Boosting Cross-problem Generalization in Diffusion-Based Neural Combinatorial Solver via Inference Time Adaptation

Haoyu Lei[1*], **Kaiwen Zhou**[2], **Yinchuan Li**[2], **Zhitang Chen**[2], **Farzan Farnia**[1]

[1] Computer Science & Engineering Department, The Chinese University of Hong Kong
[2] Noah's Ark Lab, Huawei Technologies Co., Ltd.
hylei22@cse.cuhk.edu.hk, farnia@cse.cuhk.edu.hk

Abstract

Diffusion-based Neural Combinatorial Optimization (NCO) has demonstrated effectiveness in solving NP-complete (NPC) problems by learning discrete diffusion models for solution generation, eliminating hand-crafted domain knowledge. Despite their success, existing NCO methods face significant challenges in both cross-scale and cross-problem generalization, and high training costs compared to traditional solvers. While recent studies on diffusion models have introduced training-free guidance approaches that leverage pre-defined guidance functions for conditional generation, such methodologies have not been extensively explored in combinatorial optimization. To bridge this gap, we propose a training-free inference time adaptation framework (**DIFU-Ada**) that enables both the zero-shot cross-problem transfer and cross-scale generalization capabilities of diffusion-based NCO solvers without requiring additional training. We provide theoretical analysis that helps understanding the cross-problem transfer capability. Our experimental results demonstrate that a diffusion solver, trained exclusively on the Traveling Salesman Problem (TSP), can achieve competitive zero-shot transfer performance across different problem scales on TSP variants, such as Prize Collecting TSP (PCTSP) and the Orienteering Problem (OP), through inference time adaptation.

Introduction

Combinatorial optimization (CO) problems are fundamental challenges across numerous domains, from logistics and supply chain management to network design and resource allocation. While traditional exact solvers and heuristic methods have been widely studied, they often struggle with scalability and require significant domain expertise to design problem-specific algorithms (Arora 1998; Gonzalez 2007).

Recent advances in deep learning have sparked interest in Neural Combinatorial Optimization (NCO), which aims to learn reusable solving strategies directly from data, eliminating the need for hand-crafted heuristics (Bengio, Lodi, and Prouvost 2021). Among various deep learning approaches, diffusion-based models (Ho, Jain, and Abbeel 2020; Song et al. 2020) have emerged as a particularly promising direction for solving combinatorial optimization problems. These models have demonstrated remarkable capabilities in learning complex solution distributions by adapting discrete diffusion processes to graph structures (Sun and Yang 2023). Recent works like (Li et al. 2024a,b) have achieved state-of-the-art performance on classical problems such as the Traveling Salesman Problem (TSP), showcasing the potential of diffusion-based generative approaches in combinatorial optimization.

However, the practical applicability of existing NCO approaches is limited by several generalization challenges. First, current models suffer from cross-scale generalization, with performance degrading significantly when applied to larger problem instances than those seen during training, especially for auto-regression-based solvers (Khalil et al. 2017; Kool, Van Hoof, and Welling 2018) including transformer and reinforcement learning methods. Second, these models show limited cross-problem transfer capabilities, struggling to adapt to problem variants with modified objectives or additional constraints. While several studies have attempted to enhance learning-based solvers' generalization through approaches such as training additional networks (Wang et al. 2024) and fine-tuning (Lin et al. 2024), these methods require substantial computational costs and training data for training separate models for each problem type and scale.

In parallel, recent advances in diffusion models, particularly in computer vision, have demonstrated the effectiveness of training-free guidance approaches for enhancing conditional generation (Bansal et al. 2023; Chung et al. 2022; Yu et al. 2023; Shen et al. 2024; Ye et al. 2024). These approaches leverage plug-and-play guidance functions or pre-trained networks to enable conditional generation without additional training overhead. These approaches demonstrate significant potential for modifying the inference phase to achieve controllable sampling from pre-trained diffusion models. Inspired by these developments, we explore the training-free inference time adaptation to address the cross-problem transfer challenges in neural combinatorial optimization in Figure 1.

This work introduces an inference time adaptation framework (illustrated in Figure 1) designed to enhance the generalization capabilities of diffusion-based Neural Combinato-

[*]Work completed during internship at Huawei Noah's Ark Lab.

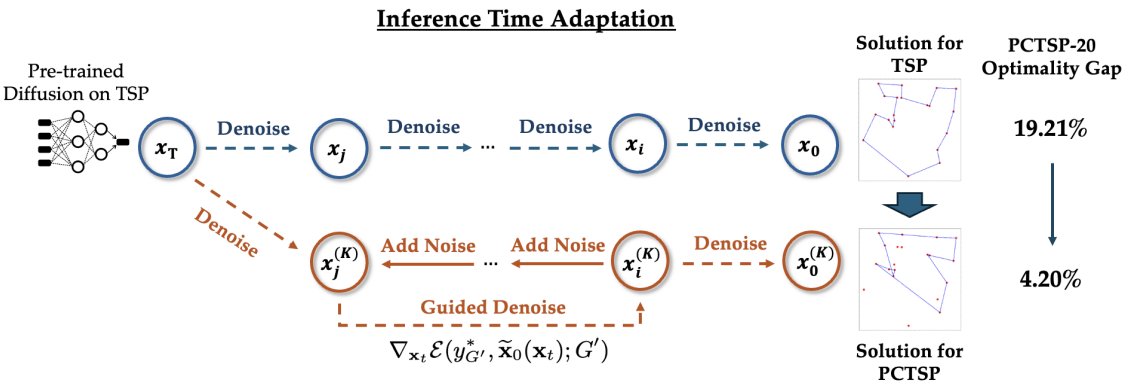

Figure 1: The proposed **Inference Time Adaptation** framework. This approach combines (1) energy-guided sampling, which incorporates problem-specific objectives and constraints, with (2) a recursive renoising-denoising travel for solution refinement, enabling zero-shot cross-problem transfer without training. The Optimality Gap (↓) on PCTSP-20 is reduced from 19.21% to 4.20%.

rial Optimization (NCO) solvers without additional training costs. We scale up the inference time by combining two key components: (1) energy-guided sampling and (2) a recursive renoising-denoising travel modelled as a Guided Langevin Dynamics. Our DIFU-Ada framework enables zero-shot cross-problem transfer while maintaining solution feasibility. Through experimental evaluation of the TSP-trained diffusion model on more complex variants, the Prize Collecting TSP (PCTSP) and the Orienteering Problem (OP), we empirically demonstrate its effective zero-shot transferability across different problems of increasing complexity, while maintaining consistent performance across different problem scales. Our work represents a significant step toward more flexible and generalizable diffusion-based CO solvers, reducing the need for problem-specific model training while maintaining competitive performance.

