Fractional Langevin Dynamics for Combinatorial Optimization via Polynomial-Time Escape

Shiyue Wang[1,3†], **Ziao Guo**[2†], **Changhong Lu**[1], **Junchi Yan**[2,3 *]
[1]School of Mathematical Sciences, Key Laboratory of MEA,
and Shanghai Key Laboratory of PMMP, East China Normal University
[2]School of Computer Science & School of Artificial Intelligence, Shanghai Jiao Tong University
[3]Shanghai Innovation Institute
{wangshiyue@stu,chlu@math}.ecnu.edu.cn {ziao.guo,yanjunchi}@sjtu.edu.cn

Abstract

Langevin dynamics (LD) and its discrete proposal have been widely applied in the field of Combinatorial Optimization (CO). Both sampling-based and data-driven approaches have benefited significantly from these methods. However, LD's reliance on Gaussian noise limits its ability to escape narrow local optima, requires costly parallel chains, and performs poorly in rugged landscapes or with non-strict constraints. These challenges have impeded the development of more advanced approaches. To address these issues, we introduce fractional Langevin dynamics (FLD) for CO, replacing Gaussian noise with α-stable Lévy noise. FLD can escape from local optima more readily via Lévy flights, and in multiple-peak CO problems with high potential barriers it exhibits a polynomial escape time that outperforms the exponential escape time of LD. Moreover, FLD coincides with LD when $\alpha = 2$, and by tuning α it can be adapted to a wider range of complex scenarios in the CO field. We provide theoretical proof that our method offers enhanced exploration capabilities and improved convergence. Experimental results on the Maximum Independent Set, Maximum Clique, and Maximum Cut problems demonstrate that incorporating FLD advances both sampling-based and data-driven approaches, achieving state-of-the-art (SOTA) performance in most of the experiments. The codes are publicly available at https://github.com/Thinklab-SJTU/FLD4CO.

1 Introduction

Combinatorial optimization (CO) problems, which involve finding an optimal solution from a finite set of possible configurations subject to a set of constraints, are of paramount importance and usefulness across fields, e.g. logistics [45], scheduling [59], network design [4], and finance [39].

There has been growing interest in developing efficient algorithms for obtaining high-quality sub-optimal solutions. Among these efforts, sampling-based methods have shown considerable promise due to their simplicity, ability to balance speed and solution quality, and training-free property. A fundamental approach is simulated annealing (SA) [28], which uses random local fluctuations guided by Metropolis-Hastings updates [36, 23] and probabilistically explores the solution space. Recent work [51] has demonstrated that incorporating Langevin dynamics (LD) and its discrete proposal [69, 50] can vastly improve the sampling efficiency, thereby advancing sampling-based approaches for CO. The core idea of LD is to leverage the gradient to guide the sampling in each iteration, resulting in a more efficient searching process. However, there are certain limitations

*Correspondence author. † denotes equal contribution. This work was partly supported by National Key R&D Program of China (Nos. 2021YFA1000300 and 2021YFA 100302) and National Natural Science Foundation of China (No. 12331014, 62222607).

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

associated with LD. Firstly, it relies on Gaussian noise as a random perturbation, with the step size being coupled to the noise amplitude. As the Gaussian noise decays exponentially at the tail, reducing the step size also diminishes the noise, making it challenging to escape from 'deep and narrow' local optima. The time required to escape local minima grows exponentially with the energy barrier height. Moreover, to maintain sample diversity, LD necessitates parallel independent chains, which can be computationally expensive. Furthermore, Gaussian noise assumes a locally smooth energy landscape, meaning that LD is less effective in scenarios where the energy function is rugged or when non-strict constraints are present. These factors limit the effectiveness of LD in more complex optimization landscapes and the development of more advanced sampling-based approaches for CO.

Another trend is the data-driven approach to learning for optimization. Early neural network (NN)-based methods [24] primarily relied on supervised learning [33, 18, 21]. Subsequent works have explored reinforcement learning [62, 60, 61] and unsupervised learning [27, 57, 52] techniques to address the challenge of collecting labeled training data. More recently, diffusion models have been introduced to the CO domain [31, 53, 44, 43], demonstrating superior performance and promising potential. These approaches also implicitly incorporate LD, as diffusion models were initially derived from LD in the field of image generation. Unlike sampling-based methods, NN-based approaches eliminate the need for explicit gradients of the problem, thereby enabling unification for a variety of problems without relying on the problem structure, utilizing the network's automatic differentiation capabilities. We leave detailed related works in Appendix 2.

In this paper, we introduce fractional Langevin dynamics (FLD) to address the propensity of conventional LD to become trapped in local optima. We incorporate symmetric α-stable ($\mathcal{S}\alpha\mathcal{S}$) noise with truncation into FLD: unlike Gaussian perturbations, $\mathcal{S}\alpha\mathcal{S}$ noise exhibits heavy-tailed jumps (Lévy flights), enabling instantaneous energy-barrier leaps that facilitate escape from local minima. Moreover, by setting $\alpha = 2$, FLD reduces to standard LD, thus retaining efficient exploration in smoother or strongly constrained settings. We propose the $\mathcal{S}\alpha\mathcal{S}$-noise FLD sampling process and present both explicit- and implicit-gradient formulations to advance both sampling-based and data-driven approaches. We adopt the mean escape time as our convergence metric, and derive theoretical upper bounds in the discrete setting, showing a polynomial-time bound for FLD versus an exponential bound for LD. Through comparative case studies on three prototypical CO problems, our methods outperform existing sampling-based and data-driven methods. Additionally, extensive sampling-trajectory experiments have been conducted to vividly illustrate the enhanced escape ability of FLD, demonstrate the impact of varying α on escape performance, and confirm the effectiveness of our truncation strategy. Finally, we perform ablation studies on the best energy-function values over iterations, thereby validating superior convergence and exploration capabilities of FLD.

2 Related Work

Data-driven Approaches for CO. They involve training NN models for CO, commonly referred to as neural solvers. Significant efforts have been made to explore supervised learning [33, 18, 21, 53, 31, 58, 32, 30, 34, 35], unsupervised learning [27, 57, 52, 56, 44, 43, 20], and reinforcement learning [40, 17]. Our FLD-IG integrates FLD with a simple reinforcement learning-based approach. FLD-IG achieves competitive performance with a simple architecture and minimal training resources, resulting in faster convergence and improved training efficiency.

Sampling-based Approaches for CO. Sampling-based approaches have been widely utilized for CO [36, 23, 37, 10, 65, 46]. However, these previous approaches are generally less efficient than data-driven approaches. Recent work by [51] has advanced sampling-based methods, achieving comparable or even superior performance to data-driven approaches by introducing the discrete LD proposal[69, 50]. [17] further develops a regularized approach on discrete LD, resulting in improved performance. Our FLD-EG enhances the sampling-based approach by integrating FLD, which can be seen as a generalization of vanilla LD, leading to faster convergence and better performance.

3 Preliminaries

Energy-Based Model (EBM). It defines an energy function $H : \mathcal{S} \to \mathbb{R}$ with the target distribution:

$$p_\tau(x) : \frac{e^{-H(x)/\tau}}{Z} \tag{1}$$

where $\mathcal{S}$ represents the energy state space, τ is a temperature parameter controlling the smoothness of the distribution, and $Z = \sum_{x \in \mathcal{S}} e^{-H(x)/\tau}$ is the partition function in statistical physics or normalization factor in probability theory.

Markov Chain Monte Carlo. Markov chain Monte Carlo (MCMC) techniques [63, 54], which are grounded in continuous diffusion processes, have gained widespread popularity owing to their demonstrated success in large-scale Bayesian machine learning [11]. The goal of the MCMC is to generate samples from a target distribution p_τ, by forming a continuous diffusion that has p_τ as a stationary distribution. Given a current state $x_t \in \mathcal{S}$, a Metropolis-Hastings (MH) sampler [36, 23] proposes a candidate state $y \in \mathcal{S}$ from a proposal distribution $g(y \mid x_t)$. Then calculate the Metropolis acceptance ratio:

$$A(y, x_t) = \min \left\{ 1, \frac{p_\tau(y) g(x_t \mid y)}{p_\tau(x_t) g(y \mid x_t)} \right\} \tag{2}$$

With generating a random number $u \sim U(0, 1)$, where $U(0, 1)$ is the uniform distribution within $[0, 1]$, if $u \leq A(y, x_t)$, then the proposal state is accepted and set $x_{t+1} = y$; otherwise, set $x_{t+1} = x_t$.

Langevin Dynamics. Langevin dynamics (LD) is an MCMC algorithm that has also been incorporated in combinatorial optimization algorithms for better exploring the landscape of the energy function $H(x)$ [47]. LD methods are based on constructing stochastic differential equations (SDEs) equipped with Brownian motion (shown as Eq. (3)), assuming that the particle is driven by an infinite number of small forces with finite variance.

$$dx_t = s(x_t)dt + \sqrt{2} dB_t \tag{3}$$

where B_t denotes the standard Brownian motion and $s(\cdot) = \nabla \log p_\tau(\cdot) = -\frac{1}{\tau} \nabla H(x)$ represents the score function of EBM. With the condition of sampling state x_t can be shown to be ergodic with $p_\tau(x_t)$. The samples can be generated from p_τ by simulating the sampling process of continuous space discrete space [41], which is given by using a first-order Euler-Maruyama discretization:

$$x_{n+1} = x_n + \eta_{n+1} s(x_n) + \sqrt{2\eta_{n+1}} \Delta B_{n+1} \tag{4}$$

where η_n denotes the step size of the sampling iteration and $\Delta B_n = \xi$ is an i.i.d. standard Gaussian random variable, $\xi \sim \mathcal{N}(0, I_{N \times N})$ when the state space $\mathcal{S} = \mathbb{R}^N$ [69].

Simulated Annealing. Simulated annealing (SA) is a variant of local search [14] that explores the solution landscape with probabilistic relaxation. As the temperature decreases, there is a tendency to sample points on the landscape to make the energy function $H(x)$ value smaller; when the temperature equals zero, the solution x will stop at the point where the $H(x)$ has the lowest value (that is, the solution obtained by the SA algorithm converges to the global optimum in probability) [55].

4 Methodology

4.1 Problem Formulation

Without loss of generality, we formulate a CO problem as follows:

$$\min_{x \in \mathcal{S} = \{0,1\}^N} a(x), \text{ s.t. } b(x) = 0 \tag{5}$$

where the solution landscape $\mathcal{S}$ is an N-dimensional vector such that each dimension takes a discrete value from $\{0, 1\}$, which is the most challenging to deal with, although it will be possible to extend.

