ML-Guided Primal Heuristics for Mixed Binary Quadratic Programs

Weimin Huang
University of Southern California, USA

Natalie M. Isenberg
Pacific Northwest National Laboratory, USA

Ján Drgoňa
Johns Hopkins University, USA

Draguna L Vrabie
Pacific Northwest National Laboratory, USA

Bistra Dilkina
University of Southern California, USA

Abstract

Mixed Binary Quadratic Programs (MBQPs) are classic problems in combinatorial optimization. As solving large-scale combinatorial optimization problems is challenging, primal heuristics have been developed to quickly identify high-quality solutions within a short amount of time. Recently, a growing body of research has also used machine learning to accelerate solution methods for challenging combinatorial optimization problems. Despite the increasing popularity of these ML-guided methods, a large body of work has focused on Mixed-Integer Linear Programs (MILPs). MBQPs are challenging to solve due to the combinatorial complexity coupled with nonlinearities. This work proposes ML-guided primal heuristics for Mixed Binary Quadratic Programs (MBQPs) by adapting and extending existing work on ML-guided MILP solution prediction to MBQPs. We propose a new neural network architecture for MBQP solution prediction and a new data collection procedure for training. Moreover, we propose to combine Binary Cross-Entropy loss and Contrastive Loss in solution prediction. We compare the methods on standard and real-world MBQP benchmarks and show that our proposed methods significantly outperform state-of-the-art solvers and existing primal heuristics.

1 Introduction

Mixed Binary Quadratic Programs (MBQPs) are discrete optimization problems with quadratic terms in the objective function subject to a set of linear constraints. MBQPs encode many important problems in Combinatorial Optimization (CO) [20, 27, 18] and cover a wide range of applications, including finance [25], machine learning [5], as well as chemical [23] and energy systems [29]. A significant body of research on CO algorithms has focused on *primal heuristics*, which are algorithms designed to find good feasible solutions quickly and without optimality guarantees [3].

Despite development in solvers and heuristics, solving large-scale COs remains challenging. In recent years, Machine Learning (ML) has been proposed to accelerate solution methods for CO problems. Motivated by the fact that CO problems sharing similar structures are solved repeatedly in many applications [16, 28], a growing body of research uses ML to guide algorithmic policies or to build new policies customized to instances that appear in specific applications. For example, [14, 24, 15] proposed ML-guided primal heuristics for Mixed Integer Linear Programs (MILPs), wherein they they predict the optimal assignment for a subset of the variables. While prior work on ML-guided

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Differentiable Learning of Combinatorial Algorithms.

CO methods has shown success across multiple algorithmic components on many challenging CO problems, existing work in this area has mainly focused on MILPs. A small body of research has used ML to advance solution methods for general nonlinear programming problems [8, 1, 13, 11], but ML-guided methods in this space are not as well developed as in MILPs.

MBQPs are even more challenging to solve than MILPs due to the combinatorial nature [22] coupled with nonlinearities. In this work, we develop ML-guided primal heuristics for MBQPs by adapting and extending existing work on ML-guided MILP solution methods. We adapt the Weighted Cross-Entropy-based and Contrastive Learning-based methods which are used in MILP solution prediction to MBQPs. We propose a novel neural network architecture that extracts MBQP features and produces variable embeddings, and a new data collection procedure that generates high-quality solutions as ground truth data for training. Furthermore, we extend existing loss functions used in solution prediction and propose to combine Binary Cross-Entropy loss and Contrastive Loss. We compare the proposed methods on standard and real-world MBQP benchmarks and show that our methods outperform state-of-the-art solvers and non-ML primal heuristics.

2 Background and Related Work

2.1 Mixed Binary Quadratic Programs

A Mixed Binary Quadratic Program (MBQP) with n decision variables is defined as

$$\min x^T Hx + c^T x \quad \text{s.t. } Ax \leq b \text{ and } x_j \in \{0, 1\}, \forall j \in B \tag{1}$$

where $H \in \mathbb{R}^{n\times n}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, and $b \in \mathbb{R}^m$. H is a real symmetric matrix that encodes quadratic terms in the objective function and is not necessarily positive semidefinite, allowing for nonconvex objective functions. $B \subseteq \{1, ..., n\}$ is the set of binary decision variables.

Solution methods MBQPs are NP-hard in general [26]. The Branch-and-Bound (BnB) algorithm is an exact tree search algorithm to solve MILPs, MBQPs and more general Mixed-Integer Nonlinear Programming problems. As large-scale MBQPs are challenging to solve with exact methods, a significant body of research has focused on *primal heuristics*, which are algorithms designed to quickly identify high-quality feasible solutions for a given optimization problem without optimality guarantees [3]. These heuristics typically involve solving a relaxation of the original problem and then creating a subproblem by fixing a subset of integer variables by rounding the relaxation values to the nearest integer values, such as RENS [3], Undercover [4], and Relax-Search [17].

2.2 Solution Prediction for MILPs

Previous work on using ML to accelerate solving CO problems has been focused on Mixed Integer Linear Programming (MILP). An MILP can be viewed as the subclass of MBQPs in Eqn. 1 where the quadratic term matrix H is the zero matrix. The goal of an MILP is to find x such that $c^T x$ is minimized, subject to $Ax \leq b$ and integrality constraints $x_j \in \{0, 1\}, \forall j \in B$. A large body of ML-guided primal heuristics for MILPs are based on predicting partial solutions [24, 14, 15].

