IMPROVING CLASSIFIER-FREE GUIDANCE IN MASKED DIFFUSION: LOW-DIM THEORETICAL INSIGHTS WITH HIGH-DIM IMPACT

Anonymous authors
Paper under double-blind review

ABSTRACT

Classifier-Free Guidance (CFG) is a widely used technique for conditional generation and improving sample quality in continuous diffusion models, and its extensions to discrete diffusion has recently started to be investigated. In order to improve the algorithms in a principled way, this paper starts by analyzing the exact effect of CFG in the context of a low-dimensional masked diffusion model, with a special emphasis on the guidance schedule. Our analysis shows that high guidance early in sampling (when inputs are heavily masked) harms generation quality, while late-stage guidance improves it. These findings provide a theoretical explanation for empirical observations in recent studies on guidance schedules. The analysis also reveals an imperfection of the current CFG implementations. These implementations can unintentionally cause imbalanced transitions, such as unmasking too rapidly during the early stages of generation, which degrades the quality of the resulting samples. To address this, we draw insight from the analysis and propose a novel classifier-free guidance mechanism. Intuitively, our method smooths the transport between the data distribution and the initial (masked) distribution, resulting in improved sample quality. Remarkably, our method is achievable via a simple one-line code change. Experiments on conditional image and text generation empirically confirm the efficacy of our method.

1 INTRODUCTION

Continuous-state diffusion models (Ho et al., 2020; Song et al.) have proven effective in both unconditional and conditional generation tasks, such as generating data from natural language prompts. Prominent examples include text-to-image and text-to-video models like Stable Diffusion, Sora, and others (Rombach et al., 2022; Esser et al., 2024; Brooks et al., 2024). More recently, progress in discrete diffusion modeling (Campbell et al., 2022; Lou et al., 2023; Huang et al., 2023; Gruver et al., 2023; Ou et al., 2024; Shi et al., 2024; Sahoo et al., 2024) has extended the applicability

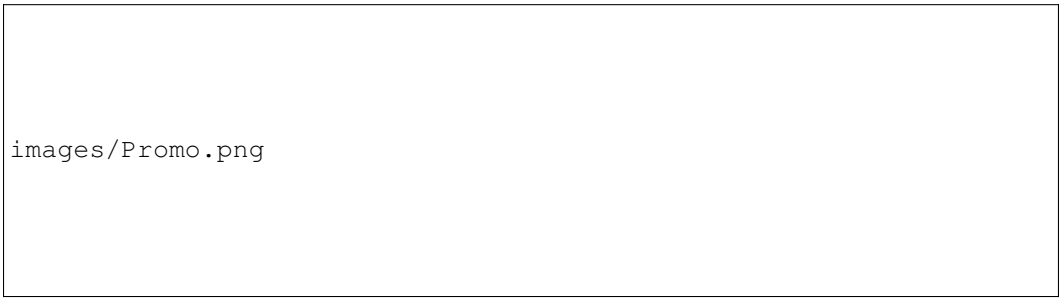


Figure 1: We proposed an improved guidance mechanism through column normalization. Our method produces sharper images while being more stable to the guidance strength. Notably, it requires only a minor code modification.

of diffusion-based generation to new domains, including molecular design, protein synthesis, and languages.

Despite their success, these models often produce outputs that lack fine detail or strong alignment with conditioning inputs (e.g., text prompts). A widely adopted technique to address this issue is classifier-free guidance (CFG) (Ho and Salimans, 2021), which improves fidelity but typically at the cost of reduced sample diversity (Karras et al., 2024).

A growing body of work has sought to understand the theoretical foundations of CFG in diffusion models (Chidambaram et al., 2024; Pavasovic et al., 2025; Bradley and Nakkiran, 2024; Ye et al., 2025), while others have developed improved guidance algorithms (Karras et al., 2024; Li et al., 2024). Classifier-free guidance has also been adapted to discrete diffusion models (Nisonoff et al., 2024; Schiff et al., 2024), yielding promising empirical gains.

Among these improvements, dynamic guidance schedules—where guidance strength varies over the generation trajectory—have shown especially effective. Strategies such as guidance intervals (Kynkäänniemi et al., 2024) and gradually increasing schedules (Xi et al., 2024) can significantly enhance sample quality and are increasingly adopted in practice (Hoogeboom et al., 2024; Yu et al., 2024; Karras et al., 2024). However, such scheduling techniques remain exclusive to the continuous setting.

While recent adaptations of CFG to discrete diffusion have improved empirical performance, defining and optimizing effective guidance strategies in discrete spaces remains a fundamentally challenging and open research problem.

In our work we aim to better understand the mechanisms by which guidance affects the sampling process in discrete diffusion. Specifically, we aim to answer the following questions:

- How does the guidance schedule affect the distribution of the generated samples?
- Is it possible to characterize properties of good guidance schedules?

To do so, we start by deriving explicit formulas for the sampled distribution under varying guidance schedules in 1 and 2 dimensions. Our analysis not only reveals flaws in current CFG implementations, but also leads to effective design principles for effective guidance schedules in masked diffusion. Our contributions can be summarized as:

- We identify a key flaw in existing discrete guidance mechanisms that complicates simulation, and provide a theoretical explanation of its cause.
- To address the flaw, we propose a novel classifier-free guidance mechanism based on a simple yet principled column normalization of the rate matrix. This change is theoretically justified, easy to implement (pseudocode in Sec.1), and compares favorably to existing approaches in practice.
- The first theoretical justifications to characterize guidance schedules and the mechanisms by which they improve sample generation

```
def normalized_guidance_euler_transition(
        x, c, t, dt, w
    ):
    uncond = model(x, cond=None)
    cond = model(x, cond=c)
    logits = w * cond + (1 - w) * uncond

    p_theta = logits.softmax(dim=-1)

    s, s_bar = sigma(t), sigma_bar(t)
    change = dt * s * (1 - exp(-s_bar))
    return sample(delta(x) + change * p_theta)
```

Listing 1: Our guidance in the special case of masked diffusion using Euler transitions. Our method is a simple one line change *but clearly motivated by theory*

.

```
def other_guidance_euler_transition(
        x, c, t, dt, w
    ):
    uncond = model(x, cond=None)
    cond = model(x, cond=c)
    logits = w * cond + (1 - w) * uncond

    p_theta = logits.exp()

    s, s_bar = sigma(t), sigma_bar(t)
    change = dt * s * (1 - exp(-s_bar))
    return sample(delta(x) + change * p_theta)
```

Listing 2: Unlocking/Simple guidance for the special case of masked diffusion using Euler transitions.

2 PRELIMINARIES

This paper considers a vocabulary of size M and state space $S = \{1, 2, \dots, M\}^d$, with each element being a sequence of tokens. The number of tokens d will also be referred to as the dimension. Each probability distribution on S is represented as a vector in $\mathbb{R}^{M^d}$ whose entries sum to one.

2.1 INTRODUCTION TO DISCRETE DIFFUSION VIA CTMC

Given an initial distribution $p \in \mathbb{R}^{M^d}$, discrete diffusion is defined by considering a rate matrix $R_t \in \mathbb{R}^{M^d \times M^d}$ and defining a continuous time Markov chain (CTMC):

$$\frac{\mathrm{d}p_t}{\mathrm{d}t} = R_t p_t, \quad p_0 = p. \tag{1}$$

we pick R_t such that when $t \to \infty$, p_t converges to a simple distribution. Additionally, R_t must satisfy that its non-diagonal entries are non-negative and each column must add up to zero. The time reversal of this process corresponds to a different CTMC given by:

$$\frac{\mathrm{d}p_{T-t}}{\mathrm{d}t} = \overline{R}_{T-t} p_{T-t}. \tag{2}$$

This process is considered as the time reversal since it has the same law as (1) for all values of t and the reverse transition matrix can be found through the following identities:

$$\overline{R}_t(y, x) = R_t(x, y) \cdot \frac{p_t(y)}{p_t(x)}, \qquad \overline{R}_t(x, x) = -\sum_{y \neq x} \overline{R}_t(y, x). \tag{3}$$

The ratios $\frac{p_t(y)}{p_t(x)}$ are called the concrete score and they enable sampling through Euler schemes, τ-leaping (Lou et al., 2023) or higher order methods (Ren et al., 2025).

Masked Discrete Diffusion is a special case of diffusion where a clean sequence x_0 is gradually corrupted over time by randomly masking some of its entries. Typically, the forward process is chosen such that at time $t = 0$, the data is completely unmasked, and at $t = T$ the data is completely masked. Formally, the distribution of each token can be written in a simple form:

$$p_t(x_t^i | x_0) = \begin{cases} x_0^i & \text{with probability } e^{-\overline{\sigma}_t} \\ M & \text{with probability } 1 - e^{-\overline{\sigma}_t} \end{cases}$$

Where $\overline{\sigma}_t$ is an increasing function that defines the unmasking schedule. The forward dynamics are defined such that tokens transition only from a clean state to a masked state, remaining masked thereafter. Generation is achieved by starting from a fully masked state and iteratively unmasking tokens until a clean sequence is recovered by following Equation (2).

Masked diffusion enjoys a simple and structured design, which has enabled its successful scaling to large practical tasks (Nie et al.; Xie et al., 2025; Ou et al., 2024; Sahoo et al., 2024; Shi et al., 2024; Campbell et al., 2022). For this reason, we adopt it as the primary setting for our analysis.

2.2 CLASSIFIER-FREE GUIDANCE

Classifier-free guidance (CFG) (Ho and Salimans, 2021) was introduced to improve conditional diffusion models, like generating images from class labels or text. Models often failed to capture fine details, which led to less accurate and misaligned samples (Karras et al., 2024).

CFG tackles this by comparing predictions with and without conditioning, and biasing generation toward the conditional signal. Formally, the method defines a reweighted distribution:

$$p^{(w)}(x|y) \propto p^w(x|y) p^{1-w}(x)$$

Where w is called the guidance strength. Setting $w = 1$ recovers the usual conditional distribution $p(x|y)$ while $w = 0$ corresponds to unconditional sampling. The crucial insight is that by setting $w > 1$ it is possible to emphasize the conditional part, effectively pulling the generation closer to satisfying the required condition. CFG is now a standard tool in conditional diffusion models, more controllable generations across tasks such as text-to-image synthesis.

While the original formulation contrasted the conditional model against its unconditional counterpart, later works recognized that this can be extended by replacing the unconditional distribution with other distributions. For example, Karras et al. (2024) used a weaker conditional model as the guiding distribution. This view has led to the understanding that the essence of guidance lies in balancing a **target distribution** p with a **guiding distribution** q.

$$p^{(w)}(x) \propto p^w(x)q^{1-w}(x) \tag{4}$$

This view highlights that the unconditional model is simply one possible choice of q. By carefully selecting q recent works (Karras et al., 2024; Li et al., 2024; Rojas et al., 2025) have proposed novel guidance strategies that further improve sample quality and control.

