GUIDED TRANSFER LEARNING FOR DISCRETE DIFFUSION MODELS

Julian Kleutgens[1,2], **Claudio Battiloro**[1], **Lingkai Kong**[1], **Benjamin Grewe**[2], **Francesca Dominici**[1], **Mauricio Tec**[1]
[1] Harvard University, [2] ETH Zürich

ABSTRACT

Discrete diffusion models (DMs) have achieved strong performance in language and other discrete domains, offering a compelling alternative to autoregressive modeling. Yet this performance typically depends on large training datasets, challenging the performance of DMs in small-data regimes—common under real-world constraints. Aimed at this challenge, recent work in continuous DMs suggests that transfer learning via classifier ratio–based guidance can adapt a pretrained DM to a related target distribution, often outperforming alternatives such as full-weight fine-tuning on the target data. By contrast, transfer learning for discrete DMs remains unexplored. We address this gap by exploring practical analogues of ratio-based transfer learning for discrete DMs. Our theoretical analysis shows that a direct extension of existing ratio-based guidance is computationally prohibitive, scaling with vocabulary size. To overcome this limitation, we introduce a scheduling mechanism that yields a practical algorithm, *Guided Transfer Learning* for discrete diffusion models (GTL). GTL enables sampling from a target distribution without modifying the pretrained denoiser and reduces the cost to linear scaling in vocabulary size, which in turn supports longer sequence generation. We evaluate GTL on sequential data, including synthetic Markov chains and language modeling tasks, and provide a detailed empirical analysis of its behavior. The results highlight a clear trade-off: when target datasets are large, weight fine-tuning is often preferable, whereas GTL becomes increasingly effective as target data shrinks. Finally, we experimentally demonstrate a key failure mode of GTL: when the source and target distributions overlap poorly, the ratio-based classifier required for guidance becomes unreliable, limiting transfer performance.

1 INTRODUCTION

Diffusion models (DMs) (Ho et al., 2020; Song et al., 2021) have become the dominant approach for generative modeling in continuous domains like image, video, and audio (Lovelace et al., 2023; Ho et al., 2022; Liu et al., 2023). More recently, discrete DMs for language have achieved competitive performance and, by denoising tokens in parallel and in any order, offer practical advantages for controllable infilling and editing compared to left-to-right autoregressive decoding (Gong et al., 2025; Li et al., 2022a). Among existing discrete diffusion approaches, masked diffusion models (MDMs) have demonstrated remarkable performance in tasks such as reasoning (Nie et al., 2025; Zheng et al., 2025), planning (Ye et al., 2025), and infilling (Gong et al., 2025), often performing on par with state-of-the-art auto-regressive language models (ALMs).

Training DMs require large amounts of data (Wang et al., 2023). And it remains an open question how to best leverage their powerful generative modeling capabilities in domains where scarse data is available (Zhang et al., 2025), a common scenario in applications like medical imaging, where privacy and limited public availability constrain dataset size (Xie et al., 2025). This problem motivates the exploration of transfer learning techniques, where the goal is leverage a DM pretrained on a large dataset for generative modeling in a related domain where data is more scarse. One possible solution is to fine-tune the pretrained DM's denoiser on the target data (Xie et al., 2023a), but this becomes ineffective in very small data regimes (Moon et al., 2022). To address this, recent works have proposed *transfer learning* to tackle the small data regime by using classifier-based guidance

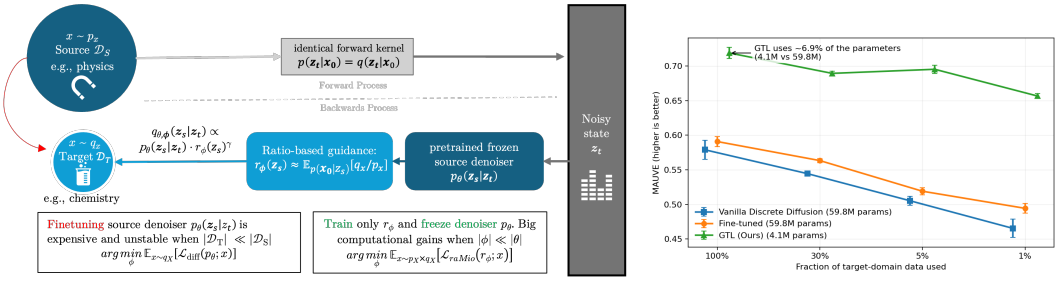

Figure 1: *Visual summary of Guided Transfer Learning (GTL) for Discrete Diffusion and Results Preview*. (Left) Method overview: adapt to the target domain by reweighting the frozen source reverse transitions using a learned ratio model. (Right) GTL outperforms finetuning with only $\sim$7% of the parameters. MAUVE (↑) vs. fraction of target-domain training data (100% = all 79,631 arXiv Physics abstracts; $1\% \approx 800$). GTL (green) achieves the highest MAUVE across all data regimes outperforming Vanilla (target-only) and Fine-tuned baselines. Points show mean and error bars show standard error over 3×3 (train$\times$sample) seeds. See Section 4.2 for setup and evaluation details.

(CBG) (Zhong et al., 2025; Ouyang et al., 2024), which estimates a density-ratio between the source and target domain and uses it as a diffusion guidance mechanism. Such methods are more effective than finetuning in scenarios of extreme scarcity, as they bypass the need of updating millions of parameters (Moon et al., 2022; Xie et al., 2023b). However, these methods have only been investigated in continuous domains (e.g., images). By contrast, transfer learning for discrete DMs such as language models, remains largely unexplored.

Outside of transfer learning, Schiff et al. (2025); Nisonoff et al. (2025) have explored CBG mechanisms for discrete DMs. These methods apply mainly to domains with small vocabulary size $|\mathcal{V}|$ and modest sequence length L, such as biological sequences, leaving scalable CBG for language modeling as an open problem. This constaints are due to the necessity of token substitutions across all positions and vocabulary items during score computation. More precisely, naive CBG requires $\mathcal{O}(L|\mathcal{V}|)$ guidance evaluations per step, which is infeasible for long sequences and large vocabularies. In this paper, we aim to leverage CBGs for transfer learning, while overcoming the limiting small vocabulary sizes.

First, we present a theoretical analysis deriving a closed-form ratio-guided reverse transition for discrete-time diffusion and extend this principle to continuous-time, score-based discrete diffusion, covering both score-based models (Lou et al., 2024; Sun et al., 2023) and discrete-time diffusion models (Austin et al., 2023). Such derivation is architecture- and corruption-agnostic—deriving the result from the evidence lower bound (ELBO). Next, we address the scaling problem of CBGs for discrete DMs. For this purpose, we introduce an efficient sampler that makes guided language modeling for masked discrete DMs practical by avoiding $\mathcal{O}(L|\mathcal{V}|)$ guidance calls, concentrating ratio evaluations on planner-selected positions and a top-n_{ratio} candidate set.

Leveraging this result, we introduce a method termed Guided Transfer Learning (GTL). The core idea of GTL is to freeze a pretrained source denoiser and learn a lightweight ratio network from mixed source/target data to estimate target-to-source density ratios; at sampling time, we use this ratio with a learned planner network as guidance to reweight the denoiser's reverse transition and generate target-domain samples without fine-tuning. This procedure allows to effectively utilize transfer learning based on CBG for discrete DMs. Previewing our results, GTL outperforms vanilla and fine-tuned diffusion across data-scarcity regimes while training only $\sim 7\%$ as many parameters (Fig. 1). In a setting common in practice—where the pretraining corpus already contains some in-domain target samples—we find that GTL performs better and yields more stable performance. Our result also reveals limitations of CBG for discrete modeling: under weak domain overlap, the interaction between guidance strength γ and candidate pruning can amplify ratio errors. In this regime, larger γ or small n_{ratio} may destabilize denoising and hurt quality.

We summarize our contributions as follows:

1. We introduce GTL, an effective transfer learning framework for discrete diffusion models. GTL keeps the source denoiser fixed while training a lightweight ratio estimator on both source and target data to enable sampling from the target distribution. GTL achieves scalability by using an unmasking position for the top-n_{ratio} candidates, dramatically reducing the per-step computational cost, without sacrificing performance in practice.
2. We provide a theoretical analysis and formal theorem of the proposed guided transfer learning method, demonstrating that ratio-based guidance correctly recovers the target reverse transition.
3. We validate GTL and the guided sampler on sequential data, including synthetic Markov chains and arXiv-abstract language modeling tasks.

2 BACKGROUND

Diffusion models are generative models defined by a forward noising process and a learned reverse denoising process. In the reverse process, a denoiser p_θ is trained to remove noise from latent variables $\mathbf{z}_t$ to recover data x. The forward process generates $\mathbf{z}_t$ by injecting noise into clean data. We consider discrete latents: $\mathbf{z}_t \in \mathcal{V}$, the set of one-hot tokens over a vocabulary $\mathcal{V}$ of size N, $\mathcal{V} := \{\mathbf{z} \in \{0,1\}^N : \sum_{i=1}^N z_i = 1\} \subset \Delta^{N-1}$, where Δ^{N-1} denotes the probability simplex over N categories. Let $\text{Cat}(\cdot;\pi)$ be the categorical distribution over N classes with the probability vector $\pi \in \Delta^{N-1}$.

Discrete diffusion Discrete Denoising Diffusion Probabilistic Models (D3PMs) (Austin et al., 2023) provide a flexible framework for diffusion over discrete state spaces. The forward Markov chain is $p(\mathbf{z}_t \mid \mathbf{z}_s) = \text{Cat}\big(\mathbf{z}_t;\, Q_{t|s}\mathbf{z}_s\big)$ where $Q_{t|s} \in \mathbb{R}^{N\times N}$ is a transition matrix whose (i,j) entry is the probability of moving to state i at time t from state j at time s. To fit the generative model p_θ to the data distribution over $\mathbf{x}_0$, we minimize the negative ELBO (NELBO). Different choices of $Q_{t|s}$ induce different corruption behaviors, e.g., uniform, absorbing, or discretized-Gaussian transitions.