Related Works

Neural Network-based Combinatorial Solvers. Neural Combinatorial Optimization (NCO) leverages neural networks to learn solution distributions for complex optimization problems (Bengio, Lodi, and Prouvost 2021; Zhang et al. 2023). Prominent approaches include autoregressive solvers (Khalil et al. 2017; Kool, Van Hoof, and Welling 2018; Kwon et al. 2020; Kim, Park, and Park 2022; Hottung, Bhandari, and Tierney 2021) and non-autoregressive methods (Joshi, Laurent, and Bresson 2019; Fu, Qiu, and Zha 2021; Qiu, Sun, and Yang 2022; Wang et al. 2024; Sun and Yang 2023; Sanokowski, Hochreiter, and Lehner 2024).
Diffusion-based Generative Modeling. Score-based diffusion models (Ho, Jain, and Abbeel 2020; Song et al. 2020; Sohl-Dickstein et al. 2015; Song and Ermon 2019; Dhariwal and Nichol 2021; Song, Meng, and Ermon 2020) have become a leading generative framework. Their application to CO was pioneered by (Sun and Yang 2023) for TSP and has been rapidly advanced by methods incorporating gradient search (Li et al. 2024a), single-step consistency models (Li et al. 2024b), large-scale solver (Zhao et al. 2025), and unsupervised frameworks (Sanokowski, Hochreiter, and Lehner 2024). However, few of existing literature focus on cross-problem generalization.
Training-free Guidance for Diffusion Models. To control generation, training-free guidance methods (Bansal et al. 2023; Chung et al. 2022; Yu et al. 2023; Shen et al. 2024) offer an efficient alternative to training-intensive classifier (Dhariwal and Nichol 2021) or classifier-free (Ho and Salimans 2022) approaches. This strategy was first adapted for discrete diffusion solvers by (Li et al. 2024a), building on the framework from (Sun and Yang 2023). Our work further demonstrates that modifying the inference phase is a powerful method for enhancing the zero-shot cross-problem transfer of diffusion-based NCO solvers.

Preliminaries

Graph-based CO Problems

Combinatorial optimization (CO) problems on graphs are fundamental to numerous real-world applications. Following recent advances (Sun and Yang 2023; Li et al. 2024a), we address these problems by formalizing graph-based CO instances. We represent each problem instance as an undirected graph $G(V, E) \in \mathcal{G}$, where V and E denote the vertex and edge sets, respectively. This representation encompasses both vertex selection and edge selection problems, covering a broad spectrum of practical CO scenarios. For any instance $G \in \mathcal{G}$, we define a binary decision variable $\mathbf{x} \in \mathcal{X}_G$, where $\mathcal{X}_G = \{0,1\}^N$ represents the feasible solution space. The optimization objective is to find the optimal solution $\mathbf{x}^*$ that minimizes a problem-specific objective function $\phi(\cdot; G) : \{0,1\}^N \to \mathbb{R}$:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}_G}{\operatorname{argmin}}\ \phi(\mathbf{x}; G), \tag{1}$$

where the objective function decomposes into a log-barrier formulation (Den Hertog, Roos, and Terlaky 1992):

$$\phi(\mathbf{x}; G) = f_{\text{cost}}(\mathbf{x}; G) - \mu \sum_{i=1}^{m} \log\big(-g_i(\mathbf{x}; G)\big), \tag{2}$$

Here $f_{\text{cost}}(\cdot; G)$ measures the solution quality, and $g_i(\cdot; G)$ enforces problem-specific constraints through a penalty coefficient $\mu > 0$. The validity functions $g_i(\cdot; G)$ returns 1 for feasible solutions and is strictly negative for infeasible ones.

Energy-based Probabilistic Modeling

To leverage recent advances in deep generative models, we reformulate the CO objective through an energy-based perspective (Lucas 2014). Specifically, we establish an energy function:

$$\mathcal{E}(\cdot; G) := \rho(y, \phi(\cdot; G)), \tag{3}$$

where ρ represents any loss functions (cross entropy loss, L2-norm), and the energy function maps each solution to its corresponding energy state. This energy-based formulation naturally leads to a probabilistic framework through the Boltzmann distribution (LeCun et al. 2006):

$$p(y|\mathbf{x}; G) = \frac{\exp\big(-\frac{1}{\tau}\mathcal{E}(y, \mathbf{x}; G)\big)}{\mathcal{Z}}, \tag{4}$$

$$\text{where } \mathcal{Z} = \sum_{\mathbf{x}} \exp\Big(-\frac{1}{\tau}\mathcal{E}(y, \mathbf{x}; G)\Big), \tag{5}$$

where τ controls the temperature of the system and $\mathcal{Z}$ denotes the partition function that normalizes the distribution. Recent works have demonstrated promising approaches to approximate this distribution using diffusion-based deep generative models by parameterizing a conditional distribution $p_\theta(\mathbf{x}|G)$ to minimize the energy function. Both supervised (Sun and Yang 2023; Li et al. 2024a,b) and unsupervised (Sanokowski, Hochreiter, and Lehner 2024) learning paradigms have shown significant advances. Since our proposed training-free guidance mechanism is applicable to any pre-trained diffusion-based solver, we focus on the supervised learning framework in this work.

Discrete Diffusion Generative Models

Given a training set $\mathcal{G} = \{G_i\}_{i=1}^k$ of i.i.d. problem instances with their optimal solutions $\mathbf{x}$ and the corresponding optimal objective values y_G^*, we use generative model and optimize the model parameters θ by maximizing the likelihood of the optimal solutions:

$$L(\theta) = \mathbb{E}_{G\sim\mathcal{G}}[-\log p_\theta(\mathbf{x}|y_G^*, G)]. \tag{6}$$

We adopt a discrete diffusion generative model (Austin et al. 2021) to effectively sample optimal solutions from the learned distribution $p_\theta(\mathbf{x}|y_G^*, G)$ (Sun and Yang 2023; Li et al. 2024a,b).

