class c_A is returned with probability p_A and the "runner-up" class is returned with probability p_B. $\underline{p_A}$ and $\overline{p_B}$ are the lower bound and upper bound of p_A and p_B, respectively. Φ^{-1} is the inverse of the standard Gaussian cumulative distribution function.

2.2 CO Problem Setting and Robustness Evaluation Metric

Problem setting. We now consider single-objective CO problems that seek an optimal solution $\boldsymbol{x}$ under a set of constraints $\{h_i(\boldsymbol{x}, Q)\}_{i=1}^{I}$:

$$\min_{\boldsymbol{x}} c(\boldsymbol{x}, Q) \text{ s.t. } h_i(\boldsymbol{x}, Q) \leq 0, i = 1, \ldots, I, \tag{3}$$

where Q denotes the problem instance, $\boldsymbol{x}$ is the discrete decision variable (i.e. solution), and $c(\boldsymbol{x}, Q)$ is the objective function. Let $\boldsymbol{x}^*$ denote an optimal solution. Given a solver f_θ, we denote its returned feasible solution by $\hat{\boldsymbol{x}} = f_\theta(Q)$. Since CO problems are often NP-hard and difficult to solve exactly, $\hat{\boldsymbol{x}}$ need not be optimal. We refer to $c(\boldsymbol{x}^*, Q)$ as the optimal cost and $c(\hat{\boldsymbol{x}}, Q)$ as the actual cost.

Robustness evaluation metric. Direct comparison between solutions is generally unsuitable because CO solutions are discrete and represented differently across tasks. We therefore use the actual cost $c\left(f_\theta(Q), Q\right)$ as the robustness evaluation metric, which provides a solver-agnostic measure of solving performance that does not rely on the structure or representation of the solution space. Accordingly, we study solver robustness through the stability of actual cost under perturbations to the input instance. This choice is also consistent with the study ROCO [25], which uses actual cost for empirical robustness evaluation under perturbations.

2.3 Perturbation Models for CO Instances

We consider perturbations to explicitly represented constraint information in a CO instance and distinguish two perturbation models according to the form of the perturbed information. A binary perturbation adds or removes a discrete constraint relation, thereby changing whether the corresponding relation is present in the instance. A tightness perturbation modifies a real-valued constraint parameter whose value controls how restrictive the corresponding feasible region is. These two perturbation models characterize the input variations considered throughout our robustness analysis.

We use four representative CO problems as running examples: directed acyclic graph (DAG) scheduling, the minimum vertex cover (MVC) problem, the asymmetric traveling salesman problem (ATSP), and the capacitated vehicle routing problem (CVRP). DAG scheduling and MVC naturally admit binary perturbations because the relevant structural information is encoded by the presence or absence

of job-dependency relations and graph edges, respectively. In contrast, ATSP and CVRP involve real-valued instance quantities that can be varied continuously. Hence, tightness perturbations are instantiated through inter-city distances and customer demands, respectively. Formal definitions of the four problems are provided in Appendix B.

3 METHODOLOGY

Our methodology addresses CO solver robustness from two complementary perspectives: certification and enhancement. Sec. 3.1 develops a solver-agnostic certification framework that extends randomized-smoothing-based robustness guarantees to CO solvers under bounded constraint perturbations. Sec. 3.2 further leverages the local-instance perspective behind smoothing to improve solving performance at inference time without modifying or retraining the solver.

3.1 CERTIFIED ROBUSTNESS OF CO SOLVERS

In this subsection, we present our theoretical contributions for certified robustness of CO solvers. Specifically, we begin by formalizing the instance notation used throughout our certification framework. Consider a reference CO instance Q. Let $\boldsymbol{H}$ denote the matrix-valued component that encodes the constraint-related information subject to perturbation, and let S collect all remaining instance information that is kept fixed during perturbation. We denote the instance by $Q_{\boldsymbol{H}} := (\boldsymbol{H}, S)$.

To accommodate the real-valued actual cost used for robustness evaluation in CO, and inspired by [44], we characterize its distribution through the cumulative distribution function (CDF). Specifically, for an instance $Q_{\boldsymbol{H}}$, we define the CDF of its actual cost $c(f_\theta(Q_{\boldsymbol{H}}), Q_{\boldsymbol{H}})$ as

$$F_{\boldsymbol{H}}(r) := \Pr[c(f_\theta(Q_{\boldsymbol{H}}), Q_{\boldsymbol{H}}) \leqslant r], \text{ where } r \in \mathbb{R}. \tag{4}$$

Based on this formulation, we define certified robustness of a CO solver in terms of the allowable variation between the actual-cost CDFs when the perturbation to the instance is confined within a prescribed range. Without loss of generality, we measure the allowable variation for CDF by the popular 1-Wasserstein (W_1) distance:

$$W_1(F_1, F_2) = \int_0^1 \left|F_1^{-1}(u) - F_2^{-1}(u)\right| du, \tag{5}$$

where F_1 and F_2 denote two such CDFs. To derive certifiable bounds on such distributional variation under the two perturbation models, we next employ two different smoothing scheme to construct smoothed CDFs tailored to binary and tightness perturbations.

For robustness certification against binary perturbation, we encode the constraint relations whose presence or absence can be modified by the perturbation using a binary constraint matrix $\boldsymbol{H}_b$. Each entry of $\boldsymbol{H}_b$ indicates whether the corresponding constraint is present: 1 represents its presence and 0 its absence. For example, in DAG scheduling, binary perturbations act on job dependencies by adding or deleting dependency relations. We therefore encode these relations in $\boldsymbol{H}_b$, where 1 represents an existing job dependency and 0 indicates no dependency between the corresponding jobs. In this setting, $\boldsymbol{H} = \boldsymbol{H}_b$, and the reference instance is denoted by $Q_{\boldsymbol{H}_b}$.

To formalize the perturbation, we represent a binary perturbation by $\boldsymbol{\delta}_b \in \{0,1\}^n$, where $(\boldsymbol{\delta}_b)_i = 1$ indicates that the i-th binary entry is inverted and $(\boldsymbol{\delta}_b)_i = 0$ leaves it unchanged. For notational convenience, we identify the n binary entries of $\boldsymbol{H}_b$ with their vectorized form in $\{0,1\}^n$ when applying such perturbations. The perturbed binary encoding is therefore $\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b$, where $\oplus$ denotes the XOR operation.

We then construct a binomial smoothing scheme by sampling a binary random vector $\boldsymbol{z} \sim \{0,1\}^n$, whose entries are independently distributed as

$$\mathbb{P}[\boldsymbol{z}_i = 0] = 1 - \beta,\ \mathbb{P}[\boldsymbol{z}_i = 1] = \beta, \quad i = 1, \cdots, n \tag{6}$$

Applying $\boldsymbol{z}$ to $\boldsymbol{H}_b$ through the element-wise XOR operation, the i-th entry of $\boldsymbol{H}_b$ is preserved with probability $1 - \beta$ and inverted with probability β. This yields the binary encoding $\boldsymbol{H}_b \oplus \boldsymbol{z}$ and the corresponding instance $Q_{\boldsymbol{H}_b \oplus \boldsymbol{z}}$, for which Eq. 4 gives the actual-cost CDF as $F_{\boldsymbol{H}_b \oplus \boldsymbol{z}}(r) = \Pr[c(f_\theta(Q_{\boldsymbol{H}_b \oplus \boldsymbol{z}}), Q_{\boldsymbol{H}_b \oplus \boldsymbol{z}}) \leqslant r]$. We then define the smoothed CDF as

$$G_{\boldsymbol{H}_b}(r) = \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \{0,1\}^n}[F_{\boldsymbol{H}_b \oplus \boldsymbol{z}}(r)] = \sum_{\boldsymbol{z} \sim \{0,1\}^n} F_{\boldsymbol{H}_b \oplus \boldsymbol{z}}(r) p(\boldsymbol{z}), \tag{7}$$

where $p(\boldsymbol{z}) = \beta^{||\boldsymbol{z}||_0}(1-\beta)^{n-||\boldsymbol{z}||_0}$ is the probability mass function of $\boldsymbol{z}$ on the discrete space $\{0,1\}^n$. We provide a guarantee for the smoothed CDF.

Theorem 3.1 (**Robustness Guarantee against Binary Perturbation**). *Given an input instance $Q_{\boldsymbol{H}_b}$ with a binary constraint matrix $\boldsymbol{H}_b$, let $G_{\boldsymbol{H}_b}(r)$ denote its smoothed CDF under the binomial smoothing above. For a binary perturbation $\boldsymbol{\delta}_b$ applied to $\boldsymbol{H}_b$, let $m = \|\boldsymbol{\delta}_b\|_0$ denote the number of entries of $\boldsymbol{H}_b$ subject to inversion. Let $K \sim \mathrm{Binomial}(m, \beta)$, $L := (\beta/(1-\beta))^{m-2K}$, and let $q_L(t) := \inf\{x : \Pr(L \leqslant x) \geqslant t\}$, $0 < t < 1$, denote the generalized quantile function of L. For $v \in [0,1]$, define*

$$\underline{T}_{m,\beta}(v) := \int_0^v q_L(t)\,dt, \qquad \overline{T}_{m,\beta}(v) := \int_{1-v}^1 q_L(t)\,dt, \tag{8}$$

and

$$B_{m,\beta}(v) := \max\left\{v - \underline{T}_{m,\beta}(v), \overline{T}_{m,\beta}(v) - v\right\}. \tag{9}$$

Then,

$$W_1\left(G_{\boldsymbol{H}_b}, G_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b}\right) \leqslant \int_{-\infty}^{\infty} B_{m,\beta}\left(G_{\boldsymbol{H}_b}(r)\right)\,dr. \tag{10}$$

Consequently, the robustness guarantee is finite whenever $\mathbb{E}|Y_{\boldsymbol{H}_b}| < \infty$, where $Y_{\boldsymbol{H}_b}$ is a real-valued random variable with CDF $G_{\boldsymbol{H}_b}$.

Similarly, for robustness certification against tightness perturbation, we encode the constraint-related numerical quantities whose values can be modified by the perturbation using a real-valued constraint matrix $\boldsymbol{H}_t$. Each entry of $\boldsymbol{H}_t$ specifies the value of one such perturbable quantity. For example, in CVRP, the customer demands are subject to tightness perturbations and are encoded by $\boldsymbol{H}_t$, where each entry specifies the demand of one customer. In this setting, $\boldsymbol{H} = \boldsymbol{H}_t$, and the reference instance is denoted by $Q_{\boldsymbol{H}_t}$.

To formalize the perturbation, we represent a tightness perturbation by $\boldsymbol{\delta}_t \in \mathbb{R}^n$, where each entry specifies the additive change applied to the corresponding entry of $\boldsymbol{H}_t$. For notational convenience, we identify the n real-valued entries of $\boldsymbol{H}_t$ with their vectorized form in $\mathbb{R}^n$ when applying such perturbations. The perturbed real-valued encoding is therefore $\boldsymbol{H}_t + \boldsymbol{\delta}_t$.