Solution Prediction Nair et al. [24] and Han et al. [14] use Weighted Cross-Entropy (WCE) loss to learn the probability distribution of the solution space of an MILP instance M. The goal is to learn from a set of multiple solutions, weighted by the quality of the solution. Specifically, for a solution $\boldsymbol{x}$, the energy function $E(\boldsymbol{x}; M)$ is defined as $c^T \boldsymbol{x}$ if $\boldsymbol{x}$ is feasible, or ∞ otherwise, assuming minimization. Given M, the conditional distribution of a solution $\boldsymbol{x}$ is modeled as

$$p(x|M) \equiv \frac{\exp(-E(x; M))}{\sum_{x'} \exp(-E(x'; M))} \tag{2}$$

, so that solutions with better objective values have higher probability. The learning task is to train a model $p_\theta(x|M)$ parameterized by θ that approximates $p(x|M)$. To collect training data, [24] and [14] obtain the set of solutions by running state-of-the-art MILP solvers for a large amount of time. Instead of using WCE loss, Huang et al. [15] learn $p_\theta(x|M)$ using Contrastive Learning (CL). The CL-based method makes discriminative predictions by contrasting the positive samples (i.e., good solutions) and negative samples (i.e., bad solutions). Positive samples are obtained by running MILP solvers, similar to [24] and [14]. Negative samples are obtained by solving another MILP that searches for bad variable assignments within some Hamming distance of the good solutions.

Inference Since the full prediction might not be feasible, ML-guided primal heuristics for MILPs involve solving another MILP at inference time. Nair et al.[24] use Neural Diving (ND), which uses the prediction of a subset of the variables and creates a smaller sub-MILP that is easier to solve after fixing the subset. The size of this sub-MILP is controlled by the ratio of variables that are fixed. Han et al. [14] and Huang et al. [15] use a Predict-and-Search (PaS) framework that searches for feasible solutions within some neighborhood of the full prediction by adding a cut to the original MILP. The degrees of freedom in PaS are controlled by the number of variables that are allowed to be different from the prediction. The ND approach allows for faster runtime at inference time as the subproblem contains a small number of variables, but the solutions returned can be more suboptimal. PAS has more freedom to correct errors from the ML predictions, but can be harder to optimize because the size of the MILP to solve at inference contains the same number of variables as the original MILP.

3 Methods

We develop ML-guided primal heuristics for MBQPs. An input MBQP (Fig. 1 (A)) is represented as a tripartite graph (Fig. 1 (B)) and then passed to a Graph Attention Network (Fig. 1 (C)) which predicts the solutions to the binary decision variables. At inference time, the predictions are used to create a sub-MBQP (Fig. 1 (F)). We introduce a new method to collect training data for MBQPs (Fig. 1 (D)). For training, we adapt the WCE and CL losses which have been used in solution prediction in MILPs to MBQPs and propose to combine Binary Cross-Entropy (BCE) and CL losses (Fig. 1 (E)).

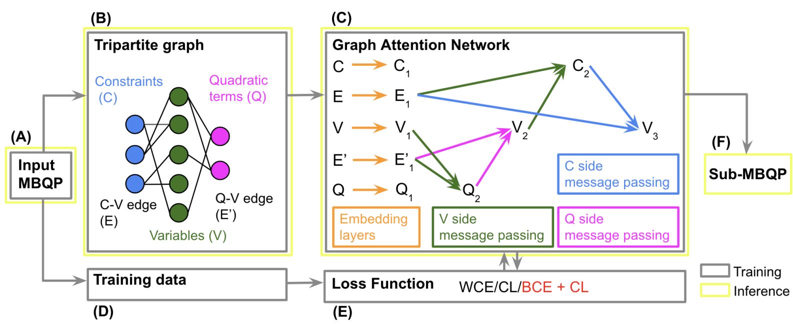

Figure 1: Training/Inference pipeline for ML-guided MBQP solving via solution prediction.

3.1 Neural Network Architecture

We propose a tripartite graph representation for MBQPs (Fig. 1 (B)). It contains three sets of nodes that represent the constraints (C), variables (V), and quadratic terms (Q). A $C - V$ edge connects a variable and a constraint if the variable has a non-zero coefficient in the constraint. A $Q - V$ edge connects two V nodes if the two variables appear in the same quadratic term. The features for the C and V nodes are adapted from solution prediction for MILPs in [14]. For the Q nodes, we propose a custom feature set that captures the characteristics of the H matrix in Eqn. (1). We learn a policy $p_\theta(x|M)$ parameterized by θ, using a Graph Attention Network (GAT) [9]. The GAT performs four rounds of message passing and produces an embedding for the variables (Fig. 1 (C)). The embeddings are then passed through a Multi-Layer Perceptron (MLP) followed by an activation layer to obtain the final output $p_\theta(x|M)$. The input features and ML module details are deferred to Appendix B.

3.2 Loss Function

The policy $p_\theta(x|M)$ that produces solution predictions can be learned with different approaches (Fig. 1 (E)). In this work, we first adapt the WCE and CL losses that have been used for MILPs to MBQPs. Then, we propose an extension that combines BCE and CL losses.

