Primal-Dual Guided Decoding for Constrained Discrete Diffusion

Federico Tomasi[*]
Spotify
federicot@spotify.com

Dmitrii Moor[*]
Spotify
dmitriim@spotify.com

Alice Wang
Spotify
alicew@spotify.com

Mounia Lalmas
Spotify
mounia@acm.org

Abstract

Discrete diffusion models generate structured sequences by progressively unmasking tokens, but enforcing global property constraints during generation remains an open challenge. We propose primal-dual guided decoding, an inference-time method that formulates constrained generation as a KL-regularised optimisation problem and solves it online via adaptive Lagrangian multipliers. At each denoising step, the method modifies token logits through an additive, constraint-dependent bias, with multipliers updated by mirror descent based on constraint violation. The bias arises as the optimal KL-regularised projection of the constraint, so the constrained distribution remains as close as possible to the model's unconstrained distribution while still satisfying the constraint. The method requires no retraining and no additional model evaluations beyond standard sampling, supports multiple simultaneous constraints, and provides formal bounds on constraint violation. We evaluate our approach on topical text generation, molecular design, and music playlist generation, showing that a single algorithm instantiated via domain-specific scoring functions improves constraint satisfaction while preserving relevant domain-specific quality metrics.

1 Introduction

Discrete diffusion models [Sahoo et al., 2024, Austin et al., 2021] have recently emerged as a powerful alternative to autoregressive (AR) models for generating structured discrete sequences. MDLM [Sahoo et al., 2024] progressively unmasks tokens from a fully masked state, enabling bidirectional context during generation. The paradigm applies to domains where outputs are discrete token sequences: natural language [Sahoo et al., 2024, Lou et al., 2024], molecular strings [Irwin et al., 2020], music playlists [Rajput et al., 2023], and protein sequences [Alamdari et al., 2023, Wang et al., 2024].

However, practical applications require generated outputs to satisfy property constraints (e.g., topical vocabulary in text, target molecular weight in molecule generation, or target genre composition in music playlists) while remaining close to the learned data distribution. This creates a fundamental tension between constraint satisfaction and distributional fidelity. Existing approaches address this trade-off only partially: predictor- or classifier-based guidance [Nisonoff et al., 2025, Schiff et al., 2025] requires training an auxiliary model per property, and constrained discrete diffusion [Cardei et al., 2025] relies on inner optimisation loops with 5–20$\times$ computational overhead.

[*]These authors contributed equally to this work.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

We propose **primal-dual guided decoding** for discrete diffusion, an inference-time method that formulates constrained generation as a KL-regularised optimisation problem and solves it online during denoising. The method is based on the stochastic primal-dual approximation technique, where tokens are revealed out of order and decisions are made under evolving context. This enables constraint enforcement to be integrated directly into the sampling process, without requiring additional trained models or modifying the diffusion model itself.

At each denoising step, we modify token log-probabilities via an additive, constraint-dependent bias derived from adaptive Lagrangian multipliers, updated by mirror descent based on constraint violation. The bias is the optimal KL-regularised projection of the constraint at each step, so the resulting sampling distribution stays as close as possible to the model's unconstrained distribution while satisfying the constraint. In contrast to approaches that rely on auxiliary predictors or inner optimisation loops, this yields a lightweight procedure that retains the computational profile of standard sampling.

We evaluate the approach on three domains: topical text steering, molecular generation, and playlist generation. A single algorithm, instantiated with domain-specific scoring functions, improves constraint satisfaction across all three while preserving the relevant per-domain quality metrics. The KL cost of constraint enforcement is exposed as a single, controllable knob: practitioners can trade fidelity for satisfaction along a smooth frontier without retraining or per-step inner-loop optimisation.

2 Related Work

Discrete diffusion models. D3PM [Austin et al., 2021] introduced discrete diffusion with structured transition matrices. We build on MDLM [Sahoo et al., 2024], which reduces this to absorbing-state (masking) dynamics with a continuous-time variational bound; SEDD [Lou et al., 2024] proposed alternative score-entropy training objectives for the same setting. Absorbing-state models naturally support hard positional constraints via inpainting (clamping a token at a known position and denoising the remainder), but inpainting alone cannot express soft or global targets such as "molecular weight at least 350" or "at least 10 ocean-related words". Addressing such global, expectation-based constraints is the focus of our method.

Constrained generation. For continuous diffusion, classifier guidance [Dhariwal and Nichol, 2021] and classifier-free guidance (CFG) [Ho and Salimans, 2021] steer generation toward desired attributes, but require either a trained classifier or retraining with label dropout, and operate in continuous state spaces that do not apply to discrete tokens. In autoregressive text generation, constrained decoding enforces lexical or structural constraints via grid beam search [Hokamp and Liu, 2017], dynamic beam allocation [Post and Vilar, 2018], look-ahead heuristics [Lu et al., 2022], or future discriminators [Yang and Klein, 2021]. These approaches rely on left-to-right generation and therefore do not apply to the parallel, partially observed denoising process of diffusion models. We adopt a similar SPDD framework as in [Moor et al., 2026] for autoregressive (AR) slate generation. While the authors assumed sequential token generation and did not directly handle the parallel unmasking dynamics of discrete diffusion, in our work we extend this framework to the diffusion setting, where decisions must be made under evolving context and partial observability.

Constrained discrete diffusion. Nisonoff et al. [2025] proposed predictor guidance for discrete diffusion, modulating transition rates via a Bayesian likelihood ratio from a trained predictor. This approach requires training a separate predictor per property and typically handles one property at a time. Schiff et al. [2025] introduced discrete classifier-based guidance (D-CBG), which guides an unconditional denoising network using a separately trained classifier. Cardei et al. [2025] introduced Constrained Discrete Diffusion (CDD), which avoids additional training but performs an inner Gumbel-softmax optimisation loop at each step, incurring 5–$20\times$ computational overhead. In contrast, our method integrates constraint handling into the sampling process via a closed-form logit bias derived from a Lagrangian formulation, requiring no auxiliary models or inner optimisation. This yields a lightweight procedure with native multi-objective support and convergence guarantees from online convex optimisation [Balseiro et al., 2020].

More broadly, existing approaches either rely on auxiliary models, incur additional optimisation cost, or assume sequential generation, whereas our method addresses all three limitations simultaneously.

3 Notation & Preliminaries

We consider sequences of length L over a vocabulary $\mathcal{V} = \{1, ..., V\}$, and we introduce a special mask token $\texttt{M}$ and define the extended vocabulary $\tilde{\mathcal{V}} = \mathcal{V} \cup \{\texttt{M}\}$. We let T denote the discrete *time horizon* of the diffusion process [Sahoo et al., 2024]. At each step t, the sequence $x_t \in \tilde{\mathcal{V}}^L$ of length L represents a partially masked version of the data, where we let x_t^ℓ be the token at position ℓ.

Let $q(x_0)$ denote the data distribution. We define a forward process that progressively masks tokens in $x_0 \sim q(x_0)$, independently across positions, increasing the probability that each token is replaced by the mask token $\texttt{M}$.

The reverse diffusion process reconstructs the original sequence by iteratively unmasking tokens. At each step t, unmasked tokens are sampled from the learned parametric model distribution $p_\theta(x|x_t)$. Generation starts from an initial sample x_T, typically taken as a fully masked sequence ($x_T = \{\texttt{M}\}_{\ell=1}^L$), and proceeds towards x_0. The goal is to recover a sample from the data distribution $q(x_0)$ by progressively removing noise.

We rely on two standard assumptions in discrete diffusion.

Assumption 1. *(Markov property) The model follows Markov factorisation:*

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t). \tag{1}$$

Assumption 2. *(Absorbing state) For any step $t = 1, ..., T$, token $j \in \mathcal{V}$ and sequence x_t, if a position is already unmasked ($x_t^\ell \neq \texttt{M}$), then:*

$$\Pr(x_{t-1}^\ell = j|x_t) = \begin{cases} 0 & \forall j \neq x_t^\ell \\ 1 & \textit{otherwise.} \end{cases} \tag{2}$$

In other words, once a token is unmasked, it remains fixed for the remainder of the reverse diffusion process. These assumptions are standard in discrete diffusion models [Sahoo et al., 2024, Austin et al., 2021].