Masked (absorbing-state) diffusion. The absorbing-state variant or often called *masked diffusion*, replaces tokens with a special `[MASK]` token (Sahoo et al., 2024; Shi et al., 2025). It can be viewed as an interpolation between the clean distribution and a noise prior over the vocabulary. For a clean token $\mathbf{x}_0$, the forward transition at time t is

$$p(\mathbf{z}_t \mid \mathbf{x}_0) = \text{Cat}(\mathbf{z}_t;\, \alpha_t \mathbf{x}_0 + (1-\alpha_t)\,\mathbf{m})\,, \tag{1}$$

where $\alpha_t \in [0,1]$ controls the corruption schedule and $\mathbf{m} \in \mathcal{V}$ is one-hot vector at the special `[MASK]` token. Making use of the masking process, the backward posterior simplifies to:

$$p(\mathbf{z}_s \mid \mathbf{z}_t, \mathbf{x}_0) = \begin{cases} \text{Cat}(\mathbf{z}_s;\, \mathbf{z}_t)\,, & \text{if } \mathbf{z}_t \neq \mathbf{m}, \\ \text{Cat}\left(\mathbf{z}_s;\, \dfrac{(1-\alpha_s)\mathbf{m} + (\alpha_s - \alpha_t)\mathbf{x}_0}{1-\alpha_t}\right), & \text{if } \mathbf{z}_t = \mathbf{m}. \end{cases} \tag{2}$$

Hence, when a token is unmasked ($\mathbf{x}_t \neq \mathbf{m}$) the step is deterministic: $\mathbf{x}_s = \mathbf{x}_t$. Otherwise, the posterior is a linear interpolation between the mask distribution and the clean token. This structure yields the simplified diffusion loss (Sahoo et al., 2024; Shi et al., 2025):

$$\mathcal{L}_{\text{diffusion}} = \sum_{i=1}^{T} D_{\text{KL}}\big(p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\big\|\, p_\theta(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big) = \sum_{i=1}^{T} \mathbb{E}_q\left[\frac{\alpha_{t(i)} - \alpha_{s(i)}}{1-\alpha_{t(i)}} \log \big\langle \mathbf{x}_\theta(\mathbf{z}_{t(i)}), \mathbf{x}_0 \big\rangle\right], \tag{3}$$

where $\mathbf{x}$ is the one-hot encoding of the clean data and $\langle\cdot,\cdot\rangle$ is the inner product. The network $\mathbf{x}_\theta$ predicts the clean token distribution from the noisy input $\mathbf{z}_{t(i)}$.

Guidance Diffusion models provide strong controllability through both classifier-based and classifier-free guidance. These two approaches correspond to different ways of expressing the backward generative process conditioned on y. In this work, we focus on the classifier-based variant, which allows us to rearrange the conditional generative process using Bayes' rule as

$p^\gamma(\mathbf{z}_s \mid \mathbf{z}_t, y) = p(y \mid \mathbf{z}_s, \mathbf{z}_t)^\gamma \, p(\mathbf{z}_s \mid \mathbf{z}_t)/Z$, with Z being a normalizing constant. (Schiff et al., 2025; Nisonoff et al., 2025)

For the extension to sequences of tokens $\mathbf{z}_s^{(1:L)}$ of length L, we assume that the distribution $p^\gamma\Big(\mathbf{z}_s^{(1:L)} \mid \mathbf{z}_t^{(1:L)}, y\Big)$ factorizes independently across tokens. This leads to the following formulation:

$$p^\gamma\Big(\mathbf{z}_s^{(1:L)} \mid \mathbf{z}_t^{(1:L)}, y\Big) = \prod_{\ell=1}^{L} \frac{p\Big(y \mid \mathbf{z}_s^{(\ell)}, \mathbf{z}_t^{(1:L)}\Big)^\gamma \, p\Big(\mathbf{z}_s^{(\ell)} \mid \mathbf{z}_t^{(1:L)}\Big)}{\sum_{\tilde{\mathbf{z}}_s^{(\ell)} \in \mathcal{V}} p\Big(y \mid \tilde{\mathbf{z}}_s^{(\ell)}, \mathbf{z}_t^{(1:L)}\Big)^\gamma \, p\Big(\tilde{\mathbf{z}}_s^{(\ell)} \mid \mathbf{z}_t^{(1:L)}\Big)} \tag{4}$$

Here, $\mathcal{V}$ denotes the vocabulary set, and $p\Big(y \mid \mathbf{z}_s^{(\ell)}, \mathbf{z}_t^{(1:L)}\Big)$ represents the classifier probability given the sequence $\mathbf{z}_t^{(1:L)}$ with the token $\mathbf{z}_s$ placed at position ℓ. However, this formulation comes with the drawback of requiring $\mathcal{O}(L \cdot |\mathcal{V}|)$ forward passes through the classifier model, which is infeasible for long sequences and large vocabularies. Because each evaluation corresponds to a distinct single-token intervention on the full sequence, this cost is not removed by straightforward parallelization.

Problem Formulation Let $\mathcal{X}$ denote the data space, with source domain distribution p_X and target domain distribution q_X. We are given source samples $\mathcal{S} = \{\mathbf{x}_i^{(1:L)}\}_{i=1}^m \sim p_X$ and target samples $\mathcal{T} = \{\mathbf{x}_i^{(1:L)}\}_{i=1}^n \sim q_X$. The goal of transfer learning is to train a network p_θ on $\mathcal{S}$ and adapt it to $\mathcal{T}$ to improve performance in the target domain. In practice, $n \ll m$, creating an imbalance between source and target data. A common strategy is to freeze most pretrained layers, extract features, and train new task-specific layers on $\mathcal{T}$. Fine-tuning relaxes this by unfreezing parameters, allowing the model to adapt more flexibly to the target distribution. In our setting, we are given a pretrained generative model p_θ trained on the source distribution p_X, and we aim to transfer its generative process to the target distribution q_X, yielding a target-domain model q_ψ. Note that the background section is introduced for the source domain diffusion.

3 Guided Transfer Learning for DDMs

We introduce *Guided Transfer Learning* (GTL), a transfer framework for discrete diffusion models that adapts a pretrained source model to a target domain *without* updating the denoiser. Our approach is orthogonal to fine-tuning: rather than updating the denoiser, GTL adapts sampling by reweighting the source reverse transitions with a learned density-ratio guidance model. In this section, we first present the core transfer identity for a single denoising step, then extend it to sequences and develop an efficient guided sampler.

Why not fine-tune? Fine-tuning adapts the source denoiser p_θ to the target domain by optimizing the diffusion objective (e.g., equation 3) on target samples. This optimization becomes expensive for large language models and tends to overfit when target data are scarce, often degrading diversity. These small-data failures motivate our approach: rather than updating millions of denoiser parameters, we reuse p_θ as-is and place the adaptation burden on a much smaller auxiliary model.

Key Assumption. This auxiliary model is meaningful only if the two domains share the same corruption mechanism. Let p_X denote the source data distribution and q_X the target data distribution over $\mathbf{x}_0$. We assume a shared forward process, $p(\mathbf{z}_t \mid \mathbf{x}_0) = q(\mathbf{z}_t \mid \mathbf{x}_0)$ for all t, which holds whenever both domains use the same discrete forward kernel (e.g., an identical masking schedule or transition family). Under this shared corruption, differences between the domains can be expressed at the data level through the density ratio $r(\mathbf{x}_0) \coloneqq q_X(\mathbf{x}_0)/p_X(\mathbf{x}_0)$, assumed well-defined on the support of interest. This ratio quantifies how much more (or less) likely a clean sequence is under the target distribution than under the source, and it is exactly the quantity needed to reweight source reverse transitions into target reverse transitions.

The next theorem makes this reweighting precise for a single reverse transition: it shows that the reverse kernel minimizing the target KL objective can be written as the source reverse kernel multiplied by a ratio-dependent weight.

Theorem 1 *Let p (source) and q (target) be two diffusion models that share the same forward process, $p(\mathbf{z}_t \mid \mathbf{x}_0) = q(\mathbf{z}_t \mid \mathbf{x}_0)$ for all t. The reverse target domain network $q_\theta(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})$ is defined*

by:

$$\arg\min_{\psi} \mathbb{E}_q \left[\sum_{i=1}^{T} D_{\mathrm{KL}}\Big(q(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\Big\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) \Big) \right] \tag{5}$$

Then we have

$$q_{\psi^\star}(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) = \frac{p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) \mathbb{E}_{\mathbf{x}_0 \sim p(\cdot \mid \mathbf{z}_{s(i)})} \left[\frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)} \right]}{\sum_{\tilde{\mathbf{z}}_{s(i)}} p(\tilde{\mathbf{z}}_{s(i)} \mid \mathbf{z}_{t(i)}) \mathbb{E}_{\mathbf{x}_0 \sim p(\cdot \mid \tilde{\mathbf{z}}_{s(i)})} \left[\frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)} \right]} \tag{6}$$

Practical implementation. Equation 6 (proved in Appendix C) expresses the target reverse transition as a reweighted version of the source reverse transition. This expression suggests a direct implementation with two components: a frozen source denoiser that provides $p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})$ and a learned ratio module that provides the ratio-dependent weight. Concretely, we approximate the source kernel with the pretrained denoiser $p_\theta(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})$, and we learn a ratio function $r_\phi(\mathbf{z}_{s(i)}) \approx \mathbb{E}_{\mathbf{x}_0 \sim p(\cdot \mid \mathbf{z}_{s(i)})} \left[\frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)} \right]$. Together, these two networks define the GTL sampler, $\psi^\star = [\theta, \phi]$, which generates target-domain samples while keeping the denoiser fixed. The ratio network is trained using a discrete-data loss adapted from Ouyang et al. (2024). Additional training details are provided in the supplementary materials.

Why ratio guidance helps in the low-data regime? The key advantage of equation 6 is that it eliminates the need to fine-tune the denoiser on scarce target data. Instead, adaptation is shifted to learning a domain discriminator (or an equivalent ratio estimator) from mixed source/target samples, which can be orders of magnitude smaller than the diffusion model. This shift is especially helpful under strong source–target imbalance: ratio estimation only needs to separate domains, rather than model the full target distribution, and it admits flexible regularization that tends to stabilize training in data-scarce settings.

Relation to TLDM and generality of our formulation. TLDM (Ouyang et al., 2024) derives a ratio-transfer rule only for *continuous* data score-based diffusion in $\mathbb{R}^d$, where guidance is expressed through Euclidean score gradients such as $\nabla_{x_t} \log q_t(x_t)$. These gradients are not available for categorical tokens, and discrete diffusion is commonly trained by minimizing KL divergences between categorical posteriors rather than score-matching MSE. Our derivation therefore works directly with discrete forward and reverse kernels and the ELBO-style objective in equation 5. This formulation is architecture- and corruption-agnostic: it applies to a range of discrete forward processes (e.g., uniform, absorbing/masked, and general D3PM transitions) and does not depend on a particular parameterization of the denoiser in benefit to TLDM.

