DDPS: Discrete Diffusion Posterior Sampling for Paths in Layered Graphs

Hao Luan[1], **See-Kiong Ng**[1,2], **and Chun Kai Ling**[1]
[1]School of Computing, National University of Singapore
[2]Institute of Data Science, National University of Singapore
haoluan@comp.nus.edu.sg, {seekiong, chunkail}@nus.edu.sg

Abstract

Diffusion models form an important class of generative models today, accounting for much of the state of the art in cutting edge AI research. While numerous extensions beyond image and video generation exist, few of such approaches address the issue of *explicit constraints* in the samples generated. In this paper, we study the problem of generating paths in a layered graph (a variant of a directed acyclic graph) using discrete diffusion models, while guaranteeing that our generated samples are indeed paths. Our approach utilizes a simple yet effective representation for paths which we call the padded adjacency-list matrix (PALM). In addition, we show how to effectively perform classifier guidance, which helps steer the sampled paths to specific preferred edges without any retraining of the diffusion model. Our preliminary results show that empirically, our method outperforms alternatives which do not explicitly account for path constraints.

1 Introduction

Diffusion models have emerged as one of the most popular methods of generative AI particularly with hyper-realistic image and video generation, often outperforming older methods like generative adversarial networks. The recent years have seen much interest in replicating this success in other domains. These domains include protein design (Frey et al., 2024), molecular conformations (Xu et al., 2022), text generation (Li et al., 2022), robotics (Chi et al., 2023; Wang et al., 2024; Feng et al., 2024; 2025), *etc.* Unlike image and video generation, These recent applications often require the restriction that the generated samples belong to some *discrete domain*. On top of that, one often requires samples to obey some form of predefined *structural constraint*, either for reasons related to safety or simply because violating those constraints would make little physical sense. For instance, certain robot poses or physical configurations cannot be achieved in the real world. Most of the existing work adopt a "data-centric" approach towards "softly" enforcing such constraints (Chi et al., 2023), relying on the assumption that such unsafe or non-physical samples do not appear in the training set and that these structural constraints would be implicitly learned.

In this paper, we study how to *explicitly* enforce these structural constraints. We focus on trying to generate paths in a *layered graph*, which are directed acyclic graphs organized in layers. Layered graphs occur regularly in domains involving planning. For this reason, we seek to be able to sample from such paths (learned from some dataset) via a diffusion model. This has applications ranging from security to logistics(Černý et al., 2024; Černỳ et al., 2024; Zhang et al., 2017). However, this requires that the generated samples from the diffusion model obey the constraint that they are indeed valid paths in the layered graph, and not just an arbitrary subset of edges/vertices.

Our contributions are as follows. First, we formulate the path learning problem in a layered graph via discrete diffusion models. We propose a simple representation of paths called padded adjacency–list matrix (PALM), which guarantees that generated output will always be paths in the underlying layered graph, unlike other diffusion based approaches. Second, we show how to perform training and inference efficiently under PALM. Third, we show how one can perform *guidance* under our proposed model, a variant of existing classifier guidance methods that allow us to favor paths which contain certain distinguished edges during inference without retraining. Lastly, we show empirically that our proposed method is superior in generating paths compared to naive alternatives. We also examine the tradeoff between strength of guidance and adherence to the learned distribution.

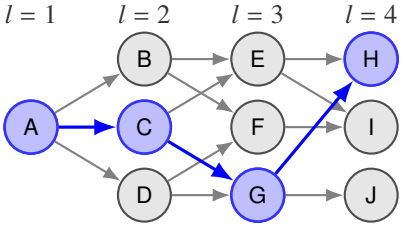


Figure 1: Example of a layered graph with $L = 4$ layers. The first layer $\mathcal{V}^1$ is a singleton ($|\mathcal{V}^1| = 1$), while subsequent layers have $|\mathcal{V}^l| = 3$ for $l = 2, 3, 4$. Edges only exist between adjacent layers ($l \to l + 1$). A specific path (A, C, G, H) is highlighted in blue.

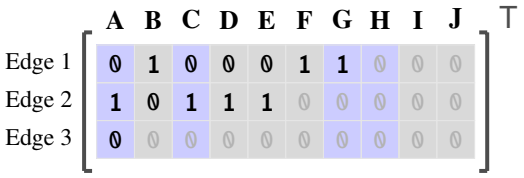


Figure 2: The PALM representation (transposed) for the path highlighted in figure 1. Columns correspond to vertices: blue indicates on-path vertices (A, C, G, H) and their specific one-hot vectors; gray indicates off-path vertices. Each gray column vector is an arbitrary one-hot selection if the vertex has outgoing edges. Faded entries are padding zeros.

2 Preliminaries and Related Work

In this paper, we work with paths in *layered graphs*. Loosely speaking, a layered graph is a directed acyclic graph with the additional structure that the vertices can be ordered by layers (Černý et al., 2024); see figure 1 for an example. Our goal is to learn and generate paths in a layered graph with a diffusion model. Furthermore, we would like the generated paths to possess some user-specified properties, *e.g.*, passing through some preferred edges. As we will see later, this will be achieved by employing discrete diffusion models and classifier guidance based posterior sampling.

2.1 Preliminaries

Definition 1 (Layered Graph, adapted from (Černý et al., 2024))**.** Let $G = (\mathcal{V}, \mathcal{E})$ be a *directed* graph defined over a finite vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. G is a layered graph if all of the following conditions are true: (1) Set $\mathcal{V}$ can be partitioned into $L > 1$ non-empty sets $\mathcal{V}^1, \ldots, \mathcal{V}^L$, which are called *layers*; (2) each edge $e \in \mathcal{E}$ is in $\mathcal{V}^l \times \mathcal{V}^{l+1}$ for some $l \in [1, L-1]$; (3) the first layer is a singleton: $|\mathcal{V}^1| = 1$; (4) every vertex $v \in \mathcal{V}^l$ for $l < L$ with zero out-degree has zero in-degree.