2.3 Guidance for discrete diffusion models

In parallel to advances in continuous domains, discrete diffusion models have emerged as powerful generative models, enabling diffusion-based approaches on modalities that were previously out of reach—most notably, text. Improving the fidelity and controllability of these models is crucial, and guidance offers a natural path forward. Extending classifier-free guidance to the discrete setting has therefore become an active line of research with two main approaches having been proposed, which we describe below, followed by a discussion in Section 3.3 comparing them to our method.

Unlocking Guidance (Nisonoff et al., 2024) introduced the first classifier-free guidance mechanisms for discrete diffusion models. Inspired by the continuous case, they constructed a guided backwards transition by interpolating between two transition matrices in equation 2, yielding

$$\overline{R}_t^{(w)}(y,x) = R_t(x,y) \cdot \left(\frac{p_t(y)}{p_t(x)}\right)^w \left(\frac{q_t(y)}{q_t(x)}\right)^{1-w}, \qquad \overline{R}_t^{(w)}(x,x) = -\sum_{y \neq x} \overline{R}_t^{(w)}(y,x), \tag{5}$$

where p_t, q_t follows the forward CTMC (1). Here $p_0 = p$ is the distribution that we want to generate from and q serves as the guiding distribution. [1]. Notice how the products mimic those present in equation 4. A useful way to interpret this is by introducing the notion of the **tilted distribution**:

$$p^{(w)}(x) = Z_w^{-1} p^w(x) \cdot q^{1-w}(x), \qquad Z_w = \sum_{y \in S} p^w(x) \cdot q^{1-w}(x).$$

The generation process follows the dynamics induced by the guided transition matrix substituted in equation 2. Nisonoff et al. (2024) showed that guidance in the discrete setting serves a role analogous to its continuous counterpart—steering the model toward more faithful conditional samples—thus providing an important step toward improving the quality of discrete diffusion generations.

Simple Guidance. Concurrently, Schiff et al. (2024) proposed an alternative formulation of classifier-free guidance for discrete diffusion. Rather than interpolating the rate matrices as in Nisonoff et al. (2024), they directly interpolate the transition probabilities themselves. Specifically, when transitioning from time t to time $s < t$, the following transition was proposed:

$$p^{(w)}_{\text{simple}}(z_s|z_t) \propto p^w(z_s|z_t)p^{1-w}q(z_s|z_t). \tag{6}$$

As before, increasing w biases towards the target distribution p. Although the construction appears different, in the limit $s \to t$ the transitions coincide with those of Nisonoff et al. (2024). In practice, however, a finite number of steps is used, and the resulting methods are distinct. To implement these transitions, one can use equation (2) together with a suitable numerical integration scheme.

2.4 Dynamic Guidance Schedules

In our work we will consider dynamic guidance schedules, i.e. making w a function of time. Such schedules have become more popular in practice. For instance, guidance interval (Kynkäänniemi et al., 2024) only applies guidance on a segment of the generation process. Doing so produces a boost in the performance of diffusion models. However, existing work on dynamic guidance

[1] In existing literature, p is usually a class-conditional distribution, and q is an unconditional distribution. We adopt the general setup since recent works have shown that q can be chosen in different ways (Karras et al., 2024; Li et al., 2024; Rojas et al., 2025).

schedules (Kynkäänniemi et al., 2024; Xi et al., 2024) has been limited to a continuous (state-space) diffusion models. It remains unclear whether such schedules are also effective in discrete state diffusion—a question that serves as the main focus of our investigation.

Specifically, this work will consider $w : [0, T] \to \mathbb{R}$, i.e. guidance strength as a function of time, referred to as the guidance schedule. The schedule induces a generative process given by:

$$\frac{\mathrm{d}p_{T-t}}{\mathrm{d}t} = \overline{R}_{T-t}^{(w_{T-t})} p_{T-t} \tag{7}$$

Understanding which schedules result in the best generation is of crucial importance to further improve the sample accuracy of discrete diffusion models.

3 Methodology

We begin by analyzing the guided process in the simplest case of a single token in Section 3.1, which already reveals a key limitation of existing guidance. We then introduce our proposed remedy in Section 3.2 via column normalization. Afterwards, we analyze the effect of guidance schedules on two tokens in Section 3.4. Finally, we present experimental results of our methods in Section 4.

3.1 Identifying an issue in the guidance of discrete diffusion

We start by studying guidance in the case where $d = 1$ where exact analysis is possible. The following result characterizes the distribution at time t under constant guidance:

Theorem 3.1. *(Informal) Along the dynamics of equation (7), starting from a fully masked state, the distribution at time t is given by:*

$$p_t = \left(1 - \left(\frac{1 - e^{-\bar{\sigma}_t}}{1 - e^{-\bar{\sigma}_T}}\right)^{Z_w}\right) \cdot p^{(w)} \tag{8}$$

We present a full proof, as well as a more general result for varying guidance schedules in Theorem B.1. This shows that for $d = 1$ the guided process exactly recovers the tilted distribution, with the unmasking speed controlled by the factor in front of $p^{(w)}$. Although low-dimensional, this result already reveals important properties of the guided backwards process.

Figure 2: We plot the unmasking rates as a function of time under guidance. Faster unmasking ($Z_w > 1$) leads to worse numerical solvers, demonstrating an issue in the existing guidance mechanism.

Crucially, the partition function Z_w appears in the exponent of the rate term, meaning that even small changes in w can result in fast changes in the sampling rate. Figure 2 shows the percentage of tokens that remain masked as a function of time $p_t(M)$ for different values of Z_w. Applying guidance can significantly accelerate unmasking rates. While this can lead to faster generation, it may also introduce stiffness (Rathinam et al., 2003) and inefficiencies if not properly controlled.

3.2 Improved guidance mechanisms for discrete diffusion via column normalization

In order to alleviate the *unintentional* fast unmasking rates, we propose a simple yet effective change to the guidance mechanism. To understand where this issue is coming from, we explicitly write the transition rates between a masked state M a nonmasked state:

Lemma 3.1. *The transition rates between a masked state and an unormalized state are given by:*

$$\bar{R}_t^{(w)}(y, M) = R_t(x, y) \frac{e^{-\bar{\sigma}_t}}{1 - e^{-\bar{\sigma}_t}} Z_w p^{(w)}(y)$$

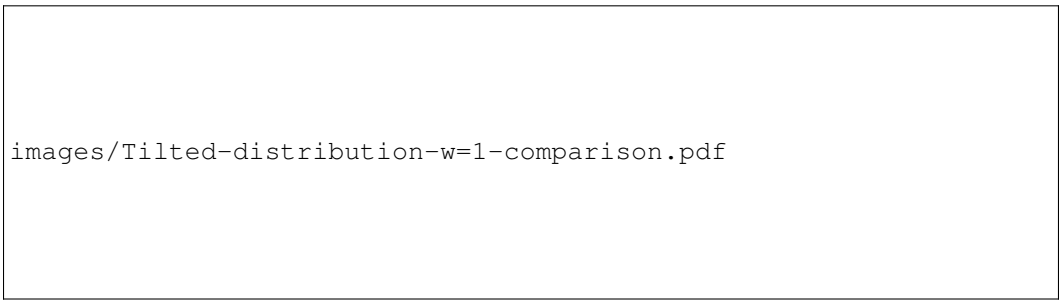


Figure 3: Tilted distributions for varying values of w. A large w produces a large concentration on one mode.

Notice how Z_w appears directly as a multiplying factor in the transition rate. However, when $w = 1$,(i.e. the conditional setup) this constant would play no role! This elucidates the effect we observe in Figure 2, the rates are being increased disproportionately due to the multiplication by the constant. To fix this, we must normalize the columns of the transition rate matrix appropriately. In the case of masked diffusion this can be achieved in a very simple fashion as follows:

$$\overline{R}^{(w)}_{\text{nor},t}(\hat{\mathbf{x}}, \mathbf{x}) = \frac{R_t(\mathbf{x}, \hat{\mathbf{x}})\, e^{-\overline{\sigma}_t}}{1 - e^{-\overline{\sigma}_t}} \operatorname{Softmax}(w \log p_0(\hat{\mathbf{x}}^i | \mathbf{x}^{\text{UM}}) + (1 - w) \log q_0(\hat{\mathbf{x}}^i | \mathbf{x}^{\text{UM}})). \quad (9)$$

The new rate matrix is normalized via the softmax function and fixes the issue introduced by the guidance mechanism. For the case of other discrete diffusions we refer the reader to Appendix E where we present a simple way of performing the normalization in general.

The normalization introduced in (9) has the effect of smoothing the transport between the starting distribution and the data distribution. This simple change stabilizes the sampling process and allows for a cleaner theory. Notably, this change can be done with a simple one line change to the code as presented in the pseudocode in 1. We further elaborate on the experimental benefits on Section 4.

3.3 Comparison of Guidance Mechanisms

We now clarify the distinctions between the various classifier-free guidance mechanisms. While some differences between our method and that of Nisonoff et al. (2024) were already discussed, we further highlight how our formulation also differs from the approach of Schiff et al. (2024). To better understand these differences, we begin by comparing the unlocking guidance mechanism of Nisonoff et al. (2024) with the simple guidance proposed by Schiff et al. (2024). For this analysis, we keep the guidance strength fixed throughout. Notice that: $p(x_s|x_t) = \exp\left(\int_s^t \overline{R}^{(w)}_\tau \mathrm{d}\tau\right) p_t$. Therefore, if p_t denotes the law of x_t, we can write the transition probabilities for each method:

$$p_{\text{unlocking}}(x_s|x_t) = \exp\left(\int_s^t \overline{R}^{w}_\tau(\cdot|c)\overline{R}^{1-w}_\tau(\cdot)\mathrm{d}\tau\right) p_t,$$

$$p_{\text{simple}}(x_s|x_t) = Z_{\text{simple}}\left(\exp\left(\int_s^t \overline{R}_\tau(\cdot|c)\mathrm{d}\tau\right) p_t\right)^{w} \left(\exp\left(\int_s^t \overline{R}_\tau(\cdot)\mathrm{d}\tau\right) p_t\right)^{1-w}.$$

where Z_{simple} is a normalizing constant. Now we look at the w-dependence inside the exponential. For $\log p_{\text{unlocking}}$, the w-dependence is *exponential* as it appears in the exponent of the rate matrices, while for $\log p_{\text{simple}}$, the w-dependence is *linear*. Therefore, the transitions induced by the unlocking guidance method get much more aggressive when w increases. On the other hand, our normalization (depending on w) normalizes the columns so that it maintains the smoothness of the transition when w increases. Therefore, our method approximates the convergence rates of the original process.