Continuous-Time GTL for Discrete Diffusion. Because the argument is expressed at the level of the ELBO/KL objective, the same ratio-transfer principle carries over to continuous-time discrete diffusion whenever the training loss can be related back to an ELBO (e.g., equation 3 and equation 36). In Appendix (Theorem 3), we instantiate this transfer rule for continuous-time, score-based diffusion over discrete state spaces by reweighting the reverse-rate matrix with the learned ratio.

Sequence Formulation. We now apply Theorem 1 to length-L token sequences. As in prior work on discrete classifier guidance, we adopt a token-factorized reverse transition, which yields a practical sampler while keeping the denoiser unchanged. Concretely, we replace the classifier term $p(y \mid \mathbf{z}_s^{(\ell)}, \mathbf{z}_t^{(1:L)})$ in equation 4 with a learned ratio model. In the sequence setting, the ratio model is evaluated on a *single-token intervention* of the current noisy sequence. Define $\mathbf{z}_{t,s^\ell}^{(1:L)} := [\, \mathbf{z}_t^{(1:\ell-1)}, \mathbf{z}_s^{(\ell)}, \mathbf{z}_t^{(\ell+1:L)} \,]$, i.e., the sequence obtained by replacing position ℓ in $\mathbf{z}_t^{(1:L)}$ by a candidate token $\mathbf{z}_s^{(\ell)}$. With guidance strength $\gamma \geq 0$, the resulting ratio-guided, factorized reverse transition is equation 7.

$$q_{\theta,\phi}\Big(\mathbf{z}_s^{(1:L)} \mid \mathbf{z}_t^{(1:L)}\Big) = \prod_{\ell=1}^{L} \frac{\Big[r_\phi\big(\mathbf{z}_{t,s^\ell}^{(1:L)}\big) \Big]^\gamma \, p_\theta\Big(\mathbf{z}_s^{(\ell)} \mid \mathbf{z}_t^{(1:L)}\Big)}{\sum_{\tilde{\mathbf{z}}_s^{(\ell)} \in \mathcal{V}} \Big[r_\phi\big(\mathbf{z}_{t,\tilde{s}^\ell}^{(1:L)}\big) \Big]^\gamma \, p_\theta\Big(\tilde{\mathbf{z}}_s^{(\ell)} \mid \mathbf{z}_t^{(1:L)}\Big)}. \tag{7}$$

Method	Per-Step Complexity	Number of Steps (T)
Ancestral Sampling	$\mathcal{O}(\lvert\mathcal{V}\rvert \cdot L)$	T
+ Caching	$\mathcal{O}(\lvert\mathcal{V}\rvert \cdot L)$	$T' \leq \min(L, T)$
+ Top-n_{ratio}	$\mathcal{O}(n_{\text{ratio}} \cdot L)$	$T' \leq \min(L, T)$
Planner Sampling		
+ Top-n_{ratio}	$\mathcal{O}(n_{\text{ratio}})$	L

Table 1: Computational cost per denoising step and total number of steps.

Algorithm 1 Ratio-Guided Denoise Step $(\mathbf{z}_t, t, \Delta t)$.

Require: noisy sequence $\mathbf{z}_t \in \mathcal{V}^L$, denoiser p_θ, ratio net r_ϕ, guidance schedule $\gamma(\cdot)$, planner ρ_ϑ

$\ell \leftarrow \rho_\vartheta(\mathbf{z}_t)$ {Get position by planner}
$\log x_\theta \leftarrow p_\theta(\mathbf{z}_t)$
$\mathcal{N} \leftarrow$ Top N tokens in $\log x_\theta[\ell, :]$
for all $v \in \mathcal{N}$ **do**
  $\tilde{\mathbf{z}} \leftarrow (\mathbf{z}_t[\ell] \leftarrow v)$
  $\log r[\ell, v] \leftarrow r_\phi(\tilde{\mathbf{z}}, \sigma_t)$ {Ratio scores}
end for
$\log q^{\text{guided}}[\ell, :] \leftarrow \log x_\theta[\ell, :] + \gamma(t) \log r[\ell, :]$
$\log q^{\text{guided}}[\ell, \mathcal{V} \setminus \mathcal{N}] \leftarrow -\infty$ {set non-Top-N tokens $-\infty$}
$\pi \leftarrow \mathrm{softmax}(\log q^{\text{guided}}[\ell, :])$
$\mathbf{z}_t^{(\ell)} \sim \mathrm{Cat}(\pi[\ell, :])$ {Sample at ℓ}
$\mathbf{z}_s \leftarrow \mathbf{z}_t$
return $\mathbf{z}_s$

A naive implementation of equation 7 is prohibitively expensive. For each masked position ℓ, the denominator requires evaluating $r_\phi(\mathbf{z}_{t,\tilde{s}^\ell}^{(1:L)})$ for all $\tilde{\mathbf{z}}_s^{(\ell)} \in \mathcal{V}$, resulting in $\mathcal{O}(L|\mathcal{V}|)$ ratio evaluations per denoising step. This is manageable for small L or $|\mathcal{V}|$ (e.g., our Markov-chain setting), but becomes prohibitive for text.

3.1 Scalability via Planner Sampling

Our goal is to reduce ratio-network evaluations from $\mathcal{O}(L|\mathcal{V}|)$ to a budget that is independent of $|\mathcal{V}|$ and ideally weakly dependent on L. Table 1 summarizes how caching, top-n_{ratio}, and planner sampling progressively reduce this cost.

Ancestral Sampling In the naive implementation, at all denoising steps, we compute logits from the denoiser and from the guidance network with $\mathcal{O}(|\mathcal{V}| \cdot L)$ complexity.
Caching. In masked diffusion, the timestep is implicitly determined by the number of masked tokens, allowing us to remove explicit time conditioning from the networks (Sahoo et al., 2024). As a result, if a denoising step does not modify the sequence, the corresponding logits can be reused. By caching logits across steps, the number of network evaluations becomes upper-bounded by the number of unmasking events. In the worst case, each position unmasks once, yielding at most L effective network calls. This optimization does not apply to uniform-noise diffusion, where token identities are mixed. ***Top-n_{ratio} pruning.*** Although caching reduces the number of steps, each step still requires evaluating the ratio network over the full vocabulary. To address this, we restrict ratio evaluations to a small candidate set. We assume that the denoiser distribution $p_\theta(\cdot \mid \mathbf{z}_t)$ is sufficiently concentrated, so that meaningful deviations between the source and target distributions occur primarily among the top-n_{ratio} tokens. Formally, we assume that for these candidates the density ratio $r(z) = q(z)/p(z)$ is bounded by a constant $C < \infty$; no such bound is required outside this set, where $p(z)$ is already negligible. Under this assumption, we evaluate the ratio term only for the top-n_{ratio} tokens proposed by the denoiser at each position. This produces position-specific candidate sets in the denominator of equation 7 and reduces the cost of ratio evaluations to $\mathcal{O}(n_{\text{ratio}}\, L)$, with $n_{\text{ratio}} \ll |\mathcal{V}|$ in practice. ***Mask-probability stabilizer.*** The interaction between top-n_{ratio} pruning and the guidance weight γ can distort probability mass, occasionally causing the

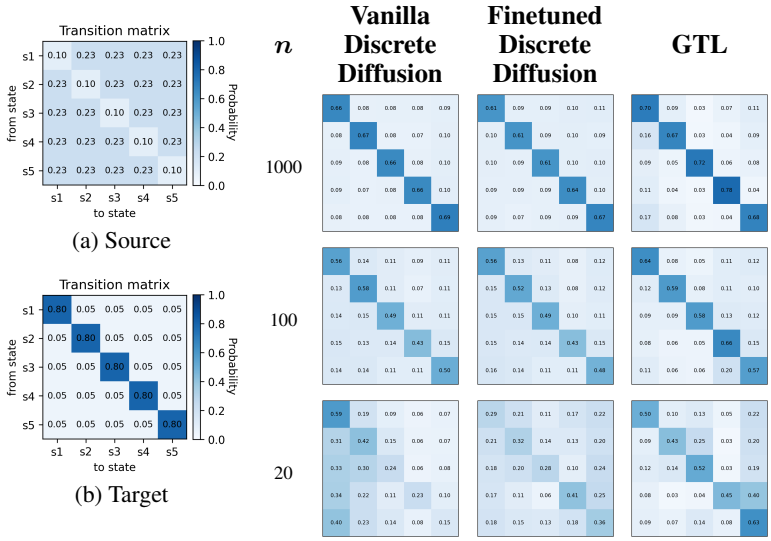

Figure 2: True source and target (left). Estimated target transition matrices for varying number of target samples n across three methods (right).

diffusion process to collapse to fully masked sequences. To prevent this failure mode, we enforce a position-independent constraint that preserves the probability of remaining masked across source and target distributions. This induces a renormalized update over non-mask tokens and stabilizes the denoising dynamics. Details of this stabilization are provided in the appendix.

Planner Sampling Even with top-n_{ratio} pruning, the overall complexity still scales linearly with L. To further reduce cost, we introduce a planner-based sampler that selects a single position to update at each step. The planner ρ_ϑ is trained to predict the next denoising position in the masked sequence, replacing the stochastic Gillespie process with a deterministic τ-leaping schedule. This strategy fixes the number of denoising steps to exactly L, which is comparable to standard masked diffusion where typically $T \approx L$, while substantially reducing guidance evaluations and wall-clock time. Algorithm 1 summarizes the resulting ratio-guided denoising step. Crucially, once the planner selects a position, ratio evaluations over the top-n_{ratio} candidates can be fully vectorized. As a result, each denoising step requires only a single forward pass through the denoiser, ratio network, and planner. For planner training, we follow simplified strategies from Peng et al. (2025); Liu et al. (2025).

4 Experiments

In this section, we present empirical results on the effectiveness of our method, Guided Transfer Learning for discrete diffusion models (GTL). We provide evidence across two applications showing that our method can successfully sample from the target distribution, in comparison to a vanilla diffusion model and standard finetuning. The first section presents a proof-of-concept on a synthetic dataset based on learning Markov chain data. The second subsection reports results on language modeling using arXiv abstracts.