Definition 2 (Path)**.** A path in a layered graph is defined as an ordered sequence $(v^1, \ldots, v^L) \in \mathcal{V}^1 \times \cdots \times \mathcal{V}^L$ of length L, such that $(v^l, v^{l+1}) \in \mathcal{E}$ for $l = 1, \ldots, L-1$.

Examples of layered graphs. In the work of Černý et al., each layer represented the possible location of an agent (typically a security patrol) at a given time; the existence of an edge between adjacent layers meant that moving between two physical locations is possible at a particular timestep. Hence, each valid path represents a different duration route L. Another example is sequence prediction. For instance, a fully connected layered graph with four vertices per layer can be used to represent all possible combinations of length L DNA sequences. Here, edges can be used to forbid certain adjacent nucleotide bases (*e.g.*, "A" cannot be followed by "G" in the 4th and 5th location).

Discrete diffusion models. A discrete diffusion model involves with a forward Markov process gradually corrupting the initial discrete data representation x_0 with noise for a finite number of time steps $t = 1, \ldots, T$, and a learned reverse Markov process gradually reconstructing x_0 from x_T. A single step in the forward chain can be formalized as

$$q(x_t \mid x_{t-1}) = \text{Cat}(Q_t x_{t-1}) \tag{1}$$

wherein $\text{Cat}(p)$ is a K-dimensional categorical distribution parameterized by the probabilities in $p \in \mathbb{R}^K$, $x_t \in \mathbb{R}^K$ is a one-hot vector, and $Q_t \in \mathbb{R}^{K \times K}$ is a transition matrix at timestep t specifying transition probabilities among the K categories. The diffusion model learns a backward process

$$p_\theta(x_{t-1} \mid x_t) = \sum_{x_0} q(x_{t-1} \mid x_t, x_0)\, p_\theta(x_0 \mid x_t)\,. \tag{2}$$

At inference time, clean samples can be generated by running the learned backward process Eq. (2) for multiple timesteps $t = T, \ldots, 1$, with $x_T \sim p_T(x)$ sampled from some prior distribution p_T.

2.2 Related Work

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019; Song et al., 2021) first gained success in tasks in continuous domains like image generation (Dhariwal & Nichol, 2021; Rombach et al., 2022). Influenced by such success, more approaches have been proposed for generating discrete data, *e.g.*, text (Li et al., 2022), graphs (Niu et al., 2020; Vignac et al., 2023; Yan et al., 2024), biological sequences (Yim et al., 2023; Avdeyev et al., 2023), *etc.* Diffusion models operating on discrete state-space (Austin et al., 2021; Hoogeboom et al., 2021) are of our interest. Campbell et al. (2022; 2024) also extended discrete diffusion models from discrete-time setting to continuous time through the lens of Continuous-time Markov chain. There are also efforts in diffusion models working on the continuous space of probabilistic simplex (Stark et al., 2024).

A favorable trait of diffusion models is their amenability to guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2021), which allows users to generate samples with desirable properties and has been widely exploited in tasks on continuous domains. For discrete diffusion, the technique of guidance is less mature. One of the pioneer attempts might be Vignac et al. (2023) using guidance in graph generation. More discrete guidance techniques were recently proposed for biological sequence (Nisonoff et al., 2025), text generation(Schiff et al., 2025) and for discrete latent diffusion models (Han et al., 2024; Murata et al., 2024). While most existing works approach guidance from the perspective of either classifier guidance (Dhariwal & Nichol, 2021) or classifier-free guidance (Ho & Salimans, 2021), recent emerging research explores a reinforcement learning approach (Rector-Brooks et al., 2025; Li et al., 2024; Wang et al., 2025; Uehara et al., 2025).

To the best of our knowledge, there has not been much work on diffusion models that can guarantee that *paths* are generated with *certainty* without significant post processing. For instance, Niu et al. (2020) and Yan et al. (2024) learn continuous representations of graph adjacency matrices and perform discretization as a form of post processing to obtain a graph — this graph is not guaranteed to be a path, even if the training data is so. More closely related to us is the work of (Shi et al., 2024), which approach path planning problems by generating sequences of vertices to traverse. However, there is no guarantee that vertices at adjacent timesteps are adjacent to each other, and the method relies on a non-trivial beam-search post-processing algorithm in order to guarantee path validity.

3 Structured Discrete Representation of Paths

Choosing an appropriate data representation of paths is important for learning and sampling with diffusion models. A direct and naive approach is to treat each path as a subgraph of the original layered graph and learn the distribution of these sub-graphs with a diffusion model. This approach may be implemented using standard continuous diffusion models to learn a *continuous relaxation* of the adjacency matrix of a graph (Niu et al., 2020; Yan et al., 2024), the output of which may then be truncated or rounded off to yield a sampled subgraph. However, as we will show in section 5.1, such an approach results in the degradation of sample quality— many subgraphs do not correspond to paths in the layered graph.The key drawback in this naive approach is that it is oblivious to the underlying *structural constraints* distinguishing a path from a general (sub-)graph and instead relies on this "pattern" in the data to be learned by the diffusion model itself. Our PALM representation seeks to overcome this issue by explicitly perform learning on a representation that maps to paths.

Definition 3 (Padded Adjacency–List Matrix (PALM))**.** Given a layered graph $G = (\mathcal{V}, \mathcal{E})$ with L layers, let $V = |\mathcal{V}|$ and $\mathcal{V} = \{1, \ldots, V\}$. $D_v = \deg(v)$, where $\deg(\cdot)$ refers to the out-degree of a vertex. The edge set $\mathcal{E}$ can also be partitioned into exactly V sets $\{\mathcal{E}_1, \ldots, \mathcal{E}_V\}$, where each edge $e \in \mathcal{E}_v$ originates at vertex v. For a path $(v^1, \cdots, v^L)$, its PALM representation is a collection of vectors $\{x^1, \ldots, x^V\}$ where $x^v \in \mathbb{R}^{D_v}$ is a one-hot vector indicating an edge $e \in \mathcal{E}_v$ for all $v \in \mathcal{V}$.

