Boosting Generalization in Diffusion-Based Neural Combinatorial Solver via Energy-guided Sampling

윤성준

Table of Contents

How to solve Combinatorial Optimizaion Problems?

Combinatorial Optimizaion Problems

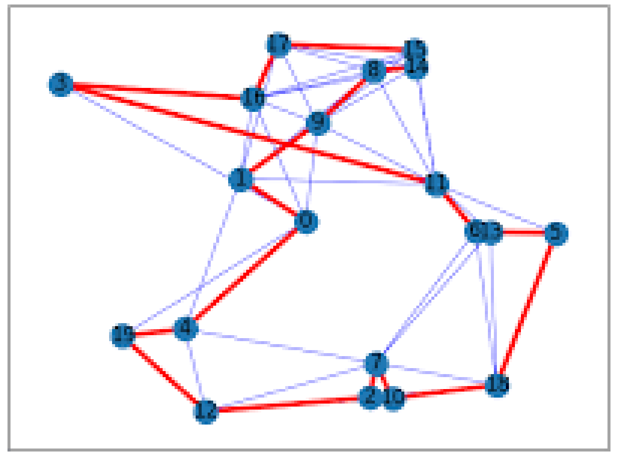
Exact Solver
- Optimal Solution
- Large computation Complexity

Heuristic & Approximation
- Find Feasible Solution faster than exact solver
- Certain level of optimality

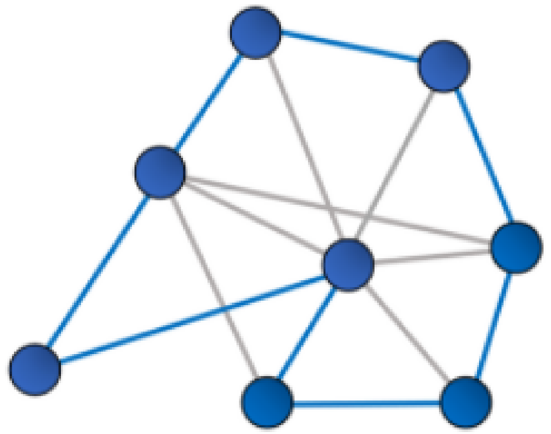
→ **Deep Learning**
- Fast Compuation time
- Less Optimality gap