We then apply Gaussian smooth by sampling $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2 I_n)$. Adding $\boldsymbol{z}$ to $\boldsymbol{H}_t$ yields the real-valued encoding $\boldsymbol{H}_t + \boldsymbol{z}$ and the corresponding instance $Q_{\boldsymbol{H}_t+\boldsymbol{z}}$, for which Eq. 4 gives the actual-cost CDF: $F_{\boldsymbol{H}_t+\boldsymbol{z}}(r) = \Pr[c(f_\theta(Q_{\boldsymbol{H}_t+\boldsymbol{z}}), Q_{\boldsymbol{H}_t+\boldsymbol{z}}) \leqslant r]$. The smoothed CDF is hence defined:

$$G_{\boldsymbol{H}_t}(r) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{N}(0,\sigma^2 I_n)}\left[F_{\boldsymbol{H}_t+\boldsymbol{z}}(r)\right] = \int F_{\boldsymbol{H}_t+\boldsymbol{z}}(r) p_\sigma(\boldsymbol{z}) d\boldsymbol{z}, \tag{11}$$

where $p_\sigma(\boldsymbol{z}) = (2\pi\sigma^2)^{-n/2}\exp(-||\boldsymbol{z}||_2^2/2\sigma^2)$ is the probability density function of $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2 I_n)$. This construction allows us to establish a corresponding theoretical robustness guarantee against tightness perturbations.

Theorem 3.2 (**Robustness Guarantee against Tightness Perturbation**). *Given an input instance $Q_{\boldsymbol{H}_t}$ with real-valued constraint matrix $\boldsymbol{H}_t$, let $G_{\boldsymbol{H}_t}(r)$ denote its smoothed CDF under Gaussian smoothing. For perturbations satisfying $||\boldsymbol{\delta}_t||_2 \leqslant \eta$, we have*

$$\sup_{||\boldsymbol{\delta}_t||_2 \leqslant \eta} W_1(G_{\boldsymbol{H}_t}, G_{\boldsymbol{H}_t+\boldsymbol{\delta}_t}) \leqslant \int_{-\infty}^{\infty} B_{\eta/\sigma}(G_{\boldsymbol{H}_t}(r))dr, \tag{12}$$

where, for $a \geqslant 0$ and $u \in (0,1)$,

$$B_a(u) := \max\{u - \Phi(\Phi^{-1}(u) - a), \Phi(\Phi^{-1}(u) + a) - u\}, \tag{13}$$

with $B_a(0) = B_a(1) = 0$ defined by continuous extension, and Φ denotes the CDF of the standard normal distribution. The robustness guarantee is finite provided that $\mathbb{E}|Y_{\boldsymbol{H}_t}|^p < \infty$ for some $p > 1$, where $Y_{\boldsymbol{H}_t}$ is a real-valued random variable with CDF $G_{\boldsymbol{H}_t}$.

Complete proofs of the above two theorems, together with the boundedness analysis of the corresponding robustness guarantees, are provided in Appendix C.

Since the certification bounds in Theorems 3.1 and 3.2 cannot be evaluated exactly in practice, we estimate them using finite smoothing and integration samples. We further employ finite-sample statistical bounds to quantify the sampling uncertainty and construct confidence intervals for the estimated bounds. The detailed estimation procedure and confidence interval construction are provided in Appendix D.

3.2 Robustness Enhancement Method

Beyond robustness certification, we further propose a training-free robustness enhancement method to improve the solving performance. The vulnerability of a CO solver manifests as substantial degradation in solving performance under limited perturbations to problem instances. Such degradation is often associated with particular configurations of $\boldsymbol{H}$, such as specific edge patterns in graph problems or particular values of numerical instance quantities. We refer to instances that expose such solver-specific vulnerability as hard instances. To mitigate this vulnerability, inspired by the smoothing idea of exploring neighboring instances, we avoid solving these hard instances directly and instead search for nearby instances from which better solutions to the hard instance can be obtained.

Consider a reference $Q_{\boldsymbol{H}}$ defined in Sec. 3.1, and denote $\mathcal{X}(Q_{\boldsymbol{H}})$ its feasible solution set. We search around $Q_{\boldsymbol{H}}$ for instances $Q_{\boldsymbol{H}'}$ satisfying two requirements: **1)** The solution obtained for $Q_{\boldsymbol{H}'}$ remain feasible for $Q_{\boldsymbol{H}}$. To facilitate the search, instead of verifying this condition only for $f_\theta(Q_{\boldsymbol{H}'})$, we impose the stronger sufficient condition $\mathcal{X}(Q_{\boldsymbol{H}'}) \subseteq \mathcal{X}(Q_{\boldsymbol{H}})$. **2)** When this solution is applied back to the reference instance, its cost should not exceed that obtained by directly solving $Q_{\boldsymbol{H}}$ i.e. $c(f_\theta(Q_{\boldsymbol{H}'}), Q_{\boldsymbol{H}}) \leqslant c(f_\theta(Q_{\boldsymbol{H}}), Q_{\boldsymbol{H}})$. Hence, we define the set of "easy instances" around $Q_{\boldsymbol{H}}$:

$$D(Q_{\boldsymbol{H}}) := \{Q_{\boldsymbol{H}'} : \mathcal{X}(Q_{\boldsymbol{H}'}) \subseteq \mathcal{X}(Q_{\boldsymbol{H}}), c(f_\theta(Q_{\boldsymbol{H}'}), Q_{\boldsymbol{H}}) \leqslant c(f_\theta(Q_{\boldsymbol{H}}), Q_{\boldsymbol{H}})\}. \tag{14}$$

Given this set, we select the instance whose solution achieves the minimum cost on $Q_{\boldsymbol{H}}$:

$$Q_{\boldsymbol{H}^*} = \underset{Q_{\boldsymbol{H}'} \in D(Q_{\boldsymbol{H}})}{\arg\min}\ c(f_\theta(Q_{\boldsymbol{H}'}), Q_{\boldsymbol{H}}), \quad \hat{\boldsymbol{x}}_e = f_\theta(Q_{\boldsymbol{H}^*}). \tag{15}$$

The resulting $\hat{\boldsymbol{x}}_e$ is used as the enhanced solution for $Q_{\boldsymbol{H}}$. In practice, rather than enumerating the entire set, we search for such easy instances efficiently by sampling. Following the same binary and tightness decomposition used for perturbation modeling in Sec. 2.3, we construct candidate instances by modifying $\boldsymbol{H}$ along two directions: adding constraints or increasing the corresponding tightness-related quantities, as these directions do not enlarge the feasible space.

Enhancement via adding constraints. For an instance $Q_{\boldsymbol{H}_b}$ with binary constraint matrix $\boldsymbol{H}_b$, we generate a candidate $Q_{\boldsymbol{H}_b'}$ by changing a limited number of zero entries in $\boldsymbol{H}_b$ to one. Formally,

$$(\boldsymbol{H}_b')_i \geqslant (\boldsymbol{H}_b)_i, \quad \|\boldsymbol{H}_b' - \boldsymbol{H}_b\|_0 \leqslant m_e, \tag{16}$$

where the inequality is element-wise and m_e denotes the maximum number of modified entries. Since these modifications only add constraints, hence $\mathcal{X}(Q_{\boldsymbol{H}_b'}) \subseteq \mathcal{X}(Q_{\boldsymbol{H}_b})$. We then solve each candidate $Q_{\boldsymbol{H}_b'}$ and evaluate its solution on $Q_{\boldsymbol{H}_b}$. Candidates satisfying $c(f_\theta(Q_{\boldsymbol{H}_b'}), Q_{\boldsymbol{H}_b}) \leqslant c(f_\theta(Q_{\boldsymbol{H}_b}), Q_{\boldsymbol{H}_b})$ are retained as easy instances, i.e., $Q_{\boldsymbol{H}_b'} \in D(Q_{\boldsymbol{H}_b})$.

Enhancement via increasing tightness quantities. For an instance $Q_{\boldsymbol{H}_t}$ with real-valued constraint $\boldsymbol{H}_t$, we generate candidates by adding non-negative modifications to the corresponding entries:

$$\boldsymbol{H}_t' = \boldsymbol{H}_t + \boldsymbol{\delta}_t^e, \quad \boldsymbol{\delta}_t^e \in \mathbb{R}^n_{\geqslant 0}, \tag{17}$$

where $\boldsymbol{\delta}_t^e$ denotes the modification. Depending on the task, we consider positive modifications that do not enlarge the feasible space, i.e., $\mathcal{X}(Q_{\boldsymbol{H}_t'}) \subseteq \mathcal{X}(Q_{\boldsymbol{H}_t})$. We again solve each candidate $Q_{\boldsymbol{H}_t'}$ and retain those satisfying $c(f_\theta(Q_{\boldsymbol{H}_t'}), Q_{\boldsymbol{H}_t}) \leqslant c(f_\theta(Q_{\boldsymbol{H}_t}), Q_{\boldsymbol{H}_t})$, which belong to $D(Q_{\boldsymbol{H}_t})$. The sampling and selection procedure is summarized in Alg. 2 in Appendix E.

4 Experiment

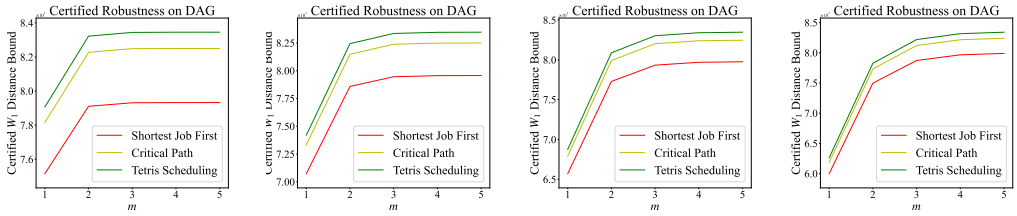
All experiments are conducted on a CPU (Intel CPU @ 3.60GHz) and a GPU (GTX 2080 Ti GPU).

4.1 Evaluation of Robustness Certification

We assess certified robustness on the four representative CO tasks introduced in Sec. 2.3, whose formal problem definitions and optimization objectives are provided in Appendix B, with the certification bounds computed using the finite-sample estimation procedure described in Appendix D.

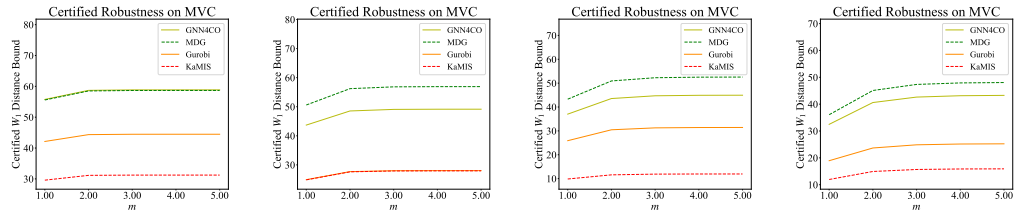
4.1.1 Robustness Certification via Binomial Smoothing

We evaluate certified robustness via binomial smoothing on two representative CO tasks: DAG scheduling and the MVC problem. In DAG scheduling, each node has two properties: execution time and resource requirement, and edges represent job dependencies. We construct a binary matrix $\boldsymbol{H}_b$, where an entry of 1 indicates a dependency between the corresponding pair of jobs, and 0 otherwise.



(a) $\beta = 0.05$ (b) $\beta = 0.10$ (c) $\beta = 0.15$ (d) $\beta = 0.20$

Figure 2: Robustness guarantee over perturbation bound m with fixed β, on TPC-H-50.



(a) $\beta = 0.05$ (b) $\beta = 0.10$ (c) $\beta = 0.15$ (d) $\beta = 0.20$

Figure 3: Robustness guarantee over perturbation bound m with fixed β, on MVC (SMALL).