3.4 Analysis of Guidance Schedules in 2D

Having addressed the existing issue we switch our focus to the analysis of guidance schedules in the case of two tokens. Although the analysis can be extended to higher-dimensions, the complexity of the problem grows exponentially with the dimension, leading to increasingly intricate expressions

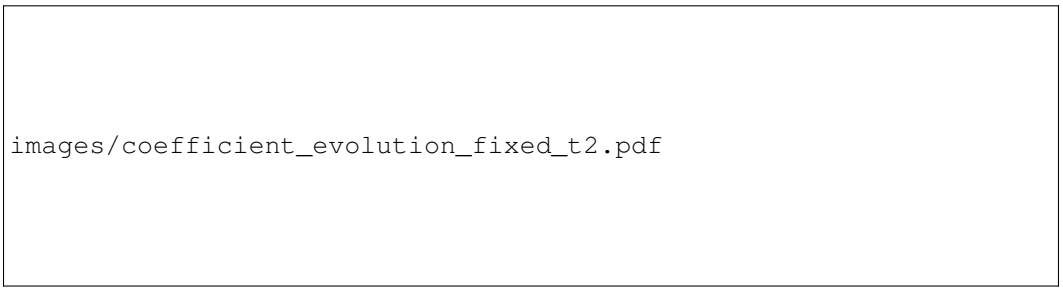


Figure 5: Evolution of the coefficients in Corollary 3.1 for different values of t_2. Notice that we must have $t_1 \leq t_2$. We observe that for moderate t_2 no coefficient dominates others, resulting in a balanced target distribution.

and reduced interpretability. This low-dimensional analysis already reveals the underlying mechanisms that define a good guidance schedule, and its impacts in high-dimensions are remarkable.

We start by stating our main theorem, in a simple to understand case that is used in practice. This simplification doesn't result in loss of generality, but significantly increases the interpretability of the results. We present a more general version in Theorem C.1.

Corollary 3.1. *Consider a time partition $0 = t_0 < t_1 < t_2 < t_3 = T$ with guidance w_i in the interval $[t_i, t_{i+1})$. With $\overline{\sigma} = -\log(1-\delta t)$ and $p_T(M, M) = 1$. Then the sampled distribution follows the following formula:*

$$p_{t_0}(i,j) = \left(\tfrac{t_3-t_2}{t_3}\right)^2 p^{(w_2)}(i,j) + \left(\tfrac{t_2-t_1}{t_3}\right)^2 p^{(w_1)}(i,j) + \left(\tfrac{t_1-t_0}{t_3}\right)^2 p^{(w_0)}(i,j)$$
$$+ \tfrac{(t_3-t_2)(t_2-t_1)}{t_3^2} p^{(w_1,w_2)}(i,j) + \tfrac{(t_3-t_2)(t_1-t_0)}{t_3^2} p^{(w_0,w_2)}(i,j) \quad + \tfrac{(t_2-t_1)(t_1-t_0)}{t_3^2} p^{(w_0,w_1)}(i,j),$$

where $p^{(w,\gamma)}(i,j) = p^{(w)}(i,j|X_1 = i)p^{(\gamma)}(X_1 = i) + p^{(w)}(i,j|X_2 = j)p^{(\gamma)}(X_2 = j)$, notice that this is not exactly a probability distribution as it is not normalized, but we will refer to it as one.

This theorem states that guidance schedules induce an interpolation of different distributions, which depend only on the guidance strengths and that the portion assigned to each one depends on the time parameters. We analyze the role of each component separately.

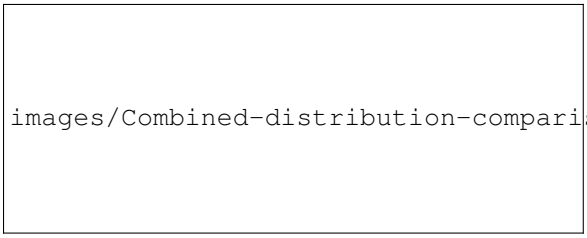
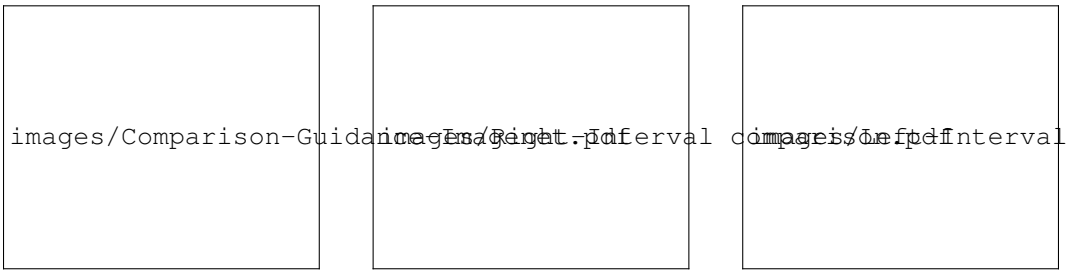


Figure 4: Notice that when $\omega < \gamma$ the combined distribution doesn't bias the leftmost mode, making this setting less efficient for guidance.

The role of guidance weights: We study a toy example in $2D$ containing 4 clusters, 2 of which are intersecting in the middle (see Appendix D for data visualizations). Figure 3 shows that increasing w leads to more concentration of mass in one of the modes. Similarly, Figure 4 shows that $p^{(w,\gamma)}$ strongly resembles the tilted distribution of w. Practically, this means that the combined distribution will be more similar to the guidance applied at the end of the generation! Therefore, effective schedules have higher guidance at the final and middle phases of the generation while keeping early guidance small.

The role of the time parameters: As observed in corollary 3.1, the time parameters set the proportion of each distribution that will contribute towards the final output. As observed in Figures 3,4, biasing just one of the distributions usually results in oversampling from a certain area. A good schedule is one that appropriately balances the contribution of each distribution.

We fix several values of t_2 and plot the coefficients as a function of t_1 in Figure 5. When $t_2 = 1$. we only have two intervals, and the curves change quickly; this implies that finding the right balance requires more careful tuning. On the other hand when $t_2 = .75$, many values of t_1 result in balanced combinations of all distributions, which ensures that we sample in a balanced way.



(a) Comparison of different guidance mechanisms.

(b) Right Interval vs Ramp-Up schedules.

(c) Left Interval vs Ramp-Down schedules

Figure 7: Evaluation of different guidance mechanisms and schedules on Imagenet

Which schedules perform best? Our theoretical analysis provides several insights into the design of effective guidance schedules. As discussed earlier, schedules that apply stronger guidance **during the middle and later stages** of the sampling process, while keeping early guidance small, tend to perform better. These selections seem to be the most critical, as they govern which distributions are mixed. Moreover, our theory predicts that using all **three intervals** (early, middle, and late) in the schedule facilitates **easier tuning** and yields more balanced output distributions. Based on these principles, we evaluate (according to our theory) various guidance schedules for discrete diffusion in Table 1, and we validate these predictions empirically in Section 4.2.

Table 1: Comparison of several guidance schedules.

	Low G. Beg	High G. Mid	High G. End	# Params Tune	Difficulty to Tune
Constant	×	✓	✓	1	High
Interval	✓	✓	×	3	Low
Increasing	✓	✓	✓	1	Low
Decreasing	×	✓	×	1	Low

4 Numerical Results

In this section, we examine whether the theoretical insights from low dimensions extend to high-dimensional image and text domains. On Section 4.1 we study the effect of our normalization and in Section 4.2 the impact of different guidance schedules. We present more details and samples of different methods in Appendix H including experiments with Show-O Xie et al..

4.1 Effect of normalization

Recall that our theory predicted that failing to normalize complicates the simulation, so normalization should improve results in practice, which we confirm below.

Testing on Imagenet: We assess MaskGIT on the ImageNet dataset (Deng et al., 2009) and evaluate FID on ImageNet-256 using 50K samples, following standard practices. For our method and for the Unlocking Guidance baseline (Nisonoff et al., 2024), we use the τ-leaping sampler. For Simple Guidance (Schiff et al., 2024), we interpolate Euler transitions. For all methods, we use 50 steps. Figure 7a shows FID as a function of guidance strength using a constant schedule. Our experiments demonstrate that *failing to normalize can substantially degrade sample quality* as suggested by our theory.

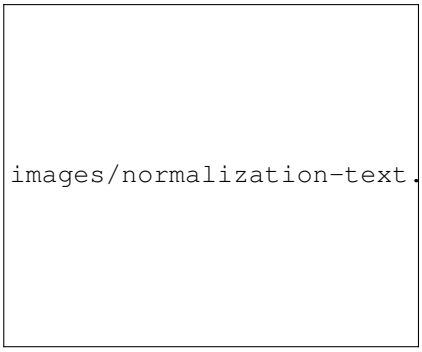


Figure 6: MATH-500 performance for LLada-8B-Instruct under a simple sampler without remasking to isolate the effect of the guidance mechanism. Normalization always yields better results.

Testing on text-to-image: We evaluate our method on the GenEval benchmark (Ghosh et al., 2023) using the pre-trained Meissonic model (Bai et al., 2024). This benchmark provides a comprehensive measure of both prompt alignment and perceptual image quality.

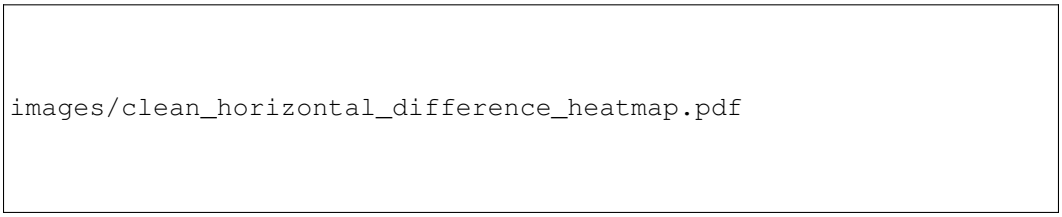


Figure 8: GenEval with and without normalization. Red denotes an improved performance due to normalization. Normalization leads to more faithful prompt adherence and image quality.

Figure 12 compares generations with and without normalization. Red regions indicate prompts where normalization improved the score. Overall, we observe consistent gains: *normalization enhances prompt adherence* and yields images that better match the target distribution.

Testing on text generation: To assess the effectiveness of normalization in the text generation domain, we evaluated using LLaDA-8B-Instruct (Nie et al.) on the MATH-500 dataset, generating up to 256 tokens. We sample autoregressively in blocks of 32 tokens using a simple Euler sampler with 32 denoising steps per block, resulting in a total of 256 steps for the full generation.