These properties define the standard reverse diffusion process used for generation. In the unconstrained setting, tokens are sampled directly from $p_\theta(x_{t-1}|x_t)$ at each step. In this work, we aim to modify this sampling process to enforce global constraints on the generated sequence while remaining close to the model's learned distribution. In the next section, we formalise this objective and derive an inference-time method for constrained generation.

4 Method: Primal-Dual Guided Denoising

Building on the diffusion framework described above, we now consider the constrained generation setting. We assume access to a trained diffusion model $p_\theta(x_{0:T})$ approximating the original data distribution $q(x_0)$ under Assumptions 1 and 2. Our goal is to modify the generation process so that sampled sequences satisfy additional constraints, while remaining as close as possible to the model's learned distribution.

To formalise this, we introduce a target distribution $r(x_0)$ that incorporates the desired constraints. Rather than sampling directly from $q(x_0)$, we aim at sampling from $r(x_0)$, which is constrained but remains close to $q(x_0)$. This corresponds to adjusting the reverse diffusion process at each step while preserving its overall structure.

We impose a Markov factorisation on $r(x_{0:T})$, mirroring the structure of the original model:

$$r(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} r(x_{t-1}|x_t), \tag{3}$$

We assume a shared prior $r(x_T) = p_\theta(x_T)$, and adopt the absorbing state Assumption 2 for $r(x_{0:T})$.

At each step of the reverse diffusion process, $r(x_t^\ell = j|x_{t+1})$ defines the probability of assigning token j to position ℓ given the current partially observed sequence x_{t+1}. We also introduce $a_{\ell j}^t \in [0, 1]$, the

probability that position ℓ is unmasked and assigned token j at step t. Formally,

$$a_{\ell j}^t = \mathbb{E}_{r(x_{t+1})}\Big[\mathbf{1}_{\{x_{t+1}^\ell = \texttt{M}\}} r(x_t^\ell = j | x_{t+1})\Big]. \tag{4}$$

In what follows, we derive how this constrained distribution translates into a simple modification of token probabilities during sampling.

Constraints. We consider linear constraints on the generated sequence x_0, defined through per-token contributions. For clarity, we present the method for a single constraint: the extension to multiple constraints is straightforward and evaluated in Appendix B.3.

Each token j contributes a non-negative amount $b_{\ell j} \geq 0$ when assigned to position ℓ. For example, in topical text generation, $b_{\ell j}$ can indicate whether token j belongs to a target vocabulary, while in molecular generation it can correspond to atomic mass contributions. The total contribution of a sequence is obtained by summing these per-token values across positions. Given a target $R \geq 0$, we aim to generate sequences whose expected total contribution meets this target. Formally,

$$\sum_{\ell=1}^{L} \sum_{j=1}^{V} b_{\ell j} r(x_0^\ell = j) \geq R. \tag{5}$$

Objective. Given the constraint formulation above, our goal is to construct a distribution $r(x_{0:T})$ that satisfies the constraint while remaining as close as possible to the original model distribution $p(x_{0:T})$. This leads to the following optimisation problem:

$$\min_{r(x_{0:T}) \in \Delta} D_{KL}\big(r(x_{0:T}) \,\big|\big|\, p_\theta(x_{0:T})\big) \tag{6}$$

$$\text{subject to } \sum_{\ell=1}^{L} \sum_{j=1}^{V} b_{\ell j} r(x_0^\ell = j) \geq R. \tag{7}$$

Intuitively, the objective encourages the constrained distribution r to stay close to the learned model p_θ, while the constraint enforces that generated sequences achieve the desired target.

Noticing that the KL-divergence is equal to the difference between the cross-entropy and the entropy, we can rewrite Objective (6) in terms of token-level decisions (full derivations in Appendix A.2):

$$\max_{r(x_{0:T}) \in \Delta} \sum_{t=0}^{T-1} \sum_{\ell=1}^{L} \sum_{j=1}^{V} \mathbb{E}_{r(x_{t+1})}\Big[r(x_t^\ell = j | x_{t+1}) \log p_\theta(x_t^\ell = j | x_{t+1})\Big] + H(r(x_{0:T})) \tag{8}$$

$$\text{subject to } \sum_{\ell=1}^{L} \sum_{j=1}^{V} b_{\ell j} r(x_0^\ell = j) \geq R. \tag{9}$$

This formulation makes explicit that the objective decomposes across time steps and token positions, aligning with the structure of the diffusion process.

The formulation above is an *offline* optimisation problem over full trajectories $r(x_{0:T})$. In practice, however, generation proceeds sequentially during the reverse diffusion process, where at each step we observe the current state x_t and must decide how to sample the next tokens.

This motivates an online formulation: at each step t, the model log probabilities $\log p_\theta(x_t^\ell = j | x_{t+1})$ are observed and fixed, and we choose the sampling distribution $r(x_t | x_{t+1})$ accordingly. Using the absorbing-state property and the definition of $a_{\ell j}^t$, we can rewrite the objective as:

$$\max_{r(x_{0:T}) \in \Delta} \sum_{t=0}^{T-1} \sum_{\ell=1}^{L} \mathbb{E}_{r(x_{t+1})}\Big[\sum_{j=1}^{V} \mathbf{1}_{\{x_{t+1}^\ell = \texttt{M}\}} r(x_t^\ell = j | x_{t+1}) \log p_\theta(x_t^\ell = j | x_{t+1})\Big] + H(r(x_{0:T})). \tag{10}$$

Since the logits are observed at each step, they can be treated as constants. Using Equation (4), we can further express the objective in terms of the unmasking probabilities $a^t_{\ell j}$, yielding:

$$\max_{a^t_{\ell j} \in \Delta} \sum_{t=0}^{T-1} \sum_{\ell=1}^{L} \sum_{j=1}^{V} a^t_{\ell j} \log p_\theta(x^\ell_t = j | x_{t+1}) + H(a^t_\ell) \tag{11}$$

$$\text{subject to } \sum_{t=0}^{T-1} \sum_{\ell=1}^{L} \sum_{j=1}^{V} b_{\ell j} a^t_{\ell j} \geq R. \tag{12}$$

This yields an online optimisation problem over token-level sampling decisions. Importantly, this formulation shows that constrained generation can be implemented by modifying token-level sampling probabilities at each denoising step. We now derive how this takes the form of a simple logit bias using a primal-dual formulation.

4.1 Primal-Dual Inference

To solve the online optimisation problem (11)-(12), we aim to derive a sampling rule that modifies token probabilities at each denoising step in a way that enforces the constraint while remaining close to the model distribution. We achieve this using a primal-dual formulation. We begin by dualising constraint (12), yielding:

$$\max_{a^t_{\ell j} \in \Delta} \Big[\sum_{t=0}^{T-1} \sum_{\ell=1}^{L} \sum_{j=1}^{V} a^t_{\ell j} \log p_\theta(x^\ell_t = j | x_{t+1}) + H(a^t_\ell) + \min_{\lambda \geq 0} \lambda \Big(\sum_{t=0}^{T-1} \sum_{\ell=1}^{L} \sum_{j=1}^{V} b_{\ell j} a^t_{\ell j} - R \Big) \Big]. \tag{13}$$

For any fixed value of the Lagrangian multiplier $\lambda = \lambda_{t+1}$, the first-order optimality conditions yield the following sampling distribution:

$$a^t_{\ell j} = \frac{p_\theta(x^\ell_t = j | x_{t+1}) e^{\lambda_{t+1} b_{\ell j}}}{\sum_{i=1}^{V} p_\theta(x^\ell_t = i | x_{t+1}) e^{\lambda_{t+1} b_{\ell i}}}. \tag{14}$$

This corresponds to modifying the model probabilities by an exponential weighting with the dual-adjusted terms $\lambda_{t+1} b_{\ell i}$. This admits an intuitive energy-based interpretation. The dual term $\lambda_{t+1} b_{\ell i}$ acts as an external field on the unbiased model's p_θ energy landscape: it makes the constraint-favoring tokens "energetically cheaper", guiding the reverse process to these new low energy states.

This shows that constrained generation can be implemented by simply adjusting the log probabilities at each denoising step, without altering the model itself. Intuitively, tokens that contribute more toward satisfying the constraint are upweighted, while others are relatively suppressed.