The diffusion process consists of two key components: a forward process that gradually corrupts the data, and a reverse process that learns to reconstruct the original distribution. The forward process $q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})$ maps clean data $\mathbf{x}_0 \sim q(\mathbf{x}_0|G)$ to a sequence of increasingly corrupted latent variables $\mathbf{x}_{1:T}$. The reverse process $p_\theta(\mathbf{x}_{0:T}|G) = p(\mathbf{x}_T)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, G)$ learns to gradually denoise these latent variables to recover the original distribution. For discrete state spaces, we define the forward process using a categorical distribution:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \text{Cat}(\mathbf{x}_t; \mathbf{p} = \widetilde{\mathbf{x}}_{t-1}\mathbf{Q}_t), \tag{7}$$

where $\widetilde{\mathbf{x}}_t \in \{0,1\}^{N\times 2}$ represents the one-hot encoding of $\mathbf{x}_t \in \{0,1\}^N$. The forward transition matrix $\mathbf{Q}_t$ is defined as:

$$\mathbf{Q}_t = \begin{bmatrix}(1-\beta_t) & \beta_t \\ \beta_t & (1-\beta_t)\end{bmatrix}, \quad \beta_t \in [0,1], \tag{8}$$

where $[\mathbf{Q}_t]_{ij}$ denotes the state transition probability from state i to state j. The t-step marginal distribution and posterior can be derived as:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \text{Cat}(\mathbf{x}_t; \mathbf{p} = \widetilde{\mathbf{x}}_0\overline{\mathbf{Q}}_t), \tag{9}$$

where $\overline{\mathbf{Q}}_t = \mathbf{Q}_1\mathbf{Q}_2\ldots\mathbf{Q}_t$ and $\odot$ denotes element-wise multiplication.

To capture the structural properties of CO problems, we employ an anisotropic graph neural network architecture (Joshi, Laurent, and Bresson 2019). For a given instance G, the network learns to predict the clean data distribution $p_\theta(\widetilde{\mathbf{x}}_0|\mathbf{x}_t, G)$. Taking TSP as an example, where G encodes the 2D Euclidean coordinates of vertices, the network outputs an adjacency matrix $p_\theta(\widetilde{\mathbf{x}}_0|\mathbf{x}_t, G) \in [0,1]^{N\times 2}$. This matrix parameterizes N independent Bernoulli distributions, each corresponding to a binary decision variable in $\widetilde{\mathbf{x}}_0$. The reverse process during sampling follows:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, G) = \sum_{\widetilde{\mathbf{x}}_0} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \widetilde{\mathbf{x}}_0)p_\theta(\widetilde{\mathbf{x}}_0|\mathbf{x}_t, G). \tag{10}$$

Method: Inference Time Adaptation

Energy-guided Sampling for Problem Transfer

While training-free guidance has been extensively studied in computer vision (Bansal et al. 2023; Chung et al. 2022; Yu et al. 2023; Shen et al. 2024; Ye et al. 2024), its application to combinatorial optimization problems has only recently been explored by (Li et al. 2024a,b). Adopted by (Li et al. 2024a,b), we extend this approach by introducing energy-based training-free guidance for zero-shot cross-problem transfer during sampling, enabling flexible incorporation of additional problem-specific objective functions and constraints into pre-trained diffusion-based CO solvers.

Let $\mathcal{G}' = \{G_i'\}_{i=1}^n$ denote a set of new problem instances which differs from the pre-trained problem set $\mathcal{G}$. For a new instance G' with its optimal solution pair $(\mathbf{x}, y_{G'}^*)$, we need to estimate the new reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, y_{G'}^*, G')$ according to (10). Following the score estimation perspective of diffusion processes (Song et al. 2020; Dhariwal and Nichol 2021), the reverse sampling for new problem instances G' can be defined in a SDE formulation:

$$d\mathbf{x} = \big[-\mathbf{f}(\mathbf{x}, t) + g^2(t)\hat{s}_\theta(\mathbf{x}, t, G')\big]\,dt' + g(t)d\mathbf{w} \tag{11}$$

where $t = T - t'$. We decompose the conditional score function $\hat{s}_\theta(\mathbf{x}, t, G')$ at time step t into two components:

$$\begin{aligned}\hat{s}_\theta(\mathbf{x}, t, G') &:= \underbrace{\nabla_{\mathbf{x}_t}\log p_\theta(\mathbf{x}|y_{G'}^*, G')}_{\text{Posterior Score}} \\ &= \underbrace{\nabla_{\mathbf{x}_t}\log p_\theta(\mathbf{x}_t|G')}_{\text{Pre-trained Prior Score}} + \underbrace{\nabla_{\mathbf{x}_t}\log p_t(y_{G'}^*|\mathbf{x}_t, G')}_{\text{Energy Potential}}.\end{aligned} \tag{12}$$

From the Bayesian perspective, $p_\theta(\mathbf{x}_t|G')$ can be understood as the *prior*, which contains knowledge of the pre-trained problems (i.e. TSP problem in our experimental settings), and $p_t(y_{G'}^*|\mathbf{x}_t, G')$ corresponds to the *Energy Potential* that incorporates additional constraints or objectives of the variant problem. We sample from the *Posterior Score*

$\hat{s}_\theta(\mathbf{x}, t, G')$ to generate high-quality solutions to the new problem G'. A theoretical analysis of how the information contained in pre-trained models can benefit the guided sampling is provided in Appendix 4.