To recast a constrained optimization problem as a sampling task, a penalty function (generally treated as the energy function of EBM) takes the form:

$$H(x) = a(x) + \lambda b(x) \tag{6}$$

where λ is the penalty factor of the constraints. Furthermore, the attempt to directly sample from $p_\tau(x)$ with the small τ makes the energy landscape highly nonsmooth; a common remedy is to incorporate the SA algorithm, progressively lowering τ toward zero as the chain evolves.

4.2 Fractional Langevin Dynamics

By Eq. (4), it can be seen that the term $(x_{n+1} - x_n - \eta_{n+1} s(x_n))/\sqrt{2\eta_{n+1}}$ follows a Gaussian. Thus the transition probability $q(x_{n+1} \mid x_n)$ in the LD algorithm can be interpreted as a Gaussian with mean $x_n + \eta_{n+1} s(x_n)$ and covariance $2\eta_{n+1} I_{N\times N}$ [69]. The discrete (gradient-based) proposal distribution with the explicit domain $\mathcal{S} = \mathbb{R}^N$ of LD is:

$$q(x_{n+1} \mid x_n) = \frac{\exp\left(-\frac{1}{4\eta_{n+1}} \|x_{n+1} - x_n - \eta_{n+1} s(x_n)\|_2^2\right)}{Z_{\mathbb{R}^N}(x_n)} \tag{7}$$

where,

$$Z_{\mathbb{R}^N}(x_n) = \sum_{x_{n+1}\in\mathbb{R}^N} \exp\left(-\frac{1}{4\eta_{n+1}} \|x_{n+1} - x_n - \eta_{n+1} s(x_n)\|_2^2\right) = (4\pi\eta_{n+1})^{N/2} \tag{8}$$

Thus, it can be factorized coordinate-wise into a set of simple categorical distributions:

$$q(x_{i,n+1} \mid x_n) \propto \exp\left\{\frac{1}{2} s(x)_i (x_{i,n+1} - x_{i,n} - \frac{(x_{i,n+1} - x_{i,n})^2}{\eta_{n+1}})\right\} \tag{9}$$

with

$$q(x_{n+1} \mid x_n) = \prod_{i=1}^{N} q(x_{i,n+1} \mid x_n) \tag{10}$$

Typically, the iteration step size $\eta_n = C$, where C is a constant, is too stable due to the combination of a fixed step size and Gaussian noise. This stability makes it difficult for LD to escape from local optima on the energy surface during the iterative process, often causing the trajectory to remain trapped near suboptimal solutions, thereby significantly degrading the quality of the final result. To address this issue, we introduce the α-stable Lévy noise [67]—a type of stochastic process with a heavy-tailed distribution. Unlike Gaussian noise, α-stable Lévy noise allows for occasional large jumps (Lévy flights), which increases the probability of escaping local optima and thus improves the exploration capability of the algorithm.

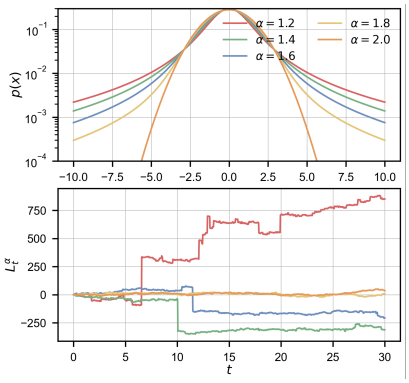


Figure 1: Pdf of $\mathcal{S}\alpha\mathcal{S}$ distribution and α-stable Lévy motion.

In this work, we are interested in the centered symmetric α-stable ($\mathcal{S}\alpha\mathcal{S}$) distribution, which is a special case of α-stable distribution. The definition of $\mathcal{S}\alpha\mathcal{S}$ random variables and $\mathcal{S}\alpha\mathcal{S}$ Lévy motion are shown as:

Definition 1 ($\mathcal{S}\alpha\mathcal{S}$ random variables [42]). *The α-stable distribution arises as the limiting distribution in the generalized Central Limit Theorem (CLT). A scalar random variable $x \in \mathbb{R}$ is said to follow a $\mathcal{S}\alpha\mathcal{S}$ distribution if its characteristic function takes the following form:*

$$\mathbb{E}[\exp(iwx)] = \exp(-|\sigma w|^\alpha) \tag{11}$$

Here, $\alpha \in (0, 2]$ is the characteristic exponent, which controls the tail heaviness of the distribution: smaller values of α result in heavier tails (shown in Figure 1). The parameter $\sigma \in \mathbb{R}^+$ is the scale parameter, reflecting the dispersion of x around zero.

Definition 2 ($\mathcal{S}\alpha\mathcal{S}$ Lévy motion [15]). *A scalar symmetric α-stable Lévy motion L_t^α, with $\alpha \in (0, 2]$, is a stochastic process satisfying the following properties:*

1. *$L_0^\alpha = 0$ almost surely.*
2. *For $t_0 < t_1 < \cdots < t_N$, the increments $(L_{t_n}^\alpha - L_{t_{n-1}}^\alpha)$ $(n = 1, 2, \cdots, N)$ are independent.*
3. *The $(L_t^\alpha - L_s^\alpha)$ and L_{t-s}^α have the same distribution $\mathcal{S}\alpha\mathcal{S}((t-s)^{1/\alpha})$ $(0 \le s < t)$.*

4. L_t^α has stochastically continuous sample paths (that is, continuous in probability):

$$\lim_{t \to s} \mathbb{P}(|L_t^\alpha - L_s^\alpha| > \delta) = 0, \ \forall \delta > 0, s \geq 0. \tag{12}$$

Similarly to the $\mathcal{S}\alpha\mathcal{S}$ distributions, the $\mathcal{S}\alpha\mathcal{S}$ Lévy motions L_t^α coincide with a scaled Brownian motion $\sqrt{2}B_t$ when $\alpha = 2$. Thus, the $\mathcal{S}\alpha\mathcal{S}$ distribution can be seen as a heavy-tailed generalization of the centered Gaussian distribution. As an important special case of $\mathcal{S}\alpha\mathcal{S}$, we obtain the Gaussian distribution $\mathcal{S}\alpha\mathcal{S}(\sigma) = \mathcal{N}(0, 2\sigma^2)$ for $\alpha = 2$.

The fractional Langevin dynamics (FLD) framework is driven by the $\mathcal{S}\alpha\mathcal{S}$ Lévy-based SDE as [47]:

$$dx_t = b(x_{t-}, \alpha)dt + dL_t^\alpha \tag{13}$$

where, $b(\cdot)$ denotes the drift function shown in Theorem 1, x_{t-} represents the left limit of the process at time t, and L_t^α is the standard $\mathcal{S}\alpha\mathcal{S}$ Lévy motion shown as Definition 2.

Theorem 1 ([47])**.** *The drift function of the SDE (13) is defined below:*

$$b(x, \alpha) \triangleq \frac{\mathcal{D}^{\alpha-2} f_{p_\tau}(x)}{\phi(x)} \triangleq c_\alpha s(x) \tag{14}$$

where, $f_{p_\tau}(x) \triangleq -\phi(x)\partial_x H(x)$, fractional integration $\mathcal{D}^{\alpha-2} \triangleq \mathcal{F}^{-1}\{|w|^{\alpha-2}\mathcal{F}(f_{p_\tau}(x))\}$, $\phi(x) = \exp\{-H(x)\}$, $c_\alpha \triangleq \Gamma(\alpha-1)/\Gamma(\alpha/2)^2$, and $\mathcal{F}$ is the Fourier transforms.

Detailed derivation and proof can be found in Appendix A.

Proposition 1. *The FLD-based SDE is the generalization of the LD-based SDE.*

Proof. When $\alpha = 2$, the SDE driven by the FLD:

$$dx_t = b(x_{t-}, 2)dt + dL_t^\alpha(\alpha \to 2) \approx c_2 s(x_t)dt + dL_t^\alpha(\alpha \to 2) \approx s(x_t)dt + \sqrt{2}dB_t \tag{15}$$

Thus, the FLD-based SDE reduces to the LD-based SDE when $\alpha = 2$, while for $\alpha \neq 2$, the FLD-based SDE exhibits heavier tails. □

Combining the Theorem 1 and Eq. (13), the approximate $\mathcal{S}\alpha\mathcal{S}$ Lévy-based SDE and the first-order Euler-Maruyama discretized $\mathcal{S}\alpha\mathcal{S}$ sampling process can be obtained as:

$$dx_t = c_\alpha s(x_t)dt + dL_t^\alpha x_{n+1} = \quad x_n + \eta_{n+1} c_\alpha s(x_n) + \eta_{n+1}^{1/\alpha} \triangle L_{n+1}^\alpha \tag{16}$$

The Proposition 1 also demonstrates that the target distribution can be sampled more accurately by adaptively adjusting α during the sampling process. In regions where the energy function is locally smooth or tightly constrained, setting $\alpha = 2$ enables efficient sampling. Conversely, when the sampling process becomes trapped in a local optimum, decreasing α increases the probability of Lévy flights, thereby facilitating escape from the local minimum.

Moreover, although the probability density function (pdf) of the $\mathcal{S}\alpha\mathcal{S}$ distribution does not have a closed-form expression, it is straightforward to generate random samples from stable distributions when $\alpha \neq 2$. The sampling of $\mathcal{S}\alpha\mathcal{S}$ is given by Theorem 2 with $\beta = 0$ by the Chambers-Mallows-Stuck method, which is shown in Theorem 3.

Theorem 2. *Let γ be uniformly distributed on $(-\frac{\pi}{2}, \frac{\pi}{2})$ and W be an independent exponential random variable with mean 1. The α-stable sampling is:*

$$Z = \begin{cases} \dfrac{\sin \alpha(\gamma - \gamma_0)}{(\cos \gamma)^{1/\alpha}} \left(\dfrac{\cos(\gamma - \alpha(\gamma - \gamma_0))}{W} \right)^{(1-\alpha)/\alpha} \triangleq \mathcal{S}_\alpha(1, \beta, 0), \ \alpha \neq 1 \\ (\dfrac{\pi}{2} + \beta\gamma)\tan\gamma - \beta \log\left(\dfrac{W \cos\gamma}{\frac{\pi}{2} + \beta\gamma} \right) \triangleq \mathcal{S}_1(1, \beta, 0), \ \alpha = 1 \end{cases} \tag{17}$$

where, $\gamma_0 = -\frac{\pi\beta K(\alpha)}{2\alpha}$, $K(\alpha) = \alpha - 1 + \text{sign}(1-\alpha)$, and $\text{sign}(\cdot)$ denotes the sign function.