As generation proceeds, the degree of constraint satisfaction may change. Therefore, the multiplier λ must be updated accordingly. We perform this update using mirror descent (derivation in Appendix A.3):

$$\lambda_t = \lambda_T \exp \Big\{ -\eta \sum_{t=0}^{T-1} \sum_{\ell=1}^{L} \sum_{j=1}^{V} \Big(b_{\ell j} - \frac{R}{L} \Big) a^t_{\ell j} \Big\}. \tag{15}$$

In words, the update tracks the cumulative constraint satisfaction during generation. When the constraint is under-satisfied, the multiplier increases, strengthening the bias toward high-contribution tokens; when the constraint is satisfied or exceeded, the multiplier decreases, reducing interference with the model distribution.[2]

Algorithm 1 summarises the resulting procedure. Starting from a fully masked sequence x_T, we iteratively perform the reverse diffusion steps. At each step, we first compute the modified sampling probabilities using Equation (14), then sample tokens for the masked positions, update the running constraint satisfaction, and adjust the Lagrangian multiplier using Equation (15). This yields a simple inference-time procedure that integrates constraint enforcement directly into the denoising process through a logit-level modification of the sampling distribution. Importantly, this modification is applied entirely at inference time and does not require retraining or additional model evaluations.

[2]For above-target objectives, the implementation clips the exponent to $[-20, 20]$ before exponentiation and caps the resulting multiplier at 10^4 to avoid numerical overflow.

Algorithm 1 Primal-Dual Guided Denoising for MDLM

Require: Model $p_\theta(x_{0:T})$, parameters $(R, \eta, b_{\ell j})$, #steps T, initial λ_T

$g_T \leftarrow 0$; $x_T \leftarrow$ [M]^L

for $t = T-1, \dots, 0$ **do**

  $a^t_{\ell j} \leftarrow p_\theta(x_t|x_{t+1}) e^{\lambda_{t+1} b_{\ell j}} \quad \forall \ell = 1, ..., L, \ \forall j = 1, ..., V$ *forward pass*

  $a^t_{\ell j} \leftarrow a^t_{\ell j} / \sum_{i=1}^V a^t_{\ell i}$ *see* Eq. (14)

  Sample $x^\ell_t \sim Cat(\cdot \mid a^t_\ell)$ for all $\ell = 1, ..., L$; update masked positions

  $g_t \leftarrow g_{t+1} + \sum_{\ell=1}^L \left(b_{\ell, x^\ell_t} - \frac{R}{L} \right) \mathbf{1}_{x^\ell_t \neq \texttt{[M]} \ \& \ x^\ell_{t+1} = \texttt{[M]}}$ *slack update*

  $\lambda_t \leftarrow \lambda_T \exp\Big\{ -\eta g_t \Big\}$ *see* Eq. (15)

end for

return x_0

4.2 Theoretical Analysis

Having introduced the primal-dual decoding algorithm, we now turn to its theoretical analysis. In particular, we bound the constraint violation reached by Algorithm 1 and characterise the resulting trade-off between constraint satisfaction and approximation error. We also discuss the computational complexity of the proposed algorithm.

Statement 1. *If Problem (11)-(12) is feasible, then for any $\eta > 0$, the expected constraint violation of Algorithm 1 is upper-bounded by $\mathcal{O}\Big(\frac{1}{\eta} \log \frac{\eta+1}{b_{max} - \frac{R}{L}}\Big)$.*

Proof. We provide the proof in Appendix A.5. □

This result shows that increasing η tightens the bound on constraint violation. Intuitively, from Equation (15), a larger η leads to stronger updates the Lagrangian multiplier λ_{t+1} when the constraint is not satisfied, which increases the sampling probability of tokens with higher contribution $b_{\ell j}$.

We now introduce a *weak temporal consistency* assumption that models how token-level log probabilities evolve across denoising steps. In particular, we decompose the logits produced by the diffusion model into a stable component and a small stochastic drift:

Assumption 3. *For any token $j \in \mathcal{V}$, step $t = T-1, ..., 0$, position $\ell = 1, ..., L$ and sequence x_{t+1}, we assume*

$$\log p_\theta(x^\ell_t = j | x_{t+1}) = \log p_\theta(x^\ell_{t+1} = j | x_{t+2}) + \epsilon_{tj\ell}, \tag{16}$$

where $\epsilon_{tj\ell} \sim \mathcal{N}(\mu_{tj\ell}, \sigma^2)$, with bounded covariance $Cov(\epsilon_{tj\ell}, \epsilon_{\tau jk}) \leq \rho$ for all $(t, \ell) \neq (\tau, k)$, and bounded drift $|\mu_{tj\ell}| \leq \bar{\mu}$.

Intuitively, this assumption states that the model's log probabilities evolve smoothly across denoising steps, with changes captured by a small stochastic perturbation. This reflects the fact that predictions become progressively more refined as the reverse diffusion process unfolds, rather than changing abruptly. Similar temporal consistency assumptions have been used in the analysis of sequential generative models, including LLMs [Maystre et al., 2025].

Statement 2. *The regret of Algorithm 1 is upper bounded by*

$$Regret_\epsilon(\mathcal{A}) \leq \frac{\lambda_T}{\eta} e^{\eta LT \max_{\ell, j} |b_{\ell j} - R/L|} + C_\delta, \tag{17}$$

where $C_\delta = \bar{\mu} LT(T+1) + \sqrt{4(\sigma^2 + LT\rho) \log \frac{2LVT(VT+1)^L}{\delta} (LT + L^2T^3)}$ with probability at least $1 - \delta$.

Proof. We provide the proof in Appendix A.4. □

This bound makes explicit the trade-off between constraint satisfaction and approximation error. In particular, increasing the step size η tightens the constraint violation bound (Statement 1), but leads to

a larger regret term in Equation (17). This reflects the tension between enforcing constraints aggressively and remaining close to the original model distribution. This trade-off is reflected empirically in Section 5, where η controls the balance between constraint satisfaction and distributional fidelity.

The second term C_δ captures stochastic variability of the model logits, as described in Assumption 3. When the variance σ^2, covariance ρ, and drift $\bar{\mu}$ are small, this term decreases, tightening the regret bound. We provide an empirical study of Assumption 3 and its effect on the bounds in Appendix B.1.

Finally, the proposed primal-dual decoding adds only constant-time operations to the inference loop of the unconstrained model (lines 3-4, 6-7 of Algorithm 1), and therefore preserves the computational complexity of standard discrete diffusion models [Sahoo et al., 2024].

4.3 Extensions: Lagrangian Update Regimes

While Constraint (12) enforces a global target R over the full sequence of length L, the Lagrangian update in Equation (15) distributes this target uniformly across positions via the per-position term R/L. This provides a stable normalised signal during generation, but may be suboptimal when constraint satisfaction is unevenly distributed across tokens or steps.

To address this, we consider alternative ways of measuring constraint satisfaction during the reverse diffusion process, leading to different update regimes for the Lagrangian multiplier. Specifically, we study three variants: (1) *Accumulated* slack follows Equation (15), aggregating constraint contributions over time and providing conservative, stable target tracking. (2) *Instantaneous* slack compares the accumulated contribution directly to the global target R , rather than the per-position target R/L, yielding $\lambda_t = \lambda_T \exp\{-\eta(\sum_{\tau=t}^{T-1} \sum_{\ell=1}^{L} \sum_{j=1}^{V} b_{\ell j} a_{\ell j}^{\tau} - R)\}$. Finally, (3) *Optimistic slack*, based on Optimistic Mirror Descent (OMD) [Rakhlin and Sridharan, 2013], incorporates a prediction of future constraint satisfaction:

$$\lambda_t = \lambda_T \exp\{-\eta \sum_{\tau=t}^{T-1} \Big[\Big(\sum_{\ell=1}^{L} \sum_{j=1}^{V} b_{\ell j} a_{\ell j}^{\tau} - R \Big) - M_\tau + M_{\tau+1} \Big] \}, \tag{18}$$

where $M_\tau = R - \sum_{\ell=1}^{L} \sum_{j=1}^{V} p_\theta(x_\tau^\ell = j | x_{\tau+1}) b_{\ell j} \mathbf{1}_{\{x_\tau^\ell = \texttt{[M]}\}}$ is the expected future shortfall estimated from the model's predictions.

Intuitively, accumulated slack provides stable but conservative updates, instantaneous slack enforces constraints aggressively, and optimistic slack balances the two by anticipating future contributions.

5 Experiments

We evaluate primal-dual guided decoding on three structurally different discrete-sequence domains: a sparse counting constraint (topical text), a dense additive property under hard syntactic constraints (molecular generation), and a noisy indirect cluster-score constraint (playlist generation). For each domain we train a single MDLM backbone once and keep it fixed across all diffusion methods.