Drawing upon this theoretical framework, we leverage the pre-trained diffusion model to estimate the first term $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_t|G')$. In the context of cross-problem transfer, while the pre-trained model yields only a *biased* score function for new problem instances, we compute the second energy-guided term $\nabla_{\mathbf{x}_t} \log p_t(y^*_{G'}|\mathbf{x}_t, G')$ to adjust using an energy function that specifically accounts for the additional objectives and constraints of the new problems:

$$\nabla_{\mathbf{x}_t} \log p_t(y^*_{G'}|\mathbf{x}_t, G') \propto -\nabla_{\mathbf{x}_t} \mathcal{E}(y^*_{G'}, \widetilde{\mathbf{x}}_0(\mathbf{x}_t); G'), \tag{13}$$

where $\rho(y^*_{G'}, \phi(\widetilde{\mathbf{x}}_0(\mathbf{x}_t); G'))$ measures the distance of energy states between the optimal value and the predicted solution. Here, $\widetilde{\mathbf{x}}_0(\mathbf{x}_t)$ represents the predicted clean sample from the current noisy state $\mathbf{x}_t$, which is estimated by the logits of N independent Bernoulli samples $\widetilde{\mathbf{x}}_0(\mathbf{x}_t) = \mathbb{E}_{\widetilde{\mathbf{x}}_0 \sim p_\theta(\widetilde{\mathbf{x}}_0|\mathbf{x}_t)}[\widetilde{\mathbf{x}}_0]$. In this work, we define $\rho(y^*_{G'}, \phi(\widetilde{\mathbf{x}}_0(\mathbf{x}_t); G')) := \|y^*_{G'} - \phi(\widetilde{\mathbf{x}}_0(\mathbf{x}_t); G')\|_2$. From Eq (13), we have:

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p_t(y^*_{G'}|\mathbf{x}_t, G') &\propto -\nabla_{\mathbf{x}_t} \rho(y^*_{G'}, \phi(\widetilde{\mathbf{x}}_0(\mathbf{x}_t); G')), \\ &\propto -\nabla_{\mathbf{x}_t} \phi(\widetilde{\mathbf{x}}_0(\mathbf{x}_t); G'), \end{aligned} \tag{14}$$

Now we obtain the *Energy Potential* by just taking the gradient of the problem-specific objective function $\phi(\widetilde{\mathbf{x}}_0; G)$. Given a pre-trained score-based diffusion model $s_\theta(\mathbf{x}, t, G) \approx \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t|G')$,the reverse process is guided by *Energy Potential* modifying the score estimate as follows:

$$\begin{aligned} d\mathbf{x} = \Big[&- \mathbf{f}(\mathbf{x}, t) + g(t)^2 \Big(\underbrace{\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|G')}_{\text{Pre-trained Score}} \\ &- \underbrace{\tau \nabla_{\mathbf{x}} \phi(\widetilde{\mathbf{x}}_0(\mathbf{x}); G')}_{\text{Energy Potential}} \Big) \Big] dt' + g(t) d\mathbf{w}, \end{aligned} \tag{15}$$

where $t = T - t'$. For discrete diffusion, each step of the reverse process involves sampling the next state, $\mathbf{x}_t$, from a multivariate Bernoulli distribution. This distribution is parameterized by the output of the guided posterior $p_\theta(\mathbf{x}|y^*_{G'}, G')$, following a standard sampling framework like DDIM (Song, Meng, and Ermon 2020).

In this work, we have the well-established log-barrier formulations for PCTSP (Definition 1) and OP (Definition 2) as the plug-and-play *Energy Potential* following (Lucas 2014):

Definition 1 (Energy Potential for Prize Collecting TSP (PCTSP))**.** *Given a complete graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}^+$, vertex prizes $r : V \to \mathbb{R}^+$, penalties $p : V \to \mathbb{R}^+$, and prize threshold R, find $\mathbf{x} \in \{0, 1\}^{|E|}$, $\mathbf{t} \in \{0, 1\}^{|V|}$ that minimizes:*

$$\begin{aligned} \phi(\mathbf{x}, \mathbf{t}; G) &= f_{cost}(\mathbf{x}, \mathbf{t}; G) - \mu \cdot \log(-g(\mathbf{x}, \mathbf{t}; G)) \\ \textit{where } f_{cost}(\mathbf{x}, \mathbf{t}; G) &= \sum_{e \in E} w_e x_e + \sum_{v \in V} p_v (1 - t_v) \\ g(\mathbf{x}, \mathbf{t}; G) &= -\max(0, R - \sum_{v \in V} r_v t_v) \end{aligned}$$

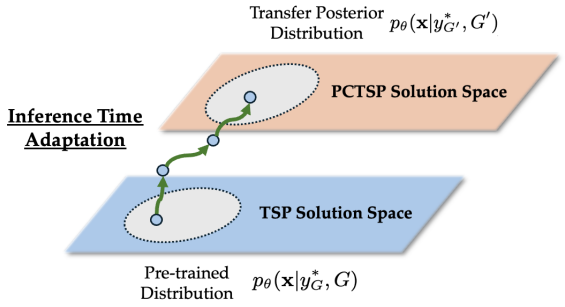

Figure 2: Overview of recursive renoising-denoising travel in **Inference Time Adaptation** for achieving zero-shot cross-problem generalization, sequentially shifting from pre-trained problem G (TSP) solution distribution to target problem distribution G' (PCTSP).

Definition 2 (Energy Potential for Orienteering Problem (OP))**.** *Given a complete graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}^+$, vertex scores $s : V \to \mathbb{R}^+$, and budget B, find $\mathbf{x} \in \{0, 1\}^{|E|}$, $\mathbf{t} \in \{0, 1\}^{|V|}$ that minimizes:*

$$\begin{aligned} \phi(\mathbf{x}, \mathbf{t}; G) &= f_{cost}(\mathbf{x}, \mathbf{t}; G) - \mu \cdot \log(-g(\mathbf{x}, \mathbf{t}; G)) \\ \textit{where } f_{cost}(\mathbf{x}, \mathbf{t}; G) &= -\sum_{v \in V} s_v t_v \\ g(\mathbf{x}, \mathbf{t}; G) &= -\max(0, \sum_{e \in E} w_e x_e - B) \end{aligned}$$

Decoding from Generated Heatmaps

Distinguished from auto-regressive solvers like transformer, diffusion-based solvers for combinatorial optimization learn to generate a time-dependent matrix of probabilities, often visualized as an $N \times N$ adjacency matrix (heatmap). This heatmap represents the likelihood of each element (e.g., an edge in a graph) being part of the final solution, rather than the discrete solution itself. Consequently, a decoding strategy is essential to translate this probabilistic output into a feasible, binary solution structure.