Theorem 3. *Let γ be uniformly distributed on $(-\frac{\pi}{2}, \frac{\pi}{2})$ and W be an independent exponential random variable with mean 1. The $\mathcal{S}\alpha\mathcal{S}$ sampling is:*

$$Z = \begin{cases} \dfrac{\sin \alpha\gamma}{(\cos\gamma)^{1/\alpha}} \left(\dfrac{\cos(\gamma - \alpha\gamma))}{W} \right)^{(1-\alpha)/\alpha} \triangleq \mathcal{S}\alpha\mathcal{S}(1), \ \alpha \neq 1 \\ \dfrac{\pi}{2}\tan\gamma \triangleq \mathcal{S}1\mathcal{S}(1), \ \alpha = 1 \end{cases} \tag{18}$$

Since $\mathcal{S}\alpha\mathcal{S}$ distributions are a special case of α-stable distributions, the detailed proof of Theorem 2 and Theorem 3 is presented together in Appendix B. Thus the discrete sampling process can be rewritten as [48], where $z_{n+1} \sim \mathcal{S}\alpha\mathcal{S}(1)$:

$$x_{n+1} = x_n - \frac{\eta_{n+1} c_\alpha}{\tau} \nabla H(x) + \eta_{n+1}^{1/\alpha} z_{n+1} \tag{19}$$

4.3 Comparative Analysis of Convergence

We compare the convergence capabilities of LD and FLD by analyzing the escape time from the local minima, which is defined as follows [5]:

Definition 3 (Escape Time)**.** *The escape time is a random variable:*

$$\tau_{y^*}^{x^*} = \inf\{t > 0 : x_t \in \mathcal{B}_\delta(y^*)\} \tag{20}$$

where, $x_0 = x^*$, two points x^* and y^* separately represent local minima under the assumption that the potential energy H has several (at least two) local minima, and $\mathcal{B}_\delta(y^*)$ denotes the ball of radius δ centered in y^*.

Under the low noise intensity ϵ, the LD-based SDE can be rewritten as:

$$dx_t = -\nabla H(x_t)dt + \sqrt{2\epsilon} dB_t \tag{21}$$

By Eyring–Kramers law, the mean escape time of LD-based SDE in the continuous space $\mathcal{S} \in \mathbb{R}^N$ is:

$$\mathbb{E}[\tau_{y^*}^{x^*}] \approx \frac{2\pi}{\lambda(z^*)} \sqrt{\frac{|\det(\nabla^2 H(z^*))|}{\det(\nabla^2 H(x^*))}} \exp\{(H(z^*) - H(x^*))/\epsilon\} \propto \exp\{(H(z^*) - H(x^*))/\epsilon\} \tag{22}$$

where z^* is a unique saddle (that is the maximum of the potential energy barrier) and $\lambda(\cdot)$ denotes the single negative eigenvalue of the Hessian matrix $\nabla^2 H(\cdot)$.

The mean escape time of FLD-based SDE [25] in the continuous space $\mathcal{S} \in \mathbb{R}^N$ is:

$$\mathbb{E}[\tau_{y^*}^{x^*}] \propto w^\alpha \tag{23}$$

where w denotes the "width" of the local minima to the boundary of a potential well.

Similarly, we provide the discrete proposal for the escape time of both LD and FLD. We state upfront that the Markov chains of LD and FLD are reversible if they satisfy the detailed balance conditions. Additionally, $p_\tau(x)$ is a positive stationary distribution, given that the symmetric proposal and the Metropolis-Hastings acceptance criterion are satisfied for constructing discrete LD and FLD. Thus, when the state space is a finite or countable set $\mathcal{S} = \{0, 1\}^N$, the symmetric Dirichlet form:

$$D_\alpha(f, f) = \frac{1}{2} \sum_{x,y \in \mathcal{S}} (f(x) - f(y))^2 p_\tau(x) P_\alpha(x, y) \tag{24}$$

where $P_\alpha(x, y)$ represents the transition matrix. Then, the conductance of an arbitrary non-empty truth subset $\mathcal{B} \subset \mathcal{S}$ is:

$$\Phi_\alpha(\mathcal{B}) = \inf_{\mathcal{B}} \frac{D_\alpha(\mathbb{1}_\mathcal{B}, \mathbb{1}_\mathcal{B})}{p_\tau(\mathcal{B})} = \frac{1}{p_\tau(\mathcal{B})} \sum_{x \in \mathcal{B}} \sum_{y \notin \mathcal{B}} p_\tau(x) P_\alpha(x, y) \tag{25}$$

The first non-trivial eigenvalue given by the Cheeger inequality of LD [29, 49] is shown as follows:

$$\lambda_1 \geq \frac{(\Phi_2(\mathcal{B}))^2}{2} \tag{26}$$

By the same reasoning, the first non-trivial eigenvalue of FLD [2, 12] is:

$$\lambda_1^\alpha \geq C_{N,\alpha} \Phi_\alpha(\mathcal{B}) = \frac{\alpha 2^{\alpha-1} \Gamma(\frac{N+\alpha}{2})}{\pi^{N/2} \Gamma(1 - \frac{\alpha}{2})} \Phi_\alpha(\mathcal{B}) \tag{27}$$

Table 1: Results of compared methods for MIS problem.

MIS		RB-[200–300]		RB-[800–1200]		ER-[700–800]		ER-[9000–11000]	
Method	**Type**	**Size** ↑	**Time** ↓	**Size** ↑	**Time** ↓	**Size** ↑	**Time** ↓	**Size** ↑	**Time** ↓
Gurobi	OR	19.98	47.57 m	40.90	2.17 h	41.38	50.00 m	—	—
KaMIS	OR	20.10	1.40 h	43.15	2.05 h	44.87	52.13 m	381.31	7.60 h
DGL	SL	17.36	12.78 m	34.50	23.90 m	37.26	22.71 m	—	—
INTEL	SL	18.47	13.07 m	34.47	20.28 m	34.86	6.06 m	284.63	5.02 m
DIFUSCO	SL	18.52	16.05 m	—	—	41.12	26.67 m	—	—
LTFT	UL	19.18	32 s	37.48	4.37 m	—	—	—	—
DiffUCO	UL	19.24	54 s	38.87	4.95 m	—	—	—	—
SDDS	UL	19.62	20 s	39.99	6.35 m	—	—	—	—
PPO	RL	19.01	1.28 m	32.32	7.55 m	—	—	—	—
DIMES	RL	—	—	—	—	42.06	12.01 m	332.80	12.72 m
RLNN	PRL	19.52	1.64 m	38.46	6.24 m	43.34	1.37 m	363.34	11.76 m
iSCO	H	19.29	2.71 m	36.96	11.26 m	42.18	1.45 m	365.37	1.10 h
RLSA	H	19.97	35 s	40.19	1.85 m	44.10	20 s	375.31	1.66 m
FLD-IG	PRL	19.72	1.08 m	39.56	6.31 m	43.50	1.35 m	365.03	11.41 m
FLD-EG	H	**20.02**	38 s	**40.25**	1.93 m	**44.37**	19 s	**377.50**	1.12 m

Derived via spectral expansion, the upper bound of mean escape time for LD and FLD is $\frac{2}{(\Phi_2(\mathcal{B}))^2}$ and $\frac{1}{C_{N,\alpha}\Phi_\alpha(\mathcal{B})}$. Due to the $\Phi_2(\mathcal{B}) \propto \exp\{-\triangle H\}$ [8, 13] and $\Phi_\alpha(\mathcal{B}) \propto w^{-\alpha}$ [26], as the potential barrier $\triangle H$ increases linearly, the escape time of the LD increases exponentially, making it prone to becoming "trapped" at high potential barriers. In contrast, FLD-based SDE, the mean escape time is no longer governed exponentially by the barrier height $\triangle H$ but is instead primarily influenced by the barrier width w in a polynomial manner. Consequently, in multiple-peak landscapes with high potential barriers, FLD-based sampling exhibits superior convergence properties compared to LD.

4.4 Enhanced Sampling-Based Approach: FLD-EG

We aim to enhance sampling-based approaches by introducing our **FLD-EG** (i.e., with explicit gradient). As discussed in Sec. 4.3, FLD exhibits a stronger ability to escape from local minima in multiple-peak and high-barrier CO landscapes compared to LD. Motivated by this, we employ FLD-based sampling to guide the assignment of variable values at each iteration. To further enhance stability and reduce the impact of outlier samples that may hinder local exploration, we apply a truncation scheme to remove extreme samples, as illustrated in Fig. 2b. This leads to a more stable and consistent sampling trajectory. Also, inspired by [17], we update only the top-d variables that have the greatest influence on $\nabla H(x)$ and determine the variable values based on the result of the sampling iteration. Specifically, a truncated version of the drift term $\frac{\eta_{n+1}c_\alpha \nabla H(x)}{\tau}$ is applied, guided by a top-d gradient indicator defined as $\text{Sigmoid}((\frac{1}{2\tau}((2x-1)\odot\nabla H(x))_i - ((2x-1)\odot\nabla H(x))_{(d)})$. Similarly, the $\mathcal{S}\alpha\mathcal{S}$ noise is truncated based on a top-d_{noise} noise indicator: $\text{Sigmoid}((\frac{1}{2\tau}((2x-1)\odot \nabla H(x))_i - ((2x-1)\odot\nabla H(x))_{(d_{\text{noise}})})$. The final update rule is given in Eq. (19). Details on the algorithmic process can be found in Appendix D.1

Consistent with standard sampling-based approaches, FLD-EG requires a closed-form gradient of the energy function for the CO problems it addresses. Case studies on the energy functions of applied CO problems are presented in Appendix C.

4.5 Enhanced Data-Driven Approach: FLD-IG

Since the gradient $\nabla H(x)$ is not available for all CO problems, we propose a data-driven implicit-gradient FLD solver named **FLD-IG**. Inspired by [27] and [17], we introduce an NN-based framework whose training procedure resembles reinforcement learning, alternating between sampling and update steps (we denote this framework as PRL). We introduce the concept of flip probability for variables, as proposed by [17], to mitigate numerical issues, and the network is designed to predict these flip probabilities. Additionally, $\mathcal{S}\alpha\mathcal{S}$ noise is incorporated into the flip decisions at each iteration. To match the FLD process, we first apply a linear transformation to rescale the noise from the range

Table 2: Results of compared methods for MaxCl and MaxCut problems.