At its core, a PALM is a stack of one-hot vectors recording a single outgoing edge of each vertex, as demonstrated in figure 2. It is easy to see that the PALM-to-path mapping is many-to-one and onto, which means (i) one PALM instance represents *exactly* one path, and (ii) all paths can be represented by at least one PALM. The path-to-PALM conversion is trivial by **Definition 3**; the PALM-to-path conversion is intuitive: starting by $v^l = v^1$ (which must be the first vertex appearing in any path per **Definition 1**), follow the edge $e = (v^l, v^{l+1})$ represented by x^l in the PALM and transit to v^{l+1}, and then repeat this process until reaching v^L.

Diffusion learning and inference with PALM. We employ the D3PM framework of (Austin et al., 2021) to learn a PALM representation. We conduct training and inference with PALM as the representation of paths. Concretely, a neural network is trained to take PALM as input and predict the logits of a distribution: $\tilde{p}_\theta(\tilde{x}_0 \mid x_t)$. We adopt the same parametrization as Austin et al. (2021):

$$p_\theta(x_{t-1} \mid x_t) \propto \sum_{\tilde{x}_0} q(x_{t-1}, x_t \mid \tilde{x}_0)\, \tilde{p}_\theta(\tilde{x}_0 \mid x_t). \tag{3}$$

To learn path distributions, we make the following changes from D3PM. First, since our dataset only stores paths but not PALM, we convert paths to PALM on the fly during training. Since the PALM-to-path mapping is many-to-one, we assign one hot vectors uniformly in rows (corresponding to vertices in the layered graph) that *do not* belong to the path. We adopt the uniform transition matrix (Austin et al., 2021; Hoogeboom et al., 2021) in the forward diffusion process, but the transition matrix at each timestep t for each node v is constructed differently according to their out-degrees, catering the varied length of vectors in PALM (see details in Appendix B). We use the common cosine noise scheduling (Nichol & Dhariwal, 2021) for the forward diffusion process. For training, we follow the combined loss proposed by Austin et al. (2021):

$$L_\gamma = \gamma L_{\text{vb}} + \mathop{\mathbb{E}}_{q(x_0)}\left[\mathop{\mathbb{E}}_{q(x_t|x_0)}\left[-\log p_\theta(\tilde{x}_0 \mid x_t)\right]\right], \tag{4}$$

where L_{vb} is the variation bound loss commonly adopted in diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Austin et al., 2021) and γ is a weighting parameter. Since we are learning distributions over paths rather than PALM, we only include losses incurred at vertices *on the path*.

We perform inference (sampling of paths) in PALM representation the same way as D3PM, with one extra step at the end that converts the sampled paths x_0 in PALM representation back to the common sequence form defined in **Definition 2**. Our implementation of D3PM is adapted from (Ryu, 2024) and (Niu et al., 2020; Yan et al., 2024).

4 Discrete Diffusion Guidance via Posterior Sampling

Guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2021) is a useful technique in diffusion-based conditional generation for steering the sampling process to data modes with the desired properties, *e.g.*, a certain class of pictures, or in our case, a collection of paths that can achieve high rewards under a reward function. In this work, we focus on the method of classifier guidance (Dhariwal & Nichol, 2021; Nichol et al., 2022) for compatibility to different external conditions (*e.g.*, different reward function instances) *without* the need for retraining the diffusion model.

Concretely, our objective of guided sampling is to generate paths that contain some or all of a set of prespecified "preferred" edges. This defines an implicit scalar reward function r over all possible paths, *i.e.*, the reward associated with a path is equal to the number of preferred edges it contains.

4.1 Guidance in Discrete Diffusion

There has not been an established principle for guidance in discrete diffusion. Unlike its counterpart in continuous domain where guidance under a given condition y can be interpreted as a combination of the Stein score $\nabla_x \log p(x)$ and the likelihood score $\nabla_x p(y \mid x)$. In discrete domains the (Stein) score is undefined, rendering this approach unfounded. The discrete generalization of the score (Meng et al., 2022; Lou et al., 2024), on the other hand, has not yet provided a principled way for guidance either. As such, we take a step back and return to the (logarithmic) Bayes' theorem *per se* to derive the posterior in discrete diffusion as recent efforts (Nisonoff et al., 2025) have attempted:

$$\underbrace{\log p(x_{t-1} \mid x_t, y)}_{\text{posterior}} = \underbrace{\log\left(p(y \mid x_t, x_{t-1}) / p(y \mid x_t)\right)}_{\text{likelihood ratio}} + \underbrace{\log p(x_{t-1} \mid x_t)}_{\text{prior}}, \tag{5}$$

where the prior $p(x_{t-1} \mid x_t)$ is approximated by a trained denoising diffusion model $p_\theta(x_{t-1} \mid x_t)$. Different yet akin to guidance in the continuous domain, the key to the posterior boils down to a *reasonable* and *efficient* approximation or direct computation of the log likelihood ratio $\log \frac{p(y|x_t, x_{t-1})}{p(y|x_t)}$.

Algorithm 1 Total Expected Reward Calculation

Inputs: PALM Logits z, Layered graph $G = (\mathcal{V}, \mathcal{E})$, Reward assignment PALM $[u]$
$[p_v] \leftarrow$ `get_transit_prob_dp`(G, z) ▷ Get transition prob. to each node v
for $v \in \mathcal{V}$ **do**
  $r_v \leftarrow p_v \cdot \mathbf{1}^\top u_v$
$\overline{R} \leftarrow \sum_{v \in V} r_v$
Return: $\overline{R}$

Algorithm 2 DDPS Inference

Inputs: Diffusion model $\hat{p}_\theta$ Reward model $\overline{R}$, Forward noise $Q_1, \cdots, Q_T$, Guidance scale λ
$x_T \sim \text{Cat}(\mathbf{1})$ ▷ Sample from uniform categorical distribution
for $t = T, T-1, \ldots, 1$ **do**
  $z \leftarrow \tilde{p}_\theta(\tilde{x}_0 | x_t)$ ▷ Predict x_0 logits with diffusion model
  $\bar{r} \leftarrow \overline{R}(z)$ ▷ Get expected rewards via **Algorithm 1**
  $g_t \leftarrow \nabla_z \bar{r}$
  $\tilde{x}_{t-1} \leftarrow (\prod_{\tau=1}^{t} Q_\tau) z$ ▷ Forward diffusion process
  $\tilde{x}'_{t-1} \leftarrow \lambda g_t + \tilde{x}_{t-1}$ ▷ Perform guidance to get posterior
  $x_{t-1} \sim \text{Cat}(\tilde{x}'_{t-1})$ ▷ Sample from the guided posterior
Return: x_0