A common and direct approach is greedy decoding, which iteratively constructs a solution by selecting the elements with the highest probabilities from the heatmap until a valid termination condition is met (e.g., forming a complete tour in the Traveling Salesperson Problem). Alternatively, the generated heatmaps can be post-processed by more sophisticated improvement heuristics, such as 2-opt (Lin 1973; Li et al. 2024a) or Monte Carlo Tree Search (Sun and Yang 2023). To ensure a fair comparison across all methods in our experiments, we apply greedy decoding.

Recursive Renoising-denoising Travel

Our preliminary experiments revealed that applying energy-guided sampling alone is often insufficient for zero-shot, cross-problem transfer. The distributional divergence between the source problem (G) and the target problem (G') can hinder the generation of high-quality, feasible solutions. To bridge this gap, we propose 2-phase Inference-Time Adaptation framework combining energy-guided sampling and recursive renoising-denoising travel, inspired by recent work (Song et al. 2023; Li et al. 2024a).

We theoretically interpret our Inference-time Adaptation as a Guided Langevin Dynamics (Song and Ermon 2019; Welling and Teh 2011) process in Figure 2. It is designed to iteratively transport the solution particle from a target distribution G towards a new problem distribution G'. Given a solution x^k in iteration k, the recursive update to next iteration $k+1$ is a discrete-time sequence of SDE:

$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} - \Big(s_\theta(\mathbf{x}^{(k)}, G) - \tau \nabla_{\mathbf{x}^{(k)}} \phi(\mathbf{x}^{(k)}; G')\Big) + \sigma_k \mathbf{z}, \tag{16}$$

where $\mathbf{z}$ represent renoising. A naive implementation approach would be to recursively simulate the full SDE (i.e., repeated full re-noising and denoising) to progressively adapt the solution to the target distribution. However, this incurs a prohibitive computational cost.

Our key insight is that the energy guidance does not need to be fully applied at every step. Based on this, we implement the recursive process to only a few renoising steps and a single step denoising in each iteration, achieving a 5-10x inference speedup compared to the full recursive approach.

Algorithm 1 outlines the framework for Inference-Time Adaptation, combining energy-guided sampling and recursive renoising-denoising travel. It provides a practical and efficient method for zero-shot cross-problem transfer with pre-trained diffusion models at no additional training cost. Furthermore, our framework is model-free and can be augmented with other heuristics, such as 2-opt (Lin 1973) or MCTS (Fu, Qiu, and Zha 2021; Sun and Yang 2023), to further enhance final solution quality.

Algorithm 1: Inference Time Adaptation for Cross-problem

Require:
p_θ: Pre-trained diffusion solver
G: Original problem
G': Target problem
T: Number of diffusion steps
K: Number of recursive travel
i: Number of renoising steps
τ: Energy guidance temperature

Ensure: Optimal tour $\mathbf{t}^{(K)}$ for problem instance G'

Initialize $\mathbf{x}_T^{(0)} \in \{0,1\}^N$
for $k = 1$ **to** K **do**
  Renoise solution $\mathbf{x}_0^{(k-1)}$ to some noise level $\mathbf{x}_i^{(k)}$
  for $t = i$ **to** 0 **do**
    Compute $p_\theta(\mathbf{x}_t^{(k)}|G')$ from pre-trained model
    Compute energy potential: $\nabla_{\mathbf{x}_t} \phi(\widetilde{\mathbf{x}}_0(\mathbf{x}_t^{(k)}); G')$
    Compute posterior $p_\theta(\mathbf{x}_{t-1}^{(k)}|\mathbf{x}_t^{(k)}, y_{G'}^*, G')$
    Update next state with Bernoulli Sampling: $\mathbf{x}_{t-1}^{(k)} \sim \text{Cat}(\mathbf{x}_{t-1}^{(k)}; p_\theta(\mathbf{x}_{t-1}^{(k)}|\mathbf{x}_t^{(k)}, y_{G'}^*, G'))$
  end for
  Decode $\mathbf{x}_0^{(k)}$ to the optimal solution $\mathbf{t}^{(k)}$
end for
return $\mathbf{t}^{(K)}$

Numerical Results

Dataset. We evaluate our approach on the commonly-used NP-complete combinatorial optimization problems: the Traveling Salesman Problem (TSP) together with its variants, the Prize Collecting Traveling Salesman Problem (PCTSP) and Orienteering Problem (OP). The detailed problem descriptions are given in Appendix 4.

Evaluation. Following (Kool, Van Hoof, and Welling 2018), we generate 10000 test instances for each problem scale: 20, 50, and 100, which denotes the node counts for PCTSP and OP. We evaluate model performance using two primary metrics: average solution cost and optimality gap relative to the exact solution solvers. Additionally, we measure computational efficiency through total training time and per-instance inference time.

Baselines. We compare our approach against multiple baseline categories. For PCTSP, (1) Exact solver: Gurobi; (2) OR-based heuristics: OR-Tools and Iterated Local Search (ILS); (3) Learning-based methods: AM (Kool, Van Hoof, and Welling 2018), MDAM (Xin et al. 2021), AM-FT (Lin et al. 2024), ASP (Wang et al. 2024), DIFUSCO (Sun and Yang 2023), T2T (Li et al. 2024a). For OP, (1) Exact solver: Gurobi; (2) OR-based heuristics: Compass (Kobeaga, Merino, and Lozano 2018) and Tsili (Tsiligirides 1984); (3) Learning-based methods: AM (Kool, Van Hoof, and Welling 2018), AM-FT (Lin et al. 2024), DIFUSCO (Sun and Yang 2023), T2T (Li et al. 2024a).

Experimental Setup. The proposed Inference Time Adaptation for Diffusion-based solvers (**DIFU-Ada**) builds upon DIFUSCO's (Sun and Yang 2023) TSP-trained checkpoints in three different scales without additional training. The backbone architecture used in the diffusion model is the same as DIFUSCO, a 12-layer Anisotropic GNN with a width of 256 hidden nodes. The results of DIFU-Ada are recorded under 100 iterations, with 5-step re-noising and 1-step guided denoising. The decoding strategy used in experiments is greedy decoding. The energy functions used in guided sampling are stated in Proposition 1 and 2. The guidance temperature τ is fixed to be 0.1, and the constraint coefficient μ is set to be 1 for all problem instances. All experiments are conducted on a single GPU with 125 TFLOPS peak FP16 performance. Experimental setup details are provided in Appendix 4.