4.1 Synthetic Dataset

Experimental Setup We construct two Markov chains with distinct transition matrices. Let the probability state vector be $\pi_i \in \Delta^{N-1}$ over a vocabulary of size $N = 5$. The transition matrix of the target domain is $\mathbf{T} \in \mathbb{R}^{N\times N}$ and that of the source domain is $\mathbf{S} \in \mathbb{R}^{N\times N}$. We sample sequences of length 20 from the target domain using the update rule $\pi_i = \mathbf{T}\,\pi_{i-1}$ with the initial state π_0 set to the uniform vector. Sampling from the source domain proceeds in the same way. Implementation details can be found in the appendix E.1.

n	Vanilla (Target-only)	Finetuned	GTL
1000	0.0476 ± 0.0064	0.0393 ± 0.0029	$\mathbf{0.0377} \pm 0.0080$
100	0.1938 ± 0.0173	0.1118 ± 0.0271	$\mathbf{0.0989} \pm 0.0180$
20	0.5842 ± 0.0244	0.4004 ± 0.0983	$\mathbf{0.3621} \pm 0.0478$

Table 2: Comparison of baseline methods and best guided results (mean $\pm$ standard error; 3 seeds) across target sample sizes for $(\text{diag}_{\text{src}} = 0.1, \text{diag}_{\text{tgt}} = 0.8)$. Reported values are the average Kullback–Leibler divergence between true and estimated transition matrices; lower is better.

Results We consider $m = 10000$ source samples and $n \in \{1000, 100, 20\}$ target samples. We estimate the transition matrix by sampling 4096 sequences, counting transitions between states, and normalizing by the total count. Figure 2 presents an example where $\mathbf{S}$ has diagonal entries 0.1 and $\mathbf{T}$ has diagonal entries 0.8. We observe that the vanilla target diffusion model and the finetuned model both degrade in performance when fewer target samples are available. Table 2 provides a direct comparison, obtained by running this example three times with different random seeds. We observe the following behaviors: (i) the diffusion model trained directly on the target dataset loses performance as the number of training samples decreases, and (ii) our method, compared to traditional finetuning, achieves better results even with fewer trainable parameters, while showing a similar trend in performance across different numbers of training samples.

4.2 Language Modeling

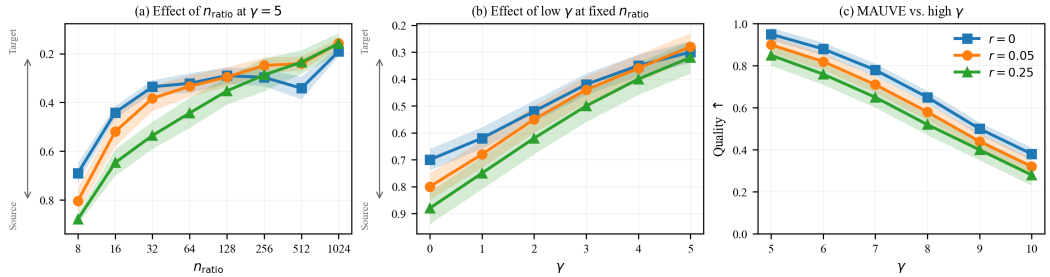

Figure 3: Sensitivity of GTL to the guidance weight γ and the candidate budget n_{ratio} on Physics validation data. The source domain is $\mathcal{S} = \text{CS} \cup \text{Math} \cup \mathcal{P}_r$ and the target domain is $\mathcal{T} = \mathcal{P} \setminus \mathcal{P}_r$, where $\mathcal{P}_r$ is a random r-subset of Physics. We use $r \in \{0, 0.05, 0.25\}$, where r denotes the fraction of Physics included in the source. For each setting, we train three independent source models and three independent ratio models. Lines show means and shaded bands show $\pm$ standard error over 3×3 train$\times$sample seeds. Increasing n_{ratio} improves target alignment, while overly strong guidance reduces quality most strongly when $r = 0$.

In this section, we apply our method to text data. We use the arXiv abstracts dataset (arXiv.org submitters, 2024), which contains 1.7 million arXiv articles with titles, authors, categories, and abstracts. We focus on Computer Science, Mathematics, and Physics. For data preparation, we concatenate all text and split it into segments of 512 tokens using the `bert-base-uncased` tokenizer with vocabulary size $N = 30{,}522$. This results in 285,946 samples in Computer Science, 176,831 in Mathematics, and 79,631 in Physics. For each category we randomly select 10% of the samples for validation. We ensure that the three category subsets are disjoint.

We run two types of experiments. First, we study the behavior of the two key parameters γ and n_{ratio}. Second, we compare three settings on the target domain. The first is a vanilla discrete diffusion model trained on the target data. The second is a finetuned version of the source model. The third is our guided transfer method.

Implementation and Evaluations We train the source diffusion model for 100,000 gradient steps using the DiT architecture (Peebles & Xie, 2023) with 59.8M trainable parameters. A log–linear noise scheduler is used for both training and sampling. The finetuned model on the target domain is trained for 10,000 gradient steps. For our guided method, we initialize a ratio model with 4.1M parameters and train it for 5,000 gradient steps. Additional training details appear in the supple-

mentary materials. For each method we generate 128 sequences of length 512. We evaluate quality with MAUVE (Pillutla et al., 2021), which captures the tradeoff between perplexity and diversity. Although generative perplexity (Gen. PPL) is often reported, we observed, consistent with prior work (Wang et al., 2025), that Gen. PPL can remain low even when the text has reduced utility, such as repeated words. In those cases, MAUVE is more sensitive and provides a clearer measure of quality. We also evaluate the domain of each sampled sequence using an independently trained classifier. Scores near zero indicate the target domain and scores near one indicate the source domain. The classifier follows the setup used for the ratio model in the supplementary materials with label smoothing set to 0.05.

4.2.1 BEHAVIOR OF γ AND n_{RATIO}

We study the sensitivity of GTL to the guidance weight γ and the candidate budget n_{ratio}. For each configuration, we train three independent source models and three ratio models using different random seeds. The source domain is defined as Computer Science $\cup$ Mathematics plus an r-fraction of Physics abstracts, and the target domain is the remaining $(1-r)$ fraction of Physics. Formally, for $r \in [0,1]$, let $\mathcal{P}_r$ be a random r-subset of Physics (sampled without replacement). We set $\mathcal{S} = \text{CS} \cup \text{Math} \cup \mathcal{P}_r$ and $\mathcal{T} = \mathcal{P} \setminus \mathcal{P}_r$. We set $r \in \{0, 0.05, 0.25\}$, thus in the case of $r = 0$ we have two distinct datasets. This setup simulates the common scenario where pretraining large corpora already includes a small portion of target-domain text. In Figure 3, the behavior of these parameters is illustrated. Intuitively, smaller candidate budgets n_{ratio} introduce higher sampling error, whereas increasing n_{ratio} monotonically reduces the domain-classifier score, yielding more target-like samples, as shown in panel (a). In Figure 3(b), we observe that the guidance score is nontrivial: a moderate $\gamma \approx 3-5$ consistently pushes the representations toward the target across all r. Panel (c) highlights two key effects: (i) excessive guidance can degrade performance when there is no overlap between source and target data, and (ii) when a small portion of target text is included in the source domain ($r = 0.05, 0.25$), MAUVE remains high for larger γ values and decays more smoothly. This indicates that a higher r improves robustness. Overall, the results suggest that $n_{\text{ratio}} \approx 256-512$ and $\gamma \approx 4-6$ achieve a favorable balance, providing strong target alignment with minimal quality loss.

4.2.2 METHOD COMPARISON AND SENSITIVITY TO DATA FRACTION

In this section, we define the union of Computer Science and Mathematics abstracts as the source domain and Physics abstracts as the target domain. We compare three methods: a fine-tuned discrete diffusion model trained on the target domain after pretraining on the source domain, a vanilla discrete diffusion model trained directly on the target domain, and our Guided Transfer Learning (GTL) method with the ratio estimator trained across target-data fractions of 100%, 30%, 5%, and 1%. Each method is trained with three random seeds, and sampling is performed with three seeds as well (nine runs per cell). Furthermore, each seed corresponds to a distinct subset of the target data. Figure 1 reports the behavior of the models across different fractions of the target training data. The observed trends align with those in Table 2. In the smallest case, we only use 796 samples of Physics abstracts. For finetuned and vanilla discrete diffusion, the number of epochs over the training set increases up to 400 when the fraction is smallest. The ratio model decreases its MAUVE score with decreasing fractions, but less dramatically than the other methods. Thus, our method demonstrates greater stability under limited target-domain data. In contrast, the finetuned and vanilla discrete diffusion models show a stepwise drop in performance as the fraction decreases. Even in the full-target-data setting, GTL outperforms the baselines because there is still a large imbalance between the number of source-domain and target-domain samples. The GTL remains relatively stable, except in the 1% case, where we also observe a sharp decline.

5 CONCLUSION

Adapting discrete diffusion models to target domains with limited data is often costly. To address this, we propose a guided transfer learning approach for discrete diffusion models that introduces a small, learned ratio network, allowing the training of significantly fewer parameters. Furthermore, we present a guidance-based sampling algorithm that enables generation with larger vocabularies and longer sequences, making guided language modeling with discrete diffusion models feasible.

We demonstrate the effectiveness of our method on synthetic Markov chains and arXiv abstracts (language modeling), particularly when the target dataset is small relative to the source.

Limitations Overall, the proposed method is promising and applicable to a wide range of tasks involving domain shifts. However, it requires access to both source and target domains. Future work could explore conditional transfer for prompt-based or attribute-controlled generation. It would also be interesting to evaluate the learned ratio on quality-filtered tasks, such as reducing toxicity in language modeling.

REFERENCES

arXiv.org submitters. arxiv dataset, 2024. URL https://www.kaggle.com/dsv/7548853.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023. URL https://arxiv.org/abs/2107.03006.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models, 2022. URL https://arxiv.org/abs/2205.14987.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards, 2024. URL https://arxiv.org/abs/2309.17400.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2305.16381.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models, 2025. URL https://arxiv.org/abs/2410.17891.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. URL https://arxiv.org/abs/2204.03458.