4.2 Guidance Signal with PALM

Given a scalar reward function r we formulate condition y in terms of reward optimality to influence edge selection probabilities. While directly increasing the sampling probabilities of preferred edges appears intuitive, this approach is inadequate since edge selection depends on the choices made in preceding layers. Consider a preferred edge $e = (v^l, v^m)$ in a path p. When its preceding edges $e' = (v^k, v^n)$, where $k < l$, have negligible selection probabilities, path p becomes effectively inaccessible regardless of the selection probability of edge e.

Thus, we opt for utilizing the expectation of total rewards $\overline{R}$ given a distribution of PALM, involving all the probabilities of selecting each edge in the graph. These probabilities are all encoded in the categorical distributions' logits z yielded by the diffusion model $\tilde{p}_\theta(\tilde{x}_0 \mid x_t)$. Further, we propose to leverage the gradient of the expected total rewards with respect to the logits z, to approximate the log likelihood ratio as guidance signal:

$$\log\left(p\left(y \mid x_t, x_{t-1}\right) / p\left(y \mid x_t\right)\right) \approx \nabla_z \overline{R}(z). \tag{6}$$

Approximation (6) results in the reward-guided sampling process

$$\log p_\theta\left(x_{t-1} \mid x_t, y\right) = \lambda \nabla_z \overline{R}(z) + \log p_\theta\left(x_{t-1} \mid x_t\right), \quad \text{where} \quad z = \tilde{p}_\theta\left(\tilde{x}_0 \mid x_t\right). \tag{7}$$

Algorithm 2 describes this guided sampling procedure.

5 Experiments

In this section, we address the following research questions:

RQ1: In terms of quality of generated samples, how does our proposed PALM representation compared to alternatives like adjacency matrix learning (Niu et al., 2020; Yan et al., 2024)?

RQ2: Is the proposed DDPS method effective for reward improvement?

RQ3: When utilizing guidance, what is the tradeoff between reward improvement and adherence to the original learned distribution?

Problem instances and data. We ran experiments on the following 3 problem instances.

- `Toy` was synthetically generated by randomly truncating edges on a layerwise-fully-connected layered graph. It features 11 layers and 41 vertices.

METHOD	Valid Rate (VR) % ↑
EDP-GNN	0.00
SWINGNN	99.7
DDPS (Ours)	**100**

Table 1: Valid rate comparisons in unconditional path generation. Each method generates 8192 samples for validity check.

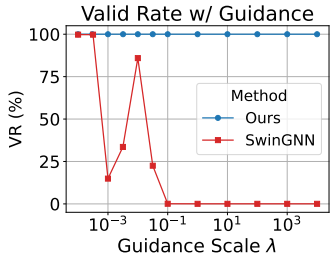


Figure 3: Valid rates for baseline and our method under different guidance strengths.

- `Heights` and `Ridge` are layered graphs converted from real-world city maps using the method of (Černý et al., 2024). `Heights` is a middle-scale graph with 12 layers and 423 vertices while `Ridge` is larger, with 13 layers and 4293 total vertices. These graphs are relatively sparse, with out-degree of each vertex mostly being less than 6.

The `Toy` dataset comprises all unique paths in the graph, while the dataset for `Heights` (~120k paths) and `Ridge` (~347k paths) were obtained by sampling them *nonuniformly* over all paths. Note that in both cases, the number of paths are orders of magnitude higher. We employ UNet-like neural networks (Ronneberger et al., 2015) as the backbone of the diffusion models, use 256 diffusion timesteps and train them with the AdamW optimizer (Loshchilov & Hutter, 2017). All experiments were run on an AMD Ryzen Threadripper PRO 5995WX 64-Core CPU, 504 GB RAM, and 2 NVIDIA RTX A6000 GPUs each with 48GB GPU memory using PyTorch (Paszke et al., 2019).

5.1 STRUCTURED DISCRETE REPRESENTATION FOR GENERATION QUALITY

To address **RQ1**, we compared the sample quality of our proposed method to diffusion baselines that operate on the continuous relaxation of the discrete data domain. By sample quality, we refer to the proportion of *valid paths* that are generated. Recall from section 3 that by construction, our proposed method generates valid paths with certainty.

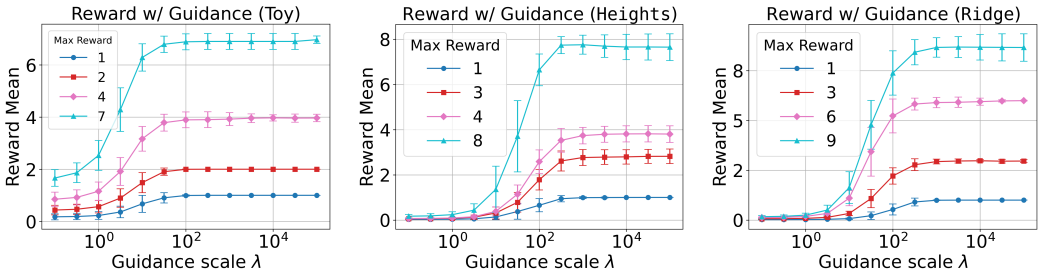
Unconditional path generation. We compared our method to baselines EDP-GNN (Niu et al., 2020) and SWINGNN (Yan et al., 2024). Both EDP-GNN and SWINGNN generate adjacency matrices rather than paths.[1] Thus, there is non-zero probability where the edges included in the adjacency matrix do not form a single path in the layered graph. For each of the 3 methods, we trained a denoising diffusion model using the same training data, with the only difference in how paths were represented internally. For this part, no guidance was used, *i.e.*, only unconditional sampling was tested. To evaluate sample quality, we generated a set S of N samples and report the valid rate (VR):

$$\text{VR} \triangleq \frac{\sum_{s \in S} \mathbb{1}\left[s \text{ is a path}\right]}{N} \times 100\%. \tag{8}$$

We used a dataset of size 1350 with at 80-20 train-validation split. The results are reported in Table 1. As expected, our method yields a VR of 100% since our approach by construction guarantees a valid path. EDP-GNN yields a VR of 0, meaning that essentially no valid paths were generated. Surprisingly, SWINGNN generates valid paths almost all the time.