MaxCl		RB-[200–300]		RB-[800–1200]		MaxCut		BA-[200–300]		BA-[800–1200]	
Method	**Type**	**Size ↑**	**Time ↓**	**Size ↑**	**Time ↓**	**Method**	**Type**	**Size ↑**	**Time ↓**	**Size ↑**	**Time ↓**
Gurobi	OR	19.05	1.92 m	33.89	19.67 m	Gurobi	OR	730.87	8.50 m	2944.38	1.28 h
ERDOES	UL	12.02	41 s	25.43	2.27 m	ERDOES	UL	693.45	46 s	2870.34	2.82 m
LTFT	UL	16.24	42 s	31.42	4.83 m	LTFT	UL	704.30	2.95 m	2864.61	21.33 m
DiffUCO	UL	16.22	1.00 m	—	—	DiffUCO	UL	727.32	1.00 m	2947.53	3.78 m
SDDS	UL	18.90	38 s	—	—	SDDS	UL	731.93	14 s	**2971.62**	1.08 m
RLNN	PRL	18.13	1.36 m	35.23	7.83 m	RLNN	PRL	729.00	1.58 m	2907.18	3.67 m
Greedy	H	13.53	25 s	26.71	25 s	Greedy	H	688.31	13 s	2786.00	3.12 m
MFA	H	14.82	27 s	27.94	2.32 m	MFA	H	704.03	1.60 m	2833.86	7.27 m
iSCO	H	18.96	54 s	40.35	11.37 m	iSCO	H	728.24	1.67 m	2919.97	4.18 m
RLSA	H	**18.97**	23 s	40.53	1.27 m	RLSA	H	733.54	27 s	2955.81	1.45 m
FLD-IG	PRL	18.52	1.40 m	37.40	6.89 m	FLD-IG	PRL	733.48	1.57 m	2922.54	3.07 m
FLD-EG	H	**18.97**	20 s	**40.63**	1.91 m	FLD-EG	H	**734.18**	25 s	2960.13	1.70 m

$[0, 1]$ to $[-1, 1]$ by multiplying $1 - 2x$. The training loss function is defined as:

$$l(\theta; \mathbf{x}, d, \lambda') = \mathbb{E}_{q_\theta(\mathbf{x}_{n+1}|\mathbf{x}_n)}[H(\mathbf{x}_n)] + \lambda' \left[\sum_{i \in V} q_\theta(\mathbf{x}_{i,n+1}|\mathbf{x}_n) - d \right]^2 \tag{28}$$

where $q_\theta(\mathbf{x}_{i,n+1}|\mathbf{x}_n)$ still satisfies the mean-field decomposition Eq. (10). Details on the network architecture and the training and inference process can be found in Appendix D.2.

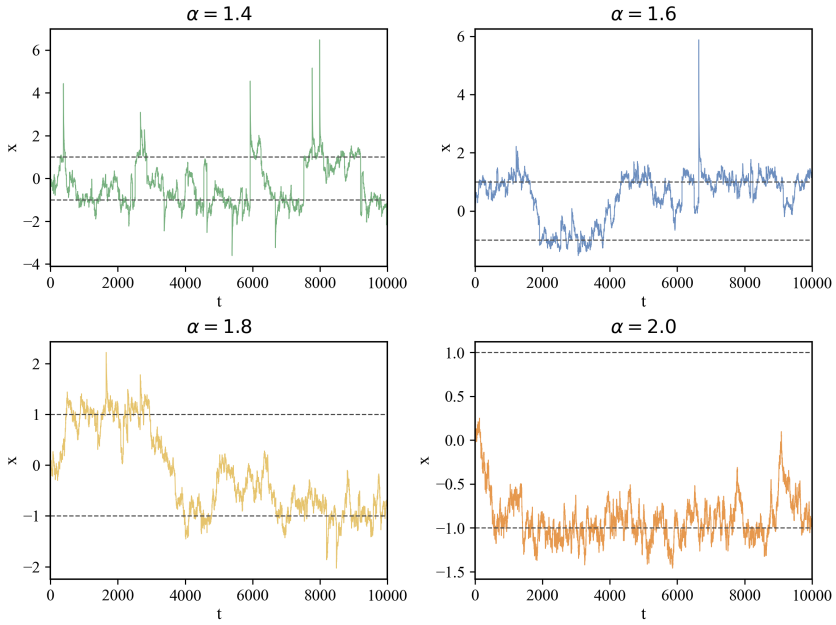
5 Experiments

We evaluate our FLD-EG and FLD-IG on three common CO problems, including maximum independent set (**MIS**), maximum clique (**MaxCl**) and max cut (**MaxCut**) problems. Furthermore, we demonstrate more analysis on the sampling trajectories and ablation studies.

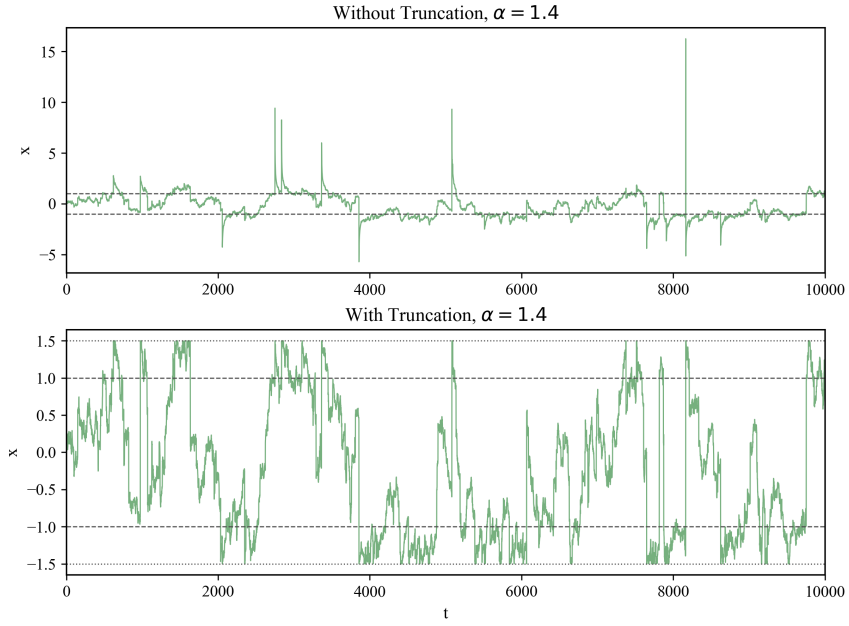
Datasets. (1) MIS datasets: Following the benchmarks in [17], we evaluate our algorithms on two graph classes: Revised Model B (RB) instances [66] and Erdős–Rényi (ER) random graphs [16] with node weight set to 1; (2) Maximum Clique dataset: we use the single RB graph which is introduced in MIS datasets for the evaluation; (3) Max Cut dataset: we compare our algorithms with the baselines on the Barabási-Albert (BA) graphs [3]. An additional point that warrants special attention is: The size of the training set and the validation set is separately 1000 and 500 graphs for all datasets except for ER-[9000, 11000] (that is, the ER graphs contain 9000 to 11000 nodes), and the test size is 500 for RB and BA graphs; 128 for ER-[700-800] and 16 for ER-[9000, 11000].

Baselines. (1) Classical methods: we categorize them into two types: operation research (OR) methods; heuristic (H) methods. In the OR type, we give the general integer linear programming representation of MIS, Maximum Clique, and Max Cut, solved by the Gurobi solver [22] as the OR baseline; especially for MIS, we additionally give the MIS problem-specific solver KAMIS [19]. For the heuristic methods, we give two sampling methods iSCO [51] and RLSA [17] with SA and discrete proposal of LD for all cases; additionally, the Greedy and MFA (Mean-Field Annealing) [6] methods are provided for Maximum Clique and Max Cut problems. (2) Learning-based methods: we classify the learning-based methods into four types: supervised learning (SL), unsupervised learning (UL), reinforcement learning (RL), and pseudo reinforcement learning (PRL) which uses the sample and update scheme similar to the RL. For SL, the INTEL with GCN and probability heatmap [33], DGL with Monte Carlo Tree Search and two GNN backbones [7], and DIFUSCO with UNet-Style diffusion model [53] are given as SL baselines for the MIS problem. In the type of UL, the LIFT with GFlownets [68], DiffUCO with UNet-Style diffusion model [44], and SDDS with discrete diffusion models [43] are presented for three case studies; beyond that, ERDOES with random graph model [16] is set for the case studies without the MIS problem. The RL methods, PPO with actor-and-critic [1] and DIMES with reinforcement optimization combined with meta-learning [40], are introduced for the MIS problem. Similar to the FLD-IG, RLNN with the discrete proposal of LD is the baseline method of the PRL for three case studies [17].

Main Results. In the main experiments for the evaluation of our FLD-EG and FLD-IG, we give two metrics: the objective value of each problem and the overall sequential testing time, which



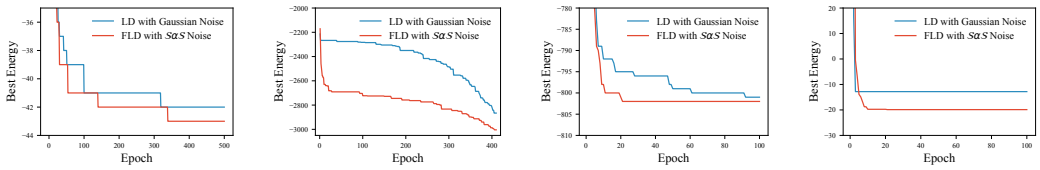
(a) Typical trajectories for $\alpha = 1.4, 1.6, 1.8, 2.0$.



(b) Trajectories w/ and w/o truncation for $\alpha = 1.4$.

Figure 2: Sampling trajectories of the FLD-based SDE.

attracted the main concern in the field of CO. For three case studies in this work, the objective values separately represent the independent set size of MIS, the clique size of Maximum Clique, and the cut size of the Max Cut (the detailed formation can be seen in Appendix C). The reported results for each learning-based method correspond to the longest runtimes and, accordingly, should also exhibit the highest objective values; with regard to heuristic methods, we fixed the number of iterations to be the same. Table 1 and 2 demonstrate the results on MIS, MaxCl and MaxCut. On most problems, our FLD-EG and FLD-IG outperform the SOTA classical and learning-based methods, respectively, achieving higher objective values in shorter or comparable runtimes. For the classical methods, since OR methods can obtain the optimal solution given sufficient runtimes, we report the results of



(a) ER-[700-800]. (b) BA-[800-1200]. (c) BA-[200-300]. (d) RB-[200-300], MaxCl.

Figure 3: Ablation for our methods (FLD with $\mathcal{S}\alpha\mathcal{S}$) and LD sampling process method with Gaussian noise. The staircase curves show how the "Best Energy" evolves as the number of iterations ("epochs"), where the "Best Energy" is the minimum energy function value between last and current epoch.