In MVC, $\boldsymbol{H}_b$ corresponds to the adjacency matrix of the graph, where 1 indicates the presence of an edge between two vertices and 0 otherwise. For both tasks, we evaluate certified robustness by computing the upper bound on $W_1(G_{\boldsymbol{H}_b}, G_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b})$ given in Eq. 10 under binary perturbations $\boldsymbol{\delta}_b$ applied to $\boldsymbol{H}_b$.

Task I: Directed Acyclic Graph Scheduling. We apply three solvers to the DAG scheduling task: the Shortest Job First (SJF) algorithm, the Critical Path (CP) algorithm, and the Tetris scheduling algorithm [13]. The SJF greedily selects the jobs with the lowest completion time. The CP identifies the jobs essential to meeting the deadline and executes them sequentially. The Tetris scheduling algorithm represents jobs as two-dimensional blocks in a Tetris game, where each block corresponds to a job and its completion time and resource requirement. We construct DAG instances based on the TPC-H benchmark [3], a decision support benchmark consisting of business-oriented query workloads. We generate three datasets of varying sizes, denoted as TPC-H-50, TPC-H-100, and TPC-H-150, corresponding to instances with different numbers of tasks.

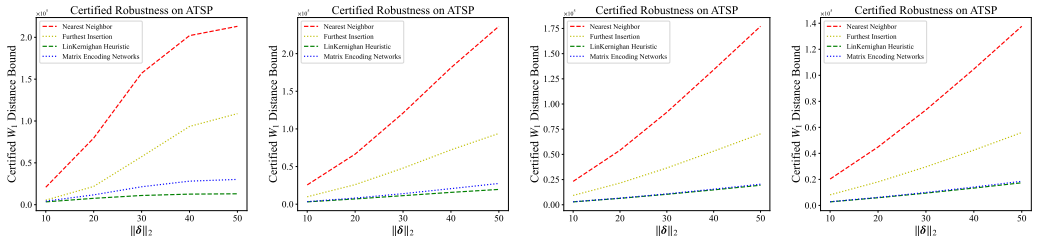
Task II: Minimum Vertex Cover. We apply four solvers: the Maximum Degree Greedy (MDG) algorithm, Karlsruhe Maximum Independent Set (KaMIS) solver [11], Gurobi Optimizer (v13.0.1) [14], and a GNN-based solver (GNN4CO) [26]. The MDG algorithm greedily selects the vertex with the highest degree and removes its incident edges. KaMIS solves the maximum independent set problem, whose complement corresponds to a minimum vertex cover. The Gurobi Optimizer solves MVC by formulating it as an integer linear programming (ILP) problem. GNN4CO employs GNNs to predict vertex selection, following the implementation provided in ML4CO-Bench-101 [26]. We use instances derived from the DIMACS benchmark [2] for evaluation, considering two scales based on the number of vertices: SMALL (200–300) and LARGE (800–1200).

Setup. Following Alg. 1, we use $k = 1000$ smoothing samples and $n = 1000$ integration samples, and vary the binary perturbation bound $m \in \{1, 2, 3, 4, 5\}$ and the smoothing parameter $\beta \in \{0.05, 0.1, 0.15, 0.20, 0.25\}$. Other parameters remain consistent with [25].

Results. Fig. 2 and Fig. 3 show the certified W_1 upper bounds as functions of the binary perturbation bound m for DAG scheduling and MVC, respectively. In both tasks, the certified bound increases as m increases, while the rate of increase gradually diminishes. This indicates that enlarging the admissible binary perturbation range leads to a looser robustness certificate, whereas the marginal effect of further increasing the perturbation bound becomes progressively smaller.

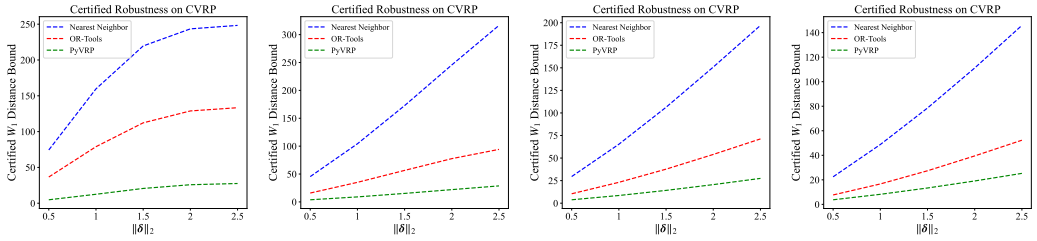
4.1.2 Robustness Certification via Gaussian Smoothing

We evaluate certified robustness via Gaussian smoothing on two tasks: asymmetric TSP and CVRP. In ATSP, a fully connected asymmetric distance matrix $\boldsymbol{H}_t$ stores the distances. In CVRP, $\boldsymbol{H}_t$ records the demand of each customer. For both tasks, we evaluate certified robustness by computing the upper bound on $\sup_{\|\boldsymbol{\delta}_t\|_2 \leq \eta} W_1(G_{\boldsymbol{H}_t}, G_{\boldsymbol{H}_t + \boldsymbol{\delta}_t})$ given in Eq. 12.



(a) $\sigma = 10$ (b) $\sigma = 20$ (c) $\sigma = 30$ (d) $\sigma = 40$

Figure 4: Robustness guarantee over perturbation bound $||\boldsymbol{\delta}_t||_2$ with fixed σ, on ATSP (20 cities).



(a) $\sigma = 0.5$ (b) $\sigma = 1.0$ (c) $\sigma = 1.5$ (d) $\sigma = 2.0$

Figure 5: Robustness guarantee over perturbation bound $||\boldsymbol{\delta}_t||_2$ with fixed σ, on CVRP (55 customers).

Task I: Asymmetric Traveling Salesman. We apply four solvers to the asymmetric TSP task: Nearest Neighbor (NN), Furthest Insertion (FI), LinKernighan Heuristic (LKH3) [16], and Matrix Encoding Networks (MatNet) [21]. The NN solver greedily chooses the closest city to add to the tour. The FI solver identifies the city with the greatest distance from the existing cities on the tour and inserts it. LKH3 is a SOTA conventional solver for asymmetric TSP. The MatNet solver is a learning-based solver. We employ the dataset from [21], including the well studied asymmetric TSP instances. Using this primary dataset, we generate three subsets with varying city counts: 20, 50, 100.

Task II: Capacitated Vehicle Routing. We apply three solvers to the CVRP task: Nearest Neighbor (NN), the Google OR-Tools routing solver (version 9.15) [29], and PyVRP [40]. The NN solver greedily extends a route by selecting the nearest unvisited customer. The OR-Tools solver combines local search with constraint programming techniques. The PyVRP solver is based on a hybrid genetic search framework for vehicle routing problems. We evaluate on instances from CVRPLIB [1], a standard benchmark repository for vehicle routing problems. We consider instances of varying sizes, grouped by the number of customers, including 50, 100, 200, and 500 nodes.

Setup. Following Alg. 1, we use $k = 1000$ smoothing samples and $n = 1000$ integration samples. Considering the different numerical scales, we set task-specific ranges for the perturbation magnitude. For ATSP, we vary the maximum ℓ_2 perturbation norm as $\|\boldsymbol{\delta}_t\|_2 \in \{10, 20, 30, 40, 50\}$ and the perturbation parameter as $\sigma \in \{10, 20, 30, 40, 50\}$. For CVRP, we adjust the ranges to $\|\boldsymbol{\delta}_t\|_2 \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$ and $\sigma \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$. Other parameter settings remain consistent with [25].

Results. Fig. 4 and Fig. 5 show the certified W_1 upper bounds as functions of the tightness perturbation bound $\|\boldsymbol{\delta}_t\|_2$ under different fixed values of σ for ATSP and CVRP, respectively. Across the evaluated smoothing parameters, the certified bound increases as $\|\boldsymbol{\delta}_t\|_2$ increases. This trend is consistent with Eq. 12, where enlarging the admissible perturbation radius increases η/σ and consequently enlarges $B_{\eta/\sigma}(G_{\boldsymbol{H}_t}(r))$ for fixed σ. The growth is generally nonlinear, reflecting the nonlinear dependence of the certification bound on η/σ. The magnitude of the bound also varies across solvers and smoothing parameters, since the smoothed CDF $G_{\boldsymbol{H}_t}$ itself depends on both the solver and σ.

4.2 Evaluation of Robustness Enhancement

We evaluate the enhancement method in terms of empirical robustness, while its effect on certified robustness is further examined in Appendix F.2. Following ROCO [25], we consider four perturbation strategies for generating hard instances: **Random Search (ROCO-RA)**, **Optimum-Guided Search (ROCO-OG)**, **Simulated Annealing (ROCO-SA)**, and **Reinforcement Learning Attacker (ROCO-RL)**. ROCO-RA randomly changes a constraint in each iteration and performs a large number of

Table 1: Evaluation for enhancement method on DAG scheduling (TPC-H-50, $m_e = 1$). **Refer to Sec. 4.2 for the meaning of the abbreviations in the table, so for other tables.**

Solver	CLEAN ($\times 10^5$)	CLEAN(E) ($\times 10^5$)	Method	Attack Method($\times 10^5$) ROCO-RA	ROCO-OG	ROCO-SA	ROCO-RL
SJF	1.0461	1.0366	ATTACK	1.0559	1.0609	1.0610	1.0604
			ATTACK(E)	1.0366	1.0472	1.0465	1.0450
CP	0.8695	0.8592	ATTACK	0.9288	0.9481	0.9496	0.9489
			ATTACK(E)	0.8691	0.8748	0.8734	0.8729
Tetris	0.8227	0.8177	ATTACK	0.8972	0.9272	0.9290	0.9391
			ATTACK(E)	0.8463	0.8751	0.8698	0.8728

Table 2: Evaluation for enhancement method on MVC (SMALL dataset, $m_e = 10$).

Solver	CLEAN	CLEAN(E)	Method	Attack Method ROCO-RA	ROCO-OG	ROCO-SA	ROCO-RL
MDG	265	263	ATTACK	267	267	266	272
			ATTACK(E)	263	263	262	265
KAMIS	261	261	ATTACK	261	261	261	262
			ATTACK(E)	261	261	261	262
GUROBI	261	261	ATTACK	263	263	262	264
			ATTACK(E)	261	261	260	261
GNN4CO	281	279	ATTACK	285	282	286	287
			ATTACK(E)	281	277	280	280

Table 3: Evaluation for enhancement method on Asymmetric TSP (20 cities, $\sigma = 1.0$).

Solver	CLEAN ($\times 10^6$)	CLEAN(E) ($\times 10^6$)	Method	Attack Method($\times 10^6$) ROCO-RA	ROCO-OG	ROCO-SA	ROCO-RL
NN	1.9354	1.9011	ATTACK	2.0873	1.9484	2.1265	2.1655
			ATTACK(E)	1.9303	1.9416	1.9800	2.1314
FI	1.5097	1.4939	ATTACK	1.5412	1.5373	1.5631	1.5272
			ATTACK(E)	1.4997	1.5183	1.4993	1.4992
LKH3	1.4605	1.4605	ATTACK	1.4595	1.4595	1.4611	1.4605
			ATTACK(E)	1.4595	1.4595	1.4547	1.4597
MatNet	1.4611	1.4611	ATTACK	1.4631	1.4663	1.4682	1.4691
			ATTACK(E)	1.4595	1.4592	1.4547	1.4602

Table 4: Evaluation for enhancement method on CVRP (55 customers, $\sigma = 0.5$).