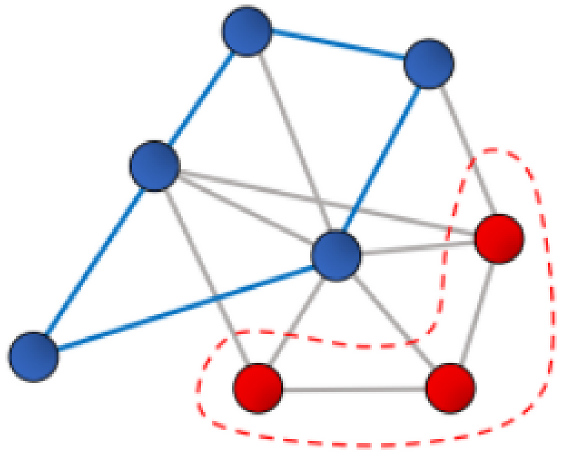
I. Introduction

1.Cross-scale problem



2.Cross-problem transfer capability





Adapter Foundation

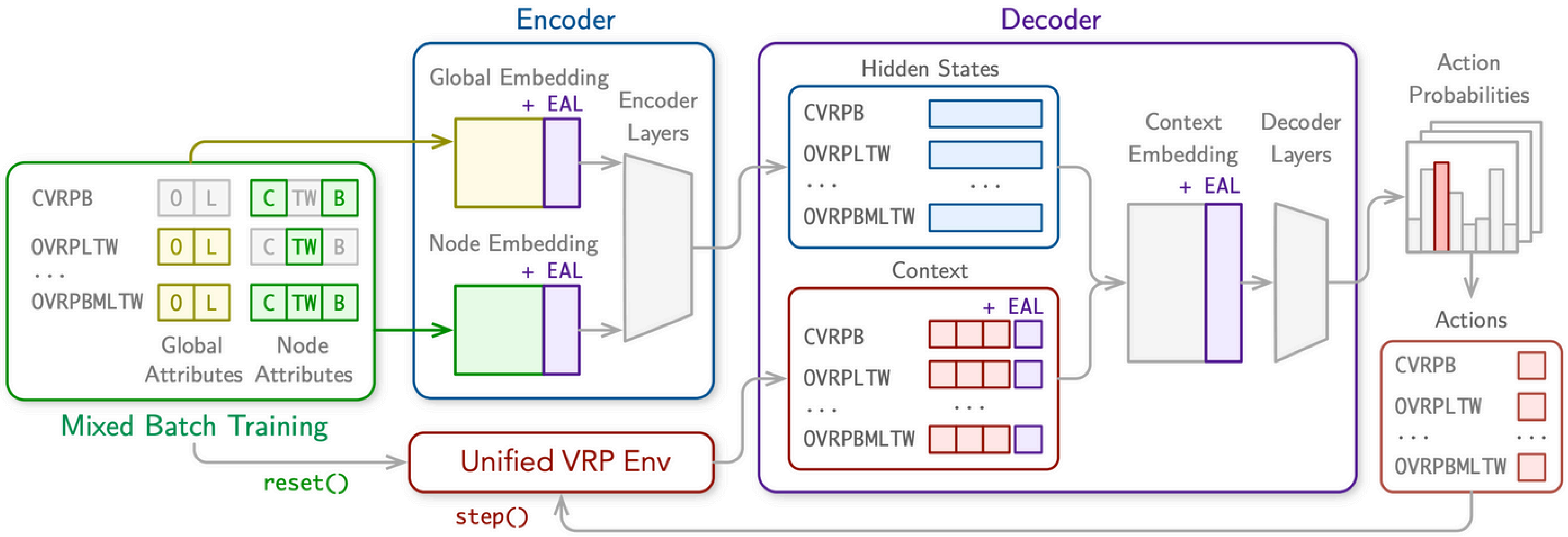
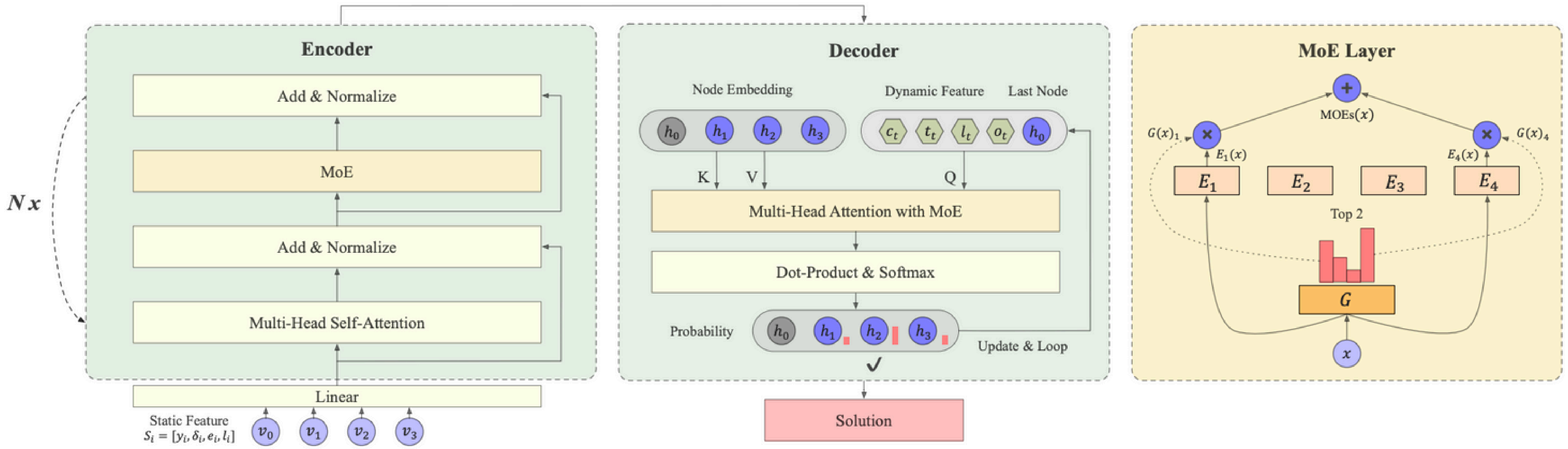


Figure 4.1: Overview of `RouteFinder`.

Reference:
RouteFinder Towards Foundation Models for Vehicle Routing Problems
https://openreview.net/pdf?id=hCiaiZ6e4G