OR methods solely to demonstrate the performance gap between the heuristic and learning-based methods relative to the optimum, rather than to make a direct comparison. In the competitive heuristic methods (i.e. iSCO, RLSA and our FLD-EG), our FLD-EG can attain equal or superior objective values within the same sampling steps, while maintaining comparable or slightly reduced runtime. In the field of learning-based methods, our FLD-IG achieves higher objective values than the others on 75% of datasets; among methods with comparable performance metrics, our FLD-IG achieves shorter runtimes on all datasets except the RB-[800-1200] instance.

Sampling Trajectories. We conduct FLD-based iterative sampling under different values of α to simulate the trajectory of a single variable x during CO solving. Owing to the symmetry of the $\mathcal{S}\alpha\mathcal{S}$, we shift and scale x from [0, 1] to [-1, 1] in our methods. As shown in Fig. 2a, when $\alpha = 2$ (i.e., the process degenerates to LD sampling), x readily becomes trapped in local minima, leading to slow convergence; as α decreases—intensifying the heavy tail of the $\mathcal{S}\alpha\mathcal{S}$ distribution—x flips more frequently between –1 and 1, promoting escape from local minima and accelerating convergence (theoretical justification is provided in Sec. 4.3). However, for very small α, the generation of large outliers can cause sampled points to deviate excessively, losing track of the underlying landscape. To remedy this, we introduce a truncation strategy. As shown in Fig. 2b, bounding the sampled points within a prescribed range yields a markedly more stable sampling process.

Ablation Studies. To rigorously validate the effectiveness of our $\mathcal{S}\alpha\mathcal{S}$-noise sampling process, we conducted ablation studies comparing its convergence behavior against that of LD with Gaussian noise when solving CO problems (cf. Sec. 4.3). Specifically, in both FLD-EG and FLD-IG, we replaced the FLD sampling process driven by $\mathcal{S}\alpha\mathcal{S}$ noise with the LD sampling process driven by Gaussian noise, and designed comparative experiments to monitor the iterative descent of the energy function. As shown in Fig. 3, two panels on the left depict the ablation studies for FLD-EG, while the others depict the ablation studies for FLD-IG. Notably, since the optimal solution is not attained, there is a gap between the best energy function value and the current objective value, which corresponds to the penalty term imposed by the constraints (i.e. $\lambda b(x)$ shown in Eq. (6)). The results in the figure indicate that, whether using explicit gradient or implicit gradient, our method markedly outperforms LD with Gaussian noise; it not only demonstrates superior ability of FLD with $\mathcal{S}\alpha\mathcal{S}$ noise to escape local optima compared to LD, but also its capacity to converge to a lower energy function value.

6 Conclusion and Outlook

In this paper, we have addressed the limitations of LD in CO, including its tendency to be trapped in local optima, slow convergence, and generally suboptimal iterative performance. To overcome these challenges, we propose a FLD sampling process driven by $\mathcal{S}\alpha\mathcal{S}$ noise, fortified with a truncation strategy to ensure sampling stability. We theoretically prove that FLD achieves a polynomial mean escape time—significantly faster than the exponential escape time of LD, which depends on the energy barrier height—thereby enabling more rapid convergence. By integrating FLD into both sampling-based and data-driven frameworks, accommodating problems with or without explicit gradient information, we demonstrate its superior convergence and exploration capabilities on three case studies: MIS, MaxCl and MaxCut. Our FLD-EG and FLD-IG achieve SOTA or near-SOTA results compared to existing methods. Although our current FLD design focuses on binary-variable CO problems, it has potential applicability to integer and continuous-variable formulations. We plan to investigate these promising extensions in future work.

References

[1] Sungsoo Ahn, Younggyo Seo, and Jinwoo Shin. Learning what to defer for maximum independent sets. In *International conference on machine learning*, pages 134–144. PMLR, 2020.

[2] David Applebaum. *Lévy processes and stochastic calculus*. Cambridge university press, 2009.

[3] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[4] Douglas Bauer, Frank Boesch, C Suffel, and Ralph Tindell. Combinatorial optimization problems in the analysis and design of probabilistic networks. *Networks*, 15(2):257–271, 1985.

[5] Nils Berglund. Kramers' law: Validity, derivations and generalisations. *arXiv preprint arXiv:1106.5799*, 2011.

[6] Griff Bilbro, Reinhold Mann, Thomas Miller, Wesley Snyder, David van den Bout, and Mark White. Optimization by mean field annealing. *Advances in neural information processing systems*, 1, 1988.

[7] Maximilian Böther, Otto Kißig, Martin Taraz, Sarel Cohen, Karen Seidel, and Tobias Friedrich. What's wrong with deep learning in tree search for combinatorial optimization. *arXiv preprint arXiv:2201.10494*, 2022.

[8] Anton Bovier, Michael Eckhoff, Véronique Gayrard, and Markus Klein. Metastability and low lying spectra¶ in reversible markov chains. *Communications in mathematical physics*, 228:219–255, 2002.

[9] Cem Çelik and Melda Duman. Crank–nicolson method for the fractional diffusion equation with the riesz fractional derivative. *Journal of computational physics*, 231(4):1743–1750, 2012.

[10] Vladimír Černỳ. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45:41–51, 1985.

[11] Changyou Chen, David Carlson, Zhe Gan, Chunyuan Li, and Lawrence Carin. Bridging the gap between stochastic gradient mcmc and stochastic optimization. In *Artificial Intelligence and Statistics*, pages 1051–1060. PMLR, 2016.

[12] Mufa Chen and Fengyu Wang. Cheeger's inequalities for general symmetric forms and existence criteria for spectral gap. *Chinese science bulletin*, 43:1516–1518, 1998.

[13] Emilio Nicola Maria Cirillo, Vanessa Jacquier, and Cristian Spitoni. Metastability of synchronous and asynchronous dynamics. *Entropy*, 24(4):450, 2022.

[14] KA Dowsland and JM Thompson. Simulated annealing. handbook of natural computing (pp. 1623–1655). *Berlin: Springer. doi*, 10:978–3, 2012.

[15] Jinqiao Duan. *An introduction to stochastic dynamics*, volume 51. Cambridge University Press, 2015.

[16] Paul Erdos, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60, 1960.

[17] Shengyu Feng and Yiming Yang. Regularized langevin dynamics for combinatorial optimization. *arXiv preprint arXiv:2502.00277*, 2025.

[18] Maxime Gasse, Didier Chételat, Nicola Ferroni, Laurent Charlin, and Andrea Lodi. Exact combinatorial optimization with graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

[19] Ernestine Großmann, Sebastian Lamm, Christian Schulz, and Darren Strash. Finding near-optimal weight independent sets at scale. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 293–302, 2023.

[20] Ziao Guo, Yang Li, Chang Liu, Wenli Ouyang, and Junchi Yan. Acm-milp: Adaptive constraint modification via grouping and selection for hardness-preserving milp instance generation. In *Forty-first International Conference on Machine Learning*, 2024.

[21] Prateek Gupta, Maxime Gasse, Elias Khalil, Pawan Mudigonda, Andrea Lodi, and Yoshua Bengio. Hybrid models for learning to branch. *Advances in neural information processing systems*, 33:18087–18097, 2020.

[22] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. Gurobi Optimization, LLC, 2023.

[23] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[24] Yixuan He, Quan Gan, David Wipf, Gesine D Reinert, Junchi Yan, and Mihai Cucuringu. Gnnrank: Learning global rankings from pairwise comparisons via directed graph neural networks. In *international conference on machine learning*, pages 8581–8612. PMLR, 2022.

[25] Peter Imkeller and Ilya Pavlyukevich. First exit times of sdes driven by stable lévy processes. *Stochastic Processes and their Applications*, 116(4):611–642, 2006.

[26] Milton Jara. Spectral gap inequality for long-range random walks. *arXiv preprint arXiv:1810.12699*, 2018.

[27] Nikolaos Karalias and Andreas Loukas. Erdos goes neural: an unsupervised learning framework for combinatorial optimization on graphs. *Advances in Neural Information Processing Systems*, 33:6659–6672, 2020.

[28] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[29] Gregory F Lawler and Alan D Sokal. Bounds on the l^2 spectrum for markov chains and markov processes: a generalization of cheeger's inequality. *Transactions of the American mathematical society*, 309(2):557–580, 1988.

[30] Yang Li, Lvda Chen, Haonan Wang, Runzhong Wang, and Junchi Yan. Generation as search operator for test-time scaling of diffusion-based combinatorial optimization. In *Advances in Neural Information Processing Systems*, 2025.

[31] Yang Li, Jinpei Guo, Runzhong Wang, and Junchi Yan. T2t: From distribution learning in training to gradient search in testing for combinatorial optimization. *Advances in Neural Information Processing Systems*, 36:50020–50040, 2023.

[32] Yang Li, Jinpei Guo, Runzhong Wang, Hongyuan Zha, and Junchi Yan. Fast t2t: Optimization consistency speeds up diffusion-based training-to-testing solving for combinatorial optimization. *Advances in Neural Information Processing Systems*, 37:30179–30206, 2024.

[33] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Combinatorial optimization with graph convolutional networks and guided tree search. *Advances in neural information processing systems*, 31, 2018.

[34] Ziang Li, Mengda Yang, Yaxin Liu, Juan Wang, Hongxin Hu, Wenzhe Yi, and Xiaoyang Xu. Gan you see me? enhanced data reconstruction attacks against split inference. *Advances in neural information processing systems*, 36:54554–54566, 2023.

[35] Ziang Li, Hongguang Zhang, Juan Wang, Meihui Chen, Hongxin Hu, Wenzhe Yi, Xiaoyang Xu, Mengda Yang, and Chenjun Ma. From head to tail: Efficient black-box model inversion attack via long-tailed learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29288–29298, 2025.

[36] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[37] Radford M Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and computing*, 6:353–366, 1996.

[38] Manuel Duarte Ortigueira. Riesz potential operators and inverses via fractional centred derivatives. *International Journal of Mathematics and Mathematical Sciences*, 2006(1):048391, 2006.

[39] Aleksandar Pekeč and Michael H Rothkopf. Combinatorial auction design. *Management science*, 49(11):1485–1503, 2003.

[40] Ruizhong Qiu, Zhiqing Sun, and Yiming Yang. Dimes: A differentiable meta solver for combinatorial optimization problems. *Advances in Neural Information Processing Systems*, 35:25531–25546, 2022.

[41] Gareth O Roberts and Osnat Stramer. Langevin diffusions and metropolis-hastings algorithms. *Methodology and computing in applied probability*, 4:337–357, 2002.

[42] Gennady Samorodnitsky, Murad S Taqqu, and RW Linde. Stable non-gaussian random processes: stochastic models with infinite variance. *Bulletin of the London Mathematical Society*, 28(134):554–555, 1996.