Solver	CLEAN	CLEAN(E)	Method	Attack Method ROCO-RA	ROCO-OG	ROCO-SA	ROCO-RL
NN	1500	1328	ATTACK	1522	1555	1602	1555
			ATTACK(E)	1324	1335	1436	1324
OR-TOOLS	1132	1086	ATTACK	1155	1153	1191	1160
			ATTACK(E)	1081	1105	1071	1081
PYVRP	1073	1073	ATTACK	1082	1076	1075	1080
			ATTACK(E)	1082	1075	1070	1080

trials. ROCO-OG exploits problem structure to guide the perturbation search. ROCO-SA adjusts the search process according to a simulated annealing strategy. ROCO-RL employs reinforcement learning to search for perturbations while treating the solver as a black box.

We employ the same tasks, solvers, and datasets as in Sec. 4.1, with the perturbation-strategy configurations following [25]. For each solver, we report four types of actual cost: **1) CLEAN:** the cost obtained by directly solving the original instance, **2) CLEAN(E):** the cost obtained on the original instance after applying the enhancement method, **3) ATTACK:** the cost obtained by directly solving the hard instance generated from the original instance using a perturbation strategy, and **4) ATTACK(E):** the cost obtained on the same hard instance after applying the enhancement method. For the enhancement procedure in Alg. 2, we set $n_e = 10$ and $m_e = 1$ for DAG scheduling, $n_e = 100$ and $m_e = 10$ for MVC, $n_e = 100$ and $\sigma = 1.0$ for ATSP, and $n_e = 50$ and $\sigma = 0.5$ for CVRP. Across the 14 evaluated solvers and four perturbation strategies, this results in 56 attacked instances for evaluating ATTACK and ATTACK(E).

Tables 1, 2, 3, and 4 report the corresponding costs for DAG scheduling, MVC, ATSP, and CVRP, respectively. The enhancement method reduces the cost on 50 of the 56 attacked instances, demonstrating its effectiveness in mitigating the performance degradation induced by different perturbation strategies. The method also reduces the cost on clean instances, indicating that useful easy instances may exist in the neighborhood of both original and hard instances. The improvement is generally more pronounced on hard instances, where the solver has already suffered performance degradation.

5 Conclusion and Outlook

We have developed a solver-agnostic framework for certifying the robustness of CO solvers under bounded perturbations and introduced a training-free robustness enhancement method that improves solving performance by exploiting nearby easy instances. Experiments across four CO problems and 14 solvers demonstrate the applicability of the certification framework and the effectiveness of the enhancement method. Our current formulation focuses on single-objective CO problems with explicitly represented constraint information, where perturbations can be modeled through binary or real-valued constraint matrix. Extending the framework to broader CO formulations and other forms of instance perturbation remains a direction for future work.

REFERENCES

[1] Cvrplib: A library of capacitated vehicle routing problem instances. http://vrp.atd-lab.inf.puc-rio.br/index.php/en/. Accessed: 2026.

[2] Dimacs challenge benchmarks. http://dimacs.rutgers.edu/.

[3] TPC-H benchmark. https://www.tpc.org/tpch/.

[4] David Adjiashvili. *Structural robustness in combinatorial optimization*. Doctoral dissertation, ETH Zurich, 2012.

[5] Joaquin Bautista and Jordi Pereira. A dynamic programming based heuristic for the assembly line balancing problem. *European Journal of Operational Research*, 194(3):787–794, 2009.

[6] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. 290(2):405–421, 2021.

[7] Dimitris Bertsimas, Dick Den Hertog, and Jean Pauphilet. Probabilistic guarantees in robust optimization. *SIAM Journal on Optimization*, 31(4):2893–2920, 2021. doi: 10.1137/21M1390967.

[8] Christoph Buchheim and Jannis Kurtz. Robust combinatorial optimization under convex and discrete cost uncertainty. *EURO Journal on Computational Optimization*, 6(3):211–238, 2018.

[9] Ping-yeh Chiang, Michael Curry, Ahmed Abdelkader, Aounon Kumar, John Dickerson, and Tom Goldstein. Detection as regression: Certified object detection with median smoothing. *Advances in Neural Information Processing Systems*, 33:1275–1286, 2020.

[10] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

[11] Jakob Dahlum, Sebastian Lamm, Peter Sanders, Christian Schulz, Darren Strash, and Renato F. Werneck. Accelerating local search for the maximum independent set problem. In *15th International Symposium on Experimental Algorithms SEA*, volume 9685 of *Lecture Notes in Computer Science*, pp. 118–133. Springer, 2016.

[12] Simon Geisler, Johanna Sommer, Jan Schuchardt, Aleksandar Bojchevski, and Stephan Günnemann. Generalization of neural combinatorial solvers through the lens of adversarial robustness. *arXiv preprint arXiv:2110.10942*, 2021.

[13] Robert Grandl, Ganesh Ananthanarayanan, Srikanth Kandula, Sriram Rao, and Aditya Akella. Multi-resource packing for cluster schedulers. *ACM SIGCOMM Computer Communication Review*, 44(4):455–466, 2014.

[14] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2026. URL https://www.gurobi.com.

[15] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. *Advances in neural information processing systems*, 30, 2017.

[16] Keld Helsgaun. An extension of the lin-kernighan-helsgaun tsp solver for constrained traveling salesman and vehicle routing problems. *Roskilde: Roskilde University*, 12:966–980, 2017.

[17] Jinyuan Jia, Binghui Wang, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of community detection against adversarial structural perturbation via randomized smoothing. In *Proceedings of The Web Conference 2020*, pp. 2718–2724, 2020.

[18] Richard M Karp and Michael Sipser. Maximum matching in sparse random graphs. In *Proceedings of the 22nd Annual Symposium on Foundations of Computer Science*, pp. 364–375, 1981.

[19] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems*, 30, 2017.

[20] Aounon Kumar and Tom Goldstein. Center smoothing: Certified robustness for networks with structured outputs. *Advances in Neural Information Processing Systems*, 34:5560–5575, 2021.

[21] Yeong-Dae Kwon, Jinho Choo, Iljoo Yoon, Minah Park, Duwon Park, and Youngjune Gwon. Matrix encoding networks for neural combinatorial optimization. *Advances in Neural Information Processing Systems*, 34:5138–5149, 2021.

[22] 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. IEEE, 2019.

[23] Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

[24] Alexander J Levine and Soheil Feizi. Improved, deterministic smoothing for l_1 certified robustness. In *International Conference on Machine Learning*, pp. 6254–6264. PMLR, 2021.

[25] Han Lu, Zenan Li, Runzhong Wang, Qibing Ren, Xijun Li, Mingxuan Yuan, Jia Zeng, Xiaokang Yang, and Junchi Yan. Roco: A general framework for evaluating robustness of combinatorial optimization solvers on graphs. In *International Conference on Learning Representations*, 2023.

[26] Jiale Ma, Wenzheng Pan, Yang Li, and Junchi Yan. Ml4co-bench-101: Benchmark machine learning for classic combinatorial problems on graphs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[27] Heinz Mühlenbein, Martina Gorges-Schleuter, and Ottmar Krämer. Evolution algorithms in combinatorial optimization. *Parallel computing*, 7(1):65–85, 1988.

[28] Fernando Peres and Mauro Castelli. Combinatorial optimization problems and metaheuristics: Review, challenges, design, and development. *Applied Sciences*, 11(14):6449, 2021.

[29] Laurent Perron and Vincent Furnon. Or-tools. URL https://developers.google.com/optimization/.

[30] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

[31] Melissa Sánchez, Jorge M Cruz-Duarte, José carlos Ortíz-Bayliss, Hector Ceballos, Hugo Terashima-Marin, and Ivan Amaya. A systematic review of hyper-heuristics on combinatorial optimization problems. *IEEE Access*, 8:128068–128095, 2020.

[32] Nithin Varma and Yuichi Yoshida. Average sensitivity of graph algorithms. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 684–703. SIAM, 2021.

[33] Binghui Wang, Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of graph neural networks against adversarial structural perturbation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1645–1653, 2021.

[34] R. Wang, Y. Zhang, Z. Guo, T. Chen, X. Yang, and J. Yan. insatnet: The positive linear satisfiability neural networks. 2013.

[35] R. Wang, J. Yan, and X. Yang. Learning combinatorial embedding networks for deep graph matching. pp. 3056–3065, 2019.

[36] Runzhong Wang, Zhigang Hua, Gan Liu, Jiayi Zhang, Junchi Yan, Feng Qi, Shuang Yang, Jun Zhou, and Xiaokang Yang. A bi-level framework for learning to solve combinatorial optimization on graphs. In *Advances in Neural Information Processing Systems*, 2021.

[37] Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. *Advances in Neural Information Processing Systems*, 34:29909–29921, 2021.

[38] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4:65–85, 1994.

[39] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pp. 5286–5295. PMLR, 2018.

[40] Niels A. Wouda, Leon Lan, and Wouter Kool. PyVRP: a high-performance VRP solver package. *INFORMS Journal on Computing*, 36(4):943–955, 2024. doi: 10.1287/ijoc.2023.0055.

[41] Min Wu, Matthew Wicker, Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. A game-based approximate verification of deep neural networks with provable guarantees. *Theoretical Computer Science*, 807:298–329, 2020.

[42] Junchi Yan, Shuang Yang, and Edwin Hancock. Learning graph matching and related combinatorial optimization problems. In *IJCAI*, 2020.

[43] Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pp. 10693–10705. PMLR, 2020.

[44] TaeHo Yoon, Youngsuk Park, Ernest K Ryu, and Yuyang Wang. Robust probabilistic time series forecasting. In *International Conference on Artificial Intelligence and Statistics*, pp. 1336–1358. PMLR, 2022.

[45] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in neural information processing systems*, 31, 2018.

A Related Work

We provide an overview of three research areas closely related to this study: combinatorial optimization (CO), the robustness of combinatorial optimization solvers, and certified robustness based on randomized smoothing.

A.1 Combinatorial Optimization

Combinatorial optimization is a rich field with numerous algorithms designed to solve various problems. For example, dynamic programming [5] decomposes problems into simpler subproblems, storing results to avoid redundancy, and is particularly useful for knapsack and shortest path problems. Greedy algorithms [18] make locally optimal choices aiming for a global optimum. These methods collectively enhance the ability to address diverse optimization challenges. However, for many NP-hard CO problems, designing effective heuristics traditionally requires considerable domain expertise and manual effort. This has motivated a shift toward data-driven, learning-based solvers [6]. Recent developments in this direction include reinforcement learning-based methods [19] and one shot networks [35; 34] tailored to specific tasks like graph matching [42]. These approaches aim to automatically discover high-quality solution strategies from data, offering promising gains in both solution quality and computational efficiency.

A.2 Robustness of Combinatorial Optimization Solvers

The robustness of CO solvers has received increasing attention, as limited perturbations to problem instances may lead to substantial degradation in solving performance [32; 12; 25]. [12] study the robustness of learning-based CO solvers under input perturbations, but their approach is tailored to differentiable neural models and therefore has limited applicability to general CO solvers. ROCO [25] further develops a solver-agnostic empirical robustness evaluation framework by testing CO solvers under several designed perturbation strategies and measuring the resulting changes in solving performance. Such empirical evaluations are inherently tied to the specific perturbation strategies considered and therefore cannot guarantee robustness against other admissible perturbations. In contrast, our work studies certified robustness and provides solver-agnostic, perturbation-strategy-independent guarantees on the variation of solving performance for perturbations within a prescribed range.