Figure 6 presents the results of such an experiment. The results clearly show that *normalization consistently improves performance across all guidance strengths*. We note that the results are not directly comparable to those reported in the LLaDA paper; we use a simple Euler sampler without remasking to better isolate the effect of guidance and normalization in a simple setting.

Empirical effect of normalization: All our empirical findings demonstrate that including normalization is a helpful step in improving the simulation of classifier-free guidance for discrete diffusion. This aligns with our low-dimensional theoretical analysis in Section 3.1, demonstrating that *low-dimensional studies can have a significant impact in high dimensions*.

4.2 Study of guidance schedules

Table 2: Description of guidance schedules.

Schedule	Formula $w(t)$
Left Interval	$w \cdot \mathbf{1}_{[0,l]}(t)$
Right Interval	$w \cdot \mathbf{1}_{[r,1]}(t)$
Ramp-Up	$\min\left(w, w \cdot \frac{1-t}{1-r}\right)$
Ramp-Down	$\min\left(w, w \cdot \frac{t}{\ell}\right)$

Previously, our theory predicted that increasing schedules improve discrete diffusion while decreasing ones degrade generation. We test this theory on Imagenet-256 with 10K samples. For precise formulas for the schedules, see Table 2. When testing increasing schedules (Ramp-Up and Right Interval) in 7b, we observe that both schedules can significantly improve the results. Furthermore, the Right Interval schedule exhibits a convex trend with respect to r, while the Ramp-Up schedule is monotone in r, and reaches a lower FID value, indicating that a gradual, linear increase in guidance outperforms abrupt alternatives. When testing the decreasing schedules (Left interval and Ramp-Down), we observe that they consistently damage the generation as seen in Figure 7c. Overall, *our experiments confirm our theory* that increasing schedules are most effective for masked diffusion.

5 Conclusions

In this work, we introduced a framework for analyzing guidance schedules in masked diffusion. Our analysis led to a novel approach for classifier-free guidance in the discrete setting. We validate the effectiveness of our method through experiments and show that guidance applied near $t = T$ is harmful to the generation quality while near $t = 0$ can improve the it. This insights enabled us to identify effective scheduling strategies. Our theoretical insights align closely with empirical observations, bridging the gap between theory and practice.

Limitations and Future work. While our framework provides a principled and tractable approach to CFG in discrete diffusion, our theoretical analysis is currently limited to masked diffusion in low-dimensional settings. Although the method is applicable to more complex real-world settings, our current theoretical study does not cover such regimes. Promising directions include extending the framework to other forms of discrete diffusion, such as uniform diffusion, scaling to higher dimensions, and analyzing the role of score estimation error in the guidance dynamics.

REFERENCES

J. Bai, T. Ye, W. Chow, E. Song, X. Li, Z. Dong, L. Zhu, and S. Yan. Meissonic: Revitalizing masked generative transformers for efficient high-resolution text-to-image synthesis. *arXiv preprint arXiv:2410.08261*, 2024.

A. Bradley and P. Nakkiran. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.

T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh. Video generation models as world simulators. https://openai.com/research/video-generation-models-as-world-simulators, 2024.

A. Campbell, J. Benton, V. De Bortoli, T. Rainforth, G. Deligiannidis, and A. Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

M. Chidambaram, K. Gatmiry, S. Chen, H. Lee, and J. Lu. What does guidance do? a fine-grained analysis in a simple setting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

D. Ghosh, H. Hajishirzi, and L. Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

N. Gruver, S. Stanton, N. Frey, T. G. Rudner, I. Hotzel, J. Lafrance-Vanasse, A. Rajpal, K. Cho, and A. G. Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36:12489–12517, 2023.

J. Ho and T. Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

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

E. Hoogeboom, T. Mensink, J. Heek, K. Lamerigts, R. Gao, and T. Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. *arXiv preprint arXiv:2410.19324*, 2024.

H. Huang, L. Sun, B. Du, and W. Lv. Conditional diffusion based on discrete graph structures for molecular graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4302–4311, 2023.

T. Karras, M. Aittala, T. Kynkäänniemi, J. Lehtinen, T. Aila, and S. Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37: 52996–53021, 2024.

T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

T. Kynkäänniemi, M. Aittala, T. Karras, S. Laine, T. Aila, and J. Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Conference on Neural Information Processing Systems*, 2024.

T. Li, W. Luo, Z. Chen, L. Ma, and G.-J. Qi. Self-guidance: Boosting flow and diffusion generation on their own. *CoRR*, 2024.

A. Lou, C. Meng, and S. Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. ZHOU, Y. Lin, J.-R. Wen, and C. Li. Large language diffusion models. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*.

H. Nisonoff, J. Xiong, S. Allenspach, and J. Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

J. Ou, S. Nie, K. Xue, F. Zhu, J. Sun, Z. Li, and C. Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

K. L. Pavasovic, J. Verbeek, G. Biroli, and M. Mezard. Understanding classifier-free guidance: High-dimensional theory and non-linear generalizations. *arXiv preprint arXiv:2502.07849*, 2025.

R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics*, 119(24):12784–12794, 2003.

Y. Ren, H. Chen, Y. Zhu, W. Guo, Y. Chen, G. M. Rotskoff, M. Tao, and L. Ying. Fast solvers for discrete diffusion models: Theory and applications of high-order algorithms. *arXiv preprint arXiv:2502.00234*, 2025.

K. Rojas, Y. Zhu, S. Zhu, F. X. Ye, and M. Tao. Diffuse everything: Multimodal diffusion models on arbitrary state spaces. In *Forty-second International Conference on Machine Learning*, 2025.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

L. Ruddigkeit, R. Van Deursen, L. C. Blum, and J.-L. Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.

S. Sahoo, M. Arriola, Y. Schiff, A. Gokaslan, E. Marroquin, J. Chiu, A. Rush, and V. Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Y. Schiff, S. S. Sahoo, H. Phung, G. Wang, S. Boshar, H. Dalla-torre, B. P. de Almeida, A. Rush, T. Pierrot, and V. Kuleshov. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.

J. Shi, K. Han, Z. Wang, A. Doucet, and M. Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

W. Xi, N. Dufour, N. Andreou, C. Marie-Paule, V. F. Abrevaya, D. Picard, and V. Kalogeiton. Analysis of classifier-free guidance weight schedulers. *Transactions on Machine Learning Research*, 2024.

J. Xie, W. Mao, Z. Bai, D. J. Zhang, W. Wang, K. Q. Lin, Y. Gu, Z. Chen, Z. Yang, and M. Z. Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *The Thirteenth International Conference on Learning Representations*.

Z. Xie, J. Ye, L. Zheng, J. Gao, J. Dong, Z. Wu, X. Zhao, S. Gong, X. Jiang, Z. Li, et al. Dream-coder 7b: An open diffusion language model for code. *arXiv preprint arXiv:2509.01142*, 2025.

H. Ye, R. Kevin, and T. Molei. What exactly does guidance do in masked discrete diffusion models. *arXiv preprint arXiv:2506.10971*, 2025.

S. Yu, S. Kwak, H. Jang, J. Jeong, J. Huang, J. Shin, and S. Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

A Notation and general results

A.1 Special Properties of Masked Diffusion

We will use the following notations specific to masked diffusion. Let $\mathbf{x}_t = (\mathbf{x}_t^1, \dots, \mathbf{x}_t^d)$ denote a random variable on S, and M be the masked token. We will write $\mathbf{x}^{\text{UM}}$ for the set of elements such that $\mathbf{x}_t^i \neq M$, meaning the entries that are not the masked token. Additionally, we will denote $\overline{\sigma}_t = \int_0^t \sigma_s \mathrm{d}s$.

Masked diffusion has several appealing properties, one being the following shown by Ou et al. (2024):

Lemma A.1. *Along the dynamics (1) given by the masked rate matrix, if $\mathbf{x}_t = (\mathbf{x}_t^1, \dots, \mathbf{x}_t^d)$ and $\hat{\mathbf{x}}_t = (\mathbf{x}_t^1, \dots, \hat{\mathbf{x}}_t^i, \dots, \mathbf{x}_t^d)$ in such a way that $\hat{\mathbf{x}}_t^i \neq M$ and $\mathbf{x}_t^i = M$, we have the following identity for the score*

$$\frac{p_t(\hat{\mathbf{x}}_t)}{p_t(\mathbf{x}_t)} = \frac{e^{-\overline{\sigma}_t}}{1 - e^{-\overline{\sigma}_t}} p_0(\hat{\mathbf{x}}_t^i | \mathbf{x}^{UM}).$$

This result is of great importance, as it tells us that it is possible to decompose the scores as a probability distribution independent of time multiplied by a time-dependent term.

B Proofs in 1D

We first prove a small lemma:

Lemma B.1. *Given a matrix of the form*

$$A = \begin{pmatrix} 0 & \dots & 0 & v_1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & v_n \end{pmatrix}$$

If $v_n \neq 0$, then its matrix exponential is given by $e^A = I + A \cdot \frac{e^{v_n} - 1}{v_n}$.

Proof. First notice that for $k > 0$ it holds that $A^k = v_n^{k-1} A$ then we can write:

$$\begin{aligned}
e^A &= I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \dots \\
&= I + A + \frac{1}{2!}Av_n + \frac{1}{3!}Av_n^2 + \dots \\
&= I + A(1 + \frac{1}{2!}v_n + \frac{1}{3!}v_n^2 + \dots) \\
&= I + A(1 + \frac{1}{v_n}(\frac{1}{2!}v_n^2 + \frac{1}{3!}v_n^3 + \dots) \\
&= I + A(1 + \frac{1}{v_n}(-1 - v_n + 1 + v_n + \frac{1}{2!}v_n^2 + \frac{1}{3!}v_n^3 + \dots) \\
&= I + A(1 + \frac{1}{v_n}(-1 - v_n + e^{v_n}) \\
&= I + A\frac{e^{v_n} - 1}{v_n}
\end{aligned}$$

As we wanted. □

We now state and prove the general version Theorem 3.1:

Theorem B.1. *Along the dynamics of equation (7). The distribution p_t is given by:*

$$p_t = \left(A_1 \cdot \frac{1 - e^A}{A}, \quad \cdots \quad, A_{M-1} \cdot \frac{1 - e^A}{A}, \quad e^A \right)^\intercal.$$

Where, for $i = 0, \dots, M-1$:

$$A_i = \int_t^T \sigma_s \frac{e^{-\overline{\sigma}_s}}{1 - e^{-\overline{\sigma}_s}} Z_{w_s} \cdot p^{z,w_s}(i) ds, \qquad A = -\sum_{i=0}^{M-1} A_i = \int_t^T \sigma_s \frac{e^{-\overline{\sigma}_s}}{1 - e^{-\overline{\sigma}_s}} Z_{w_s} ds.$$