Weighted Cross-Entropy Following the WCE approach [14], we create a training dataset that contains N MBQPs instances $\left\{\left(M^i,\ L^i\right)\right\}_{i=1}^{N}$, where $L^i \equiv \left\{x^{i,j}\right\}_{j=1}^{N_i}$ is a set of unique N_i solutions for the instance M_i. Let $P_\theta\left(x^{i,j}|M^i\right)$ denote the probability of solution $x^{i,j}$ given instance M^i as the input. We adapt the energy function $E(\boldsymbol{x}; M)$ in Eqn. (2) to the case of MBQPs to assign higher probability for better solutions. For a solution $\boldsymbol{x}$, the energy function $E(\boldsymbol{x}; M)$ is defined as $x^T H x + c^T x$ if $\boldsymbol{x}$ is feasible. During training, for instance M_i with quadratic term matrix H^i and cost vector c^i, the weight applied to the solution $x^{i,j}$ is $w^{i,j} \equiv \frac{\exp\left(-x^{i,j\,T} H^i x^{i,j} - c^{i\top} x^{i,j}\right)}{\sum_{k=1}^{N_i} \exp\left(-x^{i,k\,T} H^i x^{i,k} - c^{i\top} x^{i,k}\right)}$. Based on the Kullback-Leibler divergence which measures the distance between the conditional distribution in Equation 2 and the learned policy, the loss function to be minimized is:

$$\mathcal{L}_{\text{WCE}}(\theta) \equiv -\sum_{i=1}^{N} \sum_{j=1}^{N_i} w^{i,j} \log P_\theta \left(x^{i,j} | M^i\right). \tag{3}$$

Contrastive Learning Following the CL-based approach [15], let $\left\{\left(S_+^{M_i}, S_-^{M_i}\right)\right\}_{i=1}^{N}$ be a training dataset of N MBQP instances, where $S_+^{M_i}$ and $S_-^{M_i}$ are the sets of positive and negative samples for instance M_i, respectively. We use a form of the NT-Xent Loss [10] to learn to distinguish between positive and negative samples. We use the the $\cdot$ operator to denote the dot-product similarity. Let $p_\theta(M^i)$ be the predicted solution given instance M^i as the input. The loss function to be minimized is

$$\mathcal{L}_{\text{CL}}(\theta) = \sum_{\{(\mathcal{S}_+^{M^i}, \mathcal{S}_-^{M^i})\}_{i=1}^{N}} \frac{1}{\left|\mathcal{S}_+^{M^i}\right|} \sum_{x_+ \in \mathcal{S}_+^{M^i}} \mathcal{L}^+\left(\theta \mid \boldsymbol{x}_+, M^i\right), \tag{4}$$

where

$$\mathcal{L}^+\left(\theta \mid x_+, M^i\right) = -\log \frac{\exp\left(x_+ \cdot p_\theta(M^i) / \tau(x_+ | M^i)\right)}{\sum_{\tilde{x} \in S_-^{M^i} \cup \{x_+\}} \exp\left(\tilde{x} \cdot p_\theta(M^i) / \tau(\tilde{x} | M^i)\right)}. \tag{5}$$

Based on the dot-product similarity, the value of the loss $\mathcal{L}^+\left(\theta \mid x_+, M^i\right)$ is low when $p_\theta(M^i)$ is similar to the positive sample x_+ and dissimilar to negative samples $\tilde{x} \in S_-^{M^i}$. In the case of MILPs in [15], a sample weight of $\frac{1}{\tau(x|M^i)} = c^{iT} x / w$ where $w < 0$ is applied to minimization problems with a negative objective values, so that positive samples with lower (i.e., better) objective values are assigned higher weights. We adapt the weights and capture the objective values of MBQPs. Moreover, to also account for minimization problems with positive objective values, we transform the weights using the exponential function and set $\frac{1}{\tau(x|M^i)} = \exp(\frac{x^T H^i x + c^{iT} x}{w})$ where $w < 0$, so that better positive samples have lower weights in both minimization problems with positive objective and minimization problems with negative objective values. A discussion of sample weights applied to problems with different objective values is deferred to Appendix D.

Combining Contrastive Learning and Binary Cross-Entropy In addition to adapting the WCE and CL losses to MBQPs, we propose to combine CL and Binary Cross-Entropy (BCE) loss. It has been observed that a subset of variables often have the same assignments across different positive and negative samples in the CL-based approach. This motivates a binary classification approach for this unique subset of variables. Formally, given an MBQP instance M^i with the set of positive and negative samples $\left(\mathcal{S}_+^{M^i}, \mathcal{S}_-^{M^i}\right)$, let $B^i = \{1, ..., n\}$ be the index set of all binary decision variables. Let $x_+ \in \mathcal{S}_+^{M^i}$ be any positive sample. Let x_d^i denote the solution assignment for the d^{th} variable for instance i. Let $U^i \subseteq B^i$ be an index set for which x_d^i takes the same solution value for any $x \in S_-^{M^i} \cup \{x_+\}$. We learn the assignment of variables in U^i by classifying whether the variable is assigned 1 or 0. Let $\hat{p}_d^i \equiv p_\theta\left(x_d^i = 1 | M^i\right)$ be the probability that x_d^i takes a solution value of 1 predicted by the ML model. The classification loss for instance M^i is

$$\mathcal{L}_{\text{BCE}}^{U^i}(\theta) = -\sum_{i=1}^{N} \sum_{d \in U^i} \left[t_d^i \log(\hat{p}_d^i) + (1 - t_d^i) \log(1 - \hat{p}_d^i)\right]$$

where t_d^i is the ground truth value for x_d^i in M^i. For variables in $B^i \setminus U^i$, we apply CL loss. Let $\mathcal{L}_{CL}^{B^i \setminus U^i}(\theta)$ be the same CL loss function defined in Eqn. (4) but operate on the subset of variables $B^i \setminus U^i$ instead. Considering both CL and BCE losses, the combined loss to be minimized is

$$\mathcal{L}_{\text{CL+BCE}}(\theta) = \lambda_{\text{CL}} \mathcal{L}_{\text{CL}}^{B^i \setminus U^i}(\theta) + \mathcal{L}_{\text{BCE}}^{U^i}(\theta)$$

, where λ_{CL} is a hyperparameter that controls the weight of CL loss.