Conditional path generation with reward guidance. To further investigate this phenomenon, we consider the more complex task of conditional path generation, *i.e.*, diffusion *with guidance* and compare our method to SWINGNN for various levels of guidance scales λ. We utilize the models trained earlier for sampling under reward predictor guidance as condition generation, following the general idea of classifier guidance (Dhariwal & Nichol, 2021). We adopt a simple reward setup: a path obtains 1.0 reward if it contains a specific edge in the layered graph. Since SWINGNN does not originally support guidance, we adopt a simplified approach described by Ma et al. (2024) and

[1] EDP-GNN learns distributions of adjacency matrices of graphs by training a graph neural network via score matching and sampling with annealed Langevin dynamics. SWINGNN improves upon EDP-GNN by leveraging a graph transformer to learn the score of adjacency matrices distributions and perform sampling from the the perspective of diffusion stochastic differential equations.



(a) `Toy` (b) `Heights` (c) `Ridge`

Figure 4: Average reward values under various guidance scales for different reward configurations. Error bars denote standard deviations.

leverage the gradients of rewards achieved by noisy samples as guidance signals. Guided sampling with DDPS is implemented as described in **Algorithm 2**. [2]

The results for this setting are reported in Figure 3. When λ is small, both methods generated valid paths almost all the time, but as λ increases, SwinGNN's performance drops (surprisingly, not monotonically in λ) to zero. In contrast, our method guarantees perfect valid rate by construction regardless of λ. These results point toward the brittle nature of continuous-state approximations for inherently discrete variables.

5.2 Discrete Posterior Sampling for Reward Improvement

To assess the ability of DDPS in reward improvement, we train diffusion models for all 3 problem instances, `Toy`, `Heights`, `Ridge`.[3] We follow the approach in section 4 by assigning **binary** rewards to edges in a layered graph. An edge with a reward of 1 indicates that paths containing it are favored, and DDPS should guide the sampling process into including these favored edges. We adopt 4 reward configurations, each specified by the maximum possible reward a path can ever achieve. For example, in `Toy`, these four configurations correspond to a maximum reward of 1,2,4, or 7 (figure 4(a)). For each problem instance, we took each configuration and ran guided sampling with different reward function instances, 20 for `Toy` and `Heights`, and, 10 for `Ridge`.These classes of configurations characterize a range of reward sparsity— from nearly half of all possible paths achieving maximum rewards to the case where a path chosen u.a.r. achieves it with odds poorer than one over ten million.

For each layered graph, we used the trained discrete diffusion model as the base sampler, and perform guided sampling with DDPS. We evaluate reward improvement by taking the empirical means of rewards with 65536 samples for `Toy`, 4096 for `Heights` and 2048 for `Ridge`.

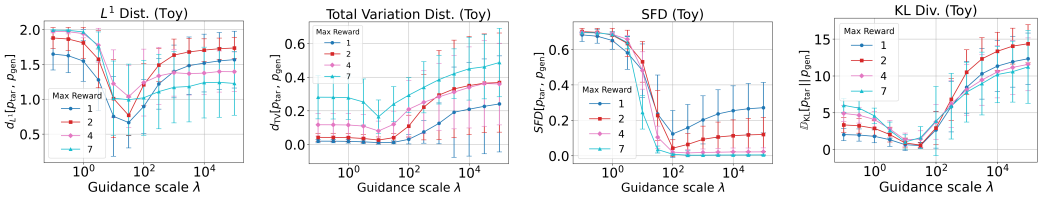
We present in figure 4 the average rewards for different reward functions and under different guidance scales of DDPS. The trends observed in figure 4 agree with what we expect: with stronger guidance, the higher reward one obtains eventually plateauing at the maximum obtainable reward. The "S"-shaped curve (note the logarithmic x-axis) suggests that when λ is small then the gradient of the obtained reward with respect to λ is small as well. Overall, our results suggest that DDPS leads to reward improvement as long as a sufficiently large guidance scale is selected.

5.3 Trade-offs between Reward Objective and Learned Distribution

While rewards obtained are highest with a large enough guidance scale λ, this comes at the expense of deviating from the original (*i.e.*, without guidance) distribution. We examine this tradeoff empirically by examining path distributions as λ varies. To do so, we introduce the idea of a *target distribution*. A desirable target distribution in our case for reward-guided sampling would be the posterior distribution of paths given the true likelihood of our desired property (*i.e.*, obtaining max rewards) and the learned prior (without guidance). In our context, we look at the distribution of

[2] Since SwinGNN uses an internal representation distinct from PALM, guidance scale technically cannot be compared across both methods.

[3] For `Toy`, the model was trained separately from section 5.1.



(a) $d_{L^1} \downarrow$ (b) $d_{\text{TV}} \downarrow$ (c) SFD $\downarrow$ (d) KL $\downarrow$

Figure 5: Metrics in path distributions for the small-size graph `Toy`. The error bars represent standard deviations across different reward function instances within the reward configuration class (20 instances per configuration). The sample size is 65536.

paths when $\lambda = 0$, *conditioned on them receiving the maximum achievable reward* (e.g., 1,2,4,7 for `Toy` in figure 4(a)) and compare that to the distribution of paths obtained when $\lambda > 0$.