Moreover, robustness certification [8] differs from robustness optimization in its object of interest. Robustness optimization studies the robustness of the resulting solutions under uncertainty, and typically achieves this by reformulating the optimization problem to explicitly account for uncertainties in problem parameters or structures (e.g., objective coefficients, constraint data, or graph connectivity [4]). While some robustness optimization studies [7] also derive probabilistic guarantees for such robust solutions , these guarantees are formulated with respect to the solution quality under uncertainty. In contrast, our work is concerned with certifying the stability of a given combinatorial optimization solver—possibly heuristic or learning-based—viewed as a black-box mapping from inputs to continuous-valued outputs, without modifying the original problem formulation.

A.3 Certified Robustness based on Randomized Smoothing (RS)

RS was first introduced by [22] and applied to train a certifiably robust classifier on ImageNet, which provides a loose robustness guarantee while being able to certify large datasets and black-box models. [10] demonstrate that using Gaussian smoothing distributions results in tight robustness guarantees. Subsequent studies propose new RS-type techniques, e.g., techniques that use smoothing distributions with different norms [24; 23; 43], and techniques that suit different tasks [9; 17; 20]. In the graph domain, [33] develop a certifiably robust method based on RS to resist structure perturbations for both graph and node level classification. Nevertheless, the applicability of these methods is still limited, as they are unable to handle the inputs and non-categorical outputs in CO. Although previous studies [19; 36] demonstrate that some inputs of CO tasks can be modeled as weighted graphs, existing RS-based certified robustness research for graph tasks [17; 33; 30] focuses on graph classification tasks and neglects tasks with non-categorical outputs as in CO tasks.

B Formal Definitions of the Evaluated CO Problems

For completeness, we provide the formal definitions of the four CO problems used as running examples and evaluation tasks in this work.

B.1 Directed Acyclic Graph (DAG) Scheduling

A DAG scheduling instance is represented by a directed acyclic graph $G = (V, E)$, where each node $i \in V$ denotes a job and each directed edge $(i, j) \in E$ indicates that job j can start only after job i is completed. Each job i has a processing time $p_i > 0$ and a resource requirement $a_i > 0$, while the total available resource capacity is $R > 0$. Let s_i denote the starting time of job i. The objective is to minimize the makespan:

$$\begin{aligned} \min_{\{s_i\}_{i\in V}} \quad & C_{\max} := \max_{i\in V}(s_i + p_i) \\ \text{s.t.} \quad & s_j \geqslant s_i + p_i, && \forall (i,j) \in E, \\ & \sum_{i\in V} a_i \mathbf{1}\{s_i \leqslant t \leqslant s_i + p_i\} \leqslant R, && \forall t \geqslant 0, \\ & s_i \geqslant 0, && \forall i \in V. \end{aligned} \tag{18}$$

The first constraint enforces the precedence relations encoded by the DAG, while the second ensures that simultaneously running jobs do not exceed the available resource capacity. Binary perturbation is applied to the job-dependency relations E, represented by the binary adjacency matrix $\boldsymbol{H}_b$.

B.2 Minimum Vertex Cover (MVC)

Given an undirected graph $G = (V, E)$, a vertex cover $C \subseteq V$ is a subset of vertices such that every edge in E has at least one endpoint in C. Let $x_i \in \{0, 1\}$ indicate whether vertex i is selected. The MVC problem is formulated as

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \sum_{i\in V} x_i \\ \text{s.t.} \quad & x_i + x_j \geqslant 1, && \forall \{i,j\} \in E, \\ & x_i \in \{0,1\}, && \forall i \in V. \end{aligned} \tag{19}$$

Binary perturbation is applied to the edge set E, represented by the binary adjacency matrix $\boldsymbol{H}_b$.

B.3 Asymmetric Traveling Salesman Problem (ATSP)

Given a set of cities $V = \{1, \ldots, N\}$ and a non-negative directed cost matrix $\boldsymbol{C} \in \mathbb{R}_{\geqslant 0}^{N\times N}$, ATSP seeks a minimum-cost directed Hamiltonian cycle that visits every city exactly once. Unlike the symmetric TSP, the travel costs need not satisfy $C_{ij} = C_{ji}$. We consider metric ATSP instances satisfying the directed triangle inequality

$$C_{ij} + C_{jk} \geqslant C_{ik}, \quad \forall \text{distinct } i, j, k \in V. \tag{20}$$

Let $\mathcal{T}(V)$ denote the set of all directed Hamiltonian cycles over V. The problem can be written as

$$\min_{T\in\mathcal{T}(V)} \sum_{(i,j)\in T} C_{ij}. \tag{21}$$

Tightness perturbation is applied to the inter-city cost matrix $\boldsymbol{C}$, which is represented by $\boldsymbol{H}_t$.

B.4 Capacitated Vehicle Routing Problem (CVRP)

Consider a set of nodes $V = \{v_0, v_1, \ldots, v_N\}$, where v_0 is the depot and $V_c = V \setminus \{v_0\}$ contains the customer nodes. Each customer $i \in V_c$ has a non-negative demand d_i, and each vehicle has capacity $B > 0$. Let $\boldsymbol{C} \in \mathbb{R}_{\geqslant 0}^{(N+1)\times(N+1)}$ denote the travel-cost matrix. A feasible solution is a collection of routes $\mathcal{R}$, where each route starts and ends at v_0, every customer is visited exactly once, and the total demand served by each route does not exceed the vehicle capacity. The problem is

$$\begin{aligned} \min_{\mathcal{R}} \quad & \sum_{r\in\mathcal{R}} \sum_{(i,j)\in r} C_{ij} \\ \text{s.t.} \quad & \{V(r)\backslash\{v_0\} : r \in \mathcal{R}\} \text{ forms a partition of } V_c, \\ & \sum_{i\in V(r)\backslash\{v_0\}} d_i \leqslant B, \quad \forall r \in \mathcal{R}. \end{aligned} \tag{22}$$

In this problem, tightness perturbation is applied to the customer-demand vector $\boldsymbol{d} = (d_1, \ldots, d_N)$, represented by $\boldsymbol{H}_t$.

C Proof Details of Theoretical Results

C.1 Proof of Theorem 3.1

We derive the robustness guarantee of CO tasks against binary perturbation. Starting from the definition of W_1 in Eq. 5,

$$W_1(F_1, F_2) = \int_0^1 |{F_1}^{-1}(u) - {F_2}^{-1}(u)|du = \int_{-\infty}^{\infty} |F_1(r) - F_2(r)|dr. \tag{23}$$

We use this integral-of-CDF form in the proof below. In the case of smoothed version G, respectively evaluated at H_b and $H_b \oplus \delta_b$, we have:

$$\begin{aligned} W_1(G_{\boldsymbol{H_b}}, G_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b}) &= \int_{-\infty}^{+\infty} |G_{\boldsymbol{H}_b}(r) - G_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b}(r)|dr \\ &= \int_{-\infty}^{+\infty} |\sum_{\boldsymbol{z} \sim \{0,1\}^n} F_{\boldsymbol{H}_b \oplus \boldsymbol{z}}(r) p(\boldsymbol{z}) - \sum_{\boldsymbol{z} \sim \{0,1\}^n} F_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b \oplus \boldsymbol{z}}(r) p(\boldsymbol{z})|dr \\ &= \int_{-\infty}^{+\infty} |\sum_{\boldsymbol{u} \sim \{0,1\}^n} F_{\boldsymbol{u}}(r) p(\boldsymbol{u} \oplus \boldsymbol{H}_b) - \sum_{\boldsymbol{u} \sim \{0,1\}^n} F_{\boldsymbol{u}}(r) p(\boldsymbol{u} \oplus \boldsymbol{H}_b \oplus \boldsymbol{\delta}_b)|dr \\ &= \int_{-\infty}^{+\infty} |\sum_{\boldsymbol{u} \sim \{0,1\}^n} F_{\boldsymbol{u}}(r)(p(\boldsymbol{H}_b \oplus \boldsymbol{u}) - p(\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b \oplus \boldsymbol{u}))|dr, \end{aligned} \tag{24}$$

where the third equality follows from the changes of variables $\boldsymbol{u} = \boldsymbol{H}_b \oplus \boldsymbol{z}$ and $\boldsymbol{u} = \boldsymbol{H}_b \oplus \boldsymbol{\delta}_b \oplus \boldsymbol{z}$ respectively. Equivalently, the two smoothed CDFs in the third line are

$$G_{\boldsymbol{H}_b}(r) = \sum_{\boldsymbol{u} \sim \{0,1\}^n} F_{\boldsymbol{u}}(r) p(\boldsymbol{H}_b \oplus \boldsymbol{u}), \quad G_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b}(r) = \sum_{\boldsymbol{u} \sim \{0,1\}^n} F_{\boldsymbol{u}}(r) p(\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b \oplus \boldsymbol{u}). \tag{25}$$

Let $m = \|\boldsymbol{\delta}_b\|_0$. For each $\boldsymbol{u} \in \{0,1\}^n$, let

$$k(\boldsymbol{u}) := |\{i : (\boldsymbol{\delta}_b)_i = 1,\ (\boldsymbol{H}_b \oplus \boldsymbol{u})_i = 1\}| \tag{26}$$

be the number of entries selected by $\boldsymbol{\delta}_b$ that differ between $\boldsymbol{u}$ and $\boldsymbol{H}_b$. Under the change of variables $\boldsymbol{u} = \boldsymbol{H}_b \oplus \boldsymbol{z}$, $k(\boldsymbol{u})$ counts the number of entries equal to one among the m corresponding entries of $\boldsymbol{z}$. Thus, by Eq. 6,

$$\sum_{\boldsymbol{u}: k(\boldsymbol{u}) = k} p(\boldsymbol{H}_b \oplus \boldsymbol{u}) = \binom{m}{k} \beta^k (1-\beta)^{m-k}, \qquad k = 0, \ldots, m. \tag{27}$$

Moreover, $\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b \oplus \boldsymbol{u}$ is obtained from $\boldsymbol{H}_b \oplus \boldsymbol{u}$ by inverting exactly these m entries. Of them, $k(\boldsymbol{u})$ change from one to zero and the other $m - k(\boldsymbol{u})$ change from zero to one, while all remaining entries are unchanged. It follows that the likelihood ratio $\ell(\boldsymbol{u})$ is

$$\ell(u) := \frac{p(\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b \oplus \boldsymbol{u})}{p(\boldsymbol{H}_b \oplus \boldsymbol{u})} = \left(\frac{\beta}{1-\beta}\right)^{m - 2k(\boldsymbol{u})}. \tag{28}$$

Let $K \sim \mathrm{Binomial}(m, \beta)$ and define

$$L := \left(\frac{\beta}{1-\beta}\right)^{m-2K}. \tag{29}$$

Equations 27 and 28 show that, with respect to the probability masses $p(\boldsymbol{H}_b \oplus \boldsymbol{u})$, $\ell(\boldsymbol{u})$ has the same distribution as L. Its mean is one, since

$$\sum_{\boldsymbol{u} \sim \{0,1\}^n} p(\boldsymbol{H}_b \oplus \boldsymbol{u}) \ell(\boldsymbol{u}) = \sum_{\boldsymbol{u} \sim \{0,1\}^n} p(\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b \oplus \boldsymbol{u}) = 1. \tag{30}$$

Let $q_L(t) := \inf\{x : \Pr(L \leqslant x) \geqslant t\}$, $0 < t < 1$, be the generalized quantile function of L. For $v \in [0,1]$, define

$$\underline{T}_{m,\beta}(v) := \int_0^v q_L(t)\, dt, \qquad \overline{T}_{m,\beta}(v) := \int_{1-v}^1 q_L(t)\, dt, \tag{31}$$

and

$$B_{m,\beta}(v) := \max\left\{v - \underline{T}_{m,\beta}(v), \overline{T}_{m,\beta}(v) - v\right\}. \tag{32}$$