Table 1 reports two cross-domain metrics. *Pass%* measures constraint satisfaction (e.g., $\geq R$ ocean tokens for text, MW ≥ 350 for molecular generation, and fraction of generated tracks tagged with the target subgenre for playlist). *KL* measures distributional divergence as the unigram KL between constrained and unconstrained samples from the same model. Together, these metrics capture the trade-off between constraint satisfaction and fidelity to the original model distribution. Domain-specific quality metrics and full hyperparameter sweeps are provided in Appendix B.

We compare against three diffusion baselines that share the same backbone: *Static logit bias* (a fixed scalar α added to per-token scores at every step), *D-CBG* [Schiff et al., 2025] (discrete classifier-based guidance using a separately trained property classifier), and *CDD* [Cardei et al., 2025] (training-free with an inner Gumbel-softmax loop per step). The appendix tables additionally include comparisons with *GPT-4.1-mini* [Achiam et al., 2023] on text and molecular, and zero-shot transfer of SPDD to *LLaDA-8B* [Nie et al., 2025] on text. Each method is reported at a representative operating point; the satisfaction–KL trade-off is monotonic in each method's primary control parameter, so no held-out tuning set is required.

SPDD reaches the highest Pass% on every domain. On text it more than doubles CDD's satisfaction; on molecular it lifts the upper-tail acceptance several points beyond the strongest baseline; on playlist

Table 1: Comparison between SPDD and baselines across three datasets, with relative inference cost.

	Text		Molecular		Playlist		Cost	
Method	Pass%	KL↓	Pass%	KL↓	Pass%	KL↓	Sample×	Train
Unconstrained	0.50%	—	29.40%	—	1.14%	—	1.00×	none
Static bias	81.90%	0.706	41.40%	0.019	1.72%	0.003	1.03×	none
D-CBG	0.00%	0.127	42.80%	0.035	1.92%	0.115	1.09×	classifier
CDD	38.00%	0.221	42.20%	0.034	9.18%	0.779	7.28×	none
SPDD (Ours)	92.10%	0.420	48.50%	0.096	9.28%	0.355	1.03×	none

it matches CDD's track-level satisfaction at less than half the KL, while preserving substantially higher token diversity (Appendix B.4). The corresponding KL is the cost of pushing further into the constrained region, and is itself a controllable knob: η traces a smooth frontier between satisfaction and fidelity (Appendix B). These gains come at near-baseline sampling speed and without auxiliary classifier training. The remainder of this section analyses the domain-specific mechanisms underlying these results.

Text results. We evaluate topical steering in text generation by generating children's stories enriched with ocean-related vocabulary, without retraining the underlying model. This tests whether the method can enforce sparse, global lexical constraints while preserving fluency and style. We train a 142M-parameter MDLM on 50k stories from TinyStories [Eldan and Li, 2023] (BPE tokenizer, vocabulary 8,192, sequence length 256), and constrain generation to include at least R=10 ocean-related tokens per story, drawn from a set of 18 target words (e.g., *ocean, whale, dolphin*).

Static bias improves satisfaction but overshoots, as it cannot reduce pressure once the constraint is satisfied. CDD improves over unconstrained sampling but requires an inner optimisation loop. D-CBG underperforms on this sparse target set: its Taylor-approximated classifier gradient is too smooth to produce a sharp bias over the 18 target words, redistributing mass within the non-ocean vocabulary without raising any target token above the sampling threshold, so Pass% remains near the unconstrained level. In contrast, the primal-dual formulation adapts its multiplier to the accumulated constraint deficit, allowing SPDD to increase satisfaction while staying much closer to the unconstrained distribution.

To check whether this effect depends on the ocean vocabulary, we repeated the counting constraint with 18-word nature and food vocabularies. At the same ten-token target, SPDD passes 71–76% of outputs across the three topics, versus 0–2% without guidance (Figure 1).



Figure 1: Cross-topic TinyStories lexical constraint satisfaction (N=100 outputs per topic and method; 18 exact single-token words per topic; R=10, T=256). SPDD uses accumulated slack, η=1, λ_0=0.5, unit token scores, and no multiplier noise. Bars are 95% Wilson binomial intervals. Across topics, SPDD GPT-2 PPL is 23.74–25.34 versus 23.97 for the shared unconstrained pool; pooled Dist-2 is 0.374–0.379 versus 0.389.

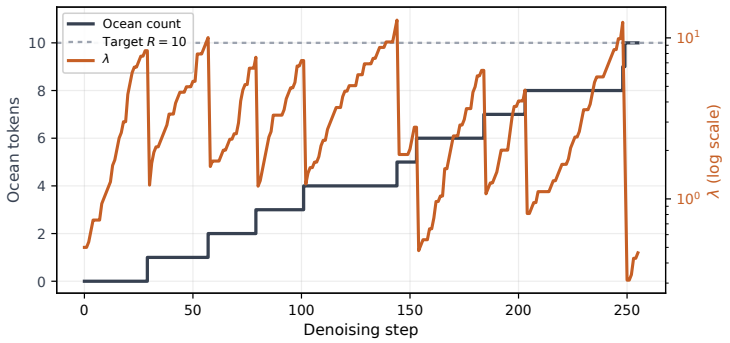

Figure 2: Accumulated-slack multiplier dynamics for a single TinyStories sample ($\eta = 2.0$, target $R = 10$). Ocean-token count and the target are shown on the left axis; the coloured curve shows λ on the right axis, rising as the accumulated deficit grows and stepping down when newly committed ocean tokens reduce it. The sample reaches the target by the end of denoising.

The same mechanism transfers zero-shot to LLaDA-8B without fine-tuning, with the caveat that the larger backbone exhibits distinct qualitative artefacts (Appendix B.2).

The slack definition controls how aggressively λ responds to the running constraint deficit. Figure 2 illustrates the dynamics of the accumulated slack used in the main experiments: λ rises as the sample falls behind its target and decreases as new ocean tokens reduce the deficit. Sweeping the slack mode (see Section 4.3) and η traces a continuum between conservative low-KL guidance and more aggressive high-satisfaction guidance (Appendix B.1).

Molecular results. We evaluate constrained molecular generation: given a diffusion model trained on drug-like molecules, we steer generation toward higher molecular weight (MW) while preserving chemical validity and diversity. This tests whether the method can control a global numerical property without breaking structural constraints. We train a 21.9M-parameter MDLM on 1M molecules from ZINC20 [Irwin et al., 2020], represented as SMILES strings, and enforce MW ≥ 350 using atomic-mass per-token scores. The above-target threshold is well into the upper tail of the unconstrained MW distribution (mean MW ≈ 330), so the constraint exercises non-trivial steering.

In this domain the unconstrained model already places non-negligible mass above the target, so the question is how much further guidance can shift the distribution without breaking chemistry. SPDD attains the highest Pass% in the column while keeping validity above 50%. D-CBG and CDD move closer to the threshold but plateau several points below SPDD; static logit bias becomes competitive only at high α, where the lack of an adaptive feedback loop already pushes mean MW well past the target. The KL of SPDD is correspondingly larger because it shifts more mass into the constrained region; the satisfaction–KL frontier is exposed via η (Appendix B.3).

Playlist results. We apply SPDD to music playlist generation, steering a diffusion model to complete playlists with tracks from a target subgenre. The model generates discrete tokens mapped back to tracks. Each token is an LSH cluster over audio embeddings, and its genre score is the fraction of cluster members tagged with the target genre.

We evaluate playlists in a held-out completion setting: the first 20 session tracks are clamped and the model generates the remaining 30. Pass% measures track-level subgenre satisfaction, the fraction of generated tracks tagged with the target subgenre.

The playlist signal is sparse and indirect: each token is an LSH cluster whose members only probabilistically exhibit the target subgenre, so guidance must amplify a weak per-token signal into a non-trivial output-distribution shift. Static is essentially passive. D-CBG, despite a trained subgenre classifier, barely lifts track-level satisfaction: its gradient is too smooth to raise any cluster's probability above the sampling threshold, the same effect we observed on the sparse text constraint. CDD achieves non-trivial track-level satisfaction but at the cost of substantial token-diversity collapse. SPDD improves on CDD satisfaction level at less than half the KL while keeping token diversity

much closer to the unconstrained baseline; the remaining trade-off is on relevance to the user's natural continuation, which any meaningful steering of the distribution must shift.