Chihan Huang and Hao Tang. Ctrldiff: Boosting large diffusion language models with dynamic block prediction and controllable generation, 2025. URL https://arxiv.org/abs/2505.14455.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation, 2022a.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation, 2022b. URL https://arxiv.org/abs/2205.14217.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. Audioldm: Text-to-audio generation with latent diffusion models, 2023. URL https://arxiv.org/abs/2301.12503.

Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising, 2025. URL https://arxiv.org/abs/2410.06264.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution, 2024. URL https://arxiv.org/abs/2310.16834.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q. Weinberger. Latent diffusion for language generation, 2023. URL https://arxiv.org/abs/2212.09462.

Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data, 2023. URL https://arxiv.org/abs/2211.00802.

Taehong Moon, Moonseok Choi, Gayoung Lee, Jung-Woo Ha, and Juho Lee. Fine-tuning diffusion models with limited data. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022. URL https://openreview.net/forum?id=0J6afk9DqrR.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text, 2025. URL https://arxiv.org/abs/2410.18514.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models, 2025. URL https://arxiv.org/abs/2406.01572.

Yidong Ouyang, Liyan Xie, Hongyuan Zha, and Guang Cheng. Transfer learning for diffusion models, 2024. URL https://arxiv.org/abs/2405.16876.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Avishek Joey Bose, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling, 2025. URL https://arxiv.org/abs/2502.03540.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers, 2021. URL https://arxiv.org/abs/2102.01454.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models, 2024. URL https://arxiv.org/abs/2406.07524.

Anirban Sarkar, Ziqi Tang, Chris Zhao, and Peter K Koo. Designing dna with tunable regulatory activity using discrete diffusion. *bioRxiv*, 2024. doi: 10.1101/2024.05.23.595630. URL https://www.biorxiv.org/content/early/2024/05/24/2024.05.23.595630.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models, 2025. URL https://arxiv.org/abs/2412.10193.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data, 2025. URL https://arxiv.org/abs/2406.04329.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models, 2023. URL https://arxiv.org/abs/2211.16750.

Jaesung Tae, Hamish Ivison, Sachin Kumar, and Arman Cohan. Tess 2: A large-scale generalist diffusion language model, 2025. URL https://arxiv.org/abs/2502.13917.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling, 2025. URL https://arxiv.org/abs/2503.00307.

Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. Patch diffusion: Faster and more data-efficient training of diffusion models, 2023. URL https://arxiv.org/abs/2304.12526.

Enze Xie, Lewei Yao, Han Shi, Zhili Liu, Daquan Zhou, Zhaoqiang Liu, Jiawei Li, and Zhenguo Li. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4207–4216, 2023a. URL https://api.semanticscholar.org/CorpusID:258108246.

Enze Xie, Lewei Yao, Han Shi, Zhili Liu, Daquan Zhou, Zhaoqiang Liu, Jiawei Li, and Zhenguo Li. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning, 2023b. URL https://arxiv.org/abs/2304.06648.

Jianhao Xie, Ziang Zhang, Zhenyu Weng, Yuesheng Zhu, and Guibo Luo. Meddiff-ft: Data-efficient diffusion model fine-tuning with structural guidance for controllable medical image synthesis, 2025. URL https://arxiv.org/abs/2507.00377.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation, 2025. URL https://arxiv.org/abs/2410.21357.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning, 2025. URL https://arxiv.org/abs/2410.14157.

Runpeng Yu, Qi Li, and Xinchao Wang. Discrete diffusion in large language and multimodal models: A survey, 2025. URL https://arxiv.org/abs/2506.13759.

Zhaoyu Zhang, Yang Hua, Guanxiong Sun, Hui Wang, and Seán McLoone. Training diffusion-based generative models with limited data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=NDH7AzljFD.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling, 2025. URL https://arxiv.org/abs/2409.02908.

Jincheng Zhong, Xiangcheng Zhang, Jianmin Wang, and Mingsheng Long. Domain guidance: A simple transfer approach for a pre-trained diffusion model, 2025. URL https://arxiv.org/abs/2504.01521.

A Appendix

B Related Work

Discrete Diffusion Models Discrete diffusion models (DDMs) have become a standard choice for generative modeling over categorical sequences, including biological sequences (DNA, RNA, proteins) and text (Sarkar et al., 2024; Yu et al., 2025). Compared to approaches that add continuous noise to embeddings or latents, fully discrete formulations avoid the projection step back to tokens, which can introduce an embedding-to-token mismatch (Li et al., 2022b; Lovelace et al., 2023). Two families of discrete diffusion methods are especially common. The first is discrete-time diffusion in the D3PM framework, where a Markov corruption process is defined by transition matrices and a denoiser is trained to invert the forward kernels; simplified masked variants such as MLDM and MD4 fall into this category (Sahoo et al., 2024; Shi et al., 2025; Austin et al., 2023). The second is continuous-time discrete diffusion, which parameterizes a discrete score (a log-ratio between neighboring-time marginals) and induces a continuous-time Markov chain (CTMC) for generation (Lou et al., 2024; Meng et al., 2023; Campbell et al., 2022).

Guided Diffusion Models Guidance is a widely used mechanism for steering diffusion sampling in both continuous and discrete domains. In discrete diffusion, classifier-free guidance typically combines conditional and unconditional predictions at inference time, requiring two forward passes of a single model trained on a mixture of conditional and unconditional examples (Huang & Tang, 2025). Classifier-based guidance is closer to our setting: it introduces an auxiliary classifier $p(y \mid x_t, t)$ and reweights the reverse transition so that samples are biased toward the desired attribute y, while the denoiser itself is trained on the full corpus (Schiff et al., 2025; Nisonoff et al.,

2025). A direct implementation scores token substitutions across all positions and vocabulary items, requiring $\mathcal{O}(L|\mathcal{V}|)$ guidance evaluations per step, which is infeasible for long sequences and large vocabularies. Because each evaluation corresponds to a distinct single-token intervention on the full sequence, this cost is not removed by straightforward parallelization. To reduce this cost, (Nisonoff et al., 2025) proposes a first-order Taylor approximation that replaces many evaluations with a forward and backward pass. Related alternatives adjust sampling using inference-time optimization: TESS2 (Tae et al., 2025) performs gradient ascent on a reward with respect to the model logits, and energy-based diffusion language models define an energy over predicted clean sequences and use it to reweight denoising updates to reduce train–sample mismatch (Xu et al., 2025).

Transfer Learning Fine-tuning and transfer learning share the goal of adapting a model to a new task or domain. Fine-tuning is computationally expensive because it updates the pretrained denoiser's weights on new data. Prior work on continuous diffusion models explores reinforcement learning (Fan et al., 2023; Clark et al., 2024) and adapter-based fine-tuning (Moon et al., 2022; Xie et al., 2023b). In contrast, transfer learning for diffusion remains comparatively underexplored. *Transfer Learning for Diffusion Models* (TLDM) (Ouyang et al. (2024)) proposes a ratio-estimation technique for continuous score-based models,but it relies on continuous scores and is evaluated in low-dimensional settings. Our work closes the gap for discrete diffusion by deriving ratio-guided reverse transitions in both discrete-time, continuous-time and score based formulations for high dimansion application.

C Proof of Theorem 1

For the proof, we use the formulation of importance reweighting as described in the following lemma.

Lemma 2 (Importance-weighted reformulation) *Fix a timestep i and define $r(\mathbf{x}_0) = q(\mathbf{x}_0)/p(\mathbf{x}_0)$. If the source and target share the same forward process, then the loss function can be rewritten as:*

$$\mathcal{L}_i(\psi) = \mathbb{E}_q\big[D_{\mathrm{KL}}\big(q(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big)\big] \tag{8}$$

$$= \mathbb{E}_{\mathbf{x}_0\sim p\; z_{t(i)}\sim p(\mathbf{z}_{t(i)}|\mathbf{x}_0)}\Big[r(\mathbf{x}_0) D_{\mathrm{KL}}\big(p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big)\Big]. \tag{9}$$

Proof of Lemma 2

$$\mathbb{E}_q[\mathcal{L}_{\text{diffusion}}] = \sum_{i=1}^{T} \mathbb{E}_q\Big[D_{\mathrm{KL}}\big(q(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\big\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big)\Big] \tag{10}$$

$$= \sum_{i=1}^{T} \sum_{\mathbf{x}_0} \sum_{\mathbf{z}_{t(i)}} q(\mathbf{x}_0, \mathbf{z}_{t(i)}) D_{\mathrm{KL}}\big(q(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\big\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big) \tag{11}$$

$$= \sum_{i=1}^{T} \sum_{\mathbf{x}_0} \sum_{\mathbf{z}_{t(i)}} q(\mathbf{x}_0)\, q(\mathbf{z}_{t(i)} \mid \mathbf{x}_0) D_{\mathrm{KL}}\big(q(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\big\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big) \tag{12}$$

$$= \sum_{i=1}^{T} \sum_{\mathbf{x}_0} \sum_{\mathbf{z}_{t(i)}} \frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)} p(\mathbf{x}_0)\, p(\mathbf{z}_{t(i)} \mid \mathbf{x}_0) D_{\mathrm{KL}}\big(q(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\big\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big) \tag{13}$$

$$= \sum_{i=1}^{T} \mathbb{E}_{\mathbf{x}_0\sim p}\, \mathbb{E}_{\mathbf{z}_{t(i)}\sim p(\mathbf{z}_{t(i)}|\mathbf{x}_0)}\Big[\frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)} D_{\mathrm{KL}}\big(p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big)\Big] \tag{14}$$

To conclude Lemma 2, we leverage the fact that the source and target domain share the same forward kernel in Equation 13 to 14 $q(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) = \frac{q(\mathbf{z}_{t(i)}|\mathbf{z}_{s(i)})\cdot q(\mathbf{z}_{s(i)}|\mathbf{x}_0)}{q(\mathbf{z}_{t(i)}|\mathbf{x}_0)} = p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0)$.