Proof. Recall that the rate matrix in the one-dimensional case is:

$$\overline{R}_t^{(w_t)} = \sigma_t \frac{e^{-\overline{\sigma}_t}}{1 - e^{-\overline{\sigma}_t}} Z_{w_t} \begin{pmatrix} 0 & 0 & \cdots & 0 & p^{(w_t)}(1) \\ 0 & 0 & \cdots & 0 & p^{(w_t)}(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & p^{(w_t)}(M-1) \\ 0 & 0 & \cdots & 0 & -1 \end{pmatrix} \tag{10}$$

By direct integration we know that:

$$p_t = \exp\Big(\int_t^T \overline{R}_\tau^{(w_\tau)} d\tau \Big) p_T.$$

Therefore applying Lemma B.1 we get that (in vector notation):

$$p_t = p_T + p_T(M) \begin{pmatrix} \int_t^T \sigma_s \frac{e^{-\overline{\sigma}_s}}{1-e^{-\overline{\sigma}_s}} Z_{w_s} \cdot p^{z,w_s} ds \cdot \dfrac{1 - e^A}{A} \\ e^A \end{pmatrix},$$

with

$$A = -\sum_{i=0}^{M-1} A_i = \int_t^T \sigma_s \frac{e^{-\overline{\sigma}_s}}{1 - e^{-\overline{\sigma}_s}} Z_{w_s} ds.$$

The result is proved. □

We can now use the previous theorem to compute the distribution under constant guidance:

Corollary B.1. *If we start with a distribution p_t and keep guidance to be constant w. Then at time s the distribution is given by:*

$$p_s(i) = p_t(i) + p_s(M) \Big(\frac{1 - e^{-\overline{\sigma}_t}}{1 - e^{-\overline{\sigma}_s}} - 1 \Big)^{Z_w} p^{(w)}(i)$$

for $i \neq M$ and $p_s(M) = \left(\frac{1-e^{-\overline{\sigma}_t}}{1-e^{-\overline{\sigma}_s}} - 1 \right)^{Z_w} p_t(M)$

Proof. The proof follows by keeping w constant in the above theorem:

$$\begin{aligned} p_s &= p_t + p_t(M) \begin{pmatrix} \int_s^t \sigma_s \frac{e^{-\overline{\sigma}_s}}{1-e^{-\overline{\sigma}_s}} Z_{w_s} \cdot p^{z,w_s} ds \cdot \dfrac{1 - e^A}{A} \\ e^A \end{pmatrix} \\ &= p_t + p_t(M) \begin{pmatrix} \int_t^T \sigma_s \frac{e^{-\overline{\sigma}_s}}{1-e^{-\overline{\sigma}_s}} Z \cdot ds \cdot \dfrac{1 - e^A}{A} p^{(w)} \\ e^A \end{pmatrix} \\ &= p_t + p_t(M) \begin{pmatrix} (1 - e^A) p^{(w)} \\ e^A \end{pmatrix} \end{aligned}$$

Substituting A gives the desired result. □

We can now chain the above argument to obtain a result for general piece-wise constant guidance schedules:

Theorem B.2. *Let $\delta = t_0 < t_1 < \cdots < t_k = T$ be a time partition and let w_i the guidance strength on the interval $(t_i, t_{i+1}]$. Along the dynamics of equation (7), the sampled distribution p_δ is given by:*

$$p_\delta = p_T + \sum_{i=0}^{k-1} p_{t_{i+1}}(M) \cdot \left(1 - \left(\frac{1 - e^{-\overline{\sigma}_{t_i}}}{1 - e^{-\overline{\sigma}_{t_{i+1}}}}\right)^{Z_{w_i}}\right) p^{(w_i)}. \tag{11}$$

Additionally, probability mass at M at different time satisfies $p_{t_i}(M) = p_{t_{i+1}}(M)\left(\frac{1 - e^{-\overline{\sigma}_{t_i}}}{1 - e^{-\overline{\sigma}_{t_{i+1}}}}\right)^{Z_{w_i}}$ for all $i = 0, 1, \cdots, k-1$.

Lemma B.2. *The transition rates between a masked state and an unnormalized state are given by:*

$$\bar{R}_t^{(w)}(y, M) = R_t(x, y) \frac{e^{-\bar{\sigma}_t}}{1 - e^{-\bar{\sigma}_t}} Z_w p^{(w)}(y)$$

Proof. Using Lemma A.1 we can write:

$$\begin{aligned}
\bar{R}_t^{(w)}(y, M) &= R_t(M, y) \cdot \left(\frac{p_t(x)}{p_t(M)}\right)^w \left(\frac{q_t(x)}{q_t(M)}\right)^{1-w} \\
&= R_t(M, y) \cdot \left(\frac{e^{-\bar{\sigma}_t}}{1 - e^{-\bar{\sigma}_t}} p_0(y)\right)^w \left(\frac{e^{-\bar{\sigma}_t}}{1 - e^{-\bar{\sigma}_t}} q_0(y)\right)^{1-w} \\
&= R_t(x, y) \frac{e^{-\bar{\sigma}_t}}{1 - e^{-\bar{\sigma}_t}} p_0^w(y) q_0^{1-w}(y) \\
&= R_t(x, y) \frac{e^{-\bar{\sigma}_t}}{1 - e^{-\bar{\sigma}_t}} Z_w p^{(w)}(y)
\end{aligned}$$

□

The results for the normalized process are identical to the ones above, so we omit them for brevity.

C Proofs in 2D

We begin by writing a simple lemma that will come in handy later.

Lemma C.1. *Given a matrix of the form*

$$A = \begin{pmatrix} 0 & a & b & 0 \\ 0 & -1 & 0 & c \\ 0 & 0 & -1 & d \\ 0 & 0 & 0 & -2 \end{pmatrix}$$

Then for any $\alpha \in \mathbb{R}$, it's matrix exponential is given by:

$$\exp(\alpha A) = \begin{pmatrix} 1 & a(1 - e^{-\alpha}) & b(1 - e^{-\alpha}) & \frac{(ac+bd)(e^\alpha - 1)^2 e^{-2\alpha}}{2} \\ 0 & e^{-\alpha} & 0 & c\left(e^\alpha - 1\right) e^{-2\alpha} \\ 0 & 0 & e^{-\alpha} & d\left(e^\alpha - 1\right) e^{-2\alpha} \\ 0 & 0 & 0 & e^{-2\alpha} \end{pmatrix}$$

Proof. The proof of the above statement is easy by noticing that $A = PDP^{-1}$ with:

$$P = \begin{pmatrix} \frac{ac}{2} + \frac{bd}{2} & -a & -b & 1 \\ -c & 1 & 0 & 0 \\ -d & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Then $\exp(\alpha A) = P \exp(\alpha D) P^{-1}$ and the result follows. □

Now for the main proof we start by explicitly writing down the rate matrix in the case of two tokens. In this case the rate matrix will have the following structure:

$$\overline{R}^{(w)}_{\text{nor},t} = \frac{\sigma_t\, e^{-\bar{\sigma}_t}}{1-e^{-\bar{\sigma}_t}} \begin{pmatrix} D_1 & \mathbf{0} & \dots & C_1 \\ \mathbf{0} & D_2 & \dots & C_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & L \end{pmatrix} \coloneqq \frac{\sigma_t\, e^{-\bar{\sigma}_t}}{1-e^{-\bar{\sigma}_t}} \overline{R}^{(w)}_{\text{nor}},$$

where each block is an $M \times M$ matrix given by the following formulas:

$$D_i = \begin{pmatrix} 0 & \dots & 0 & p^{(w)}(X_2 = 1 \mid X_1 = i) \\ 0 & \dots & 0 & p^{(w)}(X_2 = 2 \mid X_1 = i) \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & p^{(w)}(X_2 = M-1 \mid X_1 = i) \\ 0 & \dots & 0 & -1 \end{pmatrix}$$

$$C_i = \begin{pmatrix} p^{(w)}(X_1 = i \mid X_2 = 1) & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & 0 \\ 0 & \dots & p^{(w)}(X_1 = i \mid X_2 = M-1) & 0 \\ 0 & \dots & 0 & p^{(w)}(X_1 = i) \end{pmatrix}$$

$$L = \begin{pmatrix} -1 & 0 & \dots & 0 & p^{(w)}(X_2 = 1) \\ 0 & -1 & \dots & 0 & p^{(w)}(X_2 = 2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & -1 & p^{(w)}(X_2 = M-1) \\ 0 & \dots & 0 & 0 & -2 \end{pmatrix}$$

We can now state the main theorem:

Theorem C.1. *Given a starting distribution p_t following the dynamics given by (7) the distribution at time s is given by:*

$$p_s(i,j) = \begin{cases} p_t(i,j) + \left(1 - \frac{1-e^{-\bar{\sigma}_s}}{1-e^{-\bar{\sigma}_t}}\right)^2 p^{(w)}(i,j)\, p_t(M,M) \\ \quad + \left(1 - \frac{1-e^{-\bar{\sigma}_s}}{1-e^{-\bar{\sigma}_t}}\right) \Big[p^{(w)}(X_2 = j \mid X_1 = i)\, p_t(i,M) \\ \quad + \; p^{(w)}(X_1 = i \mid X_2 = j)\, p_t(M,j) \Big] & \text{if } i,j \neq M \\ \\ \left(\frac{1-e^{-\bar{\sigma}_s}}{1-e^{-\bar{\sigma}_t}}\right) p_t(i,M) \\ \quad + \left(\frac{1-e^{-\bar{\sigma}_s}}{1-e^{-\bar{\sigma}_t}}\right)^2 \left(\frac{1-e^{-\bar{\sigma}_t}}{1-e^{-\bar{\sigma}_s}} - 1\right) p^{(w)}(X_1 = i) p_t(M,M) & \text{if } i \neq M,\ j = M \\ \\ \left(\frac{1-e^{-\bar{\sigma}_s}}{1-e^{-\bar{\sigma}_t}}\right) p_t(M,j) \\ \quad + \left(\frac{1-e^{-\bar{\sigma}_s}}{1-e^{-\bar{\sigma}_t}}\right)^2 \left(\frac{1-e^{-\bar{\sigma}_t}}{1-e^{-\bar{\sigma}_s}} - 1\right) p^{(w)}(X_2 = j) p_t(M,M) & \text{if } i = M,\ j \neq M \\ \\ \left(\frac{1-e^{-\bar{\sigma}_s}}{1-e^{-\bar{\sigma}_t}}\right)^2 p_t(M,M) & \text{if } i = j = M \end{cases}$$