I. Introduction

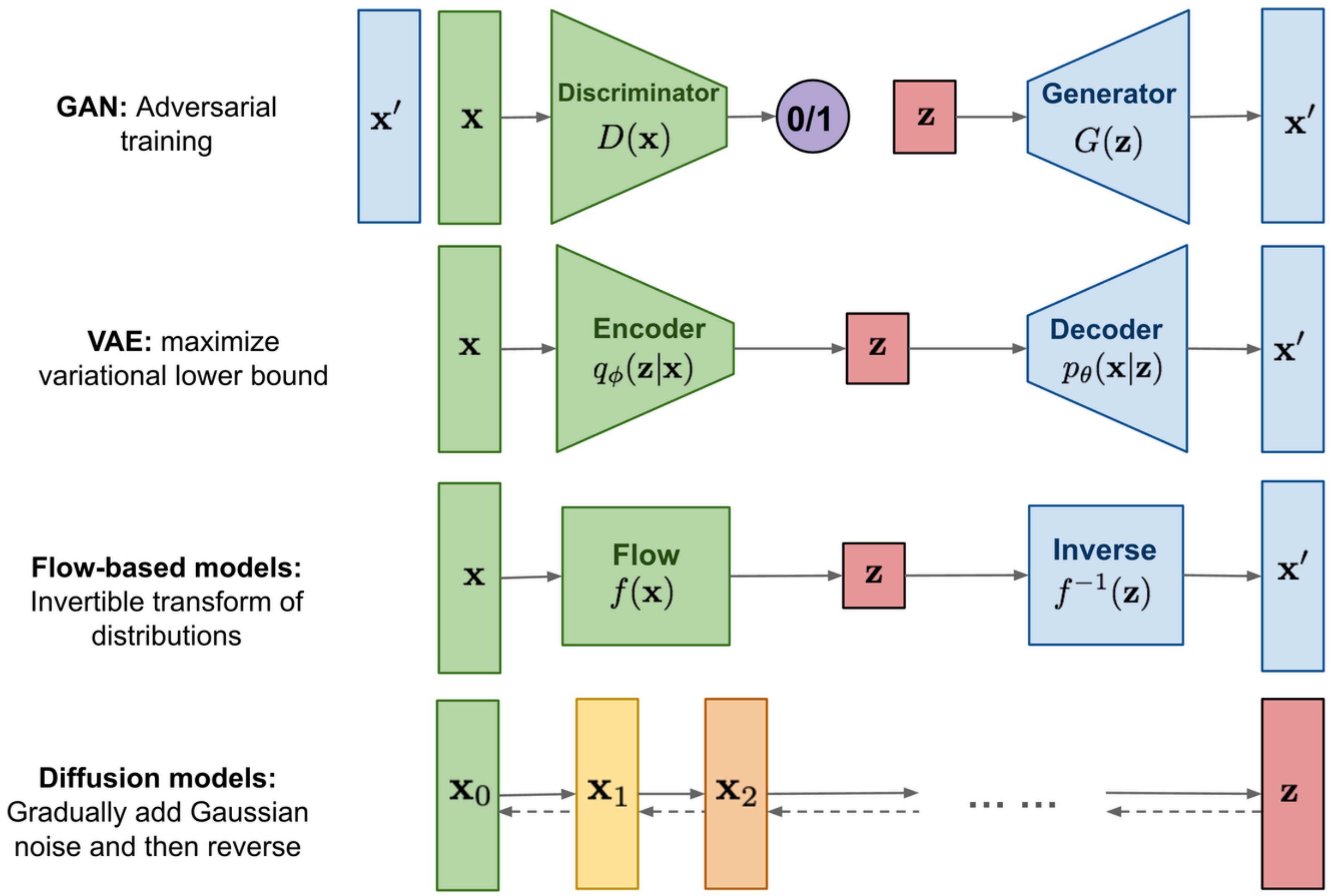
MOE Foundation



Reference:
MVMoE: Multi-Task Vehicle Routing Solver with Mixture-of-Experts
https://arxiv.org/pdf/2405.01029

II. Related works

Generative Model(Diffusion)



parameterized Markov chain trained using variational inference to produce samples matching the data after finite time

$$p_\theta(X_0) := \int p_\theta(X_{0:T})dX_{1:T}$$

II. Related works

Guided Sampling

Classifier guidance

$$p_{\theta,\phi}(x_t|x_{t+1}, y) = Zp_\theta(x_t|x_{t+1})p_\phi(y|x_t)$$

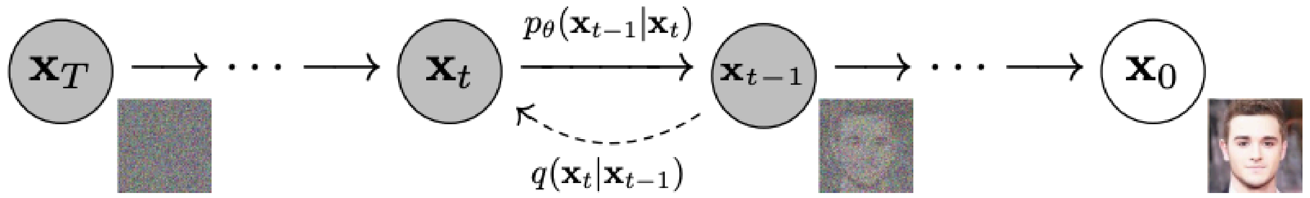


Figure 2: The directed graphical model considered in this work.

Classifier-Free guidance

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = (1 + w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda)$$

Training-Free guidance

Reference:
Diffusion Models Beat GANs on Image
https://arxiv.org/pdf/2105.05233#page=25&zoom=100,144,96
CLASSIFIER-FREE DIFFUSION GUIDANCE
https://arxiv.org/pdf/2207.12598
Universal Guidance for Diffusion Models
https://arxiv.org/pdf/2302.07121