We appeal to the following metrics between probability distributions to show the gap between the target and generation distributions $p_{\text{tar}}(x)$ and $p_{\text{gen}}(x)$,

- KL divergence $\mathbb{D}_{\text{KL}}\left[p_{\text{tar}}(x) \parallel p_{\text{gen}}(x)\right]$;
- L^1 distance: $d_{L^1}[p_{\text{tar}}(x), p_{\text{gen}}(x)] = \sum_x \left|p_{\text{tar}}(x) - p_{\text{gen}}(x)\right|$;
- Total Variation distance: $d_{\text{TV}}\left[p_{\text{tar}}(x),\ p_{\text{gen}}(x)\right] = \sup_x \left|p_{\text{tar}}(x) - p_{\text{gen}}(x)\right|$;
- Spearman's Footrule Distance: $\text{SFD}^{\pi}\left[p_{\text{tar}}(x),\ p_{\text{gen}}(x)\right]$. See **Definition 5** in Appendix B.3 for a formal definition. This metric covers the *relative significance* of elements in terms of probability.

However, as the size of layered graph increases, the number of total possible paths and the support of path distributions grows exponentially. This makes it computationally intractable to estimate $p_{\text{tar}}(x)$ and $p_{\text{gen}}(x)$ by sampling. Therefore, for such large graphs we evaluate distributions of certain *features* of the sampled paths, where these features are much lower-dimensional:

- Fréchet layered graph distance (FLGD) is an adaptation of the Fréchet Inception distance (FID) in image generation. The feature vector of a sample path is obtained by flattening the PALM representation of the sample and zeroing out all entries belonging to vertices outside the path.
- Layer Imitation Score (IS-L) aggregates across layers the marginal distributions over vertices in the each layer. We employ four variants based on the measure: IS-L-L^1, IS-L-KL, IS-L-TV, and IS-L-SF, each corresponding to the metrics used on path distributions as discussed above. A formal definition of IS-L is in **Definition 4** in Appendix B.3.

Figure 5 shows our results for `Toy` for all 4 metrics. The same trend is observed: as λ increases, we see the distributions getting closer. This trend decreases until around $\lambda = 100$, after which distances rise and eventually plateau. The shapes of the curves suggest a "sweet spot" balancing the reward objective and adherence to the learned prior distribution.

Figure 6 in Appendix C shows our results using feature distributions for `Toy`, `Heights`, and `Ridge`. By and large, they reveal a similar pattern— except for scores calculated using the SFD, the scores decay as the λ increases and achieve a "sweet spot" before rising and eventually plateauing. With the exception of IS-L-SF, the "sweet spots" are marked by the golden dashed lines, whose location approximately agree. This value of λ achieves near-optimal reward. This suggests that with the right amount of guidance, DDPS achieves a good tradeoff between reward and adherence to the original distribution (conditioned on attaining optimal reward).

6 Conclusion

In this paper we propose DDPS, which utilizes the PALM representation to guarantee that samples from a discrete diffusion model do indeed correspond to paths in a layered graph. Our preliminary results look favorable and we show that with classifier guidance, we are able to achieve high rewards given some reward function while still maintaining reasonable adherence to the learned distribution. Future work includes extending this to more general problems beyond path generation, or involving more elaborate constraints or rewards on the generated paths.

REFERENCES

Anthropic. Claude 3.5 Sonnet. Large Language Model, 2025. URL https://claude.ai.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. Advances in Neural Information Processing Systems, 34:17981–17993, 2021.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In Proceedings of the 40th International Conference on Machine Learning, volume 202 of Proceedings of Machine Learning Research, pp. 1276–1301. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/avdeyev23a.html.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. Advances in Neural Information Processing Systems, 35:28266–28279, 2022.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. arXiv preprint arXiv:2402.04997, 2024.

Jakub Černỳ, Chun Kai Ling, Darshan Chakrabarti, Jingwen Zhang, Gabriele Farina, Christian Kroer, and Garud Iyengar. Contested logistics: A game-theoretic approach. In International Conference on Decision and Game Theory for Security, pp. 124–146. Springer, 2024.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research, pp. 02783649241273668, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), Advances in Neural Information Processing Systems, volume 34, pp. 8780–8794, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

Persi Diaconis and Ronald L Graham. Spearman's footrule as a measure of disarray. Journal of the Royal Statistical Society Series B: Statistical Methodology, 39(2):262–268, 1977.

Zeyu Feng, Hao Luan, Pranav Goyal, and Harold Soh. LTLDoG: Satisfying temporally-extended symbolic constraints for safe diffusion-based planning. IEEE Robotics and Automation Letters, 9(10):8571–8578, 2024. doi: 10.1109/LRA.2024.3443501.

Zeyu Feng, Hao Luan, Kevin Yuchen Ma, and Harold Soh. Diffusion meets options: Hierarchical generative skill composition for temporally-extended tasks. In 2025 International Conference on Robotics and Automation (ICRA), 2025. URL https://openreview.net/pdf?id=WjjoYyJjJW.

Nathan C. Frey, Dan Berenberg, Karina Zadorozhny, Joseph Kleinhenz, Julien Lafrance-Vanasse, Isidro Hotzel, Yan Wu, Stephen Ra, Richard Bonneau, Kyunghyun Cho, Andreas Loukas, Vladimir Gligorijevic, and Saeed Saremi. Protein discovery with discrete walk-jump sampling. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=zMPHKOmQNb.

Jun Han, Zixiang Chen, Yongqian Li, Yiwen Kou, Eran Halperin, Robert E Tillman, and Quanquan Gu. Guided discrete diffusion for electronic health record generation. arXiv preprint arXiv:2404.12314, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), Advances in Neural Information Processing Systems, 2021. URL https://openreview.net/forum?id=6nbpPqUCIi7.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-LM improves controllable text generation. In Advances in neural information processing systems, volume 35, pp. 4328–4343, 2022.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. arXiv preprint arXiv:2408.08252, 2024. URL https://openreview.net/forum?id=2fgzf8u5fP.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pp. 32819–32848. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Jiajun Ma, Tianyang Hu, Wenjia Wang, and Jiacheng Sun. Elucidating the design space of classifier-guided diffusion generation. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=9DXXMXnIGm.

Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), Advances in Neural Information Processing Systems, 2022. URL https://openreview.net/forum?id=_RL7wtHkPJK.

Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Bac Nguyen, Stefano Ermon, and Yuki Mitsufuji. G2D2: Gradient-guided discrete diffusion for image inverse problem solving. arXiv preprint arXiv:2410.14710, 2024. URL https://openreview.net/forum?id=mZfBRjMWq0.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In International conference on machine learning, pp. 8162–8171. PMLR, 2021.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pp. 16784–16804. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/nichol22a.html.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=XsgHl54yO7.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, volume 108, pp. 4474–4484. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/niu20a.html.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32, 2019.

Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Cheng-Hao Liu, Sarthak Mittal, Nouha Dziri, Michael M. Bronstein, Pranam Chatterjee, Alexander Tong, and Joey Bose. Steering masked discrete diffusion models via discrete denoising posterior prediction. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=Ombm8S40zN.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, pp. 234–241. Springer, 2015.

Simo Ryu. Minimal implementation of a D3PM (structured denoising diffusion models in discrete state-spaces), in PyTorch. https://github.com/cloneofsimo/d3pm, 2024.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P de Almeida, Alexander M Rush, Thomas PIERROT, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=i5MrJ6g5G1.

Dingyuan Shi, Yongxin Tong, Zimu Zhou, Ke Xu, Zheng Wang, and Jieping Ye. Graph-constrained diffusion for end-to-end path planning. In The Twelfth International Conference on Learning Representations, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In International conference on machine learning, pp. 2256–2265. PMLR, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pp. 46495–46513. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/stark24b.html.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. arXiv preprint arXiv:2501.09685, 2025.

Jakub Černý, Chun Kai Ling, Christian Kroer, and Garud Iyengar. Layered graph security games. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/298. URL https://doi.org/10.24963/ijcai.2024/298.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=UaAD-Nu86WX.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Avantika Lal, Tommi Jaakkola, Sergey Levine, Aviv Regev, Hanchen, and Tommaso Biancalani. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=G328D1xt4W.

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. In 8th Annual Conference on Robot Learning, 2024. URL https://openreview.net/forum?id=wD2kUVLT1g.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=PzcvxEMzvQC.

Qi Yan, Zhengyang Liang, Yang Song, Renjie Liao, and Lele Wang. SwinGNN: Rethinking permutation invariance in diffusion models for graph generation. Transactions on Machine Learning Research, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=abfi5plvQ4.

Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation. In Proceedings of the 40th International Conference on Machine Learning, volume 202 of Proceedings of Machine Learning Research, pp. 40001–40039. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/yim23a.html.

Youzhi Zhang, Bo An, Long Tran-Thanh, Zhen Wang, Jiarui Gan, and Nicholas R Jennings. Optimal escape interdiction on transportation networks. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 3936–3944, 2017.

A Acknowledgments

We acknowledge the usage of Claude (Anthropic, 2025) to generate the term and acronym "padded adjacency-list matrix (PALM)." In addition, we used Claude (Anthropic, 2025) to polish a few paragraphs in section 4 and section B.3.

B Experiment Details

B.1 Data

Datasets used in section 5 was generated synthetically. The two small layered graphs (one used in section 5.1 and `Toy`) were obtained by first construct a layerwise-fully-connected graphs and then pruning 50% of all the edges within. The pruning process is to choose edges in the graph uniformly at random, and then remove it. Checks were performed to ensure the resulting pruned graph meets definition of layered graphs and there exists paths from the first layer to the last layer. All possible unique paths in the two pruned graphs were included in the datasets. For `Heights` and `Ridge`, we first obtain all possible unique paths in the layered graphs by searching, and then sample paths from all possible paths uniformly at random without replacement. After that, we duplicate the chosen paths by different numbers for each to create the nonuniform pattern in the dataset.

B.2 Diffusion Model Implementation Details

We provide implementation details regarding training and inference of discrete diffusion models with PALM.

Transition matrix construction For vertices v with positive out-degree(s) ($D_v > 0$):

$$[Q_t^v]_{ij} = \begin{cases} 1 - \dfrac{D_v - 1}{D_v}\beta_t, & \text{if} \quad 1 \le i = j \le D_v \\ \dfrac{1}{D_v}\beta_t, & \text{if} \quad 1 \le i \ne j \le D_v \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

For vertices with zero out-degree ($D_v = 0$):

$$Q_t^v = \boldsymbol{I}_D. \tag{10}$$

Note that for each transition matrix Q_t^v, its upper-left $D_v \times D_v$ block is a doubly stochastic square matrix.

Inference with PALM The only difference between the inference process of DDPS and that of D3PM lies in that for each node v, we only take the first D_v logits (corresponding to the D_v edges originating at v) to construct the categorical distribution for sampling at each time step, since other entries are merely paddings and do not represent valid edges in the graph.

B.3 Metrics

We define here some metrics we used in the experiments in section 5.3.

Definition 4 (Layer Imitation Scores (IS-L))**.** Let $p\left(v^{(l)}\right)$, $\forall l \in \{1, \dots, L\}$ be the marginal distribution of vertices within layer l of a layered graph with L layers in total. Given a path x, let the likelihood of vertices be the Dirac delta on the vertex on this path $x^{(l)}$ for each layer:

$$p\left(v^{(l)} \mid x\right) = \delta\left(v^{(l)} - x^{(l)}\right), \quad \forall l = 1, \dots, L. \tag{11}$$

Then, for a statistical dissimilarity measure $d[p, q] : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ between two discrete probability distributions, the IS-L is defined as

$$\text{IS-L} \triangleq \sum_{l=1}^{L} d\left[\mathop{\mathbb{E}}_{w \sim p_{\text{tar}}(x)}\left[p\left(v^{(l)} \mid w\right)\right], \mathop{\mathbb{E}}_{w' \sim p_{\text{gen}}(x)}\left[p\left(v^{(l)} \mid w'\right)\right]\right]. \tag{12}$$

Definition 5 (Spearman's Footrule Distance (SFD), adapted from (Diaconis & Graham, 1977) [4]). Given two discrete probability distributions p and q over a finite support $S = \{1, \ldots, n\}$, let π_p and π_q be the permutations of S induced by sorting p and q in descending (or ascending) order by probability mass, respectively. Specifically:

- $\pi_p(i)$ represents the position of support element i in the sorted ordering of p;
- $\pi_q(i)$ represents the position of support element i in the sorted ordering of q.