[43] Sebastian Sanokowski, Wilhelm Berghammer, Martin Ennemoser, Haoyu Peter Wang, Sepp Hochreiter, and Sebastian Lehner. Scalable discrete diffusion samplers: Combinatorial optimization and statistical physics. *arXiv preprint arXiv:2502.08696*, 2025.

[44] Sebastian Sanokowski, Sepp Hochreiter, and Sebastian Lehner. A diffusion model framework for unsupervised neural combinatorial optimization. *arXiv preprint arXiv:2406.01661*, 2024.

[45] Abdelkader Sbihi and Richard W Eglese. Combinatorial optimization and green logistics. *Annals of Operations Research*, 175:159–175, 2010.

[46] Serap Ulusam Seçkiner and Mustafa Kurt. A simulated annealing approach to the solution of job rotation scheduling problems. *Applied Mathematics and Computation*, 188(1):31–45, 2007.

[47] Umut Şimşekli. Fractional langevin monte carlo: Exploring lévy driven stochastic differential equations for markov chain monte carlo. In *International Conference on Machine Learning*, pages 3200–3209. PMLR, 2017.

[48] Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pages 5827–5837. PMLR, 2019.

[49] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Information and Computation*, 82(1):93–133, 1989.

[50] Haoran Sun, Hanjun Dai, Wei Xia, and Arun Ramamurthy. Path auxiliary proposal for mcmc in discrete space. In *International Conference on Learning Representations*, 2021.

[51] Haoran Sun, Katayoon Goshvadi, Azade Nova, Dale Schuurmans, and Hanjun Dai. Revisiting sampling for combinatorial optimization. In *International Conference on Machine Learning*, pages 32859–32874. PMLR, 2023.

[52] Haoran Sun, Etash K Guha, and Hanjun Dai. Annealed training for combinatorial optimization on graphs. *arXiv preprint arXiv:2207.11542*, 2022.

[53] Zhiqing Sun and Yiming Yang. Difusco: Graph-based diffusion solvers for combinatorial optimization. *Advances in neural information processing systems*, 36:3706–3731, 2023.

[54] Luke Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

[55] Peter JM Van Laarhoven, Emile HL Aarts, Peter JM van Laarhoven, and Emile HL Aarts. *Simulated annealing*. Springer, 1987.

[56] Haoyu Wang and Pan Li. Unsupervised learning for combinatorial optimization needs meta-learning. *arXiv preprint arXiv:2301.03116*, 2023.

[57] Haoyu Peter Wang, Nan Wu, Hang Yang, Cong Hao, and Pan Li. Unsupervised learning for combinatorial optimization with principled objective relaxation. *Advances in Neural Information Processing Systems*, 35:31444–31458, 2022.

[58] Runzhong Wang, Ziao Guo, Shaofei Jiang, Xiaokang Yang, and Junchi Yan. Deep learning of partial graph matching via differentiable top-k. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6272–6281, 2023.

[59] Shiyue Wang, Haozheng Xu, Yuhan Zhang, Jingran Lin, Changhong Lu, Xiangfeng Wang, and Wenhao Li. Where paths collide: A comprehensive survey of classic and learning-based multi-agent pathfinding. *arXiv preprint arXiv:2505.19219*, 2025.

[60] Yibin Wang, Zhimin Li, Yuhang Zang, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Unified multimodal chain-of-thought reward model through reinforcement fine-tuning. *arXiv preprint arXiv:2505.03318*, 2025.

[61] Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-grpo: Pairwise preference reward-based grpo for stable text-to-image reinforcement learning. *arXiv preprint arXiv:2508.20751*, 2025.

[62] Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

[63] Ronald L Wasserstein. Monte carlo methods, volume 1: Basics, 1989.

[64] Rafał Weron. On the chambers-mallows-stuck method for simulating skewed stable random variables. *Statistics & probability letters*, 28(2):165–171, 1996.

[65] DF Wong, Hon Wai Leong, and HW Liu. *Simulated annealing for VLSI design*, volume 42. Springer Science & Business Media, 2012.

[66] Ke Xu and Wei Li. Exact phase transitions in random constraint satisfaction problems. *Journal of Artificial Intelligence Research*, 12:93–103, 2000.

[67] Nanyang Ye and Zhanxing Zhu. Stochastic fractional hamiltonian monte carlo. In *IJCAI*, pages 3019–3025, 2018.

[68] Dinghuai Zhang, Hanjun Dai, Nikolay Malkin, Aaron C Courville, Yoshua Bengio, and Ling Pan. Let the flows tell: Solving graph combinatorial problems with gflownets. *Advances in neural information processing systems*, 36:11952–11969, 2023.

[69] Ruqi Zhang, Xingchao Liu, and Qiang Liu. A langevin-like sampler for discrete distributions. In *International Conference on Machine Learning*, pages 26375–26396. PMLR, 2022.

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: Our claims are supported by theoretical proof in Sec. 4.3 and empirical results in Sec. 5.

   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: [Yes]

   Justification: We discuss the limitations in the last part of the main paper.

   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: [Yes]

Justification: Critical theoretical proof can be found in Appendix A and B.

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: All experimental settings are included in Sec. 5.

   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: [Yes]

Justification: We will release the complete source code once the paper is accepted.

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: All experimental settings are included in Sec. 5. Appendix E.2 presents hyperparameter values.

   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: [Yes]

   Justification: The test set size is appropriate to demonstrate the statistical significance of the experiments, consistent with previous works.

   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: Included in Appendix E.1.

   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: I have read the NeurIPS Code of Ethics and I am sure that our work fully adheres to its guidelines.

   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: Discussed in Appendix F.

    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: 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: All models, data, and code used in this work are properly cited.

    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: [Yes]

    Justification: The documentation is attached to our complete source code and will be released once the paper is accepted.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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: The paper does not involve crowdsourcing nor 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: The LLM was used solely for grammar checking 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.

A Proof of Theorem 1

Theorem 4 ([47]). *Consider the SDE* (13)*, the drift* b *is defined by:*

$$b(x,\alpha) = (\mathcal{D}^{\alpha-2} f_\pi(x))/\phi(x) \tag{29}$$

$f_\pi(x) = -\phi(x)\partial_x U(x)$*, and* π *is an invariant measure of the Markov process.*

Theorem 5 ([38]). *The Riesz derivative* $\mathcal{D}^\gamma$ *of a function* $f(x)$ *can be defined as the limit of the fractional centered difference operator* $\triangle_h^\gamma$*, given:*

$$\mathcal{D}^\gamma f(x) = \lim_{h\to 0} \triangle_h^\gamma f(x) \tag{30}$$

where,

$$\triangle_h^\gamma f(x) = (1/h^\gamma) \sum_{k=-\infty}^{\infty} g_{\gamma,k} f(x-kh) \tag{31}$$

and

$$g_{\gamma,k} = (-1)^k \frac{\Gamma(\gamma+1)}{\Gamma(\gamma/2-k+1)\Gamma(\gamma/2+k+1)} \tag{32}$$

Proof. Theorem 4 guarantees the existence of the equality on the left-hand side of Eq.(13).

Next, we will give the proof of the right-hand side. The more computationally efficient variant of the first numerical scheme presented in Theorem 5 is given as follows [9]:

$$\mathcal{D}^\gamma f_\pi(x) \approx g_{\gamma,0} f_\pi(x) \tag{33}$$

where $g_{\gamma,0} = \Gamma(\gamma+1)/\Gamma(\gamma/2+1)^2$ for $x \in \mathbb{R}$.

Then we get

$$\begin{aligned} b(x,\alpha) = \frac{\mathcal{D}^{\alpha-2} f_{p_\tau}(x)}{\phi(x)} &= \frac{\mathcal{D}^{\alpha-2}(-\phi(x)\partial_x H(x))}{\phi(x)} = \mathcal{D}^{\alpha-2} s(x) \\ &= \Gamma(\alpha-1)/\Gamma(\alpha/2)^2 s(x) = c_\alpha s(x) \end{aligned} \tag{34}$$

where $c_\alpha = \Gamma(\alpha-1)/\Gamma(\alpha/2)^2$. □

B Proof of Theorem 2 and Theorem 3

Definition 4 ([64]). *A random variable* X *is* α*-stable if and only if its characteristic function is given by*

$$\log \phi(t) = \begin{cases} -\sigma_2^\alpha |t|^\alpha \exp\{-i\beta_2 \mathrm{sign}(t)\frac{\pi}{2} K(\alpha)\} + i\mu t, \ \alpha \neq 1 \\ -\sigma_2 |t|(\frac{\pi}{2} + i\beta_2 \mathrm{sign}(t)\log|t|) + i\mu t, \ \alpha = 1 \end{cases} \tag{35}$$

where, $K(\alpha) = \alpha - 1 + \mathrm{sign}(1-\alpha)$*, and* $\mathrm{sign}(\cdot)$ *denotes the sign function. The parameters* σ_2 *and* β_2 *are related to* σ *and* β*.*

Case 1: *for* $\alpha \neq 1$*,* β_2 *is such that*

$$\tan\Big(\beta_2 \frac{\pi K(\alpha)}{2}\Big) \ = \ \beta \ \tan\frac{\pi\alpha}{2} \tag{36}$$

and the new scale parameter

$$\sigma_2 \ = \ \sigma \ \Big(1 + \beta^2 \tan^2\frac{\pi\alpha}{2}\Big)^{\frac{1}{2\alpha}} \tag{37}$$

Case 2: *for* $\alpha = 1$*,* $\beta_2 = \beta$ *and* $\sigma_2 = \frac{2}{\pi}\sigma$*.*

Lemma 1 ([64]). X *is a* $\mathcal{S}_\alpha(1,\beta_2,0)$ *random variable if and only if for* $x > 0$*:*

$$\begin{cases} P\big(0 < X \leq x\big) = \dfrac{1}{\pi}\displaystyle\int_{\gamma_0}^{\frac{\pi}{2}} \exp\Big[-x^{\frac{\alpha}{\alpha-1}} \ U_\alpha(\gamma,\gamma_0)\Big]\,\mathrm{d}\gamma, \ \alpha < 1 \\ P\big(X \geq x\big) = \dfrac{1}{\pi}\displaystyle\int_{\gamma_0}^{\frac{\pi}{2}} \exp\Big[-x^{\frac{\alpha}{\alpha-1}} \ U_\alpha(\gamma,\gamma_0)\Big]\,\mathrm{d}\gamma, \ \alpha > 1 \end{cases} \tag{38}$$

where, $\gamma_0 = -\frac{\pi\beta K(\alpha)}{2\alpha}$ *and* $U_\alpha(\gamma,\gamma_0) = \Big(\frac{\sin\alpha(\gamma-\gamma_0)}{(\cos\gamma)}\Big)^{\alpha/1-\alpha} \frac{\cos(\gamma-\alpha(\gamma-\gamma_0))}{\cos\gamma}$.