III. Theoreitical Analysis

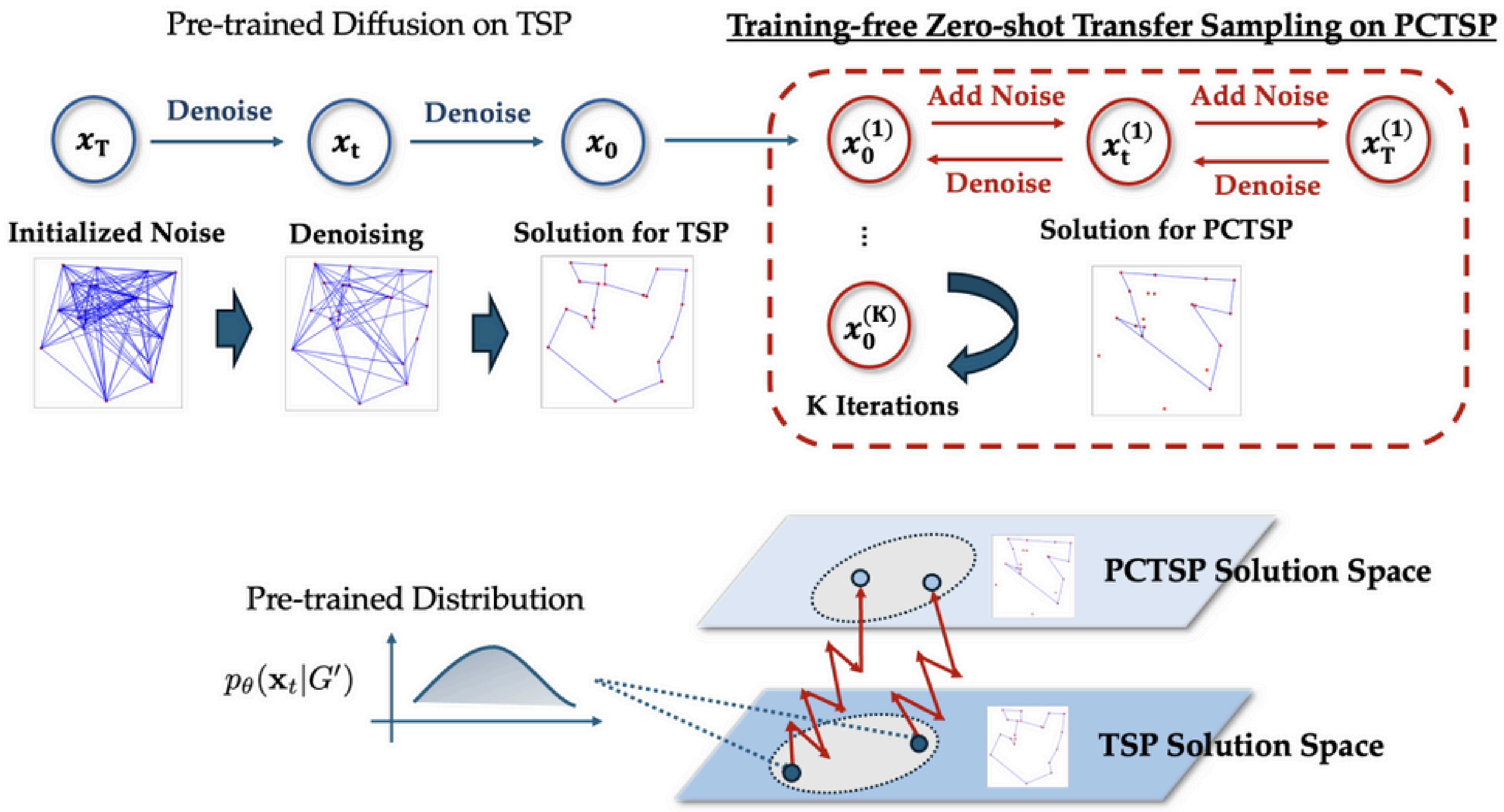
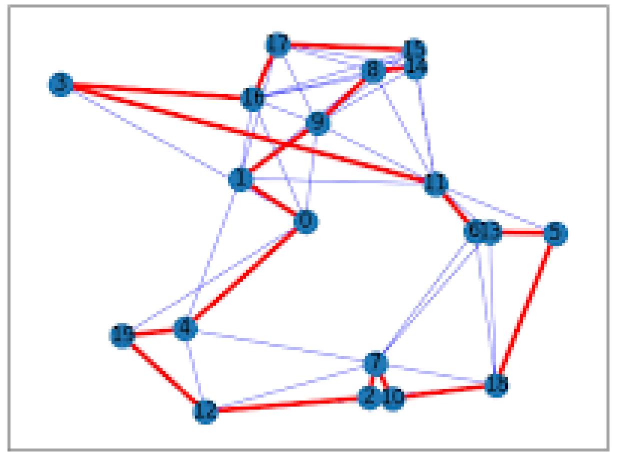


Figure 1: Overview of energy-guided sampling framework for achieving cross-problem generalization. Left: Pre-trained diffusion model performs denoising on original problem G (TSP). Right: Proposed energy-guided sampling on target problem G' (PCTSP) conducts K rewrite iterations of noise addition and guided denoising.