3.3 Training data collection for MBQPs

Training data collection (Fig. 1 (D)) consists of compiling multiple good solutions that can be used for solution prediction in MBQPs. We propose *Randomized Relax-Search*, a novel heuristic that produces a set of diverse high-quality solutions for MBQPs. *Randomized Relax-Search* is extended from the *Relax-Search* [17] heuristic, which uses a suboptimal relaxation solution of the MBQP as the basis, fixes a subset of variables using the rounded relaxation, and searches over a sub-MBQP. *Randomized Relax-Search* introduces randomization to create K sub-MBQPs, as shown in Algorithm 1. In solving the k^{th} sub-MBQP, the best solution x_+^k and the worst solution x_-^k are stored. The procedure returns the set of best solutions S_+ and worst solutions S_- after solving K sub-MBQPs.

Algorithm 1 Randomized Relax-Search for training data collection

Require: A MBQP $\mathcal{P}$ with set of binary variables $\mathcal{B}$, relaxation time limit T_r, subproblem time limit T_s, number of random seeds K, candidate fixing ratio p_1, final fixing ratio p_2 (p_1>p_2)

```
Relaxed solutions x̄ ← Compute the Nonlinear Programming relaxation of P given time limit T_r
Set of good solutions S_+ ← ∅
Set of bad solutions S_- ← ∅
Candidate set C ← select p_1 ∗ |B| variables that are least fractional variables in x̄.
for k ∈ 1, 2, ..., K do
    U_k ← Randomly and uniformly select p_2 ∗ |B| variables from C
    for i ∈ U_k do
        Fix x_i = ⌊x̄_i⌉ by rounding to the nearest integer
    end for
    x_+^k, x_-^k ← Best and worst solutions obtained by solving the k^th sub-MBQP with a complete
        solver, given time limit T_s.
    S_+ ← S_+ ∪ {x_+^k}
    S_- ← S_- ∪ {x_-^k}
end for
return S_+, S_-
```

For training with WCE loss, the set of good solutions S_+ is used. For CL losses, S_+ is used as the set of positive samples. We denote the worst solution value from S_+ as $v' = \max_{x \in S_+} x^T H x + c^T x$. For the set of negative samples in CL, we use $\{x | (x^T H x + c^T x) > v', x \in S_-\}$. In other words, we only include solutions in S_- that have worse objective values than the worst solutions in S_+.

3.4 Inference

At inference time, we choose to use the ND-based method discussed in Subsection 2.2 which reduces the original problem to a smaller sub-MBQP (Fig. 1) (F), as our goal is to develop fast primal heuristics. The PaS-based method is challenging for MBQPs because it requires solving another MBQP of the same size. After obtaining the variable predictions, we create a sub-MBQP by fixing the top p percent of binary decision variables for which the ML model is most confident with the predictions (i.e., least fractional in the predictions). The sub-MBQP is then solved with a CO solver.

4 Computational Experiments

Benchmarks We evaluate the methods on three standard benchmarks: Cardinality-constrained Binary Quadratic Programs (CBQP) [30], Cardinality-constrained Quadratic Knapsack Problem (CQKP) [19], and the Quadratic Multidimensional Knapsack Problem (QMKP) [12]. All standard benchmark instances contain 1000 binary variables and have a quadratic term density of 0.25. In addition, we test on a real-world *Wind Farm Layout Optimization* (WFLOP) problem. WFLOP seeks to identify the placement of a set of wind turbines to maximize power generation over all wind scenarios while also satisfying minimum separation constraints. We generate the WFLOP instances based on the MBQP formulation in [17], using wind data from the NOW-23 offshore wind dataset at selected locations in the California offshore region [7].

Evaluation Metrics We use the following metrics: (1) The *Primal Gap* (PG) [2] is the normalized difference between the objective value v found by a method and a best known objective value v^*, defined as PG $= \frac{|v - v^*|}{max(|v|,|v^*|)}$, when $vv^* > 0$. When no feasible solution is found or when $vv^* < 0$, PG is defined to be 1. PG is 0 when $|v| = |v^*| = 0$. (2) The *Primal Integral* (PI) [2] is the integral of the primal gap over time, which captures the speed at which better solutions are found.

Baselines First, we compare with the SCIP solver [6] with primal heuristics integrated. SCIP uses BnB as its core component and includes primal heuristics as supplementary procedures to improve the primal bound during BnB. We turn on the aggressive mode in SCIP to focus on improving the primal bound instead of proving optimality. We also compare with non-ML primal heuristics discussed in Section 2.1, including RENS [3], Undercover [4], and Relax-Search [17].

Computational Setup We set the time limit to 300s. Inference results are conducted on 100 test instances. For the ML methods, we set $p = 0.7$ and use SCIP (v8.0.1) [6] to solve the sub-MBQPs. We also perform a sensitivity analysis of $p \in \{0.65, 0.75\}$ (Appendix A). For the combined loss proposed in 3.2, we experiment with $\lambda_{CL} \in \{1, 2, 5, 7\}$. More details on the computational setup are deferred to Appendix C.

4.1 Results and Discussion

Table 1: **Primal Gap (PG) and Primal Integral (PI) results.** WCE and CL are ML-guided MBQP primal heuristics adapted from MILPs. BCE+CL, $\lambda_{CL} \in \{1, 2\}$ are the extended ML methods with the proposed combined loss. SCIP, RENS, Undercover and Relax-Search are baselines. † indicates benchmarks where there are instances for which the method did not produce a feasible solution. For CL, the number of feasible instances (out of 100) are 2, 0, and 16 for QMKP, CQKP, and WFLOP. For RENS, the number of feasible instances for CBQP and CQKP are 65 and 89, respectively. For all other methods and benchmarks, the number of feasible instances are 100. We did not include the results with $\lambda_{CL} \in \{5, 7\}$ for BCE+CL, as we observe infeasible instances with higher λ_{CL}.