This trade-off is itself controllable: varying η moves SPDD along the satisfaction–KL frontier (Appendix B.4), letting the user pick a point that fits a target fidelity or satisfaction budget. In contrast, CDD's τ does not provide a comparable increase in satisfaction in our sweep, and D-CBG's γ saturates at the classifier's decision boundary. A consistent pattern emerges across the three domains. When the constraint signal is sparse, indirect, or mediated by noisy token clusters, the primal-dual update provides the most favourable trade-off: it improves satisfaction while preserving the base sampler's computational profile and bounding distributional fidelity through η.

6 Discussion

The central contribution of primal-dual guided decoding is *enabling constrained generation with minimal distributional divergence*. The additive logit bias arises as the optimal solution to a KL-regularised Lagrangian (Equation 14), so guidance operates directly in the model's probability space rather than through re-prompting, auxiliary predictors, or inner optimisation. Empirically, SPDD keeps constrained samples close to each model's unconstrained distribution across text, molecular, and playlist generation (Table 1). The comparisons clarify the trade-offs of the alternatives: CDD is a training-free option when additional inner-loop computation is acceptable, and D-CBG offers a classifier-driven path when a property classifier is already trained. SPDD complements these methods by targeting decomposable constraints with an adaptive, low-cost update. The formal violation bound (Statement 1) makes the satisfaction–fidelity trade-off explicit through η.

The framework is lightweight: it requires no retraining, no auxiliary model, and only a per-token score vector. In the main experiments we use the accumulated slack update, which follows the mirror-descent derivation directly; alternative slack definitions are evaluated in the appendix.

Limitations. Our experiments focus on absorbing-state diffusion, where early token commitments are not revisited, making slack design and η scheduling most important in early denoising steps (Table 6). Extending the same primal-dual principle to non-absorbing discrete diffusion is a promising direction for enabling later corrections. For non-decomposable constraints (e.g., QED), improved partial-sequence slack estimators could enhance early updates, and very large backbones may benefit from standard repetition controls or adaptive λ_0 schedules. Finally, the method provides probabilistic constraint bounds (Statement 1); applications requiring strict feasibility may combine it with rejection sampling or post-hoc checks.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex Xijie Lu, Nicolo Fusi, Ava P Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, 2023. doi: 10.1101/2023.09.11.556673.

Jacob Austin, Daniel Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Santiago R Balseiro, Haihao Lu, and Vahab S Mirrokni. Dual mirror descent for online allocation problems. In *International Conference on Machine Learning (ICML)*, 2020.

Arthur E. Bryson and Yu-Chi Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. Hemisphere Publishing, Washington, DC, revised edition, 1975.

Michael Cardei, Jacob K Christopher, Bhavya Kailkhura, Thomas Hartvigsen, and Ferdinando Fioretto. Constrained discrete diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ronen Eldan and Yuanzhi Li. TinyStories: How small can language models be and still speak coherent English? *arXiv preprint arXiv:2305.07759*, 2023.

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

Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1535–1546, 2017.

John J Irwin, Khanh G Tang, Jennifer Young, Chinzorig Dandarchuluun, Benjamin R Wong, Munkhzul Khurelbaatar, Yurii S Moroz, John Mayfield, and Roger A Sayle. ZINC20 – a free ultralarge-scale chemical database for ligand discovery. *Journal of Chemical Information and Modeling*, 60(12):6065–6073, 2020.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning (ICML)*, 2024.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A Smith, and Yejin Choi. NeuroLogic A*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 780–799, 2022.

Lucas Maystre, Gabriel Barello, Tudor Berariu, Aleix Cambray, Rares Dolga, Alvaro Ortega Gonzalez, Andrei Nica, and David Barber. Incremental sequence classification with temporal consistency, 2025. URL `https://arxiv.org/abs/2505.16548`.

Dmitrii Moor, Ben Carterette, Senthilkumar Krishnamoorthy, Kyle Kretschman, Denis Beslic, Melissa Yalla, Alice Y Wang, and Mounia Lalmas. Stochastic primal-dual decoding for multiobjective generative recommender systems. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '26, page 3775–3786, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722592. doi: 10.1145/3770855.3817806. URL `https://doi.org/10.1145/3770855.3817806`.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2502.09992. Oral presentation.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *International Conference on Learning Representations (ICLR)*, 2025.

Matt Post and David Vilar. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1314–1324, 2018.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory (COLT)*, 2013. URL https://arxiv.org/abs/1208.3728.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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. In *International Conference on Learning Representations (ICLR)*, 2025.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2402.18567.

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

A Additional Proofs

A.1 Chain Rule for KL divergence, Equation (21)

$$
\begin{aligned}
&D_{KL}\Big(p(x_T)\prod_{t=1}^{T} r(x_{t-1}|x_t)\,\Big|\Big|\,p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)\Big) = \\
&\quad\sum_{x_0,\ldots,x_T} p(x_T)\prod_{t=1}^{T} r(x_{t-1}|x_t)\log\frac{\prod_{t=1}^{T} r(x_{t-1}|x_t)}{\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)} = \\
&\quad\sum_{x_0,\ldots,x_T} p(x_T)\prod_{t=1}^{T} r(x_{t-1}|x_t)\sum_{t=1}^{T}\log\frac{r(x_{t-1}|x_t)}{p_\theta(x_{t-1}|x_t)} = \\
&\quad\sum_{x_T} p(x_T)\underbrace{\sum_{x_{T-1}} r(x_{T-1}|x_T)\log\frac{r(x_{T-1}|x_T)}{p_\theta(x_{T-1}|x_T)}}_{D_{KL}\left(r(x_{T-1}|x_T)||p_\theta(x_{T-1}|x_T)\right)}\underbrace{\sum_{x_{0:T-2}}\prod_{t=1}^{T-1} r(x_{0:T-2}|x_{T-1})}_{=1} + \\
&\sum_{x_{T-1:T}} r(x_{T-1}|x_T)p(x_T)\underbrace{\sum_{x_{T-2}} r(x_{T-2}|x_{T-1})\log\frac{r(x_{T-2}|x_{T-1})}{p_\theta(x_{T-2}|x_{T-1})}}_{D_{KL}\left(r(x_{T-2}|x_{T-1})||p_\theta(x_{T-2}|x_{T-1})\right)}\underbrace{\sum_{x_{0:T-3}}\prod_{t=1}^{T-2} r(x_{0:T-3}|x_{T-2})}_{=1} + \ldots = \\
&\sum_{t=0}^{T-1}\mathbb{E}_{r(x_{t+1})}\Big[D_{KL}(r(x_t|x_{t+1})||p_\theta(x_t|x_{t+1}))\Big]. \qquad (19)
\end{aligned}
$$

A.2 Derivation of Equations (8)-(9)

We can re-write the KL-divergence term in Objective (6) as follows:

$$
\begin{aligned}
D_{KL}\big(r(x_{0:T})\,\big|\big|\,p_\theta(x_{0:T})\big) = &D_{KL}\Big(p(x_T)\prod_{t=0}^{T-1} r(x_t|x_{t+1})\,\Big|\Big|\,p(x_T)\prod_{t=0}^{T-1} p_\theta(x_t|x_{t+1})\Big) = \qquad (20)\\
&\sum_{t=0}^{T-1}\mathbb{E}_{r(x_{t+1})}D_{KL}\Big(r(x_t|x_{t+1})\,\big|\big|\,p_\theta(x_t|x_{t+1})\Big) = \qquad (21)\\
&\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\mathbb{E}_{r(x_{t+1})}D_{KL}\Big(r\big(x_t^\ell|x_{t+1}\big)\,\big|\big|\,p_\theta\big(x_t^\ell|x_{t+1}\big)\Big), \qquad (22)
\end{aligned}
$$

where Equation (21) follows from the chain rule for KL divergence (see Appendix A.1) and Equation (22) follows from independence of sampling across positions $\ell = 1, ..., L$ in the generated sequences. Expanding the KL-term we can further simplify this optimisation objective as follows:

$$
\begin{aligned}
&\min_{r(x_{0:T})\in\Delta} D_{KL}\big(r(x_{0:T})\,\big|\big|\,p_\theta(x_{0:T})\big) = \\
&\qquad\min_{r(x_{0:T})\in\Delta}\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\mathbb{E}_{r(x_{t+1})}\sum_{j=1}^{V} r(x_t^\ell = j|x_{t+1})\log\left(\frac{r(x_t^\ell = j|x_{t+1})}{p_\theta(x_t^\ell = j|x_{t+1})}\right) = \\
&\qquad\max_{r(x_{0:T})\in\Delta}\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\mathbb{E}_{r(x_{t+1})}\Big[\sum_{j=1}^{V} r(x_t^\ell = j|x_{t+1})\log p_\theta(x_t^\ell = j|x_{t+1})\Big] + H(r(x_{0:T})).
\end{aligned} \qquad (23)
$$