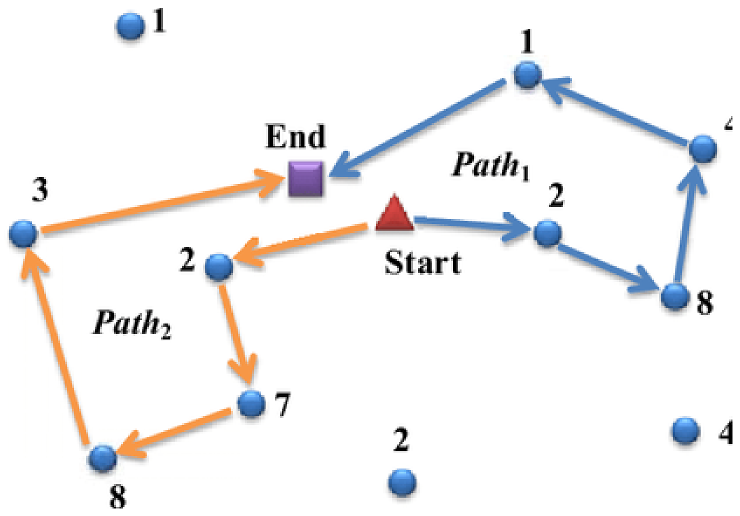
III. Theoreitical Analysis

Combinatorial Optimization problem



TSP

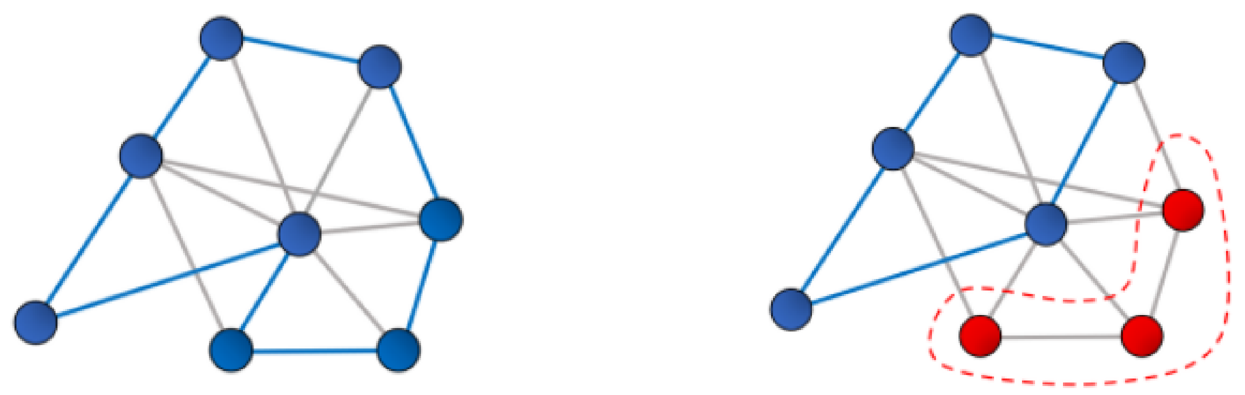
Prize Collecting TSP(PCTSP)



Orienteering Problem

III. Theoreitical Analysis

Combinatorial Optimization problem

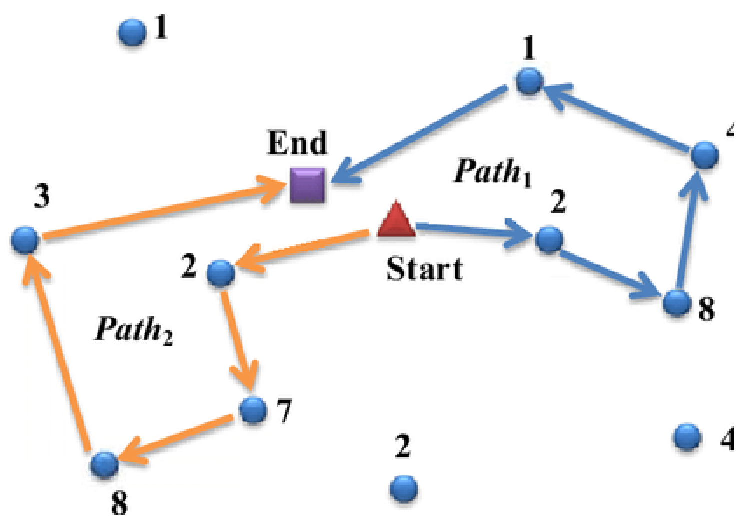


Prize Collecting TSP(PCTSP)

The goal is to collect as much prize as possible while minimizing the total travel cost.
The prize is only revealed when the node is visited.
if not visit node, each node have penalty

III. Theoreitical Analysis

Combinatorial Optimization problem



Orienteering Problem

At each step, the agent chooses a location to visit in order to maximize the collected prize.
The total length of the path must not exceed a given threshold.

III. Theoreitical Analysis

CO problem objective Formulation

$$\mathrm{x}^* = \arg\min_{\mathrm{x}\in\mathcal{X}_{\mathcal{G}}} f(\mathrm{x}; G)$$
$$f(\mathrm{x}; G) = f_{\mathrm{cost}}(\mathrm{x}; G) + \beta \cdot f_{\mathrm{valid}}(\mathrm{x}; G)$$

$f_{\mathrm{cost}}(\cdot; G)$: meausres the solution quality
$f_{\mathrm{valid}}(\cdot; G)$: problem- specific constraints through a penalty coefficient
$\mathrm{x} \in \{0,1\}^N$: decision variable

III. Theoreitical Analysis

CO problem objective Formulation

$$x^* = \arg\min_{x \in \mathcal{X}_\mathcal{G}} f(x; G)$$

$$f(x; G) = f_{\text{cost}}(x; G) + \beta \cdot f_{\text{valid}}(x; G)$$

Proposition A.5 (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 $\mathbf{x} \in \{0,1\}^{|E|}$, $\mathbf{y} \in \{0,1\}^{|V|}$ that minimizes:*

$$f(\mathbf{x}, \mathbf{y}; G) = f_{cost}(\mathbf{x}, \mathbf{y}; G) + \beta \cdot f_{valid}(\mathbf{x}, \mathbf{y}; G)$$

$$\textit{where} \quad f_{cost}(\mathbf{x}, \mathbf{y}; G) = \sum_{e \in E} w_e x_e + \sum_{v \in V} p_v(1 - y_v)$$

$$f_{valid}(\mathbf{x}, \mathbf{y}; G) = \max(0, R - \sum_{v \in V} r_v y_v)$$

$f_{\text{cost}}(\cdot; G)$: meausres the solution quality

$f_{\text{valid}}(\cdot; G)$: problem- specific constraints through a penalty coefficient

$x \in \{0, 1\}^N$: decision variable

III. Theoreitical Analysis

CO problem objective Formulation

$$x^* = \arg\min_{x \in \mathcal{X}_{\mathcal{G}}} f(x; G)$$

$$f(x; G) = f_{\text{cost}}(x; G) + \beta \cdot f_{\text{valid}}(x; G)$$

$f_{\text{cost}}(\cdot; G)$: meausres the solution quality

$f_{\text{valid}}(\cdot; G)$: problem- specific constraints through a penalty coefficient