		CBQP		**QMKP**		**CQKP**		**WFLOP**	
	Method	**PG**	**PI**	**PG**	**PI**	**PG**	**PI**	**PG**	**PI**
Adapted	WCE	**0.04**	**50.2**	0.17	78.22	0.09	68.38	0.05	66.28
	CL	0.28	109.62	$0.98^\dagger$	$294.77^\dagger$	$1^\dagger$	$300^\dagger$	$0.78^\dagger$	$274.94^\dagger$
Extended	BCE+CL ($\lambda_{CL} = 1$)	**0.04**	52.53	0.15	67.16	0.11	74.84	**0.04**	65.6
	BCE+CL ($\lambda_{CL} = 2$)	**0.04**	52.25	**0.11**	**55.17**	**0.06**	**62.13**	0.05	**64.98**
Baselines	SCIP	0.89	278.22	0.85	265.9	0.99	298.27	0.12	153.69
	RENS	$0.76^\dagger$	$276.02^\dagger$	0.85	282.59	$0.93^\dagger$	$292.86^\dagger$	0.23	116.25
	Undercover	1	300	0.99	299.79	0.99	298.36	0.49	262.56
	Relax-Search	0.57	183.6	0.53	182.76	0.49	163.56	0.35	150.09

As shown in Table. 1, all ML-guided MBQP primal heuristics other than the pure CL-based method outperform the best-performing non-ML baseline. The extended methods with combined BCE and CL losses perform best in terms of PG for all benchmarks. The best choice of the λ_{CL} hyperparameter differs for each benchmark. In terms of PI, BCE+CL ($\lambda_{CL} = 2$) performs the best in three of the four benchmarks (QMKP, CQKP, and WFLOP). Compared to the adapted pure CL approach, our extension that combines CL and BCE significantly improves both feasibility and solution quality.

5 Conclusion

We present ML-guided primal heuristics for MBQPs based on solution prediction. We adapt existing methods on ML-guided MILP primal heuristics to MBQPs by introducing a new neural network architecture for feature extraction and a new data collection procedure for collecting high-quality training data for MBQPs. Moreover, we extend existing loss functions used in CO solution prediction and propose to combine Binary Cross-Entropy loss and Contrastive Loss. Experimental results show that the adapted and extended ML-guided methods significantly outperform non-ML primal heuristics in primal gap and primal integral. More importantly, our extended loss function significantly improves the feasibility and solution quality compared to the pure Contrastive Learning method.

Acknowledgments and Disclosure of Funding

This work was done during Weimin Huang's internship at the Pacific Northwest National Laboratory (PNNL) and at the University of Southern California. PNNL is a multi-program national laboratory operated by Battelle Memorial Institute for the U.S. Department of Energy (DOE) under Contract No. DE-AC05-76RL0-1830. The research is partially supported by the National Science Foundation (NSF) grant 2112533: "NSF Artificial Intelligence (AI) Research Institute for Advances in Optimization (AI4OPT)", the U.S. Department of Energy, Office of Science Energy Earthshot Initiative, as part of the Addressing Challenges in Energy: Floating Wind in a Changing Climate Energy Earthshot Research Center at PNNL, and the Ralph S. O'Connor Sustainable Energy Institute (ROSEI) at Johns Hopkins University.

References

[1] Bagga, P. S. and Delarue, A. (2023). Solving the quadratic assignment problem using deep reinforcement learning. *arXiv preprint arXiv:2310.01604*.

[2] Berthold, T. (2013). Measuring the impact of primal heuristics. *Operations Research Letters*, 41(6):611–614.

[3] Berthold, T. (2014). Rens: the optimal rounding. *Mathematical Programming Computation*, 6:33–54.

[4] Berthold, T. and Gleixner, A. M. (2014). Undercover: a primal minlp heuristic exploring a largest sub-mip. *Mathematical Programming*, 144:315–346.

[5] Bertsimas, D. and Shioda, R. (2009). Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 43(1):1–22.

[6] Bestuzheva, K., Besançon, M., Chen, W.-K., and et al (2021). The SCIP Optimization Suite 8.0. Technical report, Optimization Online.

[7] Bodini, N., Optis, M., Redfern, S., Rosencrans, D., Rybchuk, A., Lundquist, J. K., Pronk, V., Castagneri, S., Purkayastha, A., Draxl, C., et al. (2023). The 2023 national offshore wind data set (now-23). *Earth System Science Data Discussions*, 2023:1–57.

[8] Bonami, P., Lodi, A., and Zarpellon, G. (2018). Learning a classification of mixed-integer quadratic programming problems. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 15th International Conference, CPAIOR 2018, Delft, The Netherlands, June 26–29, 2018, Proceedings 15*, pages 595–604. Springer.

[9] Brody, S., Alon, U., and Yahav, E. (2021). How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.

[10] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

[11] Ferber, A. M., Huang, T., Zha, D., Schubert, M., Steiner, B., Dilkina, B., and Tian, Y. (2023). Surco: Learning linear surrogates for combinatorial nonlinear optimization problems. In *International Conference on Machine Learning*, pages 10034–10052. PMLR.

[12] Forrester, R. J. and Hunt-Isaak, N. (2020). Computational comparison of exact solution methods for 0-1 quadratic programs: Recommendations for practitioners. *Journal of Applied Mathematics*, 2020(1):5974820.