A.3 Mirror Descent

To derive the update rule for λ_t we rely on the mirror descent applied to the inner minimisation problem in Equation (13). In particular, we let

$$\lambda_t = \arg\min_{\lambda \geq 0} \Big\{ \sum_{t=0}^{T-1} \sum_{\ell=1}^{L} \sum_{j=1}^{V} \big(b_{\ell j} - \frac{R}{L}\big) a_{\ell j}^t + D_h(\lambda || \lambda_{t+1}) \Big\}, \tag{24}$$

where $D_h(\lambda||\lambda_{t+1})$ is the Bregman divergence that is induced by the negative entropy function $h(\lambda) = \lambda \ln \lambda - \lambda$. Solving Problem (24) we obtain:

$$\lambda_t = \lambda_{t+1} \exp\Big\{ -\eta \sum_{\ell=1}^{L} \sum_{j=1}^{V} \big(b_{\ell j} - \frac{R}{L}\big) a_{\ell j}^t \Big\}, \tag{25}$$

where $\eta > 0$ is a calibration step-size parameter. Equivalently, we obtain:

$$\lambda_t = \lambda_T \exp\Big\{ -\eta \sum_{t=0}^{T-1} \sum_{\ell=1}^{L} \sum_{j=1}^{V} \big(b_{\ell j} - \frac{R}{L}\big) a_{\ell j}^t \Big\}. \tag{26}$$

A.4 Expected Regret Analysis

In this section, we derive an upper bound on the optimal reward. To this end, we start with analysing the stochastic process defined in Assumption 3. Specifically, from Assumption 3 we can write:

$$\log p_\theta(x_t^\ell = j | x_{t+1}) = \log p_\theta(x_T^\ell = j) + \sum_{\tau=t}^{T-1} \epsilon_{\tau j \ell}, \quad \forall t = 1, ..., T-1, \ell = 1, ..., L, j \in \mathcal{V}, \tag{27}$$

where the logits computed at time step t of the reverse diffusion process are modelled as a stochastic process that steers the prior logits $\log p_\theta(x_T^\ell = j)$ with a Gaussian process $\sum_{\tau=t}^{T-1} \epsilon_{tj\ell}$. We decompose $\epsilon_{tj\ell} = \mu_{tj\ell} + \eta_{tj\ell}$ into the random noise $\eta_{tj\ell} \sim \mathcal{N}(0, \sigma^2)$ and the constant average drift $\mu_{tj\ell} \in \mathbb{R}$.

Now, let $\mathcal{H}_{tj\ell} = \sum_{\tau=t}^{T-1} \eta_{\tau j \ell}$ and $\mathcal{M}_{tj\ell} = \sum_{\tau=t}^{T-1} \mu_{\tau j \ell}$ be the cumulative random and the drift components reached by step t of the reverse process. Notice, that $\mathcal{H}_{tj\ell}$ is a zero-mean process for $t = T-1, ..., 0$, s.t.,

$$Var(\mathcal{H}_{tj\ell}) \leq (T-t)\sigma^2 + (T-t)(T-t-1)\rho, \tag{28}$$

and

$$Cov(\mathcal{H}_{tj\ell}, \mathcal{H}_{tjk}) \leq (T-t)^2 \rho \quad \forall \ell \neq k, \forall t = 1, ..., T, \forall j \in \mathcal{V}. \tag{29}$$

Furthermore, using the fact that $\mu_{tj\ell} \leq \bar{\mu}$ (see Assumption 3) we can bound

$$\mathcal{M}_{tj\ell} \leq \bar{\mu} \cdot (T-t). \tag{30}$$

We let $Reward(a, \epsilon)$ be the reward reached by the randomised allocation $a \sim a^*(\epsilon)$ drawn from the optimal policy $a^*(\epsilon)$ given randomness ϵ defined in Assumption 3. Observe, that the expected reward $\mathbb{E}_{a \sim a^*(\epsilon)}[Reward(a, \epsilon) \,|\, \epsilon\,]$ reached with this randomised allocation a is itself a random variable as it depends on ϵ. We will, therefore, derive an upper bound for this expected reward that must hold with high probability given the randomness generated by ϵ.

Statement 3. *Expected reward $\mathbb{E}_{a \sim a^*(\epsilon)}[Reward(a, \epsilon) \,|\, \epsilon\,]$ of the optimal allocation in Problem (11)-(12) is upper bounded by*

$$\mathbb{E}_{a \sim a^*(\epsilon)}[Reward(a, \epsilon) \,|\, \epsilon\,] \leq \min_{\lambda \geq 0} T \sum_{\ell=1}^{L} \log \sum_{i=1}^{V} p_\theta(x_T^\ell = i) e^{\lambda(b_{\ell i} - R/L)} + \bar{\mu} LT + \sqrt{2LT(\sigma^2 + LT\rho)(\log \frac{1}{\delta} + L \log(VT+1))} \tag{31}$$

with probability at least $1 - \delta$.

Proof. From Equations (11) and (27) we obtain:

$$\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \log p_\theta(x_t^\ell = j|x_{t+1}) = \sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \log p_\theta(x_T^\ell = j) + \underbrace{\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \mathcal{M}_{tj\ell}}_{\text{systematic drift}} + \underbrace{\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \mathcal{H}_{tj\ell}}_{\text{random component}}. \tag{32}$$

Remember that in the MDLM setting once the token is allocated (i.e., $a_{\ell j}^t = 1$), it stays fixed until the end of generation, i.e.,

$$\sum_{t=0}^{T-1}\sum_{j=1}^{V} a_{\ell j}^t \leq 1 \quad \forall \ell = 1, ..., L. \tag{33}$$

Therefore, using Equations (30) and (33) the systematic drift can be upper bounded as follows:

$$\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \mathcal{M}_{tj\ell} \leq \sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \bar{\mu} \cdot (T-t) \leq \bar{\mu} LT. \tag{34}$$

Now, for any position ℓ let $\mathcal{H}_{\ell t} = \sum_{j=1}^{V} a_{\ell j}^t \mathcal{H}_{tj\ell} \sim \mathcal{N}\big(0,\, Var(\mathcal{H}_{\ell t})\big)$, where

$$Var(\mathcal{H}_{\ell t}) = \sum_{i,j=1}^{V} a_{\ell j}^t a_{\ell i}^t Cov(\mathcal{H}_{tj\ell}, \mathcal{H}_{ti\ell}) = \sum_{j=1}^{V} a_{\ell j}^t Var(\mathcal{H}_{tj\ell}) \leq (T-t)\sigma^2 + (T-t)(T-t-1)\rho.$$

Here, we used the fact that we can allocate at most one token to position ℓ, i.e., $\sum_{i=1}^{V} a_{\ell i}^t \leq 1$. Now, observe that $\mathcal{H} = \sum_{t=0}^{T-1}\sum_{\ell=1}^{L} \mathcal{H}_{\ell t}$ is Gaussian with zero mean. Remember, that since only a single token j can be allocated to a specific position ℓ (Equation (33)), then for any fixed feasible allocation $a_{\ell j}^t$ we must have:

$$\mathcal{H} = \sum_{\tau=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} b_{\ell j}^\tau(a)\eta_{\tau j\ell}, \tag{35}$$

where $b_{\ell j}^\tau(a) = \sum_{t=1}^{\tau} a_{\ell j}^t$ indicates whether token j is allocated to position ℓ by time step τ (in this case, all the following random contributions $\eta_{\tau j\ell}$ must have non-zero coefficients). Noticing that $\sum_{\tau,j,\ell} b_{\ell j}^\tau(a) = \sum_{t\ell j} a_{\ell j}^t (T-t) \leq L(T-1)$ we can bound:

$$Var(\mathcal{H}) \leq LT\sigma^2 + L^2T^2\rho.$$

Therefore, using the concentration inequality we can bound:

$$\Pr(\mathcal{H} \geq \delta) \leq \exp\Big\{ - \frac{\delta^2}{2LT(\sigma^2 + LT\rho)} \Big\}.$$

Notice that we want to bound $\mathcal{H}$ for any possible allocations. As $\mathcal{H}$ is linear in $a_{\ell j}^t$ and we optimise over a convex feasible set, the supremum of $\mathcal{H}$ is attained at the deterministic allocations $a_{\ell j}^t$. Therefore, using the union bound we obtain:

$$\Pr\Big(\sup_a \mathcal{H} \geq \delta\Big) \leq (VT+1)^L \exp\Big\{ - \frac{\delta^2}{2LT(\sigma^2 + LT\rho)} \Big\},$$

where $(VT+1)^L$ is the number of possible deterministic allocations. Finally,

$$\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \log p_\theta(x_t^\ell = j|x_{t+1}) \leq \sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \log p_\theta(x_T^\ell = j) + \bar{\mu} LT + \sqrt{2LT(\sigma^2 + LT\rho)(\log\frac{1}{\delta} + L\log(VT+1))}. \tag{36}$$

with probability at least $1-\delta$.