$x \in \{0,1\}^N$: decision variable

Proposition A.6 (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 $\mathbf{x} \in \{0,1\}^{|E|}$, $\mathbf{y} \in \{0,1\}^{|V|}$ that minimizes:*

$$f(\mathbf{x}, \mathbf{y}; G) = f_{cost}(\mathbf{x}, \mathbf{y}; G) + \beta \cdot f_{valid}(\mathbf{x}, \mathbf{y}; G)$$

$$\textit{where} \quad f_{cost}(\mathbf{x}, \mathbf{y}; G) = -\sum_{v \in V} s_v y_v$$

$$f_{valid}(\mathbf{x}, \mathbf{y}; G) = \max(0, \sum_{e \in E} w_e x_e - B)$$

III. Theoreitical Analysis

Analysis transfer of Combinatorial Optimization problem (PCTSP)

Definition 4.1 (Marginal Decrease)**.** For a non-empty subset of nodes $S \subseteq V$, let $\mathrm{TSP}(S)$ denote the cost of the optimal TSP tour visiting all nodes in S. The marginal decrease of a subset S is defined as

$$\Delta(S) = \mathrm{TSP}(V) - \mathrm{TSP}(V \setminus S).$$

The marginal decrease measures the cost reduction of not visiting a subset of nodes S, which helps quantify the difference between the optimal tours. Take PCTSP as an example. If for any non-empty subset of nodes $S \subseteq V$, the penalty of not visiting the nodes in S satisfies $\sum_{i \in S} p_i \geq \Delta(S)$, then PCTSP and TSP share the same optimal tours. Based on this notion, we formalize the structural similarities in the following theorem.

TSP(V) : cost of the optimal TSP

TSP(V\S): cost of the optimal TSP except for subset S

\Delta(S): cost reduction TSP except for subset S

III. Theoreitical Analysis

Analysis transfer of Combinatorial Optimization problem (PCTSP)

Under these assumptions, we compare the optimal solutions of PCTSP/OP with that of TSP as follows.

(i) For PCTSP, if for any non-empty subset of nodes $S \subseteq V$, the penalty of not visiting the nodes in S satisfies

$$\sum_{i \in S} p_i \geq \Delta(S),$$

then PCTSP and TSP share the same optimal solutions, since not visiting any subset of nodes would result in a higher total cost. We can thus formulate the optimal cost (tour length + penalty) of PCTSP as

$$\mathrm{PCTSP}(V) = \mathrm{TSP}(V) + \min_{S \subseteq V} \sum_{i \in S} p_i - \Delta(S). \tag{10}$$

And we have the optimal solutions of PCTSP being the tours in $\mathrm{TSP}(V \setminus S)$ with

$$S \in \arg\min_{S \subseteq V} \sum_{i \in S} p_i - \Delta(S),$$

completing the proof for PCTSP.

III. Theoreitical Analysis

Analysis transfer of Combinatorial Optimization problem

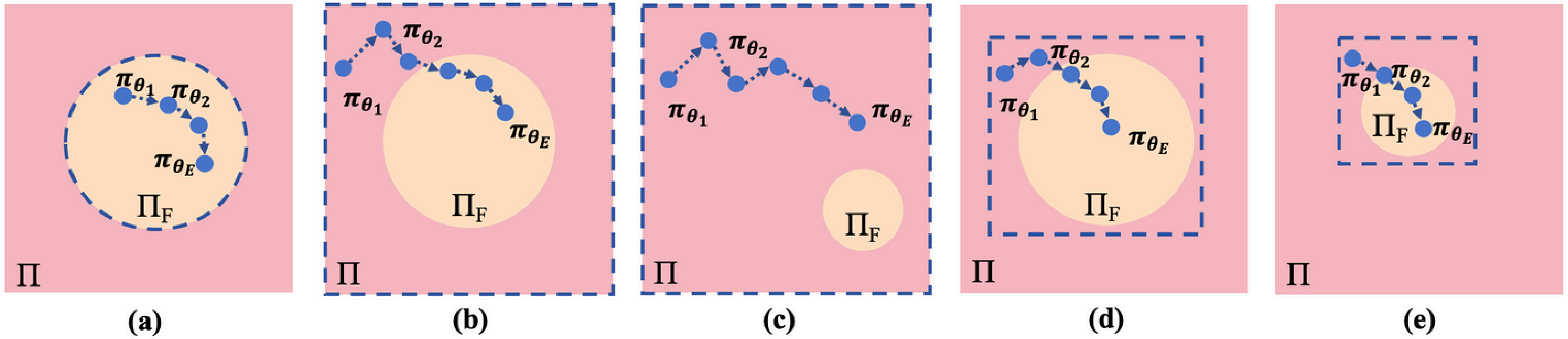


Figure 2: Illustration of policy optimization trajectories on VRP with varying difficulty levels - (a)(b)(d) easy and (c)(e) hard, and different constraint handling schemes - (a) feasibility masking, (b)(c) Lagrangian multiplier, and (d)(e) our PIP. The orange-filled circle denotes the feasible policy space Π_F, while the dotted frame represents the actual search space of the neural policies π_θ.