[13] Ghaddar, B., Gómez-Casares, I., González-Díaz, J., González-Rodríguez, B., Pateiro-López, B., and Rodríguez-Ballesteros, S. (2023). Learning for spatial branching: An algorithm selection approach. *INFORMS Journal on Computing*, 35(5):1024–1043.

[14] Han, Q., Yang, L., Chen, Q., Zhou, X., Zhang, D., Wang, A., Sun, R., and Luo, X. (2023). A GNN-guided predict-and-search framework for mixed-integer linear programming. In *The Eleventh International Conference on Learning Representations*.

[15] Huang, T., Ferber, A. M., Zharmagambetov, A., Tian, Y., and Dilkina, B. (2024a). Contrastive predict-and-search for mixed integer linear programs. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F., editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 19757–19771. PMLR.

[16] Huang, W., Huang, T., Ferber, A. M., and Dilkina, B. (2024b). Distributional miplib: a multi-domain library for advancing ml-guided milp methods. *arXiv preprint arXiv:2406.06954*.

[17] Huang, W., Isenberg, N. M., Drgoňa, J., Vrabie, D. L., and Dilkina, B. (2025). Efficient primal heuristics for mixed binary quadratic programs using suboptimal rounding guidance. *Proceedings of the International Symposium on Combinatorial Search*, 18(1):74–82.

[18] Kochenberger, G. A., Glover, F., Alidaee, B., and Rego, C. (2005). An unconstrained quadratic binary programming approach to the vertex coloring problem. *Annals of Operations Research*, 139:229–241.

[19] Létocart, L., Plateau, M.-C., and Plateau, G. (2014). An efficient hybrid heuristic method for the 0-1 exact k-item quadratic knapsack problem. *Pesquisa Operacional*, 34.

[20] Loiola, E. M., De Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., and Querido, T. (2007). A survey for the quadratic assignment problem. *European journal of operational research*, 176(2):657–690.

[21] Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

[22] Magnanti, T. L. (1981). Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. *Networks*, 11(2):179–213.

[23] Misener, R. and Floudas, C. A. (2013). Glomiqo: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, 57(1):3–50.

[24] Nair, V., Bartunov, S., Gimeno, F., Von Glehn, I., Lichocki, P., Lobov, I., O'Donoghue, B., Sonnerat, N., Tjandraatmadja, C., Wang, P., et al. (2020). Solving mixed integer programs using neural networks. *arXiv preprint arXiv:2012.13349*.

[25] Parpas, P. and Rustem, B. (2006). Global optimization of the scenario generation and portfolio selection problems. In *International Conference on Computational Science and Its Applications*, pages 908–917. Springer.

[26] Pia, A. D., Dey, S. S., and Molinaro, M. (2017). Mixed-integer quadratic programming is in np. *Mathematical Programming*, 162:225–240.

[27] Rebennack, S. (2024). Stable set problem: Branch & cut algorithms. In *Encyclopedia of optimization*, pages 1–14. Springer.

[28] Scavuzzo, L., Aardal, K., Lodi, A., and Yorke-Smith, N. (2024). Machine learning augmented branch and bound for mixed integer linear programming. *Mathematical Programming*, pages 1–44.

[29] Turner, S., Romero, D., Zhang, P., Amon, C., and Chan, T. (2014). A new mathematical programming approach to optimize wind farm layouts. *Renewable Energy*, 63:674–680.

[30] Zheng, X., Sun, X., Li, D., and Sun, J. (2012). Successive convex approximations to cardinality-constrained quadratic programs: a dc approach. Technical report, Tech. rep., School of Management, Fudan University.

A Sensitivity Analysis

We perform a sensitivity analysis of $p \in \{0.65, 0.75\}$, as shown in Table 2 and Table 3.

Table 2: **Primal Gap (PG) and Primal Integral (PI) results with $p = 0.65$.** For CL, the number of feasible instances (out of 100) are 2, 0, and 19 for QMKP, CQKP, and WFLOP. For RENS, the number of feasible instances for CBQP and CQKP are 65 and 89, respectively. For all other methods and benchmarks, the number of feasible instances are 100. We did not include the results with $\lambda_{CL} \in \{5, 7\}$ for BCE+CL, as we observe infeasible instances with higher λ_{CL}.

		CBQP		QMKP		CQKP		WFLOP	
	Method	**PG**	**PI**	**PG**	**PI**	**PG**	**PI**	**PG**	**PI**
Adapted	WCE	0.05	**67.94**	0.14	82.26	0.11	91.54	**0.06**	97.89
	CL	0.22	96.96	$0.98^{\dagger}$	$294.85^{\dagger}$	$1^{\dagger}$	$300^{\dagger}$	$0.77^{\dagger}$	$270.21^{\dagger}$
Extended	BCE+CL ($\lambda_{CL} = 1$)	**0.04**	69.01	0.12	70.91	0.1	86.1	**0.06**	**94.64**
	BCE+CL ($\lambda_{CL} = 2$)	**0.04**	69.75	**0.08**	**60.35**	**0.09**	**77.66**	**0.06**	96.85
Baselines	SCIP	0.89	278.22	0.85	265.9	0.99	298.27	0.12	153.69
	RENS	$0.76^{\dagger}$	$276.02^{\dagger}$	0.85	282.59	$0.93^{\dagger}$	$292.86^{\dagger}$	0.23	116.25
	Undercover	1	300	0.99	299.79	0.99	298.36	0.49	262.56
	Relax-Search	0.57	183.6	0.53	182.76	0.49	163.56	0.35	150.09

Table 3: **Primal Gap (PG) and Primal Integral (PI) results with $p = 0.75$.** For CL, the number of feasible instances (out of 100) are 2, 1, and 14 for QMKP, CQKP, and WFLOP. For RENS, the number of feasible instances for CBQP and CQKP are 65 and 89, respectively. For BCE+CL ($\lambda_{CL} = 2$), the number of feasible instances in QMKP is 92. For all other methods and benchmarks, the number of feasible instances are 100. We did not include the results with $\lambda_{CL} \in \{5, 7\}$ for BCE+CL, as we observe infeasible instances for higher λ_{CL}.