Cross-problem and Cross-scale Generalization

To validate the effectiveness of our inference time adaptation framework in zero-shot cross-problem transfer scenarios, we first conduct experiments while maintaining consistent problem scales. As demonstrated in Table 1, our method achieves significant zero-shot performance improvements when ap-

	PCTSP-20			PCTSP-50			PCTSP-100		
Method	Cost↓	Gap↓	Time↓	Cost↓	Gap↓	Time↓	Cost↓	Gap↓	Time↓
DIFUSCO	3.78	19.21%	1.04s	5.20	15.97%	1.35s	7.85	32.31%	2.02s
T2T	3.64	14.82%	2.02s	4.90	9.38%	2.18s	7.23	21.92%	2.70s
DIFU-Ada	**3.30**	**4.20%**	1.72s	**4.63**	**3.57%**	1.99s	**6.51**	**9.61%**	2.74s
	OP-20			**OP-50**			**OP-100**		
Method	Prize↑	Gap↓	Time↓	Prize↑	Gap↓	Time↓	Prize↑	Gap↓	Time↓
DIFUSCO	9.25	12.48%	1.51s	25.60	13.45%	1.88s	45.66	20.02%	2.04s
T2T	9.67	8.51%	2.70s	26.89	9.09%	3.04s	48.72	14.70%	3.45s
DIFU-Ada	**10.24**	**3.11%**	2.09s	**28.21**	**4.63%**	2.35s	**54.56**	**8.06%**	2.91s

Table 1: Zero-shot cross-problem transfer performance comparison between TSP-trained DIFUSCO, T2T and DIFU-Ada (Ours) approach on PCTSP and OP instances. The results show solution cost, optimality gap, and the average inference time for one instance across three problem scales (20, 50, and 100 nodes).

plied to the pre-trained DIFUSCO (Sun and Yang 2023) and T2T (Li et al. 2024a) model on TSP. These improvements are evident across all PCTSP and OP problem scales, in both enhanced solution quality and reduced optimality gaps compared to the exact OR solvers.

Following the evaluation strategy following (Wang et al. 2024), we conducted experiments of DIFU-Ada's transferability across diverse combinatorial optimization (CO) problem types and scales. Our evaluation framework specifically assesses two aspects of generalization: 1) **Cross-problem Generalization**: We assess the model's adaptability to different CO problem variants, such as transferring from TSP to PCTSP or OP. 2) **Cross-scale Generalization**: We evaluate the performance of a single, optimally trained model across various problem scales (e.g., from small to large instances of the same problem type). Unlike existing learning-based methods that necessitate problem-specific retraining for new problem types or scales (e.g., for PCTSP and OP instances), our DIFU-Ada framework achieves zero-shot generation at inference time. This rigorous out-of-distribution evaluation strategy presents a more challenging benchmark than conventional in-distribution assessments.

The experimental results presented in Table 2 (PCTSP) and Table 3 (OP) show the cross-problem and cross-scale generalization capabilities of DIFU-Ada. Our framework achieves competitive average optimality gaps (9.70% for PCTSP and 7.26% for OP) across all scales, while maintaining its zero-shot cross-problem transfer capability. This performance is achieved without the training costs (e.g., 3-5 days of retraining reported for existing methods) or the need for well-labeled training datasets. While OR-based heuristics like Compass exhibit strong performance on narrowly defined problem types, DIFU-Ada's ability to leverage pre-trained diffusion models across different CO problem variants without additional training highlights its flexibility and broader applicability. Besides, the results on large-scale PCTSP-500 and PCTSP-1000 in Appendix , Table 4 demonstrate the scalability of DIFU-Ada.

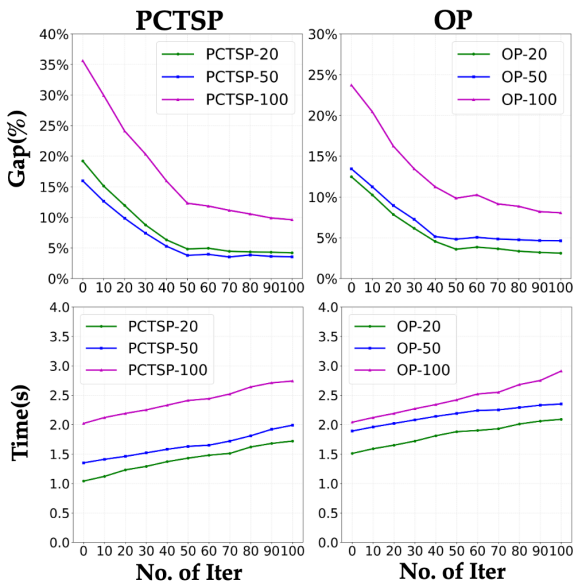

Figure 3: Ablation studies of the number of recursive travel steps on the trade-off between optimality gap (%) and inference time (s) for PCTSP and OP.

Ablation Studies

The DIFU-Ada framework integrates two key components: energy-guided sampling and recursive renoising-denoising travel. To assess the contribution of each, we conducted ablation studies. Figure 3 quantitatively illustrates the impact of the recursive renoising-denoising, demonstrating a progressive enhancement in cross-problem transfer capabilities as the number of recursive iterations increases. Figure 4 presents an analysis of performance with or without energy-guided sampling during the recursive iterations. The performance gap observed in these studies shows the efficacy of both components and highlights their distinct roles in the effectiveness to achieve zero-shot cross-problem transfer.