Reference:
Learning to Handle Complex Constraints for Vehicle Routing Problems
https://arxiv.org/pdf/2410.21066

III. Theoreitical Analysis

Probabilistic Modeling for CO

$$p_{\theta,\phi}(x_t|x_{t+1},y) = Zp_\theta(x_t|x_{t+1})\boxed{p_\phi(y|x_t)}$$

$$\mathcal{E}(\cdot|;\mathcal{G}) \doteq |y - f(\cdot;G)|$$

$$p(y|\mathbf{x};G) = \frac{\exp\left(-\frac{1}{\tau}\mathcal{E}(y,\mathbf{x};G)\right)}{\mathcal{Z}}, \text{ where } \mathcal{Z} = \sum_{\mathbf{x}} \exp\left(-\frac{1}{\tau}\mathcal{E}(y,\mathbf{x};G)\right),$$

tau: controls the temperature of system
Z : partition function for pmf condition
x : optimal solution

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

III. Theoreitical Analysis

Discrete Diffusion Generation Modeling

Loss

$$
\begin{aligned}
L(\theta) &= \mathbb{E}_{G\sim\mathcal{G}}[-\log p_\theta(\mathbf{x}_0|G)] \\
&\leq \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[D_{KL}[q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,G)] - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1,G)\right] + C.
\end{aligned}
$$

III. Theoreitical Analysis

Discrete Diffusion Generation Modeling(DIFUSCO)

$$q(x_{1:T}|x_0) = \Pi_{t=1}^{T} q(x_t|x_{t-1})$$

$$p_\theta(x_{0:T}|G) = p(x_T)\Pi_{t=1}^{T} p_\theta(x_{t-1}|x_t, G)$$

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

forward process

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

backward process

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

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathrm{Cat}\left(\mathbf{x}_{t-1}; \mathbf{p} = \frac{\widetilde{\mathbf{x}}_t\mathbf{Q}_t^\top \odot \widetilde{\mathbf{x}}_0\overline{\mathbf{Q}}_{t-1}}{\widetilde{\mathbf{x}}_0\overline{\mathbf{Q}}_t\widetilde{\mathbf{x}}_t^\top}\right),$$

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

III. Theoreitical Analysis

Energy-guided Sampling for Cross-problem

$$\nabla_x \log p(x|y) = \nabla_x \log p(x) + \nabla_x \log p(y|x)$$

$$\underbrace{\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t|y^*_{G'}, G')}_{\text{posterior score}} = \underbrace{\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t|G')}_{\text{pre-trained prior score}} + \underbrace{\nabla_{\mathbf{x}_t} \log p_t(y^*_{G'}|\mathbf{x}_t, G')}_{\text{energy-guided score}}.$$

III. Theoreitical Analysis

Energy-guided Sampling for Cross-problem

$$\nabla_x \log p(x|y) = \nabla_x \log p(x) + \nabla_x \log p(y|x)$$

$$\underbrace{\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t | y^*_{G'}, G')}_{\text{posterior score}} = \underbrace{\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t | G')}_{\text{pre-trained prior score}} + \underbrace{\nabla_{\mathbf{x}_t} \log p_t(y^*_{G'} | \mathbf{x}_t, G')}_{\text{energy-guided score}}.$$

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

$$\mathcal{E}(\cdot|;\mathcal{G}) \doteq |y - f(\cdot;G)|$$

III. Theoreitical Analysis

Energy-guided Sampling for Cross-problem

$$\nabla_x \log p(x|y) = \nabla_x \log p(x) + \nabla_x \log p(y|x)$$

$$\underbrace{\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t|y^*_{G'}, G')}_{\text{posterior score}} = \underbrace{\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t|G')}_{\text{pre-trained prior score}} + \underbrace{\nabla_{\mathbf{x}_t} \log p_t(y^*_{G'}|\mathbf{x}_t, G')}_{\text{energy-guided score}}.$$

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

$$\mathcal{E}(\cdot|; \mathcal{G}) \doteq |y - f(\cdot; G)|$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, y^*_{G'}, G') \propto p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, G')p(y^*_{G'}|\mathbf{x}_t, G'),$$

$$p(y^*_{G'}|\mathbf{x}_t, G') = \exp\left(-\frac{1}{\tau}\nabla_{\mathbf{x}_t} f\left(\widetilde{\mathbf{x}}_0(\mathbf{x}_t); G'\right)\right).$$

Reference:

Diffusion Models Beat GANs on Image

https://arxiv.org/pdf/2105.05233#page=25&zoom=100,144,96

IV. Proposed Approach

Algorithm 1: Energy-guided Diffusion Sampling for Cross-problem Transfer

Input:
p_θ: Pre-trained diffusion model
G': Target problem instance
T: Number of diffusion steps
τ: Energy guidance temperature
K: Number of re-inference iterations

Output: Optimal solution $\mathbf{t}_K$ for problem instance G'

Initialize $\mathbf{x}_T$ with random binary values;
for $k = 1$ **to** K **do**
    Initialize $\mathbf{x}_T$ with previous best solution $\mathbf{t}_{k-1}$
    for $t = T$ **to** 1 **do**
        Compute $p_\theta(\mathbf{x}_t|G')$ from pre-trained model;
        Compute energy gradient: $\nabla_{\mathbf{x}_t} f(\widetilde{\mathbf{x}}_0(\mathbf{x}_t); G')$;
        Compute $p_t(y^*_{G'}|\mathbf{x}_t, G') \propto \exp(-\nabla_{\mathbf{x}_t} f/\tau)$;
        Compute guided posterior: $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, y^*_{G'}, G') \propto p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, G') p_t(y^*_{G'}|\mathbf{x}_t, G')$;
        Update next state with Bernoulli Sampling: $\mathbf{x}_{t-1} \sim \mathrm{Cat}\left(\mathbf{x}_{t-1}; p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, y^*_{G'}, G')\right)$;
    end for
    Decode $\mathbf{x}_0$ to the best solution $\mathbf{t}_k$;
end for
return $\mathbf{t}_K$