		CBQP		QMKP		CQKP		WFLOP	
	Method	**PG**	**PI**	**PG**	**PI**	**PG**	**PI**	**PG**	**PI**
Adapted	WCE	**0.03**	**39.86**	0.16	75.17	0.09	61.78	**0.06**	84.4
	CL	0.34	115.12	$0.98^{\dagger}$	$294.77^{\dagger}$	$1^{\dagger}$	$300^{\dagger}$	$0.79^{\dagger}$	$277.54^{\dagger}$
Extended	BCE+CL ($\lambda_{CL} = 1$)	0.04	45.55	**0.13**	**67.32**	0.1	71.39	**0.06**	**83.23**
	BCE+CL ($\lambda_{CL} = 2$)	0.04	51.25	$0.19^{\dagger}$	$77.57^{\dagger}$	**0.06**	**54.39**	0.07	83.73
Baselines	SCIP	0.89	278.22	0.85	265.9	0.99	298.27	0.12	153.69
	RENS	$0.76^{\dagger}$	$276.02^{\dagger}$	0.85	282.59	$0.93^{\dagger}$	$292.86^{\dagger}$	0.23	116.25
	Undercover	1	300	0.99	299.79	0.99	298.36	0.49	262.56
	Relax-Search	0.57	183.6	0.53	182.76	0.49	163.56	0.35	150.09

B Neural Network Aarchitecture

List of features The full list of features for the tripartite graph is shown in Table 6.

GAT module details For the embedding layers, we use 2-layer MLPs with 64 hidden units per layer and ReLU as the activation function to map the node and edge features (C, E, V, E', Q) to new embeddings $(C_1, E_1, V_1, E'_1, Q_1)$ in $\mathbb{R}^d$ where $d = 64$. The GAT performs four rounds of message passing, as shown in Fig. 1 (C). In round one, each quadratic term node in Q_1 attends over its neighbors in V_1 using H attention heads to produce updated quadratic term embeddings Q_2. In round two, each variable node in V_1 attends over its neighbors (using a separate set of H heads) to produce updated variable embeddings V_2. In round three, each constraint node in C_1 attends over its neighbors in V_2 to produce updated constraint embeddings C_2. In the final round, each variable node in V_2 attends over its neighbors in C_2 to produce the final variable embeddings V_3. We use $H = 8$ attention heads.

C Training and Inference Setup

For each MBQP benchmark, 800 instances are used for training and 100 are used for validation. For data collection, we set a relaxation time limit of $T_r = 1000s$, a subproblem time limit of $T_s = 1000s$, number of random seeds $K = 10$, candidate fixing ratio of $p_1 = 0.9$, and final fixing ratio of $p_2 = 0.7$.

Trainings are done on an NVIDIA A100 GPU with 40 GB of memory. For training, we use the AdamW optimizer [21] with learning rate 10^{-5}. We use a batch size of 16 and train for 2000 epochs.

Testing (ML inference and non-ML primal heuristics) is conducted on 2.90 GHz AMD epyc-7542 CPUs with 10 GB RAM.

D Sample Weights in Contrastive Learning

We denote the objective value of instance M^i given x as the solution as obj$(x|M^i)$, so that this discussion applies to both MILP and MBQPs. For MILPs, obj$(x|M^i) = c_i^T x$. For MBQPs, obj$(x|M^i) = x^T H^i x + c^{iT} x$.

We assume minimization problems and consider two positive samples x_+^1 and x_+^2 for the same instance M^i, with obj$(x_+^1|M^i) <$ obj$(x_+^2|M^i)$. Since this is a minimization problem, the solution quality of x_+^1 is higher than x_+^2. Let $p_\theta(M^i)$ be the prediction from the ML model. According to Eqn. 5, the value of the loss $\mathcal{L}^+ \left(\theta \mid x_+, M^i\right)$ should be low when the values of $x_+^1 \cdot p_\theta(M^i)$ and $x_+^2 \cdot p_\theta(M^i)$ are high, so that the predictions become similar to the positive samples when the training loss decreases. Moreover, the sample weight function $\frac{1}{\tau(x|M^i)}$ should be set so that $\mathcal{L}^+ \left(\theta \mid x_+^1, M^i\right) > \mathcal{L}^+ \left(\theta \mid x_+^2, M^i\right)$. In other words, positive samples with higher objective values are assigned higher weights during training. In the case when $x_+^1 \cdot p_\theta(M^i) > x_+^2 \cdot p_\theta(M^i) > 0$, it should hold that

$$x_+^1 \cdot p_\theta(M^i) \frac{1}{\tau(x_+^1|M^i)} > x_+^2 \cdot p_\theta(M^i) \frac{1}{\tau(x|M^i)} \geq 0 \tag{6}$$

so that the function $\frac{1}{\tau(x|M^i)}$ does not change the signs of dot product, and that the weighted dot product for x_+^1 is higher than x_+^2.

Now we compare the effects of different choices of the $\frac{1}{\tau(x|M^i)}$ function. We consider two scenarios: (1) minimization problems with negative objective values (i.e., obj$(x_+^1|M^i) <$ obj$(x_+^2|M^i) \leq 0$) and (2) minimization problems with positive objective values (i.e., $0 \leq$ obj$(x_+^1|M^i) <$ obj$(x_+^2|M^i)$).