Now, we use Lemma 2:

$$\mathcal{L}_i(\psi) = \mathbb{E}_{\mathbf{x}_0 \sim p}\, \mathbb{E}_{\mathbf{z}_{t(i)} \sim p(\mathbf{z}_{t(i)}|\mathbf{x}_0)} \Big[\underbrace{\frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)}}_{r(\mathbf{x}_0)} D_{\mathrm{KL}}\big(p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big) \Big] \tag{15}$$

$$= \sum_{\mathbf{x}_0} \sum_{\mathbf{z}_{t(i)}} \underbrace{p(\mathbf{z}_{t(i)} \mid \mathbf{x}_0)\, p(\mathbf{x}_0)}_{p(\mathbf{x}_0|\mathbf{z}_{t(i)})\, p(\mathbf{z}_{t(i)})} r(\mathbf{x}_0) D_{\mathrm{KL}}\big(p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big) \tag{16}$$

$$= \sum_{\mathbf{z}_{t(i)}} p(\mathbf{z}_{t(i)}) \sum_{\mathbf{x}_0} \underbrace{p(\mathbf{x}_0 \mid \mathbf{z}_{t(i)}) r(\mathbf{x}_0)}_{\alpha(\mathbf{x}_0, \mathbf{z}_{t(i)})} D_{\mathrm{KL}}\big(p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big) \tag{17}$$

$$= \sum_{\mathbf{z}_{t(i)}} p(\mathbf{z}_{t(i)}) \sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)})\; D_{\mathrm{KL}}\big(p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big) \tag{18}$$

Now, we take the derivative in regard to ψ:

$$0 = \frac{\partial}{\partial \psi} \mathcal{L}_i(\psi) \tag{19}$$

$$\implies \qquad 0 = \frac{\partial}{\partial \psi} \sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)})\; D_{\mathrm{KL}}\big(p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \,\|\, q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big) \tag{20}$$

To solve this, we set up a Lagrangian equation by using the constraint $\sum_{\mathbf{z}_{s(i)}} q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) = 1$

$$\begin{aligned} \mathcal{L}(\psi, \lambda) \;=\; & \sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)}) \sum_{\mathbf{z}_{s(i)}} p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \big[\log p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) - \log q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})\big] \\ & + \lambda \Big(\sum_{\mathbf{z}_{s(i)}} q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) - 1 \Big) \end{aligned} \tag{21}$$

Stationary condition for each $\mathbf{z}_{s(i)}$:

$$\frac{\partial \mathcal{L}}{\partial \psi} = -\frac{\sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)})\, p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0)}{q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)})} \;+\; \lambda = 0 \tag{22}$$

$$\implies \quad q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) = \frac{\sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)})\, p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0)}{\lambda} \tag{23}$$

Now we determine λ from the constraint

$$1 = \sum_{\mathbf{z}_{s(i)}} q_\psi(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) = \sum_{\mathbf{z}_{s(i)}} \frac{1}{\lambda} \sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)})\, p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \tag{24}$$

$$\iff \qquad \lambda = \sum_{\mathbf{z}_{s(i)}} \sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)})\, p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \tag{25}$$

Putting all together:

$$q_{\psi^\star}(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) = \frac{\sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)})\, p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0)}{\sum_{\tilde{\mathbf{z}}_{s(i)}} \sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)}) p(\tilde{\mathbf{z}}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0)} \tag{26}$$

$$= \frac{p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) \frac{q(\mathbf{z}_{s(i)})}{p(\mathbf{z}_{s(i)})}}{\sum_{\tilde{\mathbf{z}}_{s(i)}} p(\tilde{\mathbf{z}}_{s(i)} \mid \mathbf{z}_{t(i)}) \frac{q(\tilde{\mathbf{z}}_{s(i)})}{p(\tilde{\mathbf{z}}_{s(i)})}} \tag{27}$$

$$= \frac{p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) \mathbb{E}_{\mathbf{x}_0 \sim p(\cdot|\mathbf{z}_{s(i)})} \left[\frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)} \right]}{\sum_{\tilde{\mathbf{z}}_{s(i)}} p(\tilde{\mathbf{z}}_{s(i)} \mid \mathbf{z}_{t(i)}) \mathbb{E}_{\mathbf{x}_0 \sim p(\cdot|\tilde{\mathbf{z}}_{s(i)})} \left[\frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)} \right]} \tag{28}$$

For 26 to 27 we show that $\sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)})\, p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) = p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}) \frac{q(\mathbf{z}_{s(i)})}{p(\mathbf{z}_{s(i)})}$:

$$\sum_{\mathbf{x}_0} \alpha(\mathbf{x}_0, \mathbf{z}_{t(i)})\, p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) = \sum_{\mathbf{x}_0} p(\mathbf{x}_0 \mid \mathbf{z}_{t(i)})\, \frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)}\, p(\mathbf{z}_{s(i)} \mid \mathbf{z}_{t(i)}, \mathbf{x}_0) \tag{29}$$

$$= \sum_{\mathbf{x}_0} p(\mathbf{z}_{t(i)} \mid \mathbf{x}_0)\, \frac{p(\mathbf{x}_0)}{p(\mathbf{z}_{t(i)})}\, \frac{q(\mathbf{x}_0)}{p(\mathbf{x}_0)}\, \frac{p(\mathbf{z}_{t(i)} \mid \mathbf{z}_{s(i)}) \cdot p(\mathbf{z}_{s(i)} \mid \mathbf{x}_0)}{p(\mathbf{z}_{t(i)} \mid \mathbf{x}_0)} \tag{30}$$

$$= \frac{p(\mathbf{z}_{t(i)} \mid \mathbf{z}_{s(i)})}{p(\mathbf{z}_{t(i)})} \sum_{\mathbf{x}_0} q(\mathbf{x}_0) p(\mathbf{z}_{s(i)} \mid \mathbf{x}_0) = p(\mathbf{z}_{t(i)} \mid \mathbf{z}_{s(i)}) \frac{q(\mathbf{z}_{s(i)})}{p(\mathbf{z}_{t(i)})} \tag{31}$$

For 27 to 28 we show that $\frac{q(\mathbf{z}_{s(i)})}{p(\mathbf{z}_{s(i)})} = \mathbb{E}_{\mathbf{x}_0 \sim p(\cdot \mid \mathbf{z}_{s(i)})}\big[q(\mathbf{x}_0)/p(\mathbf{x}_0)\big]$:

$$\frac{q(\mathbf{z}_{s(i)})}{p(\mathbf{z}_{s(i)})} = \frac{\sum_{\mathbf{x}_0} q(\mathbf{z}_{s(i)} \mid \mathbf{x}_0)\, q(\mathbf{x}_0)}{p(\mathbf{z}_{s(i)})} \tag{32}$$

$$= \frac{\sum_{\mathbf{x}_0} q(\mathbf{z}_{s(i)} \mid \mathbf{x}_0)\, p(\mathbf{x}_0)\; r(\mathbf{x}_0)}{p(\mathbf{z}_{s(i)})} = \sum_{\mathbf{x}_0} \frac{p(\mathbf{z}_{s(i)} \mid \mathbf{x}_0)\, p(\mathbf{x}_0)}{p(\mathbf{z}_{s(i)})}\; r(\mathbf{x}_0) \tag{33}$$

$$= \sum_{\mathbf{x}_0} p(\mathbf{x}_0 \mid \mathbf{z}_{s(i)})\; r(\mathbf{x}_0) = \mathbb{E}_{\mathbf{x}_0 \sim p(\cdot \mid \mathbf{z}_{s(i)})}\big[q(\mathbf{x}_0)/p(\mathbf{x}_0)\big] \tag{34}$$

D GUIDED τ-SAMPLING

Algorithm 2 NAIVE GUIDED DENOISE STEP $(\mathbf{z}_t,\ t,\ \Delta t)$

Require: noisy sequence $\mathbf{z}_t \in \mathcal{V}^L$, time t, step $\Delta t > 0$, denoiser p_θ, ratio net r_ϕ, mask index m, guidance schedule $\gamma(\cdot)$, noise schedule $\sigma(\cdot)$
Ensure: updated sequence $\mathbf{z}_s$

```
σ_t ← σ(t)
log x_θ ← p_θ(z_t, σ_t) {L × |V| logits}
E ← {ℓ ∈ {1, ..., L} : z_t^(ℓ) = m} {masked positions}
z_s ← z_t
for all ℓ ∈ E do
    for all v ∈ V do
        z̃ ← z_t
        z̃^(ℓ) ← v
        log r[ℓ, v] ← log r_φ(z̃, σ_t)
    end for
    log q^guided[ℓ, :] ← log x_θ[ℓ, :] + γ(t) log r[ℓ, :]
    π^(ℓ) ← softmax(log q^guided[ℓ, :])
    z_s^(ℓ) ~ Cat(π^(ℓ))
end for

return z_s
```

In the experiments, we observe that the combination of top-n_{ratio} selection and the guidance weight γ can misallocate probability mass, potentially causing the diffusion process to collapse into a fully masked sequence at the final step. For instance, when the top-n_{ratio} is small, the denoiser tends to propose tokens that are highly likely to belong to the source domain, which then receive negative logits from the guidance ratio network. This effect is especially pronounced in high-noise regions, where all tokens experience a reduction in their probability of being unmasked, though some are affected more than others. Consequently, the overall probability of unmasking decreases, while the probability of remaining masked is incorrectly increased. This leads to the issue of ending up

with a higher number of masked positions in the final step. To mitigate this problem, we apply a position-independent normalization:

$$q^{\gamma}_{\theta,\phi}(\mathbf{z}_s = \mathbf{m} \mid \mathbf{z}_t) = p(\mathbf{z}_s = \mathbf{m} \mid \mathbf{z}_t) = p_{\mathbf{m}}, \qquad \sum_{\mathbf{v}\in\mathcal{V}\setminus\{\mathbf{m}\}} q^{\gamma}_{\theta,\phi}(\mathbf{z}_s = \mathbf{v} \mid \mathbf{z}_t) = 1 - p_{\mathbf{m}},$$

which implies the renormalized update over non-mask tokens, thus stabilizing the denoising diffusion process.

$$q^{\gamma}_{\theta,\phi}(\mathbf{z}_s = \mathbf{v} \mid \mathbf{z}_t) = \frac{(1-p_{\mathbf{m}})\, p_\theta(\mathbf{z}_s = \mathbf{v} \mid \mathbf{z}_t)\, r_\phi(\mathbf{z}_s = \mathbf{v})^{\gamma}}{\sum_{\tilde{\mathbf{v}}\in\mathcal{V}\setminus\{\mathbf{m}\}} p_\theta(\mathbf{z}_s = \tilde{\mathbf{v}} \mid \mathbf{z}_t)\, r_\phi(\mathbf{z}_s = \tilde{\mathbf{v}})^{\gamma}} \quad \text{for all } \mathbf{v} \in \mathcal{V}\setminus\{\mathbf{m}\}.$$

We use this algorithm in the first experiments section.