We next bound the CDF difference at a fixed $r \in \mathbb{R}$. By Eq. 24,

$$G_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b}(r) = \sum_{\boldsymbol{u} \sim \{0,1\}^n} F_{\boldsymbol{u}}(r) p(\boldsymbol{H}_b \oplus \boldsymbol{u}) \ell(\boldsymbol{u}), \tag{33}$$

where $0 \leqslant F_{\boldsymbol{u}}(r) \leqslant 1$ and

$$\sum_{\boldsymbol{u} \sim \{0,1\}^n} F_{\boldsymbol{u}}(r) p(\boldsymbol{H}_b \oplus \boldsymbol{u}) = G_{\boldsymbol{H}_b}(r). \tag{34}$$

For fixed r, consider the following two finite-dimensional optimization problems, where $a_{\boldsymbol{u}} \in [0,1]$ are the optimization variables:

$$\min / \max_{\{a_{\boldsymbol{u}}\}} \sum_{\boldsymbol{u} \sim \{0,1\}^n} a_{\boldsymbol{u}} p(\boldsymbol{H}_b \oplus \boldsymbol{u}) \ell(\boldsymbol{u}) \quad \text{subject to} \quad \sum_{\boldsymbol{u} \sim \{0,1\}^n} a_{\boldsymbol{u}} p(\boldsymbol{H}_b \oplus \boldsymbol{u}) = G_{\boldsymbol{H}_b}(r). \tag{35}$$

The minimum is obtained by assigning $a_{\boldsymbol{u}} = 1$ in increasing order of $\ell(\boldsymbol{u})$, and the maximum by assigning $a_{\boldsymbol{u}} = 1$ in decreasing order, with a fractional value at the last likelihood-ratio level if necessary. Indeed, if $\ell(\boldsymbol{u}_1) < \ell(\boldsymbol{u}_2)$, $a_{\boldsymbol{u}_1} < 1$, and $a_{\boldsymbol{u}_2} > 0$, transferring an amount ε of probability mass from $\boldsymbol{u}_2$ to $\boldsymbol{u}_1$ preserves the constraint and changes the objective by $\varepsilon(\ell(\boldsymbol{u}_1) - \ell(\boldsymbol{u}_2)) < 0$. Repeating this exchange proves the minimizing arrangement; the maximizing arrangement follows by reversing the exchange.

Since $\ell(\boldsymbol{u})$ has distribution L under the probability masses $p(\boldsymbol{H}_b \oplus \boldsymbol{u})$, the minimum and maximum in Eq. 35 are respectively $\underline{T}_{m,\beta}(G_{\boldsymbol{H}_b}(r))$ and $\overline{T}_{m,\beta}(G_{\boldsymbol{H}_b}(r))$. Taking $a_{\boldsymbol{u}} = F_{\boldsymbol{u}}(r)$, which is feasible by Eq. 34, Eq. 33 therefore gives

$$\underline{T}_{m,\beta}(G_{\boldsymbol{H}_b}(r)) \leqslant G_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b}(r) \leqslant \overline{T}_{m,\beta}(G_{\boldsymbol{H}_b}(r)). \tag{36}$$

By Eq. 32, Eq. 36 yields

$$|G_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b}(r) - G_{\boldsymbol{H}_b}(r)| \leqslant B_{m,\beta}(G_{\boldsymbol{H}_b}(r)). \tag{37}$$

Using the CDF representation in Eq. 23, understood as an extended-real identity until finiteness is established, and integrating Eq. 37, we obtain

$$\begin{aligned} W_1(G_{\boldsymbol{H}_b}, G_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b}) &= \int_{-\infty}^{\infty} |G_{\boldsymbol{H}_b}(r) - G_{\boldsymbol{H}_b \oplus \boldsymbol{\delta}_b}(r)|\, dr \\ &\leqslant \int_{-\infty}^{\infty} B_{m,\beta}(G_{\boldsymbol{H}_b}(r))\, dr. \end{aligned} \tag{38}$$

This establishes the upper bound in Theorem 3.1. When $m = 0$ or $\beta = 1/2$, $L = 1$ almost surely and hence $B_{m,\beta} \equiv 0$, so the same conclusion follows directly. The finiteness of the right-hand side under the stated moment assumption is proved next.

C.2 Boundedness of the Robustness Guarantee in Theorem 3.1

We show that the right-hand side of Eq. 38 is finite under the moment condition in Theorem 3.1. Let $Y_{\boldsymbol{H}_b}$ be a real-valued random variable with CDF $G_{\boldsymbol{H}_b}$, i.e., $G_{\boldsymbol{H}_b}(r) = \Pr(Y_{\boldsymbol{H}_b} \leqslant r)$, and suppose that $\mathbb{E}|Y_{\boldsymbol{H}_b}| < \infty$. For $0 < \beta < 1$, define the finite constant

$$C_{m,\beta} := \max_{0 \leqslant k \leqslant m} \left| \left(\frac{\beta}{1-\beta} \right)^{m-2k} - 1 \right|. \tag{39}$$

For any feasible assignment in Eq. 35, write $v := \sum_{\boldsymbol{u} \sim \{0,1\}^n} a_{\boldsymbol{u}} p(\boldsymbol{H}_b \oplus \boldsymbol{u})$. Then Eqs. 30 and 39 give

$$\begin{aligned} &\left| \sum_{\boldsymbol{u} \sim \{0,1\}^n} a_{\boldsymbol{u}} p(\boldsymbol{H}_b \oplus \boldsymbol{u})(\ell(\boldsymbol{u}) - 1) \right| \leqslant C_{m,\beta} v, \\ &\left| \sum_{\boldsymbol{u} \sim \{0,1\}^n} a_{u} p(\boldsymbol{H}_b \oplus \boldsymbol{u})(\ell(\boldsymbol{u}) - 1) \right| = \left| \sum_{\boldsymbol{u} \sim \{0,1\}^n} (1 - a_{\boldsymbol{u}}) p(\boldsymbol{H}_b \oplus \boldsymbol{u})(\ell(\boldsymbol{u}) - 1) \right| \\ &\qquad \leqslant C_{m,\beta}(1 - v). \end{aligned} \tag{40}$$

Since $B_{m,\beta}(v)$ is the largest absolute change over these feasible assignments, it follows that

$$B_{m,\beta}(v) \leqslant C_{m,\beta} \min\{v, 1-v\}. \tag{41}$$

Splitting the integral at $r = 0$ and using the tail-integral representation of the absolute first moment gives

$$\begin{aligned}\int_{-\infty}^{\infty} B_{m,\beta}(G_{\boldsymbol{H}_b}(r))\, dr &\leqslant C_{m,\beta}\left[\int_{-\infty}^{0} G_{\boldsymbol{H}_b}(r)\, dr + \int_{0}^{\infty}(1 - G_{\boldsymbol{H}_b}(r))\, dr\right] \\ &= C_{m,\beta}\mathbb{E}|Y_{\boldsymbol{H}_b}| < \infty.\end{aligned} \tag{42}$$

Therefore, the robustness guarantee in Theorem 3.1 is finite whenever $\mathbb{E}|Y_{\boldsymbol{H}_b}| < \infty$. For $m = 0$ or $\beta = 1/2$, this also follows immediately from $B_{m,\beta} \equiv 0$.

C.3 Proof of Theorem 3.2

We derive the robustness guarantee of CO tasks against binary perturbation. For a measurable function $h : \mathbb{R}^n \to [0, 1]$, define

$$\hat{h}_\sigma(\boldsymbol{x}) := \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{N}(0,\sigma^2 I_n)}[h(\boldsymbol{x} + \boldsymbol{z})]. \tag{43}$$

Applying Lemma 6 of [44] with variance σ^2, the mapping $\boldsymbol{x} \mapsto \Phi^{-1}((\hat{h}_\sigma(\boldsymbol{x})))$ is $(1/\sigma)$-Lipschitz, where Φ denotes the CDF of the standard normal distribution. For each fixed $r \in \mathbb{R}$, define $h_r(\boldsymbol{H}_t) := F_{\boldsymbol{H}_t}(r)$. Since $F_{\boldsymbol{H}_t}(r) \in [0, 1]$, Gaussian smoothing gives

$$G_{\boldsymbol{H}_t}(r) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{N}(0,\sigma^2 I_n)}[F_{\boldsymbol{H}_t+\boldsymbol{z}}(r)] = \hat{h}_{r,\sigma}(\boldsymbol{H}_t). \tag{44}$$

Hence, $\boldsymbol{H}_t \to \Phi^{-1}(G_{\boldsymbol{H}_t}(r))$ is $(1/\sigma)$-Lipschitz. For any $||\boldsymbol{\delta}_t||_2 \leqslant \eta$ we therefore have

$$|\Phi^{-1}(G_{\boldsymbol{H}_t+\boldsymbol{\delta}_t}(r)) - \Phi^{-1}(G_{\boldsymbol{H}_t}(r))| \leqslant \frac{\eta}{\sigma}. \tag{45}$$

Since Φ is monotone increasing,

$$\Phi(\Phi^{-1}(G_{\boldsymbol{H}_t}(r)) - \frac{\eta}{\sigma}) \leqslant G_{\boldsymbol{H}_t+\boldsymbol{\delta}_t}(r) \leqslant \Phi(\Phi^{-1}(G_{\boldsymbol{H}_t}(r)) + \frac{\eta}{\sigma}). \tag{46}$$

For $a \geqslant 0$ and $u \in (0, 1)$, define

$$B_a(u) := \max\{u - \Phi(\Phi^{-1}(u) - a), \Phi(\Phi^{-1}(u) + a) - u\}, \tag{47}$$

with $B_a(0) = B_a(1) = 0$ defined by continuous extension. It follows from Eq. 46 that, for every $||\boldsymbol{\delta}_t||_2 \leqslant \eta$

$$|G_{\boldsymbol{H}_t+\boldsymbol{\delta}_t}(r) - G_{\boldsymbol{H}_t}(r)| \leqslant B_{\eta/\sigma}(G_{\boldsymbol{H}_t}(r)). \tag{48}$$

Therefore,

$$W_1(G_{\boldsymbol{H}_t}, G_{\boldsymbol{H}_t+\boldsymbol{\delta}_t}) = \int_{-\infty}^{\infty} |G_{\boldsymbol{H}_t}(r) - G_{\boldsymbol{H}_t+\boldsymbol{\delta}_t}(r)|dr \leqslant \int_{-\infty}^{\infty} B_{\eta/\sigma}(G_{\boldsymbol{H}_t}(r))dr. \tag{49}$$

Since the right-hand side does not depend on the particular $\boldsymbol{\delta}_t$, taking the supremum over all admissible perturbations gives

$$\sup_{||\boldsymbol{\delta}_t||_2\leqslant\eta} W_1(G_{\boldsymbol{H}_t}, G_{\boldsymbol{H}_t+\boldsymbol{\delta}_t}) \leqslant \int_{-\infty}^{\infty} B_{\eta/\sigma}(G_{\boldsymbol{H}_t}(r))dr. \tag{50}$$

C.4 Boundedness of the Robustness Guarantee in Theorem 3.2

For any finite $a \geqslant 0$, define

$$I_a := \int_{-\infty}^{\infty} B_a(G_{\boldsymbol{H}_t}(r))dr. \tag{51}$$

In particular, the bound in Eq. 50 corresponds to $a = \eta/\sigma$. We show below that $I_a < \infty$ whenever $\mathbb{E}|Y_{\boldsymbol{H}_t}|^p < \infty$ for some $p > 1$, where $Y_{\boldsymbol{H}_t}$ is a real-valued random variable with CDF $G_{\boldsymbol{H}_t}$.