From Equations (13) and (36) we now obtain:

$$\mathbb{E}_{a\sim a^*(\epsilon)}[Reward(a,\epsilon)\,|\,\epsilon] \leq \min_{\lambda\geq 0}\max_{a^t_{\ell j}\in\Delta}\Big[\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a^t_{\ell j}\Big(\log p_\theta(x^\ell_T=j)+\lambda(b_{\ell j}-\frac{R}{L})\Big)+H(a^t_\ell)\Big] + \bar{\mu}LT+\sqrt{2LT(\sigma^2+LT\rho)(\log\frac{1}{\delta}+L\log(VT+1))} \tag{37}$$

with probability at least $1-\delta$. From the first-order conditions we obtain the optimal sampling probabilities:

$$a^*_{\ell j}=\frac{p_\theta(x^\ell_T=j)e^{\lambda(b_{\ell j}-R/L)}}{\sum_{i=1}^{V}p_\theta(x^\ell_T=i)e^{\lambda(b_{\ell i}-R/L)}}. \tag{38}$$

Importantly, as the coefficients $\log p_\theta(x^\ell_T=j)$ of the simplified dualised Problem (37) do not depend on t, the resulting solution in Equation (38) does not depend on t as well. We now obtain:

$$\mathbb{E}_{a\sim a^*(\epsilon)}[Reward(a,\epsilon)\,|\,\epsilon] \leq \min_{\lambda\geq 0} T\cdot\sum_{\ell=1}^{L}\log\sum_{i=1}^{V}p_\theta(x^\ell_T=i)e^{\lambda(b_{\ell i}-R/L)} + \bar{\mu}LT+\sqrt{2LT(\sigma^2+LT\rho)(\log\frac{1}{\delta}+L\log(VT+1))}$$

with probability at least $1-\delta$. Q.E.D. □

From Statement 3 it follows that in domains with the small (co-)variance values ρ, σ and the small drift values $\bar{\mu}$ (see Section B.1) the approximation bounds of primal-dual inference may get significantly tighter.

We now switch to deriving the lower bound on the objective value reached by Algorithm 1.

Statement 4. *Expected reward of Algorithm 1 is lower bounded by*

$$\mathbb{E}_{a(\epsilon)}[Reward(\mathcal{A},\epsilon)|\epsilon] \geq \sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\log\sum_{i=1}^{V}p_\theta(x^\ell_T=i)e^{\lambda_{t+1}(b_{\ell i}-R/L)}-\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V}a^t_{\ell j}\lambda_{t+1}(b_{\ell j}-\frac{R}{L}) - LT^2\bar{\mu}-LT\sqrt{2\log\frac{LVT}{\delta}T(\sigma^2+(T-1)\rho)} \tag{39}$$

with probability at least $1-\delta$.

Proof. Consider the expected reward produced by Algorithm 1 at iteration t:

$$\mathbb{E}_{a(\epsilon)}[\pi_t|\epsilon]=\sum_{\ell=1}^{L}\sum_{j=1}^{V}a^t_{\ell j}\log p_\theta(x^\ell_t=j|x_{t+1})+H(a^t_\ell), \tag{40}$$

where the sampling probability $a^t_{\ell j}$ is defined in Equation (14). Substituting $a^t_{\ell j}$ in Equation (40) we obtain:

$$\mathbb{E}_{a(\epsilon)}[\pi_t|\epsilon]=\sum_{\ell=1}^{L}\log\sum_{i=1}^{V}p_\theta(x^\ell_t=i|x_{t+1})e^{\lambda_{t+1}(b_{\ell i}-R/L)}-\sum_{\ell=1}^{L}\sum_{j=1}^{V}a^t_{\ell j}\lambda_{t+1}(b_{\ell j}-\frac{R}{L}). \tag{41}$$

We now rely on Assumption 3 to simplify the expression above as follows:

$$\mathbb{E}_{a(\epsilon)}[\pi_t|\epsilon]=\sum_{\ell=1}^{L}\log\sum_{i=1}^{V}p_\theta(x^\ell_T=i)e^{\lambda_{t+1}(b_{\ell i}-R/L)}e^{\sum_{\tau=t}^{T-1}\epsilon_{\tau i\ell}}-\sum_{\ell=1}^{L}\sum_{j=1}^{V}a^t_{\ell j}\lambda_{t+1}(b_{\ell j}-\frac{R}{L}).$$

Observe, that for any real scalar value $X_{i\ell}>0$ we have

$$\Pr\Big[\sum_{\tau=t}^{T-1}\epsilon_{\tau i\ell}\geq -X_{i\ell}\Big]\geq\Pr\Big[\mathcal{H}_{ti\ell}\geq(T-t)\bar{\mu}-X_{i\ell}\Big]=1-\Pr\Big[\mathcal{H}_{ti\ell}<(T-t)\bar{\mu}-X_{i\ell}\Big].$$

Now, if we let $X_{i\ell} = (T-t)\bar{\mu} + \gamma$, then we can use the concentration bound to obtain:

$$\Pr\Big[\sum_{\tau=t}^{T-1} \epsilon_{\tau i\ell} \geq -X_{i\ell}\Big] \geq 1 - \exp\Big\{-\frac{\gamma^2}{2(T-t)(\sigma^2 + (T-t-1)\rho)}\Big\}.$$

Consequently, using the union bound across i and ℓ we can bound

$$\mathbb{E}_{a(\epsilon)}[\pi_t|\epsilon] \geq \sum_{\ell=1}^{L} \log \sum_{i=1}^{V} p_\theta(x_T^\ell = i) e^{\lambda_{t+1}(b_{\ell i} - R/L)} - \sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \lambda_{t+1}(b_{\ell j} - \frac{R}{L})$$
$$- L(T-t)\bar{\mu} - L\sqrt{2\log\frac{LV}{\delta}(T-t)(\sigma^2 + (T-t-1)\rho)}$$

with probability at least $1-\delta$. Finally, using the union bound across the time steps t we obtain:

$$\mathbb{E}_{a(\epsilon)}\Big[\sum_{t=0}^{T-1}\pi_t\Big|\epsilon\Big] \geq \sum_{t=0}^{T-1}\sum_{\ell=1}^{L} \log \sum_{i=1}^{V} p_\theta(x_T^\ell = i) e^{\lambda_{t+1}(b_{\ell i} - R/L)} - \sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \lambda_{t+1}(b_{\ell j} - \frac{R}{L})$$
$$- LT^2\bar{\mu} - LT\sqrt{2\log\frac{LVT}{\delta}T(\sigma^2 + (T-1)\rho)} \tag{42}$$

with probability at least $1-\delta$. Q.E.D. □

Now, once we derived the upper bound on the expected optimal reward (Statement 3) and the lower bound on the expected reward reached by Algorithm 1 (Statement 4), we can derive an upper bound on the regret of Algorithm 1. First, we define the expected regret as

$$Regret_\epsilon(\mathcal{A}) = \mathbb{E}_{a\sim a^*(\epsilon)}[Reward(a,\epsilon)\,|\,\epsilon\,] - \mathbb{E}_{a(\epsilon)}[Reward(\mathcal{A},\epsilon)|\,\epsilon\,] \tag{43}$$

We start with the following lemma:

Lemma 1. *The regret of Algorithm 1 is at most*

$$Regret_\epsilon(\mathcal{A}) \leq \sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \cdot \lambda_{t+1}(b_{\ell j} - \frac{R}{L}) + \bar{\mu}LT(T+1) + \tag{44}$$
$$\sqrt{4(\sigma^2 + LT\rho)\log\frac{2LVT(VT+1)^L}{\delta}(LT + L^2T^3)}$$

with probability at least $(1-\delta)$.