Proof. By direct integration we know that:

$$p_s = \exp\Big(\int_s^t \frac{\sigma_\tau e^{-\bar{\sigma}_\tau}}{1-e^{-\bar{\sigma}_\tau}} d\tau \overline{R}^{(w)}_{\text{nor}} \Big) = \exp\Big(\ln\Big(\frac{1-e^{-\bar{\sigma}_t}}{1-e^{-\bar{\sigma}_s}}\Big) \overline{R}^{(w)}_{\text{nor}} \Big).$$

Due to the block structure of $\overline{R}^{(w)}_{\text{nor}}$, it is enough to be able to compute the exponential of:

$$\begin{pmatrix} D_i & C_i \\ \mathbf{0} & L \end{pmatrix} = \left[\begin{array}{ccc|c|ccc} 0 & \dots & 0 & p^{(w)}(X_2 = 1 \mid X_1 = i) & p^{(w)}(X_1 = i \mid X_2 = 1) & 0 & \dots & 0 \\ 0 & \dots & 0 & p^{(w)}(X_2 = 2 \mid X_1 = i) & \vdots & \ddots & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & \dots & & 0 \\ 0 & \dots & 0 & -1 & 0 & \dots & 0 & p^{(w)}(X_1 = i) \\ \hline 0 & \dots & 0 & 0 & -1 & 0 & \dots & p^{(w)}(X_2 = 1) \\ 0 & \dots & 0 & 0 & 0 & -1 & \dots & p^{(w)}(X_2 = 2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & -2 \end{array}\right]$$

where once again we can exploit the structured form of the matrix to simplify the calculation. It is clear that when computing products of this matrix, coordinates will only get affected by the smaller sublocks:

$$\left(\begin{array}{cc|cc} 0 & p^{(w)}(X_2 = j \mid X_1 = i) & p^{(w)}(X_1 = i \mid X_2 = j) & 0 \\ 0 & -1 & 0 & p^{(w)}(X_1 = i) \\ \hline 0 & 0 & -1 & p^{(w)}(X_2 = j) \\ 0 & 0 & 0 & -2 \end{array}\right)$$

This is not only clear from the structure, but it also reveals a true intuitive understanding. The probability mass at a given position can only be affected by those states that are reachable from the current state by masking or unmasking the entries. We can now use Lemma C.1 to find the exponential:

$$\begin{pmatrix} 1 & p^{(w)}(X_2 = j \mid X_1 = i)(1 - e^{-\alpha}) & p^{(w)}(X_1 = i \mid X_2 = j)(1 - e^{-\alpha}) & p^{(w)}(i, j)(e^{\alpha} - 1)^2 e^{-2\alpha} \\ 0 & e^{-\alpha} & 0 & p^{(w)}(X_1 = i)\,(e^{\alpha} - 1)\,e^{-2\alpha} \\ 0 & 0 & e^{-\alpha} & p^{(w)}(X_2 = j)\,(e^{\alpha} - 1)\,e^{-2\alpha} \\ 0 & 0 & 0 & e^{-2\alpha} \end{pmatrix}$$

where $\alpha = \ln\left(\frac{1 - e^{-\overline{\sigma}_t}}{1 - e^{-\overline{\sigma}_s}}\right)$ and we used that $2p^{(w)}(i, j) = p^{(w)}(X_2 = j \mid X_1 = i)p^{(w)}(X_1 = i) + p^{(w)}(X_1 = i \mid X_2 = j)p^{(w)}(X_2 = j)$. Putting this together, we get that exponentiation each block we get:

$$\left[\begin{array}{ccc|c|ccc} 1 & \dots & 0 & p^{(w)}(X_2 = 1 \mid X_1 = i)(1 - e^{-\alpha}) & p^{(w)}(X_1 = i \mid X_2 = 1)(1 - e^{-\alpha}) & \dots & p^{(w)}(i, 1)(e^{\alpha} - 1)^2 e^{-2\alpha} \\ 0 & \dots & 0 & p^{(w)}(X_2 = 2 \mid X_1 = i)(1 - e^{-\alpha}) & \vdots & \ddots & p^{(w)}(i, 1)(e^{\alpha} - 1)^2 e^{-2\alpha} \\ \vdots & \vdots & \vdots & \vdots & 0 & & \vdots \\ 0 & \dots & 0 & e^{-\alpha} & 0 & \dots & p^{(w)}(X_1 = i)(1 - e^{\alpha})e^{-2\alpha} \\ \hline 0 & \dots & 0 & 0 & e^{-\alpha} & \dots & p^{(w)}(X_2 = 1)(1 - e^{\alpha})e^{-2\alpha} \\ 0 & \dots & 0 & 0 & 0 & e^{-\alpha} \dots & p^{(w)}(X_2 = 2)(1 - e^{\alpha})e^{-2\alpha} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & e^{-2\alpha} \end{array}\right]$$

With this, we have a full characterization of the matrix exponential. Therefore, we can simply write down the probability distribution by multiplying by p_t:

$$p_s(i,j) = \begin{cases} p_t(i,j) + (1-e^{-\alpha})^2\, p^{(w)}(i,j)\, p_t(M,M) \\ \quad + (1-e^{-\alpha})\Big[p^{(w)}(X_2 = j \mid X_1 = i)\, p_t(i,M) \\ \quad + \; p^{(w)}(X_1 = i \mid X_2 = j)\, p_t(M,j)\Big] & \text{if } i,j \neq M \\ e^{-\alpha} p_t(i,M) \\ \quad + e^{-2\alpha}(e^{\alpha}-1)\, p^{(w)}(X_1 = i)\, p_t(M,M) & \text{if } i \neq M,\ j = M \\ e^{-\alpha} p_t(M,j) \\ \quad + e^{-2\alpha}(e^{\alpha}-1)\, p^{(w)}(X_2 = j)\, p_t(M,M) & \text{if } i = M,\ j \neq M \\ e^{-2\alpha} p_t(M,M) & \text{if } i = j = M \end{cases}$$

We can now replace $\alpha = \ln\left(\frac{1-e^{-\overline{\sigma}_t}}{1-e^{-\overline{\sigma}_s}}\right)$ into the formula above to obtain the result.

□

Corollary C.1. *Given a starting distribution p_t following the dynamics given by (7) with $\overline{\sigma}_t = -\log(1-\delta t)$ the distribution at time s is given by:*

$$p_s(i,j) = \begin{cases} p_t(i,j) + \left(\frac{t-s}{t}\right)^2 p^{(w)}(i,j)\, p_t(M,M) \\ \quad + \left(\frac{t-s}{t}\right)\Big[p^{(w)}(X_2 = j \mid X_1 = i)\, p_t(i,M) \\ \quad + \; p^{(w)}(X_1 = i \mid X_2 = j)\, p_t(M,j)\Big] & \textit{if } i,j \neq M \\ \frac{s}{t}\cdot p_t(i,M) + \left(\frac{s}{t}\right)^2 \left(\frac{t-s}{s}\right) p^{(w)}(X_1 = i) p_t(M,M) & \textit{if } i \neq M,\ j = M \\ \frac{s}{t}\cdot p_t(M,j) + \left(\frac{s}{t}\right)^2 \left(\frac{t-s}{s}\right) p^{(w)}(X_2 = j) p_t(M,M) & \textit{if } i = M,\ j \neq M \\ \left(\frac{s}{t}\right)^2 p_t(M,M) & \textit{if } i = j = M \end{cases}$$

Proof. Notice that under this schedule we have that:

$$\frac{1-e^{-\bar{\sigma}_s}}{1-e^{-\bar{\sigma}_t}} = \frac{\delta s}{\delta t} = \frac{s}{t}$$

Substituting this in gives the corollary above. □

Proof of Corollary 3.1. We track the changes in the distribution in every time interval. This can be found by plugging in the result of the corollary above. Firstly, on the interval $T \to t_2$ we obtain:

$$p_{t_2}(M,M) = \left(\frac{t_2}{T}\right)^2$$

$$p_{t_2}(M,j) = \left(\frac{t_2}{T}\right)^2 \left(\frac{T-t_2}{t_2}\right) p^{(w_2)}(X_2 = j)$$
$$p_{t_2}(i,M) = \left(\frac{t_2}{T}\right)^2 \left(\frac{T-t_2}{t_2}\right) p^{(w_2)}(X_1 = i)$$
$$p_{t_2}(i,j) = \left(\frac{T-t_2}{T}\right)^2 p^{(w_2)}(i,j)$$

Then on the interval from $t_2 \to t_1$ we get:

$$p_{t_1}(M,M) = \left(\frac{t_1}{T}\right)^2$$
$$p_{t_1}(M,j) = \left(\frac{t_1}{t_2}\right) p_{t_2}(M,j) + \left(\frac{t_1}{t_2}\right)^2 \left(\frac{t_2-t_1}{t_1}\right) p^{(w_1)}(X_2 = j) p_{t_2}(M,M)$$
$$p_{t_1}(i,M) = \left(\frac{t_1}{t_2}\right) p_{t_2}(i,M) + \left(\frac{t_1}{t_2}\right)^2 \left(\frac{t_2-t_1}{t_1}\right) p^{(w_1)}(X_1 = i) p_{t_2}(M,M)$$
$$p_{t_1}(i,j) = p_{t_2}(i,j) + \left(\frac{t_2-t_1}{t_2}\right)^2 p^{(w_1)}(i,j) p_{t_2}(M,M)$$
$$+ \left(\frac{t_2-t_1}{t_2}\right) [p^{(w_1}(X_2 = j|X_1 = i) p_{t_2}(i,M) + p^{(w_1)}(X_1 = i|X_2 = j) p_{t_2}(M,j)]$$

Replacing the values for p_{t_1} into this equation we get:

$$p_{t_1}(M,M) = \left(\frac{t_1}{T}\right)^2$$
$$p_{t_1}(M,j) = \left(\frac{t_1(T-t_2)}{T^2}\right) p^{(w_2)}(X_2 = j) + \left(\frac{t_1}{T}\right)^2 \left(\frac{t_2-t_1}{t_1}\right) p^{(w_2)}(X_2 = j)$$
$$p_{t_1}(i,M) = \left(\frac{t_1(T-t_2)}{T^2}\right) p^{(w_2)}(X_1 = i) + \left(\frac{t_1}{T}\right)^2 \left(\frac{t_2-t_1}{t_1}\right) p^{(w_2)}(X_1 = i)$$
$$p_{t_1}(i,j) = \left(\frac{T-t_2}{T}\right)^2 p^{(w_2)}(i,j) + \left(\frac{t_2-t_1}{t_2}\right)^2 \left(\frac{t_2}{T}\right)^2 p^{(w_1)}(i,j)$$
$$+ \left(\frac{t_2-t_1}{t_2}\right) \left(\frac{t_2}{T}\right)^2 \left(\frac{T-t_2}{t_2}\right) p^{(w_1)}(X_2 = j|X_1 = i) p^{(w_2)}(X_1 = i)$$
$$+ \left(\frac{t_2-t_1}{t_2}\right) \left(\frac{t_2}{T}\right)^2 \left(\frac{T-t_2}{t_2}\right) p^{(w_1)}(X_1 = i|X_2 = j) p^{(w_2)}(X_2 = j)$$
$$= \left(\frac{T-t_2}{T}\right)^2 p^{(w_2)}(i,j) + \left(\frac{t_2-t_1}{T}\right)^2 p^{(w_1)}(i,j) + \frac{(t_2-t_1)(T-t_2)}{T^2} p^{(w_1,w_2)}$$