Proof. When $\gamma > \gamma_0$ then the right hand side of Eq. (17) ($\alpha \neq 1$) is positive and can be expressed as [64]:

$$\Big(\frac{a(\gamma)}{W}\Big)^{(1-\alpha)/\alpha}, \tag{39}$$

where

$$a(\gamma) = \Big(\frac{\sin\big(\alpha(\gamma-\gamma_0)\big)}{\cos\gamma}\Big)^{\alpha/(1-\alpha)} \; \frac{\cos\big(\gamma-\alpha(\gamma-\gamma_0)\big)}{\cos\gamma}. \tag{40}$$

Case 1: When $0 < \alpha < 1$, Eq. (17) ($\alpha \neq 1$) implies that $X > 0$ if and only if $\gamma > \gamma_0$. Since $\frac{1-\alpha}{\alpha} > 0$, we can write

$$\begin{aligned}
P(0 < X \leq x) &= P\big(0 < X \leq x,\ \gamma > \gamma_0\big) \\
&= P\Big(0 < \big(a(\gamma)/W\big)^{(1-\alpha)/\alpha} \leq x,\ \gamma > \gamma_0\Big) \\
&= P\Big(W \geq x^{\alpha/(\alpha-1)}\, a(\gamma),\ \gamma > \gamma_0\Big) \\
&= E_\gamma\Big[\exp\big(-x^{\alpha/(\alpha-1)}\, a(\gamma)\big)\, \mathbf{1}_{\{\gamma>\gamma_0\}}\Big] \\
&= \frac{1}{\pi}\int_{\gamma_0}^{\pi/2} \exp\big(-x^{\alpha/(\alpha-1)}\, a(\gamma)\big)\, d\gamma.
\end{aligned}$$

From Lemma 1, we conclude that $X \sim S_\alpha(1, \beta_2, 0)$.

Case 2: For $1 < \alpha \leq 2$, noting that $\frac{\alpha-1}{\alpha} > 0$, we similarly deduce that for all $x > 0$,

$$\begin{aligned}
P(X \geq x) &= P\big(X \geq x,\ \gamma > \gamma_0\big) \\
&= P\Big(\big(a(\gamma)/W\big)^{(1-\alpha)/\alpha} \geq x,\ \gamma > \gamma_0\Big) \\
&= P\Big(\big(W/a(\gamma)\big)^{(\alpha-1)/\alpha} \geq x,\ \gamma > \gamma_0\Big) \\
&= P\Big(W \geq x^{\alpha/(\alpha-1)}\, a(\gamma),\ \gamma > \gamma_0\Big) \\
&= E_\gamma\Big[\exp\big(-x^{\alpha/(\alpha-1)}\, a(\gamma)\big)\, \mathbf{1}_{\{\gamma>\gamma_0\}}\Big] \\
&= \frac{1}{\pi}\int_{\gamma_0}^{\pi/2} \exp\big(-x^{\alpha/(\alpha-1)}\, a(\gamma)\big)\, d\gamma.
\end{aligned}$$

Again by Lemma 1 we get $X \sim S_\alpha(1, \beta_2, 0)$.

Case 3: For the case $\alpha = 1$, when $\beta_2 = 0$ the right hand side of Eq. (17) ($\alpha = 1$) simplifies to $\frac{\pi}{2}\tan\gamma$, which has a Cauchy law (cf. Eq. (35)). If $\beta_2 \neq 0$, it can instead be written as

$$\beta_2 \log\Big(\frac{a_1(\gamma)}{W}\Big), \tag{41}$$

where

$$a_1(\gamma) = \frac{\frac{\pi}{2}+\beta_2\gamma}{\cos\gamma}\; \exp\Big(\frac{1}{\beta_2}\,\big(\tfrac{\pi}{2}+\beta_2\gamma\big)\tan\gamma\Big). \tag{42}$$

For $\beta_2 > 0$, we have

$$\begin{aligned}
P(X \leq x) &= P\big(\beta_2 \log(a_1(\gamma)/W) \leq x\big) \\
&= P\big(W \geq e^{-x/\beta_2}\, a_1(\gamma)\big) \\
&= E_\gamma\Big[\exp\big(-e^{-x/\beta_2}\, a_1(\gamma)\big)\Big] \\
&= \frac{1}{\pi}\int_{-\pi/2}^{\pi/2} \exp\big(-e^{-x/\beta_2}\, a_1(\gamma)\big)\, d\gamma.
\end{aligned}$$

Finally, we conclude that for all β_2, $X \sim S_1(1, \beta_2, 0)$.

Due to the $\mathcal{S}\alpha\mathcal{S}$ distribution being the special case of α-stable distribution when $\beta = 0$, thus Theorem 2 and Theorem 3 have been proven together. □

C Case Study on Energy Functions

The problem formulations utilized in this paper are given below for three common case studies with the closed form of the energy function $H(x)$, which means $H(x)$ is first-order derivable. Let $G(V, E)$ be an undirected and unweighted graph, where $V = \{1, 2, \cdots, N\}$ denotes the node set of the graph G and $E \subseteq V \times V$ represents the edge set. The problem descriptions, energy function, and its gradient are given.

Case Study 1: Maximum Independent Set (MIS). The independent set is S satisfied that $\forall i, j \in S \subseteq V$ and $e(i, j) \in E$, then $i = j$. Thus the definition of MIS $S^* = \arg\max_{S \subseteq V} |S|$ ($|\cdot|$ denotes the size of $\cdot$). The formulation of MIS is:

$$\max_{x \in \{0,1\}^N} \sum_{i \in V} c_i x_i, \text{ s.t. } \sum_{e(i,j) \in E} x_i x_j = 0 \tag{43}$$

The energy function of MIS formed as Eq. (6) is:

$$H(x) = -\sum_{i \in V} c_i x_i + \lambda \sum_{e(i,j) \in E} x_i x_j = -\mathbf{c}^\top \mathbf{x} + \lambda \frac{\mathbf{x}^\top \mathbf{A} \mathbf{x}}{2} \tag{44}$$

where c_i $(i \in V)$ denotes the node weights of graph G and the content to the right of the last equal sign is the energy function under the matrix form representation (the same goes for the following two cases). The gradient of the energy function can be presented readily:

$$\nabla H(x) = -\mathbf{c} + \lambda \mathbf{A} \mathbf{x} \tag{45}$$

Case Study 2: Maximum Clique. The clique is the set $C \subseteq V$ satisfied that $\forall i, j \in C,\ i \neq j$, then $e(i, j) \in E$. Therefore, the definition of maximum clique $C^* = \arg\max_{S \subseteq V} |C|$. The formulation of maximum clique is:

$$\max_{x \in \{0,1\}^N} \sum_{i \in V} c_i x_i, \text{ s.t. } \sum_{e(i,j) \notin E} x_i x_j = 0 \tag{46}$$

The energy function of maximum clique formed as Eq. (6) is:

$$H(x) = -\sum_{i \in V} c_i x_i + \lambda \sum_{e(i,j) \notin E} x_i x_j = -\mathbf{c}^\top \mathbf{x} + \frac{\lambda}{2}((\mathbf{1}^\top x)^2 - \mathbf{x}^\top \mathbf{x} - \mathbf{x}^\top \mathbf{A} \mathbf{x}) \tag{47}$$

The gradient of $H(x)$ is shown as:

$$\nabla H(x) = -\mathbf{c} + \lambda((\mathbf{1}^\top \mathbf{x})\mathbf{1} - \mathbf{x} - \mathbf{A}\mathbf{x}) \tag{48}$$

Case Study 3: Max Cut. The max cut problem seeks a partition $(S, \bar{S})$ that maximizes the number of crossing edges:

$$\max_{S \subseteq V} |\{e(i, j) \in E | i \in S, j \in \bar{S} = V \setminus S\}| \tag{49}$$

The mathematical formulation, energy function, and its gradient can be presented as:

$$\max_{x \in \{0,1\}^N} \sum_{e(i,j) \in E} \frac{1 - (2x_i - 1)(2x_j - 1)}{2} \tag{50}$$

$$H(x) = -\sum_{e(i,j) \in E} \frac{1 - (2x_i - 1)(2x_j - 1)}{2} = \mathbf{x}^\top \mathbf{A} \mathbf{x} - \mathbf{1}^\top \mathbf{A} \mathbf{x} \tag{51}$$

$$\nabla H(x) = \mathbf{A}(2\mathbf{x} - \mathbf{1}) \tag{52}$$

D Details on FLD-EG and FLD-IG

D.1 Algorithmic Process of FLD-EG

We now present the detailed implementation of the FLD-EG algorithm in Alg. 1. The algorithm takes as input the maximum number of iterations T, the number K of independent SA processes, the truncation parameter d for the closed-form gradient ∇H, the truncation parameter d_{noise} for the $\mathcal{S}\alpha\mathcal{S}$ noise, the initial temperature τ_0 for SA, the stability parameter α of the FLD sampling process, and the stepsize schedule parameters a_η and b_η. (For the specific values of these hyperparameters, see Appendix E.2.)

The FLD-EG algorithm employs a near–continuous sampling procedure to guide the assignment of binary variables. First, both the binary variable vector $\mathbf{x}$ and the auxiliary continuous variable vector $\mathbf{h}$ are initialized, and the initial energy is computed via the problem-specific function $\text{energy\_func}()$. Each iteration then comprises two stages: (1) a sampling update for $\mathbf{h}$, and (2) an update for $\mathbf{x}$ based on the newly sampled $\mathbf{h}$. After each update of $\mathbf{x}$, we record the best observed energy. Once the maximum iteration count T is reached, a greedy decoding step produces a final, constraint-satisfying solution.

The sampling update for $\mathbf{h}$ consists of four substeps:

1. **Noise sampling.** Sample the noise variable z_iter from an $\mathcal{S}\alpha\mathcal{S}(1)$ distribution as prescribed by Theorem 3, with the sampling mechanism defined in Eq. (18). In practice, for convenience, we replace exponentially distributed sampling with uniformly distributed sampling for U.
2. **Gradient truncation.** Apply the Top-d truncated indicator to ∇H, ensuring that only the d components with the largest magnitude influence the update.
3. **Noise truncation.** Apply the Top-d_{noise} truncated indicator to the sampled noise vector, truncating extreme values to stabilize the sampling process.
4. **State update.** Update $\mathbf{h}$ according to the FLD update rule in Eq. (19).

Finally, since the solution may still violate some problem constraints, we perform a greedy decoding step on $\mathbf{x}$ until all constraints are satisfied.