We first bound $B_a(u)$ for $u \in [0,1]$. Fix any $\gamma \in (0,1)$. For every finite $a \geqslant 0$, there exists a finite constant $K_{a,\gamma} \geqslant 1$ such that

$$B_a(u) \leqslant K_{a,\gamma} \begin{cases} u^\gamma, & 0 \leqslant u \leqslant \frac{1}{2}, \\ (1-u)^\gamma, & \frac{1}{2} \leqslant u \leqslant 1. \end{cases} \tag{52}$$

To prove Eq. 52, consider first $0 \leqslant u \leqslant 1/2$, and let $x := \Phi^{-1}(u) \leqslant 0$. For the first term in Eq. 47,

$$u - \Phi(x-a) \leqslant u \leqslant u^\gamma. \tag{53}$$

For the second term, $\Phi(x+a) - u \leqslant \Phi(x+a)$. Consider the ratio $R_a(x) := \Phi(x+a)/\Phi(x)^\gamma, x \leqslant 0$. As $x \to -\infty$, the Gaussian tail satisfies $\log \Phi(x) = -x^2/2 + O(\log|x|)$. Thus,

$$\log R(x) = -\frac{1-\gamma}{2}x^2 - ax + O(\log|x|) \to -\infty. \tag{54}$$

Hence, $\lim_{x\to-\infty} R(x) = 0$. Since R_a is continuous on $(-\infty, 0]$, it is bounded on this interval. Therefore, there exists a finite constant $K_{a,\gamma} \geqslant 1$ such that

$$\Phi(x+a) \leqslant K_{a,\gamma}\Phi(x)^\gamma = K_{a,\gamma}u^\gamma. \tag{55}$$

Combining Eqs. 53 and 55 gives $B_a(u) \leqslant K_{a,\gamma}u^\gamma, 0 \leqslant u \leqslant 1/2$. Moreover, by the symmetry of the standard normal distribution we have $B_a(u) = B_a(1-u)$. Therefore, $B_a(u) \leqslant K_{a,\gamma}(1-u)^\gamma, 1/2 \leqslant u \leqslant 1$, which proves Eq. 52.

Next, let $Y_{\boldsymbol{H}_t}$ be a real-valued random variable with CDF $G_{\boldsymbol{H}_t}$, i.e., $G_{\boldsymbol{H}_t}(r) = \Pr(Y_{\boldsymbol{H}_t} \leqslant r), r \in \mathbb{R}$. Suppose that, for some $p > 1$, $M_p := \mathbb{E}|Y_{\boldsymbol{H}_t}|^p < \infty$. By Markov's inequality,

$$\Pr(|Y_{\boldsymbol{H}_t}| \geqslant r) \leqslant \frac{M_p}{r^p}, \text{where } r > 0. \tag{56}$$

Choose $\gamma \in (1/p, 1)$, so that $p\gamma > 1$. We further choose $R > 0$ sufficiently large such that $R \geqslant (2M_p)^{1/p}$. Then, for $r \geqslant R$, $G_{H_t}(r) \geqslant 1/2$, while for $r \leqslant -R$, $G_{H_t}(r) \leqslant 1/2$. For the right tail $r \geqslant R$, Eqs. 52 and 56 give

$$\begin{aligned} B_a(G_{\boldsymbol{H}_t}(r)) &\leqslant K_{a,\gamma}(1 - G_{\boldsymbol{H}_t}(r))^\gamma \\ &= K_{a,\gamma}\Pr(Y_{\boldsymbol{H}_t} > r)^\gamma \leqslant K_{a,\gamma}\Pr(|Y_{\boldsymbol{H}_t}| \geqslant |r|)^\gamma \leqslant K_{a,\gamma}{M_p}^\gamma r^{-p\gamma}. \end{aligned} \tag{57}$$

Hence,

$$\int_R^\infty B_a(G_{\boldsymbol{H}_t}(r))dr \leqslant K_{a,\gamma}{M_p}^\gamma \int_R^\infty r^{-p\gamma}dr = \frac{K_{a,\gamma}{M_p}^\gamma}{p\gamma - 1}R^{1-p\gamma} < \infty. \tag{58}$$

Similarly, for the left tail $r \leqslant -R$,

$$\begin{aligned} B_a(G_{\boldsymbol{H}_t}(r)) &\leqslant K_{a,\gamma}G_{\boldsymbol{H}_t}(r)^\gamma \\ &= K_{a,\gamma}\Pr(Y_{\boldsymbol{H}_t} \leqslant r)^\gamma \leqslant K_{a,\gamma}\Pr(|Y_{\boldsymbol{H}_t}| \geqslant |r|)^\gamma \leqslant K_{a,\gamma}{M_p}^\gamma |r|^{-p\gamma}. \end{aligned} \tag{59}$$

Therefore,

$$\int_{-\infty}^{-R} B_a(G_{H_t}(r))dr \leqslant K_{a,\gamma}{M_p}^\gamma \int_{-\infty}^{-R} |r|^{-p\gamma}dr = \frac{K_{a,\gamma}{M_p}^\gamma}{p\gamma - 1}R^{1-p\gamma} < \infty. \tag{60}$$

Finally, since $0 \leqslant B_a(u) \leqslant 1, u \in [0,1]$, the integral over the bounded interval satisfies

$$\int_{-R}^{R} B_a(G_{\boldsymbol{H}_t}(r))dr \leqslant 2R < \infty \tag{61}$$

Combining Eqs. 58, 60, and 61, we obtain

$$I_a = \int_{-\infty}^{\infty} B_a(G_{\boldsymbol{H}_t}(r))dr \leqslant 2R + \frac{2K_{a,\gamma}{M_p}^\gamma}{p\gamma - 1}R^{1-p\gamma} < \infty. \tag{62}$$

Therefore, the robustness bound in Theorem 3.2 is finite provided that $\mathbb{E}|Y_{\boldsymbol{H}_t}|^p < \infty$ for some $p > 1$.

Algorithm 1 Finite-Sample Estimation of Certification Bounds

Require: Solver f_θ, reference instance Q_H, smoothing parameter β or σ, perturbation bound m or η, smoothing sample number k, integration sample number n, actual-cost range $[r_{\min}, r_{\max}]$.

Ensure: Estimated certification bound $\widehat{W}_1$ and its $(1-\delta)$ confidence interval.

for $j = 1$ to k **do**
  if $\boldsymbol{H} = \boldsymbol{H}_b$ **then**
    Sample $\boldsymbol{z}^{(j)}$ according to Eq. (6). $\quad \boldsymbol{H}^{(j)} \leftarrow \boldsymbol{H}_b \oplus \boldsymbol{z}^{(j)}$.
  else if $\boldsymbol{H} = \boldsymbol{H}_t$ **then**
    Sample $\boldsymbol{z}^{(j)} \sim \mathcal{N}(0, \sigma^2 I_n)$. $\quad \boldsymbol{H}^{(j)} \leftarrow \boldsymbol{H}_t + \boldsymbol{z}^{(j)}$.
  end if
  $y^{(j)} \leftarrow c(f_\theta(Q_{\boldsymbol{H}^{(j)}}), Q_{\boldsymbol{H}^{(j)}})$.
end for
Sample $r_i \sim U(r_{\min}, r_{\max})$, $i = 1, \ldots, n$.
for $i = 1$ to n **do**
  Compute $\widehat{G}_{\boldsymbol{H}}(r_i)$ using Eq. (64).
  if $\boldsymbol{H} = \boldsymbol{H}_b$ **then**
    $b_i \leftarrow B_{m,\beta}(\widehat{G}_{\boldsymbol{H}}(r_i))$.
  else if $\boldsymbol{H} = \boldsymbol{H}_t$ **then**
    $b_i \leftarrow B_{\eta/\sigma}(\widehat{G}_{\boldsymbol{H}}(r_i))$.
  end if
  Compute B_i^- and B_i^+ using Eq. (71).
end for
$\widehat{W}_1 \leftarrow \dfrac{r_{\max} - r_{\min}}{n} \displaystyle\sum_{i=1}^{n} b_i$.
Compute the $(1-\delta)$ confidence interval using Eq. (73).
return $\widehat{W}_1$ and the $(1-\delta)$ confidence interval.

D Finite-Sample Estimation of Certification Bounds

The certification bounds in Theorems 3.1 and 3.2 depend on the smoothed CDF $G_{\boldsymbol{H}}$, which cannot be evaluated exactly in practice. We therefore estimate $G_{\boldsymbol{H}}$ from a finite number of smoothing samples and numerically evaluate the corresponding integral over the actual-cost range.

Given a reference instance $Q_{\boldsymbol{H}}$, we draw k independent smoothing samples according to the smoothing scheme in Sec. 3.1. For binary perturbation, $\boldsymbol{z}^{(j)}$ is sampled according to Eq. 6; for tightness perturbation, $\boldsymbol{z}^{(j)} \sim \mathcal{N}(0, \sigma^2 I_n)$. Each smoothing sample produces an actual cost

$$y^{(j)} = c(f_\theta(Q_{\boldsymbol{H}^{(j)}}), Q_{\boldsymbol{H}^{(j)}}), \tag{63}$$

where

$$\boldsymbol{H}^{(j)} = \begin{cases} \boldsymbol{H}_b \oplus \boldsymbol{z}^{(j)}, & \boldsymbol{H} = \boldsymbol{H}_b, \\ \boldsymbol{H}_t + \boldsymbol{z}^{(j)}, & \boldsymbol{H} = \boldsymbol{H}_t. \end{cases}$$

The smoothed CDF is then estimated by

$$\widehat{G}_{\boldsymbol{H}}(r) = \frac{1}{k} \sum_{j=1}^{k} \mathbf{1}\left\{ y^{(j)} \le r \right\}. \tag{64}$$

Let $[r_{\min}, r_{\max}]$ denote the actual-cost range used for numerical integration. We independently sample $r_i \sim U(r_{\min}, r_{\max})$, $i = 1, \ldots, n$. The certification bound is estimated as

$$\widehat{W}_1 = \frac{r_{\max} - r_{\min}}{n} \sum_{i=1}^{n} B\left(\widehat{G}_{\boldsymbol{H}}(r_i) \right), \tag{65}$$

where

$$B(v) = \begin{cases} B_{m,\beta}(v), & \boldsymbol{H} = \boldsymbol{H}_b, \\ B_{\eta/\sigma}(v), & \boldsymbol{H} = \boldsymbol{H}_t. \end{cases} \tag{66}$$

Here, $B_{m,\beta}$ and $B_{\eta/\sigma}$ are defined in Theorems 3.1 and 3.2, respectively. Thus, the same finite-sample estimation procedure applies to both certification settings.

Algorithm 2 Robustness Enhancement for CO.