Finally, we can proceed with the final step from $t_1 \to t_0$. In this case, we have:

$$p_{t_0}(M,M) = 0$$
$$p_{t_0}(M,j) = 0$$
$$p_{t_0}(i,M) = 0$$
$$p_{t_0}(i,j) = p_{t_1}(i,j) + \left(\frac{t_1-t_0}{t_1}\right)^2 p^{(w_0)}(i,j) p_{t_1}(M,M)$$
$$+ \left(\frac{t_1-t_0}{t_1}\right) [p^{(w_0}(X_2 = j|X_1 = i) p_{t_1}(i,M) + p^{(w_0)}(X_1 = i|X_2 = j) p_{t_1}(M,j)]$$

Then substituting in the previous results:

$$p_{t_0}(i,j) = p_{t_1}(i,j) + \left(\frac{t_1-t_0}{t_1}\right)^2 \left(\frac{t_1}{T}\right)^2 p^{(w_0)}(i,j)$$

$$+\left(\frac{t_1-t_0}{t_1}\right)\Bigg[p^{(w_0}(X_2=j|X_1=i)$$
$$\left(\frac{t_1(T-t_2)}{T^2}p^{(w_2)}(X_1=i)+\left(\frac{t_1}{T}\right)^2\left(\frac{t_2-t_1}{t_1}\right)p^{(w_2)}(X_1=i)\right)$$
$$+p^{(w_0)}(X_1=i|X_2=j)$$
$$\left(\frac{t_1(T-t_2)}{T^2}p^{(w_2)}(X_2=j)+\left(\frac{t_1}{T}\right)^2\left(\frac{t_2-t_1}{t_1}\right)p^{(w_2)}(X_2=j)\right)\Bigg]$$

Grouping by coefficient we get:

$$p_{t_0}(i,j)=p_{t_1}(i,j)+\left(\frac{t_1-t_0}{t_1}\right)^2\left(\frac{t_1}{T}\right)^2p^{(w_0)}(i,j)$$
$$+\left(\frac{t_1-t_0}{t_1}\right)\cdot$$
$$\Bigg[\left(\frac{t_1(T-t_2)}{T^2}\right)[p^{(w_0)}(X_2=j|X_1=i)p^{(w_2)}(X_1=i)+p^{(w_0)}(X_1=i|X_2=j)p^{(w_2)}(X_2=j)]$$
$$+\left(\frac{t_1}{T}\right)^2\left(\frac{t_2-t_1}{t_1}\right)[p^{(w_0)}(X_2=j|X_1=i)p^{(w_1)}(X_1=i)+p^{(w_0)}(X_1=i|X_2=j)p^{(w_1)}(X_2=j)]\Bigg]$$
$$=p_{t_1}(i,j)+\left(\frac{t_1-t_0}{t_1}\right)^2\left(\frac{t_1}{T}\right)^2p^{(w_0)}(i,j)$$
$$+\left(\frac{t_1-t_0}{t_1}\right)\cdot\left[\frac{t_1(T-t_2)}{T^2}p^{(w_0,w_2)}+\left(\frac{t_1}{T}\right)^2\left(\frac{t_2-t_1}{t_1}\right)p^{(w_0,w_1)}\right]$$

Simplifying and substituting the term of p_{t_1} this becomes:

$$p_{t_0}(i,j)=\left(\frac{t_3-t_2}{t_3}\right)^2p^{(w_2)}(i,j)+\left(\frac{t_2-t_1}{t_3}\right)^2p^{(w_1)}(i,j)+\left(\frac{t_1-t_0}{t_3}\right)^2p^{(w_0)}(i,j)$$
$$+\frac{(t_3-t_2)(t_2-t_1)}{t_3^2}p^{(w_1,w_2)}(i,j)+\frac{(t_3-t_2)(t_1-t_0)}{t_3^2}p^{(w_0,w_2)}(i,j)$$
$$+\frac{(t_2-t_1)(t_1-t_0)}{t_3^2}p^{(w_0,w_1)}(i,j).$$

□

D DETAILS ON TOY EXAMPLE

We now present the details of the toy example that we used to demonstrate our theoretical results. In figure 9 we present plots of each class and the full data distribution. Each cluster is defined via the following matrix

$$\begin{bmatrix}0.1&0.2&0.3&0.4&0.5&0.4&0.3&0.2&0.1\\0.2&0.4&0.6&0.7&0.8&0.7&0.6&0.4&0.2\\0.3&0.6&0.8&0.9&1.0&0.9&0.8&0.6&0.3\\0.4&0.7&0.9&1.0&1.0&1.0&0.9&0.7&0.4\\0.5&0.8&1.0&1.0&1.0&1.0&1.0&0.8&0.5\\0.4&0.7&0.9&1.0&1.0&1.0&0.9&0.7&0.4\\0.3&0.6&0.8&0.9&1.0&0.9&0.8&0.6&0.3\\0.2&0.4&0.6&0.7&0.8&0.7&0.6&0.4&0.2\\0.1&0.2&0.3&0.4&0.5&0.4&0.3&0.2&0.1\end{bmatrix}$$

Each class is equally weighted. A pseudocode for generating the above dataset is:

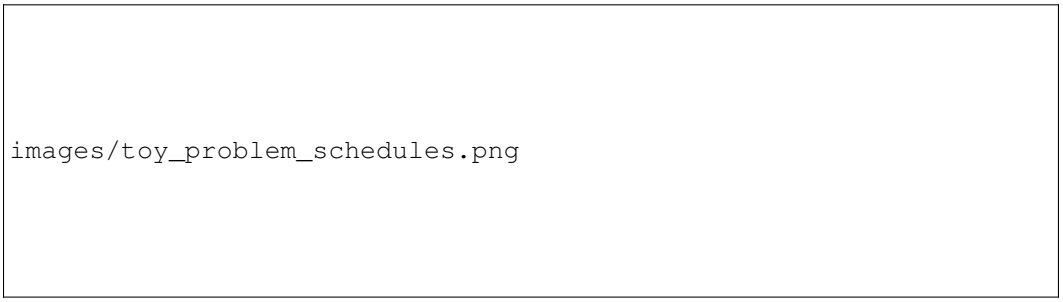


Figure 9: Definitions of the class and unconditional distributions for the toy problem.

```
    height, width = 30, 30

    matrix1 = torch.zeros((height, width))
    matrix1[1:10, 1:10] = torch.tensor(cluster)
    matrix1[9:18, 9:18] = torch.tensor(cluster)

    matrix2 = torch.zeros((height, width))
    matrix2[11:20, 11:20] = torch.tensor(cluster)
    matrix2[19:28, 19:28] = torch.tensor(cluster)
```

Listing 3: Code to generate our toy dataset

E NORMALIZATION FOR GENERAL DIFFUSION PROCESSES

In this section we demonstrate how it is possible to extend our normalization to the general class of diffusion processes. We propose a simple but effective per-column normalization that applies to different discrete diffusion models (masked/uniform).

$$\underbrace{\begin{bmatrix} \frac{p_t^w(1)q_t^{1-w}(1)}{p_t^w(x)q_t^{1-w}(x)} \\ \frac{p_t^w(2)q_t^{1-w}(2)}{p_t^w(x)q_t^{1-w}(x)} \\ \vdots \\ -\sum_{i\neq x} \frac{p_t^w(i)q_t^{1-w}(i)}{p_t^w(x)q_t^{1-w}(x)} \\ \vdots \\ \frac{p_t^w(M)p_t^{1-w}(M)}{p_t^w(x)p_t^{1-w}(x)} \end{bmatrix}}_{\text{Unnormalized vector}} \implies \underbrace{\frac{\left(\sum_{i\neq x} p_t(i)\right)^w \left(\sum_{i\neq x} q_t(i)\right)^{1-w}}{\sum_{i\neq x} p_t(i)^w q_t(i)^{1-w}} \cdot \begin{bmatrix} \frac{p_t^w(1)q_t^{1-w}(1)}{p_t^w(x)q_t^{1-w}(x)} \\ \frac{p_t^w(2)q_t^{1-w}(2)}{p_t^w(x)q_t^{1-w}(x)} \\ \vdots \\ -\sum_{i\neq x} \frac{p_t^w(i)q_t^{1-w}(i)}{p_t^w(x)q_t^{1-w}(x)} \\ \vdots \\ \frac{p_t^w(M)p_t^{1-w}(M)}{p_t^w(x)p_t^{1-w}(x)} \end{bmatrix}}_{\text{Normalized vector}} . \tag{12}$$

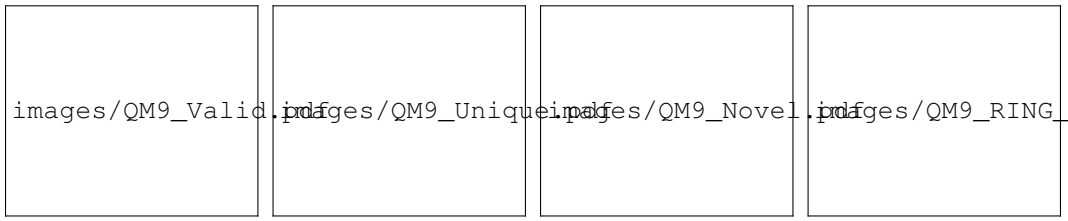
As seen in (12), our normalization applies to general diffusion models as long as we have access the scores for models of p, q. For the multiple-token case ($d > 1$), due to the fact that we only use one column vector that corresponds to a single-dimension jump every time, the normalization for one-token in (12) can be applied to that column vector.

```
def get_normalized_rate(
        x, c, t, dt, w
    ):
    # Get scores
    log_score_c = get_score(x, t, cond=c)
    logs_core_u = get_score(x, t, cond=None)
    log_score_w = w * log_score_c + (1-w) * log_score_u

    score_c = log_score_c.exp()
    score_u = log_score_u.exp()
    score_w = log_score_w.exp()

    # Set diagonal terms
    score_c.scatter_(-1, x[..., None], torch.zeros_like(score_c))
    score_u.scatter_(-1, x[..., None], torch.zeros_like(score_u))
    score_w.scatter_(-1, x[..., None], torch.zeros_like(score_w))

    normalized_rate = edge * score_w
    normalized_rate.scatter_(-1, x[..., None], -normalized_rate.sum(dim=-1, keepdim=True))

    # Normalize appropriately
    sum_c = score_c.sum(-1,keepdim=True) ** w
    sum_u = score_u.sum(-1,keepdim=True)
    sum_u = torch.where(sum_u > 0, sum_u**(1-w), 0)
    sum_w = score_w.sum(-1,keepdim=True)
    normalized_rate = (sum_c * sum_u / sum_w) * normalized_rate


    return sample(delta(x) + dt * sigma(t) * normalized_rate)
```