The Spearman's footrule distance between distributions p and q is then defined as:

$$\mathrm{SFD}^{\pi}[p, q] \triangleq \frac{1}{M(n)} \sum_{i \in S} \left| \pi_p(i) - \pi_q(i) \right|, \tag{13}$$

where $M(n)$ is the maximal possible distance, which occurs when one permutation is the reverse of the other:

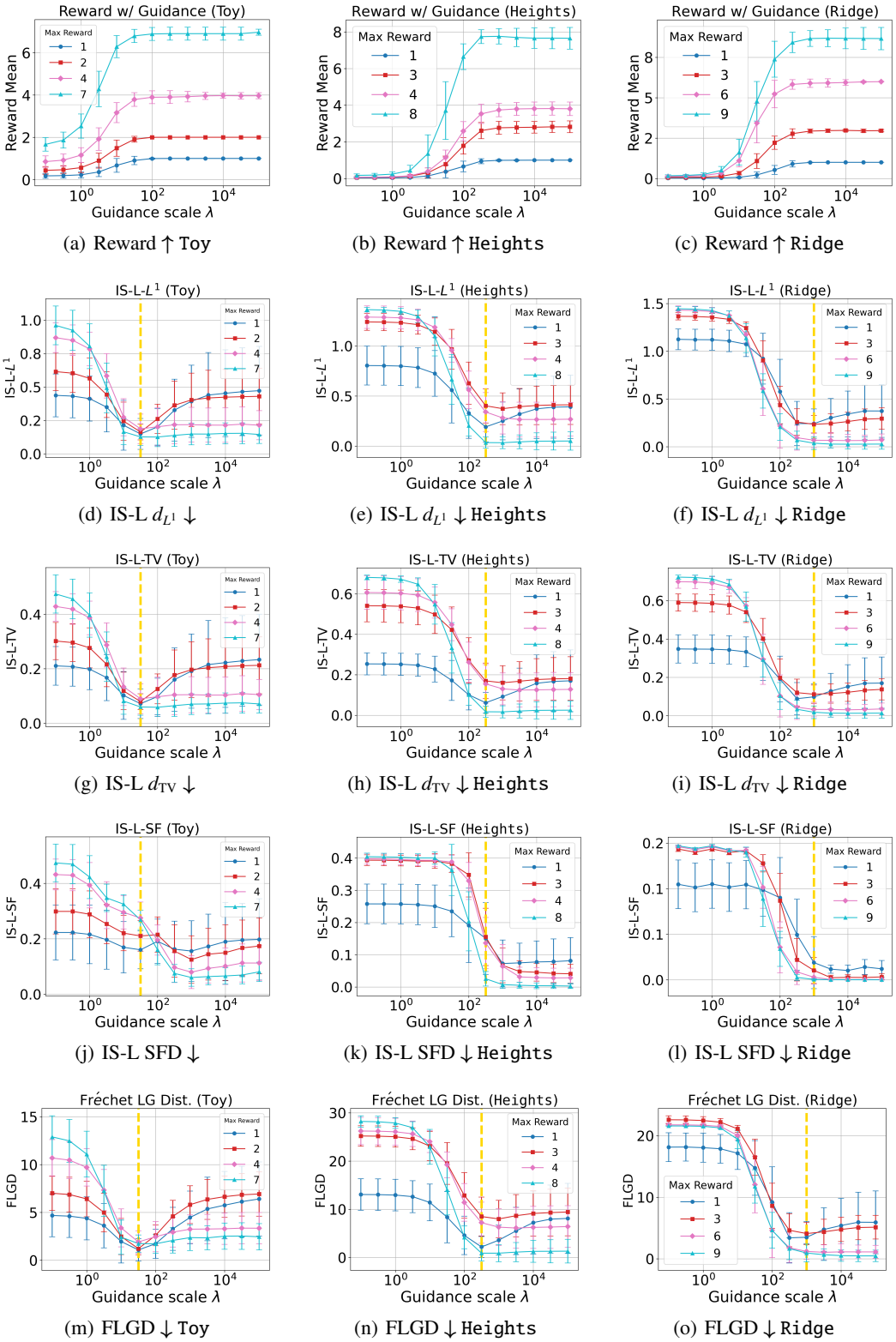
$$M(n) = \begin{cases} \frac{n^2}{2}, & \text{if } n \text{ is even} \\ \frac{n^2-1}{2}, & \text{if } n \text{ is odd} \end{cases}$$

Remark 1. We acknowledge the SFD is *not* invariant to permutations of elements with same probabilities. As such, we utilize the same tie-breaking mechanism across all experiments to make fair comparisons.

C Additional Experiment Results

We include experiment results involving all feature-based metrics in this section.

[4] The SFD defined in **Definition 5** is adapted from, but slightly different to the Spearman's footrule in (Diaconis & Graham, 1977, Eq.(1.1)) which is defined over two non-parametric permutations. In contrast, the two permutations involved in the SFD herein are parameterized by two probability distributions.



(a) Reward ↑ `Toy` (b) Reward ↑ `Heights` (c) Reward ↑ `Ridge`

(d) IS-L d_{L^1} ↓ (e) IS-L d_{L^1} ↓ `Heights` (f) IS-L d_{L^1} ↓ `Ridge`

(g) IS-L d_{TV} ↓ (h) IS-L d_{TV} ↓ `Heights` (i) IS-L d_{TV} ↓ `Ridge`

(j) IS-L SFD ↓ (k) IS-L SFD ↓ `Heights` (l) IS-L SFD ↓ `Ridge`

(m) FLGD ↓ `Toy` (n) FLGD ↓ `Heights` (o) FLGD ↓ `Ridge`

Figure 6: Additional experiments with guidance on `Toy`, `Heights` and `Ridge`. The first row contains the rewards obtained. The next 4 rows are based on FLGD and IS-L metrics.