Sample weights in [15]. $\frac{1}{\tau(x|M^i)} = \text{obj}(x|M^i)/w$ with $w < 0$.

Scenario	Sample weight	Weighted dot product
obj$(x_+^1\|M^i) <$ obj$(x_+^2\|M^i) \leq 0$	$\frac{1}{\tau(x_+^1\|M^i)} > \frac{1}{\tau(x_+^2\|M^i)} \geq 0$	$x_+^1 \cdot p_\theta(M^i) \frac{1}{\tau(x_+^1\|x_+^1, M^i)} > x_+^2 \cdot p_\theta(M^i) \frac{1}{\tau(x\|M^i)} \geq 0$
$0 \leq$ obj$(x_+^1\|M^i) <$ obj$(x_+^2\|M^i)$	$\frac{1}{\tau(x_+^2\|M^i)} < \frac{1}{\tau(x_+^1\|M^i)} \leq 0$	$x_+^1 \cdot p_\theta(M^i) \frac{1}{\tau(x\|M^i)} \gtrless x_+^2 \cdot p_\theta(M^i) \frac{1}{\tau(x\|M^i)} \leq 0$

Table 4: Relationship between sample weight and weighted dot product in [15].

Proposed sample weights. $\frac{1}{\tau(x|M^i)} = \exp(\frac{\text{obj}(x|M^i)}{w})$ with $w < 0$.

Scenario	Sample weight	Weighted dot product
obj$(x_+^1\|M^i) <$ obj$(x_+^2\|M^i) \leq 0$	$\frac{1}{\tau(x_+^1\|M^i)} > \frac{1}{\tau(x_+^2\|M^i)} \geq 0$	$x_+^1 \cdot p_\theta(M^i) \frac{1}{\tau(x_+^1\|x_+^1, M^i)} > x_+^2 \cdot p_\theta(M^i) \frac{1}{\tau(x\|M^i)} \geq 0$
$0 \leq$ obj$(x_+^1\|M^i) <$ obj$(x_+^2\|M^i)$	$\frac{1}{\tau(x_+^1\|M^i)} > \frac{1}{\tau(x_+^2\|M^i)} \geq 0$	$x_+^1 \cdot p_\theta(M^i) \frac{1}{\tau(x_+^1\|x_+^1, M^i)} > x_+^2 \cdot p_\theta(M^i) \frac{1}{\tau(x\|M^i)} \geq 0$

Table 5: Relationship between sample weight and weighted dot product with proposed sample weight function.

As shown in Table. 4, Eqn. 6 fails to hold in scenario (2) with the function $\frac{1}{\tau(x|M^i)} = \text{obj}(x|M^i)/w$, as the relationship between sample weights are filpped for minimization problems with positive objective values. Our proposed $\tau(x|M^i)$ function addresses this issue by converting the weights to positive values, regardless of the signs of obj$(x|M^i)$.

Nodes	Features	Source
C	avg. coefficients in the constraint	[14]
	min. coefficients in the constraint	new
	max. coefficients in the constraint	new
	variance of coefficients in the constraint	new
	# of variables in the constraint	[14]
	left-hand side or right-hand side	[14]
	constraint sense in one-hot encoding (3) $(=, >, <)$	new
V-C edge	coefficient of variables in constraints	[14]
V	normalized coefficient in obj (among linear terms)	[14]
	avg. coefficient in constraints	[14]
	# of times it appear in linear constraints	[14]
	variance of. coefficient in constraints	new
	max. coefficient in constraints	[14]
	min. coefficient in constraints	[14]
	binary variable indicator	[14]
	LP relaxation value in MILP reformulation	new
	# times it appears in quadratic terms	new
	avg. coefficient in quadratic terms that it appears in	new
	max. coefficient in quadratic terms that it appears in	new
	min. coefficient in quadratic terms that it appears in	new
	variance of coefficient in quadratic terms that it appears in	new
	avg. # times its neighbors appears in quadratic terms	new
	max. # times its neighbors appears in quadratic terms	new
	min. # times its neighbors appears in quadratic terms	new
	variance of # times its neighbors appears in quadratic terms	new
	Eigenvalue centrality in Hessian graph	new
Q	coefficient of quadratic term in objective function	new
	LP relaxation value of reformulated variable $z_{ij} = x_i x_j$	new
	LP relaxation violation	new
	Edge centrality in Hessian graph	new
V-Q edge	None	new

Table 6: Features of MBQP tripartite graph representation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: This paper introduces a ML-guided primal heuristics for MBQPs. The abstract and introduction (Section 1) reflects this.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

Justification: We do not have enough space for this discussion. However, one of the limitations is that feasibility is not guaranteed with ML-methods.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We do not have theoretical results in this paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Details on training the ML model architecture are provided in Section 3 and Appendix B. Details on data collection, ML model training, and inference are provided in Section 4 and Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: As for now, we cannot provide open access to the data and code before a software disclosure request is approved. But the data and code will be publicly available once the disclosure request is approved.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Details on training data collection, ML model training, and inference are provided in Section 4 and Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We did not provide error bars, but we performed a sensitivity analysis in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Details on training resources are provided in Section 4 (CPUs) and Appendix C (GPUs).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed and followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The experiments (Section 4) in this paper include a real-world wind farm layout optimization problem. Methods in this work has the potential to increase the efficiency of wind energy production. Authors are not aware of negative societal impact of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: As far as the authors are aware, the paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We generate the WFLOP instances based on the MBQP formulation in [17], using wind data from the NOW-23 offshore wind dataset at selected locations in the California offshore region [7]. We credited these sources.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve research with human subjects.**[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were only used for formmating purposes in this paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.