D.2 Details on FLD-IG

Training and Inference Process. In Alg. 2, we present the detailed training procedure for FLD-IG. The algorithm takes as input the maximum number of iterations T' for training and T for inference, the number of independent parallel sampling processes K' for training and K for inference, the truncation parameter d for both the gradient and the $\mathcal{S}\alpha\mathcal{S}$ noise, the initial temperature τ_0 for the sampling process, and the stability parameter α of the FLD sampling process. (For the specific values of these hyperparameters, see Appendix E.2.) At each iteration, we sample $\mathcal{S}\alpha\mathcal{S}$ noise and incorporate it into the proposal distribution, allowing $\mathbf{x}'$ to be drawn from this noise-augmented proposal so as to compute flip probabilities and update $\mathbf{x}$. Unlike FLD-EG, which applies truncation to both the gradient and the noise terms during sampling, and given that the noise samples are independent at each iteration and no closed-form formulation exists for the energy-function gradient, we instead perform a unified truncation of the noise-augmented proposal distribution directly within the loss function. After each update, we record the best-observed energy. Once the maximum training iteration step T' is reached, save the best parameters for the inference process.

Similar to the training process, we first sample the $\mathcal{S}\alpha\mathcal{S}$ noise and incorporate it into the proposal distribution in the inference process. Next, we draw $\mathbf{x}'$ by sampling from this noise-augmented proposal distribution to compute flip probabilities for updating $\mathbf{x}'$, and we record the best energy observed after each update. Once the maximum inference iteration count T is reached, a final greedy decoding step generates a constraint-satisfying solution which is similar to the greedy decoding step of FLD-EG.

Network Architecture. For the implementation of the network architecture, we adopt a five-layer GCN with 128 hidden dimensions, which is consistent with [17]. We observe that our FLD-IG converges faster than [17]. Therefore, we set the number of training epochs to 30, which is adequate for convergence.

Algorithm 1 FLD-EG

Require: T, K, d, d_{noise}, τ_0, α, a_η, and b_η.

Initialize $\mathbf{x} \in \{0,1\}^{N\times k}$; $h \leftarrow \mathbf{x}$; $\mathbf{x}^* \leftarrow \mathbf{x}$; $c_\alpha = \Gamma(\alpha-1)/\Gamma(\alpha/2)^2)$
Build adjacency matrix A from (edge_index, edge_weight)
(energy, ∇H) $\leftarrow$ energy_func$(A, b, \mathbf{x}, \mathit{True})$
best_sol $\leftarrow \mathbf{x}$; **best_energy** $\leftarrow$ energy
for $t = 1, 2, \cdots, T$ **do**
  $\tau \leftarrow \tau_0\Big(1 - \dfrac{t}{T}\Big)$
  for $i = 1, 2, \cdots, N$ **do**
    Sample $W \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$, $U \sim \mathcal{U}(0,1)$
    $z\_iter \leftarrow \dfrac{\sin(\alpha W)}{\cos(W)^{1/\alpha}} \Big(\dfrac{\cos((1-\alpha)W)}{-\ln U}\Big)^{\frac{1-\alpha}{\alpha}}$ ▷ $\mathcal{S}\alpha\mathcal{S}$ Noise Sampling
    $\eta \leftarrow \left(\dfrac{a_\eta}{t+1}\right)^{b_\eta}$
    $t_g \leftarrow -\text{kthvalue}(-\frac{1}{2}\left((2\mathbf{x}-1)\odot\nabla H\right)_{(i)}, d, \text{dim} = 0)$ ▷ Top-d Truncated Indicator
    $p_g \leftarrow \text{Sigmoid}\big((\frac{1}{2}\left((2\mathbf{x}-1)\odot\nabla H\right)_{(i)} - t_g)/\tau\big)$
    Sample $\mathbf{I}_g \sim \text{Bernoulli}(p_g)$
    $t_z \leftarrow -\text{kthvalue}(-\frac{1}{2}\left((2\mathbf{x}-1)\odot\nabla H\right)_{(i)}, d_{\text{noise}}, \text{dim} = 0)$ ▷ Top-d_{noise} Truncated Indicator
    $p_z \leftarrow \text{Sigmoid}\big((-\frac{1}{2}\left((2\mathbf{x}-1)\odot\nabla H\right)_{(i)} - t_z)/\tau\big)$
    Sample $\mathbf{I}_z \sim \text{Bernoulli}(p_z)$
    $grad\_iter \leftarrow \eta\, c_\alpha \left(-\frac{1}{\tau}\right) \nabla H$
    $z\_iter \leftarrow \eta^{1/\alpha}\, z\_iter$
    $h_i \leftarrow h_i - \mathbf{I}_g \odot grad\_iter + \mathbf{I}_z \odot z\_iter$ ▷ Sampling Iterative Process
    $h_i \leftarrow \text{Clamp}(h_i, 0, 1)$
    $x_i \leftarrow \text{where}\big(\text{rand}() < h_i, 0, 1\big)$
  end for
  (energy, ∇H) $\leftarrow$ energy_func$(A, b, \mathbf{x}, \mathit{epoch} < T)$
  if energy $<$ **best_energy then**
    best_sol $\leftarrow \mathbf{x}$; **best_energy** $\leftarrow$ energy
  end if
end for
return **best_sol** (**or** return min **best_energy** if skip-decode)

E Details on Experiments

E.1 Hardware and Software Devices

Experiments are conducted on a Linux workstation using an H100 GPU and an Intel(R) Xeon(R) Platinum 8468 CPU, with programs implemented in *PyTorch*.

E.2 Hyperparameters

We show the utilized hyperparameter values of FLD-EG and FLD-IG in Table 3 and Table 4, respectively. The selection of hyperparameter values partly refers to [17].

F Broader Impacts

The FLD framework we introduce has the potential to substantially advance both the practical application and theoretical understanding of CO. By enabling reliable escape from deep local optima—and doing so in polynomial time across a range of problem landscapes—FLD can drive more efficient logistic networks, reducing transportation costs and carbon emissions through better routing; streamline scheduling in manufacturing and cloud computing, leading to higher resource utilization

Algorithm 2 FLD-IG (Training)

Require: T', K', d, λ', α, τ_0

Initialize θ
while stopping criterion not met **do**
  Initialize $x \in \{0,1\}^N, \quad \mathcal{D} \leftarrow \{x\}$
  for $t = 1, 2, \cdots, T'$ **do**
    $\tau \leftarrow \tau_0 \left(1 - \frac{t-1}{T'}\right)$
    Sample $\mathbf{W}_{(i)} \overset{\text{i.i.d.}}{\sim} \mathcal{U}\big(-\frac{\pi}{2}, \frac{\pi}{2}\big), \quad i = 1, \ldots, N$
    Sample $\mathbf{U}_{(i)} \overset{\text{i.i.d.}}{\sim} \mathcal{U}\big(0, 1\big), \quad i = 1, \ldots, N$
    Compute

$$\mathbf{Z}_{(i)} \leftarrow \frac{\sin(\alpha \mathbf{W}_{(i)})}{\cos(\mathbf{W}_{(i)})^{1/\alpha}} \left(\frac{\cos((1-\alpha)\mathbf{W}_{(i)})}{-\ln \mathbf{U}_{(i)}} \right)^{\frac{1-\alpha}{\alpha}}, \quad i = 1, \ldots, N$$

▷ $\mathcal{S}\alpha\mathcal{S}$ Noise Sampling

    $\mathbf{x}' \sim q_\theta\big(\mathbf{x}' \mid \mathbf{x} + \ \tau\, \mathbf{Z}\,(1 - 2\mathbf{x})\big)$ ▷ Sample from proposal distribution with $\mathcal{S}\alpha\mathcal{S}$ Noise
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{x'\}$
    $x \leftarrow x'$
  end for
  $\theta \leftarrow \arg\min_\theta \mathbb{E}_{x \in \mathcal{D}} \big[\ell_{\text{FLD}}(\theta; x, d, \lambda', \alpha, \tau_0)\big]$
end while

return θ

Table 3: Hyperparameters used by FLD-EG on all datasets.

Problem	Dataset	τ_0	d	K	T	λ	d_{noise}	α	a_η	b_η
MIS	RB-[200–300]	0.01	5	200	300	1.02	9	1.2	4	0.6
	RB-[800–1200]	0.01	5	200	500	1.02	12	1.3	0.1	0.6
	ER-[700–800]	0.01	20	200	500	1.001	20	1.5	0.1	0.6
	ER-[9000–1100]	0.01	20	200	5000	1.001	60	1.1	0.1	0.6
MaxCl	RB-[200–300]	4	2	200	100	1.02	4	1.5	30	0.6
	RB-[800–1200]	4	2	200	500	1.02	2	1.7	0.1	0.6
MaxCut	BA-[200–300]	5	20	200	200	1.02	33	1.01	200	0.6
	BA-[800–1200]	5	20	200	500	1.02	35	1.01	200	0.6

Table 4: Hyperparameters used by FLD-IG on all datasets.

Problem	Dataset	τ_0	d	K	T	λ	K'	T'	λ'	α
MIS	RB-[200–300]	0.25	5	20	100	1.02	10	50	0.5	1.7
	RB-[800–1200]	0.25	5	20	200	1.02	10	300	0.5	1.7
	ER-[700–800]	0.6	20	20	200	1.001	10	500	0.5	1.7
	ER-[9000–1100]	0.9	20	20	800	1.001	–	–	–	1.7
MaxCl	RB-[200–300]	0.25	2	20	100	1.02	10	100	0.5	1.7
	RB-[800–1200]	0.25	2	20	200	1.02	10	300	0.5	1.7
MaxCut	BA-[200–300]	0.25	20	20	100	1.02	10	50	0.5	1.7
	BA-[800–1200]	0.25	20	20	200	1.02	10	300	0.5	1.7

and energy savings; and enhance network-design and financial-optimization tools, yielding more robust communication infrastructures and investment strategies. Moreover, because FLD naturally integrates with data-driven pipelines via its implicit-gradient formulation, it can be seamlessly

incorporated into emerging machine-learning platforms for applications such as automated materials discovery, bioinformatics, and large-scale resource allocation, fostering interdisciplinary innovation.

At the same time, we acknowledge that any powerful optimization technology carries risks. Unchecked, FLD could be used to accelerate adversarial planning—such as in cybersecurity, market manipulation, or autonomous weaponry—by quickly finding worst-case configurations. To mitigate misuse, we recommend that practitioners pair FLD with domain-specific ethical guidelines and transparency mechanisms (e.g., logging and audit trails for critical decision systems), and that the research community pursue formal verification methods to ensure that FLD-based solutions adhere to safety and fairness constraints. By proactively addressing these considerations, we believe FLD can serve as a force for positive impact—improving efficiency and sustainability in industrial and scientific applications while minimizing potential harms.