V. Numerical Results

	Method	PCTSP-20		PCTSP-50		PCTSP-100				
		Gap ↓	Time ↓	Gap ↓	Time ↓	Gap ↓	Time ↓	Avg Gap ↓	Training-based	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%	—	—
Auto-reg*	AM (Greedy)	2.88%	0.02s	17.95%	0.06s	29.24%	0.14s	16.69%	✓	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
	AM (ASP)	12.05%	0.03s	10.34%	0.08s	**11.56%**	0.18s	11.32%	✓	1.1 days
	AM-FT (Sampling)	**1.02%**	2.51s	14.11%	8.02s	25.19%	18.21s	13.44%	✓	4.9 days
Diff.	DIFUSCO (TSP)	19.21%	1.04s	18.61%	1.69s	43.42%	2.34s	27.08%	✗	1.5 days
	DIF-guide (Ours)	4.83%	4.58s	**7.58%**	6.37s	18.1%	10.79s	**10.17%**	✗	**0 days**

Table 1: 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), autoregressive models (AM, MDAM, AM-ASP, AM-FT), and diffusion-based approaches (DIFUSCO, DIF-guide). Performance metrics include optimality gap, inference time, and training time. The best results marked with ∗ are reported from (Wang et al. 2024). The proposed DIF-guide achieves competitive performance while requiring no training.

V. Numerical Results

Method	PCTSP-20			PCTSP-50			PCTSP-100		
	Cost	Gap	Time	Cost	Gap	Time	Cost	Gap	Time
DIFUSCO	3.78	19.21%	**1.04s**	5.20	15.97%	**1.35s**	8.14	35.61%	**2.01s**
DIF-Guide (Ours)	**3.32**	**4.83%**	5.58s	**4.69**	**3.51%**	7.51s	**6.67**	**13.32%**	12.20s

Table 2: Zero-shot cross-problem transfer performance comparison between baseline DIFUSCO and our DIF-Guide approach on PCTSP instances. Results show solution cost, optimality gap, and computation time across three problem scales (20, 50, and 100 nodes). The energy-guided sampling uses 50 iterations with greedy decoding strategy.

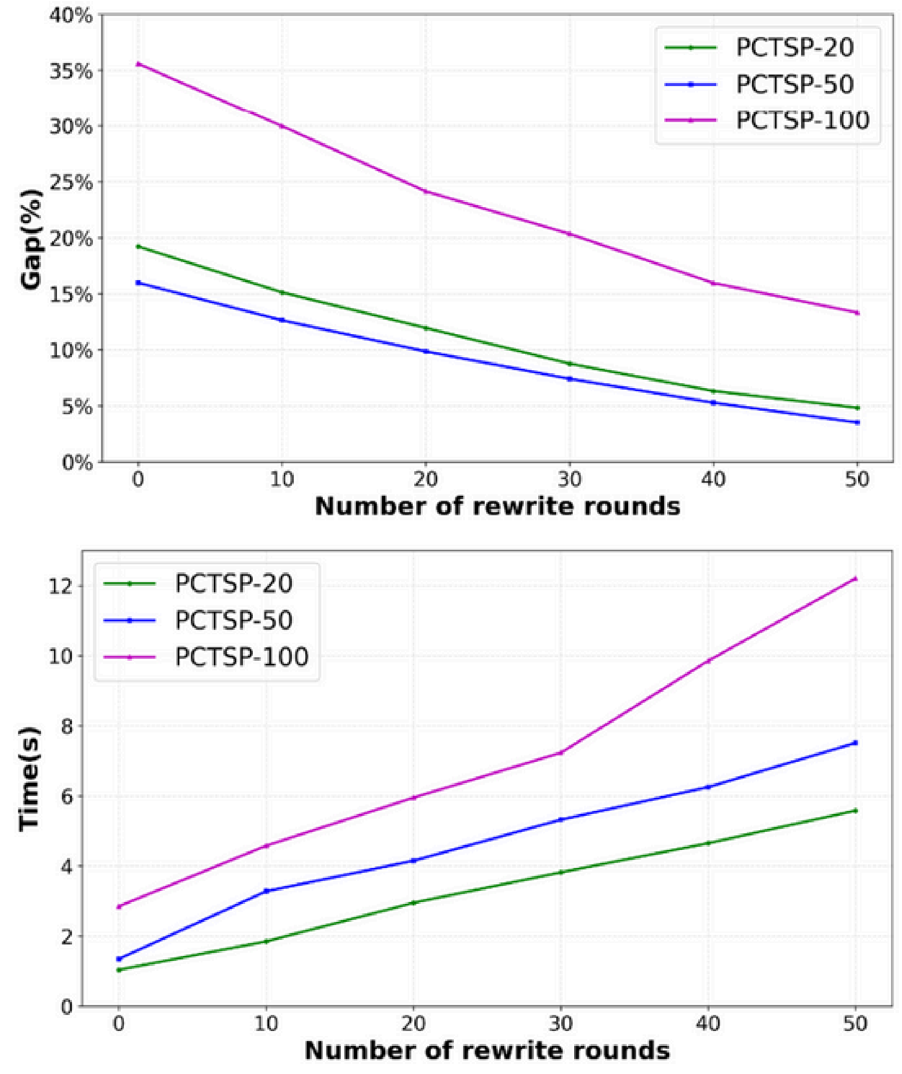


Figure 2: Trade-off between performance and inference time of energy-guided sampling with respect to rewriting rounds.