Moreover, Figure 5 shows the effect of the temperature parameter in controlling the guidance strength of adaptation. Higher temperature effectively preserve the structural in-

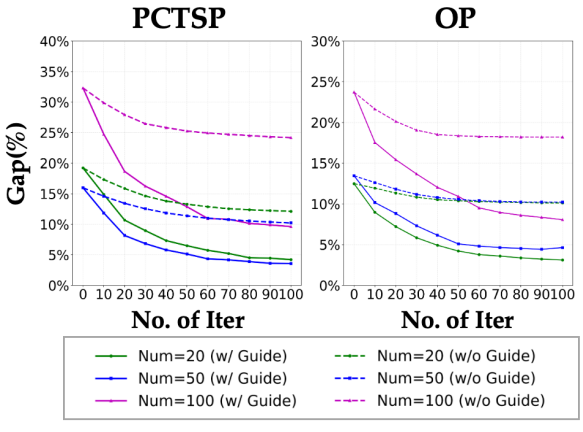

Figure 4: The Optimality Gap of performing recursive travel with or without energy-guided sampling on PCTSP and OP.

	Method	PCTSP-20 Gap ↓	PCTSP-20 Time ↓	PCTSP-50 Gap ↓	PCTSP-50 Time ↓	PCTSP-100 Gap ↓	PCTSP-100 Time ↓	Avg Gap ↓	Training-free	Training Time ↓
OR	Gurobi	**0.00%**	3.10s	—	—	—	—	—	—	—
	OR-Tools	2.13%	12.31s	4.85%	2.02m	10.33%	5.84m	5.77%	—	—
	ILS (C++)	1.07%	2.13s	**0.00%**	18.30s	**0.00%**	56.11s	0.36%	—	—
	ILS (Python 10x)*	63.23%	3.05s	148.05%	4.70s	209.78%	5.27s	140.35%	—	—
Learning	AM (Greedy)	2.76%	0.02s	18.20%	0.07s	28.98%	0.15s	16.65%	✗	3.5 days
	AM (Sampling)	2.54%	2.43s	14.58%	7.08s	22.20%	15.13s	13.11%	✗	3.5 days
	MDAM* (Greedy)	11.76%	41.10s	24.73%	1.31m	30.07%	1.96m	22.19%	✗	4.3 days
	MDAM* (Beam Search)	5.88%	2.70m	18.81%	4.77m	26.09%	6.97m	16.93%	✗	4.3 days
	ASP*	12.05%	0.03s	10.34%	0.08s	**11.56%**	0.18s	11.32%	✗	4.6 days
	AM-FT (greedy)	2.11%	0.03s	16.58%	0.07s	29.08%	0.16s	15.92%	✗	4.9 days
	AM-FT (Sampling)	**1.02%**	2.51s	14.11%	8.02s	25.19%	17.21s	13.44%	✗	4.9 days
	DIFU-Ada (Ours)	4.45%	1.80s	**8.43%**	1.92s	16.22%	2.87s	**9.70%**	✓	**0 day**

Table 2: Comprehensive evaluation of cross-scale generalization capabilities across different solver categories on PCTSP instances. Comparison includes exact solvers (Gurobi), OR-based heuristics (OR-Tools, ILS), learning-based models (AM, MDAM, ASP, AM-FT, DIFU-Ada). Performance metrics include optimality gap, inference time, and training time. The results marked with $*$ are reported from (Wang et al. 2024). The proposed DIFU-Ada achieves competitive performance while requiring no training.

	Method	OP-20 Gap ↓	OP-20 Time ↓	OP-50 Gap ↓	OP-50 Time ↓	OP-100 Gap ↓	OP-100 Time ↓	Avg Gap ↓	Training-free	Training Time ↓
OR	Gurobi	**0.00%**	10.22s	—	—	—	—	—	—	—
	Compass	0.15%	0.27s	**0.00%**	1.02s	**0.00%**	6.74s	0.05%	—	—
	Tsili (Greedy)*	16.58%	0.02s	19.22%	0.03s	19.71%	0.03s	18.50%	—	—
	Tsili (Sampling)*	0.85%	0.55s	4.46%	2.45s	8.56%	9.08s	4.62%	—	—
Learning	AM (Greedy)	5.78%	0.04s	12.10%	0.09s	28.75%	0.19s	15.54%	✗	3.2 days
	AM (Sampling)	2.62%	2.87s	8.81%	8.12s	22.17%	14.98s	11.20%	✗	3.2 days
	AM-FT (Greedy)	5.50%	0.04s	11.51%	0.08s	23.40%	0.18s	13.47%	✗	4.7 days
	AM-FT (Sampling)	**1.01%**	2.76s	7.19%	8.44s	19.51%	18.15s	9.24%	✗	4.7 days
	DIFU-Ada (Ours)	3.44%	2.01s	**6.11%**	2.29s	**12.21%**	2.85s	**7.26%**	✓	**0 day**

Table 3: Comprehensive evaluation of cross-scale generalization capabilities across different solver categories on OP instances. Comparison includes exact solvers (Gurobi), OR-based heuristics (Compass, Tsili), learning-based models (AM, AM-FT, DIFU-Ada). The results marked with $*$ are reported from (Lin et al. 2024).

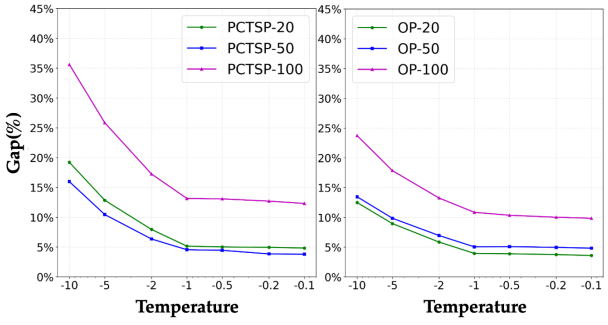

Figure 5: Optimality gap changes with respect to the guided temperature ($-\tau$ as the x-axis label) on PCTSP and OP.

formation in the original problem instances, whereas lower temperatures adapt to the characteristics of new problem instances. This temperature-dependent behavior enables fine-grained control over the balance between maintaining prior knowledge and new problem features. In Appendix , Figure 6, the results on varying hyperparameter μ in energy function show that the performances are not sensitive to the choice of μ, thus we choose the optimal $\mu = 1$ empirically for all problems.

Conclusions and Limitations

We introduce DIFU-Ada, an inference-time adaptation framework for zero-shot cross-problem transfer in diffusion-based solvers. By integrating energy-guided sampling with a recursive denoising strategy, our approach adapts pre-trained models to generate high-quality solutions for new TSP variants without retraining. Experiments confirm competitive zero-shot performance against existing methods. This work highlights the potential of adapting diffusion solvers for real-world combinatorial optimization problems with dynamic constraints. While promising on TSP variants, future work should test its applicability across broader domains.