E Training Details

The Code will be released upon acceptance. In this section, we briefly describe the training setup for reproducibility. All training networks share the same learning schedule, which includes cosine decay with warmup and a high learning rate of 3×10^{-4}. The Adam optimizer is used, and we apply a dropout rate of 0.1. The batch size is set to 256 across all training algorithms. The time-independent and time-dependent Classifier is trained with 0.1 label smoothing and 4000 gradient steps. Importantly, the classifier is trained with binary labels, where the target domain is labeled as zero and the source domain as one. The ratio network is trained with λ set to 0.1, as outlined in the following pseudo-algorithm 3.

Algorithm 3 Training loop for the ratio network $r_\phi(x_t, t)$

Require: source data $p(x)$ and target data $q(x)$, frozen classifier $d_\omega : \mathcal{V}^L \to [0,1]$,
frozen time-dependent classifier $d_\omega : \mathcal{V}^L, t \to [0,1]$ forward kernel $\textsc{Corrupt}(x_0, \sigma_t)$ with schedule $\sigma(t)$,
batch size b, learning rate η, regularizer rate λ

repeat
  Sample mini-batch x_0 of size b **from** $\mathbf{p}$ and **draw** $t \sim \text{Uniform}(0,1)$
  $x_t \leftarrow \textsc{Corrupt}(x_0, \sigma(t))$ {forward diffusion}
  $\mathcal{L}_{\text{guidance}} = \frac{1}{b} \sum \| r_\phi(x_t, t) - \frac{1 - d_\omega(x_0)}{d_\omega(x_0)} \|_2^2$ {Guidance loss}

  Sample mini-batch x_0 of size b **from** $\mathbf{q}$ and **draw** $t \sim \text{Uniform}(0,1)$
  $x_t \leftarrow \textsc{Corrupt}(x_0, \sigma(t))$ {forward diffusion}
  $\mathcal{L}_{\text{cycle}} = \frac{1}{b} \sum \| r_\phi(x_t, t) - \frac{1 - d_\omega(x_t, t)}{d_\omega(x_t, t)} \|_2^2$ {Guidance loss}

  $\mathcal{L} = \mathcal{L}_{\text{guidance}} + \lambda \cdot \mathcal{L}_{\text{cycle}}$
  $\psi \leftarrow \psi - \eta \nabla_\psi \mathcal{L}$ {gradient update}
until convergence

return trained parameters ψ

For sample corruption, we use a log-linear schedule with $\sigma_{\text{max}} = 20$ and $\sigma_{\text{min}} = 0.0001$. The planner is trained following Algorithm 4, using the same training configuration as described above, except that it is trained for 100k update steps. On the held-out validation set from the source domain, the planner achieves an average accuracy of 70%. In high-noise regions, it becomes considerably more challenging for the planner to predict which positions will be correctly denoised, due to the high variability in token unmasking.

E.1 Implementation on Markov Chain

We use the sampling process of (Sahoo et al., 2024) for the source, finetuned, and target discrete diffusion models. The backbone is a Diffusion Transformer (DiT) (Peebles & Xie, 2023) with about

Algorithm 4 Training loop for the planner $\rho_\vartheta(x_t, t)$

Require: training corpus $\mathcal{D}$, frozen source denoiser $x_\theta : \mathcal{V}^L \to \mathbb{R}^{L\times|\mathcal{V}|}$ (per-position logits/probs), forward kernel $\textsc{Corrupt}(x_0, \sigma(t))$ with schedule $\sigma(t)$, batch size b, learning rate η

repeat
  Sample mini-batch x_0 of size b from $\mathcal{D}$; draw $t \sim \mathrm{Uniform}(0, 1)$
  $x_t \leftarrow \textsc{Corrupt}(x_0, \sigma(t))$ {(1) corrupt the sample}
  $\log x_\theta \leftarrow x_\theta(x_t)$; $\hat{x}_0 \leftarrow \arg\max \log x_\theta$ {(2) pass to source denoiser}
  $M \leftarrow (x_t = \texttt{[MASK]})$; $y \leftarrow \mathbf{1}[\hat{x}_0 = x_0]$ {(3) where denoiser is correct (masked positions)}
  $s \leftarrow \rho_\vartheta(x_t)$ {(4) planner logits per position}
  $\mathcal{L} \leftarrow \mathrm{BCEWithLogits}(s[M],\, y[M])$ {(5) loss over masked positions only}
  $\vartheta \leftarrow \vartheta - \eta\nabla_\vartheta\mathcal{L}$ {gradient update}
until convergence

return trained planner parameters ϑ

five million learnable parameters. The ratio estimator and the classifier use about half as many parameters with a final mean and fully connected layer that maps the output to a scalar. The exact implementation appears in our codebase. For guided sampling, we follow the ancestral sampling method. Since the vocabulary is small we set $n_{\text{ratio}} = N$ and use 20 sampling steps. We train a masked diffusion model on the source domain for 30 epochs. We then compare a vanilla masked diffusion model trained directly on the target samples (60 epochs), a finetuned diffusion model (90 epochs), and our GTL method (60 epochs). The finetuned model and our method share the same source diffusion model.

F Application to continuous-time Score-based Models

F.1 Background

Campbell et al. (2022) introduces a continuous-time framework for discrete denoising models formulated as a continuous-time Markov chain (CTMC). Unlike the discrete-time setting of (Austin et al., 2023) with steps $t = 0, 1, \ldots, T$, here $t \in [0, T]$ is continuous.

In this section, we switch notation. The forward kernel is $p_{t|s}(\mathbf{y}' \mid \mathbf{y}) := p(\mathbf{z}_t = \mathbf{y}' \mid \mathbf{z}_s = \mathbf{y})$, the backward kernel is $p_{s|t}(\mathbf{y}' \mid \mathbf{y}) := p(\mathbf{z}_s = \mathbf{y}' \mid \mathbf{z}_t = \mathbf{y})$, the noising kernel is $p_t(\mathbf{y} \mid \mathbf{x}) := p(\mathbf{z}_t = \mathbf{y} \mid \mathbf{x}_0 = \mathbf{x})$, and the true backward transition is $p(\mathbf{y}' \mid \mathbf{y}, \mathbf{x}) := p(\mathbf{z}_s = \mathbf{y}' \mid \mathbf{z}_t = \mathbf{y}, \mathbf{x}_0 = \mathbf{x})$. This CTMC formulation allows transitions at arbitrary times. The forward corruption is specified by a rate matrix $R_t \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, and the reverse process by a reverse rate matrix $\widetilde{R}_t$:

$$p_{t|s}(\mathbf{y}' \mid \mathbf{y}) = \delta_{\mathbf{y}',\mathbf{y}} + R_t(\mathbf{y}', \mathbf{y})\,\Delta t + \mathcal{O}\big(\Delta t^2\big)\,,$$
$$p_{s|t}(\mathbf{y}' \mid \mathbf{y}, \mathbf{x}) = \delta_{\mathbf{y}',\mathbf{y}} + \widetilde{R}_t(\mathbf{y}', \mathbf{y})\,\Delta t + \mathcal{O}\big(\Delta t^2\big)\,.$$

Here Δt denotes a small time increment when discretizing the continuous-time process, with $s = t - \Delta t$. Here, $\delta_{i,j}$ denotes the Kronecker delta (equal to 1 if $i = j$ and 0 otherwise), and $\mathcal{O}(\Delta t^2)$ denotes terms of order Δt^2 or higher. The reverse rate can be written in terms of the forward rate and the marginals $q_t(\cdot \mid \mathbf{x})$:

$$\widetilde{R}_t(\mathbf{y}', \mathbf{y}) = \begin{cases} \dfrac{q_t(\mathbf{y}' \mid \mathbf{x})}{q_t(\mathbf{y} \mid \mathbf{x})}\, R_t(\mathbf{y}, \mathbf{y}'), & \mathbf{y}' \neq \mathbf{y}, \\ -\displaystyle\sum_{\tilde{\mathbf{y}}\neq\mathbf{y}} \dfrac{q_t(\tilde{\mathbf{y}} \mid \mathbf{x})}{q_t(\mathbf{y} \mid \mathbf{x})}\, R_t(\mathbf{y}, \tilde{\mathbf{y}}), & \mathbf{y}' = \mathbf{y}. \end{cases}$$

Concrete score. We can express the reverse process in terms of the forward transition rates and a *concrete score* defined by $s_\theta(\mathbf{y})_{\mathbf{y}'} \approx \frac{q_t(\mathbf{y}'|\mathbf{x})}{q_t(\mathbf{y}|\mathbf{x})}$, parameterized by a neural network $s_\theta : \mathcal{V} \to \mathbb{R}^{|\mathcal{V}|}_{>0}$. This enables training via concrete score matching (CSM) (Meng et al., 2023), directly analogous to score-based models in continuous domains (Song et al., 2021). Lou et al. (2024) propose the

diffusion-weighted denoising score entropy (DWDSE) loss (their CDM-based objective) and note it can also be derived from the continuous-time likelihood framework of (Campbell et al., 2022); a detailed derivation appears in Appendix C.3 of (Sahoo et al., 2024). Concretely,

$$\lim_{T\to\infty} \mathbb{E}_{\substack{t\in\{\frac{1}{T},\frac{2}{T},\dots,1\}\\ \mathbf{y}\sim q_t(\cdot|\mathbf{x})}}\big[\mathrm{D}_{\mathrm{KL}}\big(p_{s|t}(\mathbf{y}' \mid \mathbf{y},\mathbf{x}) \,\|\, p_{s|t}(\mathbf{y}' \mid \mathbf{y})\big)\big] \tag{35}$$

$$= \mathbb{E}_{\substack{t\in[0,1]\\ \mathbf{y}\sim q_t(\cdot|\mathbf{x})}}\Big[\sum_{\mathbf{y}'\neq\mathbf{y}} R_t(\mathbf{y},\mathbf{y}')\Big(s_\theta(\mathbf{y})_{\mathbf{y}'} - \frac{q_t(\mathbf{y}' \mid \mathbf{x})}{q_t(\mathbf{y} \mid \mathbf{x})}\log s_\theta(\mathbf{y})_{\mathbf{y}'} + K\Big(\tfrac{q_t(\mathbf{y}'|\mathbf{x})}{q_t(\mathbf{y}|\mathbf{x})}\Big)\Big)\Big], \tag{36}$$

where $K(a) = a\log a - a$.