Listing 4: Our guidance in the general case using Euler transitions

E.1 RESULTS ON QM9

We present results using our guidance mechanism in the context of uniform diffusion, applied to the QM9 small molecule dataset (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014). QM9 is a dataset containing small organic molecules containing up to 9 heavy atoms. We train a conditional model on QM9 using uniform diffusion, based on the official implementation of Schiff et al. (2024), without modifying the architecture or hyperparameters. The model is conditioned on the number of rings in each molecule (ring count). Unlike ImageNet, evaluation on QM9 is more nuanced: we generate 1,024 samples and assess several metrics. First, generated molecules must satisfy chemical constraints to be considered valid. Second, a key goal of generative models is to produce novel



(a) Percentage of valid generated molecules

(b) Percentage of unique generated molecules

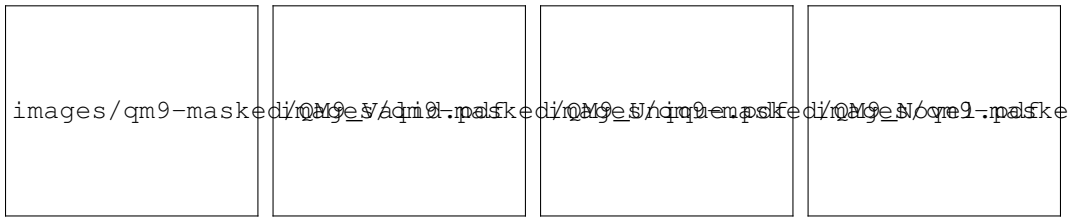
(c) Percentage of novel generated molecules

(d) Ring Count Mean as a function of guidance strength

Figure 10: We display the percentage of valid, unique, and novel molecules. We find that our method is the most robust to an increase in guidance strength.

molecules not found in the training data. Accordingly, we report both validity and novelty, along with the mean ring count (i.e., the conditioning signal), in Figure 10.

We find that our method is the most robust to increases in guidance strength. However, in general, all methods perform comparably well across the full range of strengths. This suggests that normalization may have a less pronounced effect in the uniform diffusion setting. Due to the complexity of evaluating on QM9, further experiments on additional discrete datasets are needed to more conclusively determine the optimal guidance mechanism. We leave this for future work.



(a) Percentage of valid generated molecules (b) Percentage of unique generated molecules (c) Percentage of novel generated molecules (d) Ring Count Mean as a function of guidance strength

Figure 11: We display the percentage of valid, unique, and novel molecules. We find that our method is the most robust to an increase in guidance strength.

F Extra experiments on QM9 for Masked Diffusion

We present similar results using our guidance mechanism but in the context of masked diffusion, applied to the QM9 small molecule dataset (Ruddigkeit et al., 2012; Ramakrishnan et al., 2014). We train a conditional model on QM9 using uniform diffusion, based on the official implementation of Schiff et al. (2024), without modifying the architecture or hyperparameters. The model is conditioned on the number of rings in each molecule (ring count). We report statistics under the same setting as in Appendix E.

We plot the results in Figure 11, we find that our method is the most robust to increases in guidance strength and generally achieves better results across various guidance strengths. Suggesting that normalization is also a helpful technique on this domain.

G Analysis of Diversity tradeoffs

We present a simple study of the diversity tradeoff when using different guidance mechanisms for discrete diffusion. We leverage the Imagenet dataset as in the main paper, using the same set of hyperparameters. We analyze the precision-recall Kynkäänniemi et al. (2019) and report the results in Table 3. Precision measures the proportion of generated images that lie close to the real data manifold (fidelity), while recall quantifies the coverage of the real data distribution (diversity), allowing us to understand this correctly.

As guidance strength increases from w=1 to w=5, all methods exhibit stable recall, ranging from 0.72 to 0.79. The precision statistic is more revealing: for Unlocking and Simple guidance, using $w \geq 1$ always results in degradated precision (worse fidelity) while our method is capable of improving on it (better fidelity). Putting this together, all methods show similar diversity across guidance strengths. However, only our method can improve the fidelity while maintaining diverse samples.

H Generated Samples Text to Image

To generate these samples we made use of a single node with 8 NVIDIA A100 GPUs. We present samples to compare our method against other guidance methods as well as the detailed results of the GenEval benchmark in Table 4 and 5. The results demonstrate that normalization is key in order to improve the sample quality.

Table 3: Performance across guidance strengths w. Each cell shows *Precision/Recall*.

Method / Strength	w=1 (no guidance)	w=2	w=3	w=4	w=5
Our Method	0.48 / 0.73	0.51 / 0.75	0.51 / 0.75	0.52 / 0.76	0.18 / 0.72
Unlocking Guidance	0.48 / 0.73	0.43 / 0.79	0.29 / 0.77	0.19 / 0.77	0.12 / 0.76
Simple Guidance	0.46 / 0.72	0.45 / 0.79	0.34 / 0.76	0.24 / 0.74	0.19 / 0.72

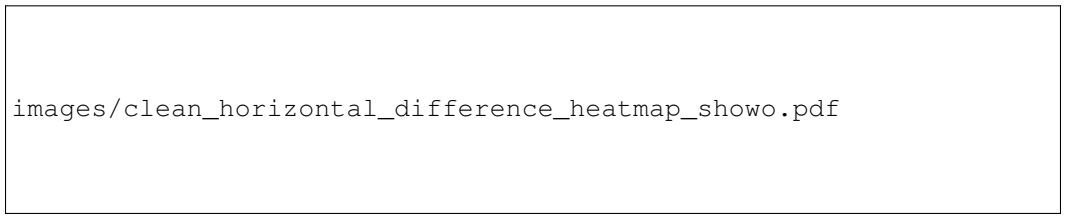


Figure 12: GenEval with and without normalization using Show-o as a base model. Red denotes an improved performance due to normalization. Normalization leads to more faithful prompt adherence and image quality.

Table 4: Performance comparison across different guidance weights using Meissonic as a base model

	Ours				Unlocking			
Metric	$w=1$	$w=3$	$w=6$	$w=9$	$w=1$	$w=3$	$w=6$	$w=9$
Overall	8.2	40.8	45.9	44.7	8.2	43.1	28.5	19.9
Objects Single	23.8	89.4	91.9	91.2	23.8	88.4	80.0	64.1
Objects Two	3.0	36.9	48.2	48.7	3.0	47.5	23.2	18.2
Counting	0.6	27.2	33.8	28.4	0.6	33.1	11.9	3.1
Position	20.7	72.3	77.4	77.7	20.7	72.3	48.7	29.0
Color Attribution	0.2	8.5	7.8	7.8	0.2	6.2	7.8	3.8
Colors	1.0	10.5	16.5	14.5	1.0	10.8	2.8	1.0

Table 5: Performance comparison across different guidance weights using Show-o as a base model

	Ours				Unlocking			
Metric	$w=2$	$w=4$	$w=6$	$w=8$	$w=2$	$w=4$	$w=6$	$w=8$
Overall	56.42	62.46	63.13	63.39	53.73	52.84	52.89	43.96
Objects Single	96.88	98.75	99.06	98.75	95.94	98.44	97.19	86.88
Objects Two	65.66	75.76	78.28	80.05	64.14	61.36	60.61	60.10
Counting	41.56	50.00	50.94	50.94	39.69	35.00	35.00	31.25
Position	78.19	81.38	79.79	81.12	73.14	75.00	76.60	51.33
Color Attribution	22.75	28.5	30.25	27.25	26.25	22.75	22.00	20.25
Colors	33.5	41.5	40.5	42.25	23.25	24.50	26.00	14.00

H.1 Generated Samples from Meissonic

We now present some samples from our method:

Figure 13: Comparison of samples generated by different guidance methods across various seeds and configurations. Using the prompts: A photo of a bench (top), A photo of a cow (bottom).

H.2 Generated Samples from Show-o

We now present some samples from our method using Show-o Xie et al.:

Figure 14: Comparison of samples generated by different guidance methods across various seeds and configurations. Using the prompts: A photo of a bike (top), A photo of a clock (bottom).

I GENERATED SAMPLES IMAGENET

I.1 GUIDANCE STRENGTH $w = 2$

Figure 15: Comparison of samples generated by different guidance methods across various seeds and configurations. Using the prompts: A photo of a carrot (top), A photo of a suitcase (bottom).

I.2 GUIDANCE STRENGTH $w = 3$

Figure 16: Comparison of samples generated by different guidance methods across various seeds and configurations. Using the prompts: A photo of a fork (top), A photo of a surfboard (bottom).

I.3 GUIDANCE STRENGTH $w = 4$

Figure 17: Comparison of samples generated by different guidance methods across various seeds and configurations. Using the prompt: A photo of a refrigerator (top), A photo of a cup (bottom)

I.4 GUIDANCE STRENGTH $w = 5$

Figure 18: Comparison of samples generated by different guidance methods across various seeds and configurations. Using the prompts: A photo of a dog (top), A photo of a tie (bottom).

STATEMENT ON THE USE OF LARGE LANGUAGE MODELS

This work made use of large language models to assist with proofreading and improving the clarity of the writing. All research ideas, theoretical development, and experiments were carried out solely by the authors. When used for coding, it was solely used for plotting purposes.

Figure 19: Comparison of samples generated by different guidance methods across various seeds and configurations. Using the prompts: A photo of a laptop (top), A photo of a computer mouse (bottom).

Figure 20: Comparison of samples generated by different guidance methods across various seeds and configurations. Using the prompts: A photo of a sandwich (top), A photo of a baseball bat (bottom).

Figure 21: Comparison of samples generated by different guidance methods across various seeds and configurations. Using the prompts: A photo of a train (top), A photo of a cell phone (bottom).

Figure 22: Comparison of samples generated by different guidance methods across various seeds and configurations. Using the prompt: A photo of a chair (top), A photo of a tv (bottom)

Figure 23: Comparison of samples generated by different guidance methods across various seeds or configurations.

Figure 24: Comparison of samples generated by different guidance methods across various seeds or configurations.

Figure 25: Comparison of samples generated by different guidance methods across various seeds or configurations.

Figure 26: Comparison of samples generated by different guidance methods across various seeds or configurations.