Acknowledgments

The work of Farzan Farnia is partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China, Project 14209920, and is partially supported by CUHK Direct Research Grants with CUHK Project No. 4055164 and 4937054. Also, the authors would like to thank the anonymous reviewers for their useful comments and suggestions.

References

Arora, S. 1998. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5): 753–782.

Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and Van Den Berg, R. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34: 17981–17993.

Balas, E. 1989. The prize collecting traveling salesman problem. *Networks*, 19(6): 621–636.

Bansal, A.; Chu, H.-M.; Schwarzschild, A.; Sengupta, S.; Goldblum, M.; Geiping, J.; and Goldstein, T. 2023. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 843–852.

Bengio, Y.; Lodi, A.; and Prouvost, A. 2021. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2): 405–421.

Chung, H.; Kim, J.; Mccann, M. T.; Klasky, M. L.; and Ye, J. C. 2022. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*.

Den Hertog, D.; Roos, C.; and Terlaky, T. 1992. On the classical logarithmic barrier function method for a class of smooth convex programming problems. *Journal of Optimization Theory and Applications*, 73(1): 1–25.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Fu, Z.-H.; Qiu, K.-B.; and Zha, H. 2021. Generalize a small pre-trained model to arbitrarily large tsp instances. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 7474–7482.

Golden, B. L.; Levy, L.; and Vohra, R. 1987. The orienteering problem. *Naval Research Logistics (NRL)*, 34(3): 307–318.

Gonzalez, T. F. 2007. *Handbook of approximation algorithms and metaheuristics*. Chapman and Hall/CRC.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Hottung, A.; Bhandari, B.; and Tierney, K. 2021. Learning a latent search space for routing problems using variational autoencoders. In *International Conference on Learning Representations*.

Joshi, C. K.; Laurent, T.; and Bresson, X. 2019. An efficient graph convolutional network technique for the travelling salesman problem. *arXiv preprint arXiv:1906.01227*.

Khalil, E.; Dai, H.; Zhang, Y.; Dilkina, B.; and Song, L. 2017. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems*, 30.

Kim, M.; Park, J.; and Park, J. 2022. Sym-nco: Leveraging symmetricity for neural combinatorial optimization. *Advances in Neural Information Processing Systems*, 35: 1936–1949.

Kobeaga, G.; Merino, M.; and Lozano, J. A. 2018. An efficient evolutionary algorithm for the orienteering problem. *Computers & Operations Research*, 90: 42–59.

Kool, W.; Van Hoof, H.; and Welling, M. 2018. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*.

Kwon, Y.-D.; Choo, J.; Kim, B.; Yoon, I.; Gwon, Y.; and Min, S. 2020. Pomo: Policy optimization with multiple optima for reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 21188–21198.

LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; Huang, F.; et al. 2006. A tutorial on energy-based learning. *Predicting structured data*, 1(0).

Li, Y.; Guo, J.; Wang, R.; and Yan, J. 2024a. From distribution learning in training to gradient search in testing for combinatorial optimization. *Advances in Neural Information Processing Systems*, 36.

Li, Y.; Guo, J.; Wang, R.; Zha, H.; and Yan, J. 2024b. Fast t2t: Optimization consistency speeds up diffusion-based training-to-testing solving for combinatorial optimization. *Advances in Neural Information Processing Systems*, 37: 30179–30206.

Lin, S. 1973. An efficient heuristic algorithm for the traveling salesman problem. *Oper. Res.*, 21: 498–516.

Lin, Z.; Wu, Y.; Zhou, B.; Cao, Z.; Song, W.; Zhang, Y.; and Jayavelu, S. 2024. Cross-problem learning for solving vehicle routing problems. *arXiv preprint arXiv:2404.11677*.

Lucas, A. 2014. Ising formulations of many NP problems. *Frontiers in physics*, 2: 5.

Qiu, R.; Sun, Z.; and Yang, Y. 2022. Dimes: A differentiable meta solver for combinatorial optimization problems. *Advances in Neural Information Processing Systems*, 35: 25531–25546.

Sanokowski, S.; Hochreiter, S.; and Lehner, S. 2024. A Diffusion Model Framework for Unsupervised Neural Combinatorial Optimization. *arXiv preprint arXiv:2406.01661*.

Shen, Y.; JIANG, X.; Yang, Y.; Wang, Y.; Han, D.; and Li, D. 2024. Understanding and Improving Training-free Loss-based Diffusion Guidance. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Dhariwal, P.; Chen, M.; and Sutskever, I. 2023. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, 32211–32252.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Sun, Z.; and Yang, Y. 2023. Difusco: Graph-based diffusion solvers for combinatorial optimization. *Advances in Neural Information Processing Systems*, 36: 3706–3731.

Tsiligirides, T. 1984. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9): 797–809.

Wang, C.; Yu, Z.; McAleer, S.; Yu, T.; and Yang, Y. 2024. Asp: Learn a universal neural solver! *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Welling, M.; and Teh, Y. W. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 681–688.

Xin, L.; Song, W.; Cao, Z.; and Zhang, J. 2021. Multi-decoder attention model with embedding glimpse for solving vehicle routing problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 12042–12049.

Ye, H.; Lin, H.; Han, J.; Xu, M.; Liu, S.; Liang, Y.; Ma, J.; Zou, J. Y.; and Ermon, S. 2024. Tfg: Unified training-free guidance for diffusion models. *Advances in Neural Information Processing Systems*, 37: 22370–22417.

Yu, J.; Wang, Y.; Zhao, C.; Ghanem, B.; and Zhang, J. 2023. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 23174–23184.

Zhang, J.; Liu, C.; Li, X.; Zhen, H.-L.; Yuan, M.; Li, Y.; and Yan, J. 2023. A survey for solving mixed integer programming via machine learning. *Neurocomputing*, 519: 205–217.

Zhao, H.; Yu, K.; Huang, Y.; Yi, R.; Zhu, C.; and Xu, K. 2025. DISCO: Efficient diffusion solver for large-scale combinatorial optimization problems. *Graphical Models*, 141: 101284.