F.2 Theorem

In the previous section, we introduced the diffusion process and the source-domain objective on which the score function s_θ is trained. We now state a theorem that enables sampling from the target distribution *without* retraining s_θ under the objective in 36. Let $r_\phi(\mathbf{y})$ denote the learned density-ratio guidance term defined earlier.

Theorem 3 (Score-Based Ratio Transfer Learning) *Let $s_\theta(\mathbf{y})_{\mathbf{y}'}$ be a score-based model trained on the source distribution p, and let $\widetilde{R}_t^\theta(\cdot,\cdot)$ be the reverse-rate matrix induced by s_θ. Then, sampling from the target distribution q is given by the target reverse-rate matrix:*

$$\widetilde{R}_t^\psi(\mathbf{y}',\mathbf{y}) = \begin{cases} \dfrac{r_\phi(\mathbf{y}')}{r_\phi(\mathbf{y})}\,\widetilde{R}_t^\theta(\mathbf{y}',\mathbf{y}), & \mathbf{y}'\neq\mathbf{y},\\ -\displaystyle\sum_{\tilde{\mathbf{y}}\neq\mathbf{y}} \dfrac{r_\phi(\tilde{\mathbf{y}})}{r_\phi(\mathbf{y})}\,\widetilde{R}_t^\theta(\tilde{\mathbf{y}},\mathbf{y}), & \mathbf{y}'=\mathbf{y}. \end{cases} \tag{37}$$

F.3 Proof of Theorem 3

For the proof, we take the loss for a small step Δt with $s = t - \Delta t$ from equation 35 and apply Theorem 1, which yields the following expression:

$$q_{s|t}^{\psi^\star}(\mathbf{y}' \mid \mathbf{y}) = \frac{p_{s|t}(\mathbf{y}' \mid \mathbf{y})\,\mathbb{E}_{\mathbf{x}\sim p(\cdot|\mathbf{y}')}\big[\frac{q(\mathbf{x})}{p(\mathbf{x})}\big]}{\sum_{\widetilde{\mathbf{y}}} p_{s|t}(\widetilde{\mathbf{y}} \mid \mathbf{y})\,\mathbb{E}_{\mathbf{x}\sim p(\cdot|\widetilde{\mathbf{y}})}\big[\frac{q(\mathbf{x})}{p(\mathbf{x})}\big]} \tag{38}$$

$$= \frac{p_{s|t}(\mathbf{y}' \mid \mathbf{y})\,r_\phi(\mathbf{y}')}{\sum_{\widetilde{\mathbf{y}}} p_{s|t}(\widetilde{\mathbf{y}} \mid \mathbf{y})\,r_\phi(\widetilde{\mathbf{y}})} \tag{39}$$

We can already assume that the source backward kernel is approximated by $p_{s|t}(\mathbf{y}' \mid \mathbf{y}) = \delta_{\mathbf{y}',\mathbf{y}} + \widetilde{R}_t^\theta(\mathbf{y}',\mathbf{y})\,\Delta t + \mathcal{O}(\Delta t^2)$:

$$q_{s|t}^{\psi}(\mathbf{y}' \mid \mathbf{y}) = \frac{\big[\delta_{\mathbf{y}',\mathbf{y}} + \widetilde{R}_t^\theta(\mathbf{y}',\mathbf{y})\Delta t\big]\,r_\phi(\mathbf{y}')}{\sum_{\tilde{\mathbf{y}}}\big[\delta_{\tilde{\mathbf{y}},\mathbf{y}} + \widetilde{R}_t^\theta(\tilde{\mathbf{y}},\mathbf{y})\Delta t\big]\,r_\phi(\tilde{\mathbf{y}})} \;+\; \mathcal{O}(\Delta t^2)$$

We simplify the denominator by using the Taylor expansion of $1/(1+v)$ around $v=0$: $[\frac{1}{1+v} = 1 - v + \mathcal{O}(v^2)]$ and replacing $S := \sum_{\tilde{\mathbf{y}}} \widetilde{R}_t^\theta(\tilde{\mathbf{y}},\mathbf{y})\,r_\phi(\tilde{\mathbf{y}})$:

$$\frac{1}{\sum_{\tilde{\mathbf{y}}}\big[\delta_{\tilde{\mathbf{y}},\mathbf{y}} + \widetilde{R}_t^\theta(\tilde{\mathbf{y}},\mathbf{y})\Delta t\big]\,r_\phi(\tilde{\mathbf{y}})} = \frac{1}{r_\phi(\mathbf{y}) + \Delta t\,S} = \frac{1}{r_\phi(\mathbf{y})}\Big(1 + \frac{\Delta t\,S}{r_\phi(\mathbf{y})}\Big)^{-1} = \frac{1}{r_\phi(\mathbf{y})} - \frac{\Delta t\,S}{r_\phi(\mathbf{y})^2} + \mathcal{O}(\Delta t^2)$$

1. For the Off–diagonal case $\mathbf{y}' \neq \mathbf{y}$, hence $\delta_{\mathbf{y}',\mathbf{y}} = 0$:

$$\begin{aligned} q^{\psi}_{s|t}(\mathbf{y}' \mid \mathbf{y}) &= \frac{\left[\delta_{\mathbf{y}',\mathbf{y}} + \widetilde{R}^{\theta}_t(\mathbf{y}',\mathbf{y})\Delta t\right] r_\phi(\mathbf{y}')}{\sum_{\tilde{\mathbf{y}}}\left[\delta_{\tilde{\mathbf{y}},\mathbf{y}} + \widetilde{R}^{\theta}_t(\tilde{\mathbf{y}},\mathbf{y})\Delta t\right] r_\phi(\tilde{\mathbf{y}})} \;+\; \mathcal{O}(\Delta t^2) \\ &= \widetilde{R}^{\theta}_t(\mathbf{y}',\mathbf{y})\, r_\phi(\mathbf{y}')\, \Delta t \left(\frac{1}{r_\phi(\mathbf{y})} - \frac{\Delta t\, S}{r_\phi(\mathbf{y})^2} + \mathcal{O}(\Delta t^2)\right) \\ &= \frac{\widetilde{R}^{\theta}_t(\mathbf{y}',\mathbf{y})\, r_\phi(\mathbf{y}')}{r_\phi(\mathbf{y})}\, \Delta t \;+\; \mathcal{O}(\Delta t^2), \end{aligned}$$

2. For the diagonal case $\mathbf{y}' = \mathbf{y}$:

$$\begin{aligned} q^{\psi}_{s|t}(\mathbf{y} \mid \mathbf{y}) &= \frac{\left[1 + \widetilde{R}^{\theta}_t(\mathbf{y},\mathbf{y})\,\Delta t\right] r_\phi(\mathbf{y})}{r_\phi(\mathbf{y}) + \Delta t\, S} + \mathcal{O}(\Delta t^2) = \frac{1 + \widetilde{R}^{\theta}_t(\mathbf{y},\mathbf{y})\,\Delta t}{1 + \frac{\Delta t\, S}{r_\phi(\mathbf{y})}} + \mathcal{O}(\Delta t^2) \\ &= \Big(1 + \widetilde{R}^{\theta}_t(\mathbf{y},\mathbf{y})\,\Delta t\Big)\Big(1 - \frac{\Delta t\, S}{r_\phi(\mathbf{y})}\Big) + \mathcal{O}(\Delta t^2) = 1 + \Big[\widetilde{R}^{\theta}_t(\mathbf{y},\mathbf{y}) - \frac{S}{r_\phi(\mathbf{y})}\Big]\Delta t + \mathcal{O}(\Delta t^2). \end{aligned}$$

Putting these two cases together, we arrive at the following:

$$\begin{aligned} q^{\psi}_{s|t}(\mathbf{y}' \mid \mathbf{y}) &= \delta_{\mathbf{y}',\mathbf{y}}\Big[1 + \big(\widetilde{R}^{\theta}_t(\mathbf{y},\mathbf{y}) - \sum_{\tilde{\mathbf{y}}} \widetilde{R}^{\theta}_t(\tilde{\mathbf{y}},\mathbf{y})\, \tfrac{r_\phi(\tilde{\mathbf{y}})}{r_\phi(\mathbf{y})}\big)\Delta t\Big] + (1 - \delta_{\mathbf{y}',\mathbf{y}})\Big[\widetilde{R}^{\theta}_t(\mathbf{y}',\mathbf{y})\, \tfrac{r_\phi(\mathbf{y}')}{r_\phi(\mathbf{y})}\, \Delta t\Big] + \mathcal{O}(\Delta t^2) \\ &= \delta_{\mathbf{y}',\mathbf{y}} + \Delta t\Big[\widetilde{R}^{\theta}_t(\mathbf{y}',\mathbf{y})\, \frac{r_\phi(\mathbf{y}')}{r_\phi(\mathbf{y})} - \delta_{\mathbf{y}',\mathbf{y}} \sum_{\tilde{\mathbf{y}}} \widetilde{R}^{\theta}_t(\tilde{\mathbf{y}},\mathbf{y})\, \frac{r_\phi(\tilde{\mathbf{y}})}{r_\phi(\mathbf{y})}\Big] + \mathcal{O}(\Delta t^2), \\ &= \delta_{\mathbf{y}',\mathbf{y}} + \Delta t\, \widetilde{R}^{\psi}_t(\mathbf{y}',\mathbf{y}) \;+ \mathcal{O}(\Delta t^2). \end{aligned}$$

Thus concluding the proof.

$$\widetilde{R}^{\psi}_t(\mathbf{y}',\mathbf{y}) := \widetilde{R}^{\theta}_t(\mathbf{y}',\mathbf{y})\, \frac{r_\phi(\mathbf{y}')}{r_\phi(\mathbf{y})} - \delta_{\mathbf{y}',\mathbf{y}}\, \frac{1}{r_\phi(\mathbf{y})} \sum_{\tilde{\mathbf{y}}} \widetilde{R}^{\theta}_t(\tilde{\mathbf{y}},\mathbf{y})\, r_\phi(\tilde{\mathbf{y}}).$$

The work from (Nisonoff et al., 2025) in equation (2) arrives at a similar term just for classifier-based guidance.