Input: solver f_θ, reference instance $Q_{\boldsymbol{H}}$, number of samples n_e, and either the maximum number of added constraints m_e or the Gaussian sampling parameter σ.
Output: enhanced solution $\hat{\boldsymbol{x}}_e$ and its cost $c(\hat{\boldsymbol{x}}_e, Q_{\boldsymbol{H}})$.
$Q_{\boldsymbol{H}^*} \leftarrow Q_{\boldsymbol{H}}$, $\hat{\boldsymbol{x}}_e \leftarrow f_\theta(Q_{\boldsymbol{H}})$, $c_{\min} \leftarrow c(\hat{\boldsymbol{x}}_e, Q_{\boldsymbol{H}})$.
for $i = 1$ **to** n_e **do**
  if $\boldsymbol{H} = \boldsymbol{H}_b$ **then**
    generate $\boldsymbol{H}'^{(i)}$ by changing at most m_e zero entries in $\boldsymbol{H}_b$ to one.
  else if $\boldsymbol{H} = \boldsymbol{H}_t$ **then**
    sample $\boldsymbol{z}^{(i)} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 I_n)$ and set $\boldsymbol{H}'^{(i)} \leftarrow \boldsymbol{H}_t + |\boldsymbol{z}^{(i)}|$ element-wise.
  end if
  $\hat{\boldsymbol{x}}^{(i)} \leftarrow f_\theta(Q_{\boldsymbol{H}'^{(i)}})$.
  $c_i \leftarrow c(\hat{\boldsymbol{x}}^{(i)}, Q_{\boldsymbol{H}})$.
  if $c_i \leqslant c_{\min}$ **then**
    $c_{\min} \leftarrow c_i$, $Q_{\boldsymbol{H}^*} \leftarrow Q_{\boldsymbol{H}'^{(i)}}$, $\hat{\boldsymbol{x}}_e \leftarrow \hat{\boldsymbol{x}}^{(i)}$.
  end if
end for
return $\hat{\boldsymbol{x}}_e$ and $c_{\min}$.

Confidence interval. We further account for the sampling uncertainty from estimating $G_{\boldsymbol{H}}$ and from numerically evaluating the integral over r. Let $1-\delta$ denote the desired overall confidence level, and choose $\delta_G, \delta_r > 0$ such that $\delta_G + \delta_r = \delta$. For the k smoothing samples, the Dvoretzky–Kiefer–Wolfowitz inequality gives

$$\Pr\left[\sup_{r\in\mathbb{R}} \left|\widehat{G}_{\boldsymbol{H}}(r) - G_{\boldsymbol{H}}(r)\right| \le \epsilon_G\right] \ge 1-\delta_G, \tag{67}$$

where

$$\epsilon_G = \sqrt{\frac{\log(2/\delta_G)}{2k}}. \tag{68}$$

Hence, simultaneously for all r,

$$G_{\boldsymbol{H}}(r) \in \left[\max\left\{0, \widehat{G}_{\boldsymbol{H}}(r) - \epsilon_G\right\}, \min\left\{1, \widehat{G}_{\boldsymbol{H}}(r) + \epsilon_G\right\}\right] \tag{69}$$

with probability at least $1-\delta_G$. For each sampled r_i, we define

$$B_i^- = \min_{v\in\left[\max\{0,\widehat{G}_{\boldsymbol{H}}(r_i)-\epsilon_G\},\, \min\{1,\widehat{G}_{\boldsymbol{H}}(r_i)+\epsilon_G\}\right]} B(v), \tag{70}$$

$$B_i^+ = \max_{v\in\left[\max\{0,\widehat{G}_{\boldsymbol{H}}(r_i)-\epsilon_G\},\, \min\{1,\widehat{G}_{\boldsymbol{H}}(r_i)+\epsilon_G\}\right]} B(v). \tag{71}$$

Since $0 \le B(v) \le 1$, Hoeffding's inequality gives the numerical-integration error

$$\epsilon_r = \sqrt{\frac{\log(2/\delta_r)}{2n}}. \tag{72}$$

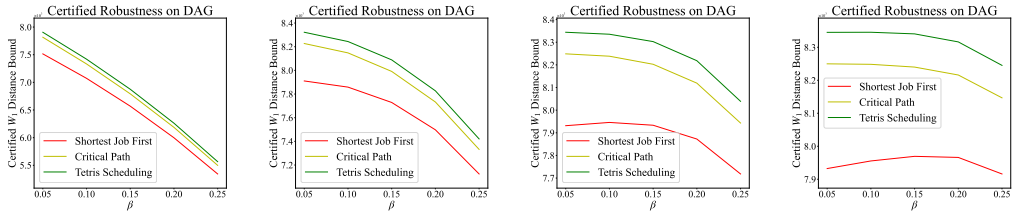
Combining the two sampling stages by the union bound, the certification bound lies, with probability at least $1-\delta$, in

$$\left[(r_{\max}-r_{\min})\max\left\{0,\ \frac{1}{n}\sum_{i=1}^{n} B_i^- - \epsilon_r\right\},\ (r_{\max}-r_{\min})\min\left\{1,\ \frac{1}{n}\sum_{i=1}^{n} B_i^+ + \epsilon_r\right\}\right]. \tag{73}$$

In our experiments, we use $k = n = 1000$ and set the overall confidence level to $1-\delta = 0.95$, with $\delta_G = \delta_r = \delta/2 = 0.025$.

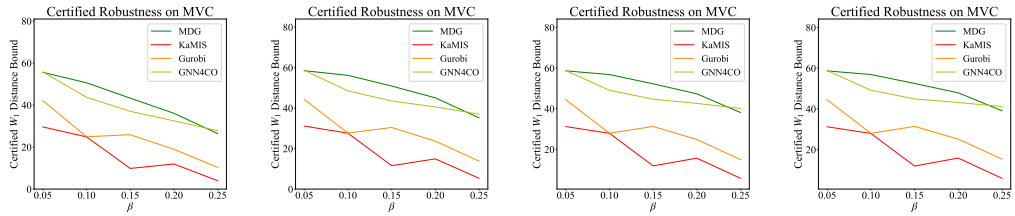
E Algorithm Details

The details of the two proposed algorithms are given in the two algorithm charts: Alg. 1 and Alg. 2.



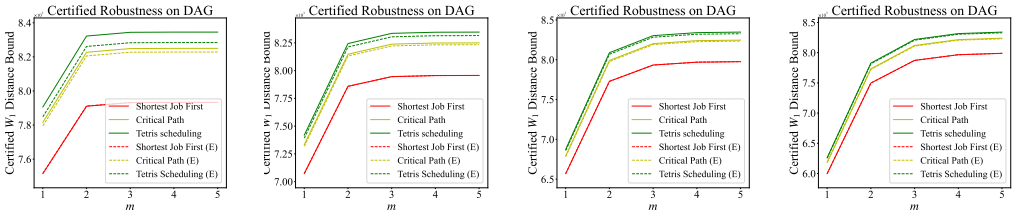
(a) $m = 1$ (b) $m = 2$ (c) $m = 3$ (d) $m = 4$

Figure 6: Robustness guarantee by varying perturbation β w/ fixed m on TPC-H-50.



(a) $m = 1$ (b) $m = 2$ (c) $m = 3$ (d) $m = 4$

Figure 7: Robustness by varying perturbation β with fixed m, on MVC (SMALL).



(a) $\beta = 0.05$ (b) $\beta = 0.1$ (c) $\beta = 0.15$ (d) $\beta = 0.2$

Figure 8: Robustness enhancement on certified robustness by varying m w/ fixed β, on TPC-H-50.

F Further Experimental Results

We provide additional analysis and description of the experiments.

F.1 Additional Evaluation for Certified Robustness

We further examine the certified W_1 upper bound as a function of the smoothing parameter β. Fig. 6 and Fig. 7 show that the certified bound generally decreases as β increases, with the reduction being more pronounced for smaller m. As β approaches 0.5, the likelihood-ratio random variable L defined in Theorem 3.1 becomes increasingly concentrated around 1, which reduces the deviation characterized by $B_{m,\beta}$. For larger m, this reduction becomes less pronounced in our experiments. Meanwhile, since the smoothed CDF $G_{\boldsymbol{H}_b}$ itself also depends on β, the resulting certification bound reflects the combined effect of the smoothing distribution and $B_{m,\beta}$.

F.2 Further Evaluation for Robustness Enhancement

Enhancement for Certified Robustness. We contrast the certified robustness outcomes pre- and post-application of the robustness enhancement method as depicted in Fig. 8. Solvers equipped with the enhancement method are denoted by E. Notably, aside from the SJF method, the W_1 distance for enhanced models in both TETRIS and CP is markedly reduced compared to their original solvers.

Enhancement for Empirical Robustness. We detail its performance across varying dataset dimensions and parameters. First, we present the outcomes for instances with varying city counts in Tab. 5, affirming the enhancement method's efficacy in bolstering robustness across datasets of diverse dimensions. Second, we investigate the effect of the parameters m_e and σ on the robustness

Table 5: Evaluation of the robustness enhancement method on the Asymmetric TSP for different datasets using NN solver.

City #	CLEAN ($\times 10^6$)	CLEAN(E) ($\times 10^6$)	Method	Attack Method($\times 10^6$) ROCO-RA	ROCO-OG	ROCO-SA	ROCO-RL
20	1.9354	1.9011	ATTACK	2.0873	1.9484	2.1265	2.1655
			ATTACK(E)	1.9303	1.9416	1.9800	2.1314
50	2.2246	2.1436	ATTACK	2.3504	2.2268	2.3599	2.3832
			ATTACK(E)	2.1940	2.1961	2.2139	2.1666
100	2.1456	2.0630	ATTACK	2.2209	2.1456	2.2250	2.2450
			ATTACK(E)	2.0974	2.0976	2.1087	2.0639

Table 6: Evaluation of the robustness enhancement method on DAG scheduling using SJF solver by maximum perturbation number m_e.

m_e	CLEAN ($\times 10^5$)	CLEAN(E) ($\times 10^5$)	Method	Attack Method($\times 10^5$) ROCO-RA	ROCO-OG	ROCO-SA	ROCO-RL
1	1.0461	1.0366	ATTACK	1.0559	1.0609	1.0610	1.0604
			ATTACK(E)	1.0366	1.0472	1.0465	1.0450
2	1.0461	1.0353	ATTACK	1.0559	1.0609	1.0610	1.0604
			ATTACK(E)	1.0358	1.0424	1.0344	1.0445
3	1.0461	1.0352	ATTACK	1.0559	1.0609	1.0610	1.0604
			ATTACK(E)	1.0353	1.0405	1.0400	1.0434
4	1.0461	1.0366	ATTACK	1.0559	1.0609	1.0610	1.0604
			ATTACK(E)	1.0335	1.0396	1.0403	1.0402

Table 7: Evaluation of our empirical defense method on Asymmetric TSP using NN solver with σ.

σ	CLEAN ($\times 10^6$)	CLEAN(E) ($\times 10^6$)	Method	Attack Method($\times 10^6$) ROCO-RA	ROCO-OG	ROCO-SA	ROCO-RL
1	1.9354	1.9011	ATTACK	2.0873	1.9484	2.1265	2.1655
			ATTACK(E)	1.9303	1.9416	1.9800	2.1314
2	1.9354	1.9181	ATTACK	2.0873	1.9484	2.1265	2.1655
			ATTACK(E)	1.9180	1.9306	1.9647	2.1228
3	1.9354	1.9183	ATTACK	2.0873	1.9484	2.1265	2.1655
			ATTACK(E)	1.9182	1.9301	1.9649	2.1237
4	1.9354	1.9186	ATTACK	2.0873	1.9484	2.1265	2.1655
			ATTACK(E)	1.9185	1.9302	1.9642	2.0978

enhancement method. Tab. 6 demonstrates that an increase in m_e enhances the model's empirical robustness. Tab. 7 suggests that an increase in σ enhances the model's empirical robustness, up to a critical threshold beyond which efficacy may diminish.