Proof. Let

$$\lambda^* \in \arg\min_{\lambda} \sum_{\ell=1}^{L} \log \sum_{i=1}^{V} p_\theta(x_T^\ell = i) e^{\lambda(b_{\ell i} - R/L)},$$

see Equation (31). This implies that

$$\sum_{\ell=1}^{L} \log \sum_{i=1}^{V} p_\theta(x_T^\ell = i) e^{\lambda^*(b_{\ell i} - R/L)} \leq \sum_{\ell=1}^{L} \log \sum_{i=1}^{V} p_\theta(x_T^\ell = i) e^{\lambda_t(b_{\ell i} - R/L)}, \;\; \forall t = 1, ..., T.$$

Consequently,

$$T\sum_{\ell=1}^{L} \log \sum_{i=1}^{V} p_\theta(x_T^\ell = i) e^{\lambda^*(b_{\ell i} - R/L)} \leq \sum_{t=0}^{T-1}\sum_{\ell=1}^{L} \log \sum_{i=1}^{V} p_\theta(x_T^\ell = i) e^{\lambda_t(b_{\ell i} - R/L)}.$$

Therefore, from Statements 3 and 4 it follows:

$$\mathbb{E}_{a\sim a^*(\epsilon)}[Reward(a,\epsilon)\,|\,\epsilon\,] \leq \mathbb{E}_{a(\epsilon)}[Reward(\mathcal{A},\epsilon)|\,\epsilon\,] + \sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \cdot \lambda_{t+1}(b_{\ell j} - \frac{R}{L})$$
$$+\bar{\mu}LT(T+1) + \sqrt{4(\sigma^2 + LT\rho)\log\frac{2LVT(VT+1)^L}{\delta}(LT + L^2T^3)}$$

with probability at least $1-\delta$. Q.E.D. □

We are now ready to prove our final regret bound:

Statement 2. *The regret of Algorithm 1 is upper bounded by*

$$Regret_{\epsilon}(\mathcal{A}) \leq \frac{\lambda_T}{\eta} e^{\eta LT \max_{\ell,j} |b_{\ell j} - R/L|} + C_{\delta}, \tag{17}$$

where $C_{\delta} = \bar{\mu}LT(T+1) + \sqrt{4(\sigma^2 + LT\rho)\log\frac{2LVT(VT+1)^L}{\delta}(LT + L^2T^3)}$ *with probability at least* $1-\delta$.

Proof. Let δ_t be a per-step reward in Equation (15), i.e.,

$$\delta_t = \sum_{\ell=1}^{L}\sum_{j=1}^{V}(b_{\ell j} - \frac{R}{L})a_{\ell j}^t, \tag{45}$$

that is defined for any deterministic allocation $a_{\ell j}^t$ at time step t of the reverse diffusion process. In this case, we can re-write Equation (15) as follows:

$$\lambda_t = \lambda_T \exp\Big\{ -\eta \sum_{t=t}^{T-1} \delta_t \Big\}. \tag{46}$$

Now, we can bound the first sum in Equation (44) as follows:

$$\sum_{t=0}^{T-1}\sum_{\ell=1}^{L}\sum_{j=1}^{V} a_{\ell j}^t \cdot \lambda_{t+1}(b_{\ell j} - \frac{R}{L}) = \sum_{t=0}^{T-1}\lambda_{t+1}\delta_t \leq \sum_{t=0}^{T-1}\lambda_{t+1}|\delta_t|. \tag{47}$$

Let $B = L \cdot \max_{\ell,j}|b_{\ell j} - \frac{R}{L}|$, so that $|\delta_t| \leq B$ for all $t = 0, ..., T-1$. Consequently, we can bound:

$$\sum_{t=0}^{T-1}\lambda_{t+1}|\delta_t| \leq B\lambda_T \sum_{t=0}^{T-1}\exp\{\eta B(T-t-1)\} \leq B\lambda_T \frac{e^{\eta BT}-1}{e^{\eta B}-1} \leq \frac{\lambda_T}{\eta}e^{\eta BT}. \tag{48}$$

□

A.5 Bounds on Constraint Violation

We now illustrate how to bound the expected value of constraint violations reached by Algorithm 1.

Statement 1. *If Problem (11)-(12) is feasible, then for any* $\eta > 0$*, the expected constraint violation of Algorithm 1 is upper-bounded by* $\mathcal{O}\Big(\frac{1}{\eta}\log\frac{\eta+1}{b_{max}-\frac{R}{L}}\Big)$.

Proof. We let (Π_{min}, Π_{max}) be the support of the log probabilities distribution, and let b_{max} be the maximal contribution of the tokens towards the constraint, i.e.,

$$b_{max} = \max_{\ell,j} b_{\ell j}. \tag{49}$$

We let r_t be the *marginal reward* obtained by Algorithm 1 at iteration t, i.e.,

$$r_t = \sum_{\ell=1}^{L}\Big(b_{\ell j(t)} - \frac{R}{L}\Big), \tag{50}$$

where $j(t)$ is the index of the token allocated to position ℓ at step t. Remember, that the expected number of masked tokens at time $t+1$ of the reverse diffusion process can be computed as $L(1-\alpha_{t+1})$, and the probability that the token gets unmasked at step t is $\frac{\alpha_t - \alpha_{t+1}}{1-\alpha_{t+1}}$. Therefore, the trivial lower bound on the expected marginal reward r_t at time t is

$$\mathbb{E}_{p_\theta}[r_t] \geq -\frac{R}{L} \cdot \underbrace{\frac{\alpha_t - \alpha_{t+1}}{1-\alpha_{t+1}} \cdot L(1-\alpha_{t+1})}_{\substack{\text{Expected number of tokens} \\ \text{unmasked at time } t}} = -R \cdot (\alpha_t - \alpha_{t+1}). \tag{51}$$

Now, let us choose some $\beta \in (0, b_{max})$, and assume that for all steps $t \geq \tilde{t}$ we have $\lambda_t \geq (\eta+1) \cdot \frac{\Pi_{max}-\Pi_{min}}{b_{max}-\beta}$. Then, for any tokens i and j s.t., $b_{max} = b_{\ell j} > \beta \geq b_{\ell i}$ we have:

$$\underbrace{\big(\log p_\theta(j) + \lambda_t b_{\ell j}\big)}_{s_{\ell j}} - \underbrace{\big(\log p_\theta(i) + \lambda_t b_{\ell i}\big)}_{s_{\ell i}} = \log p_\theta(j) - \log p_\theta(i) + \lambda_t\big(b_{\ell j} - b_{\ell i}\big) \geq$$

$$-(\Pi_{max} - \Pi_{min}) + (\eta+1) \cdot \frac{\Pi_{max} - \Pi_{min}}{b_{max} - \beta}\big(b_{max} - b_{\ell i}\big) \geq$$

$$(\Pi_{max} - \Pi_{min})\eta\Big[\frac{b_{max} - \beta}{b_{max} - \beta}\Big] \geq \eta(\Pi_{max} - \Pi_{min}). \tag{52}$$

This means that for any time step $t = 1, ..., \tilde{t}$ the score $s_{\ell j}$ of the token with the largest attribute b_{max} is at least $\eta(\Pi_{max} - \Pi_{min})$ larger that the score of any token with attribute $b_{\ell i} \leq \beta$, i.e., $s_{\ell i} \leq s_{\ell j} - \eta(\Pi_{max} - \Pi_{min})$. Therefore, the probability of sampling tokens with attributes at least β can be lower bounded as follows:

$$\frac{\exp\{s_{\ell j}\}}{\sum_{i=1}^{V}\exp\{s_{\ell i}\}} \geq \frac{e^{s_{\ell j}}}{\exp\{s_{\ell j}\} + (V-1)e^{s_{\ell j} - \eta(\Pi_{max} - \Pi_{min})}} = \frac{1}{1 + (V-1)e^{-\eta(\Pi_{max} - \Pi_{min})}}. \tag{53}$$

Consequently, we can lower bound the expected marginal reward for any time steps 1,...,$\tilde{t}$:

$$\mathbb{E}_{p_\theta}[r_t] \geq \frac{(L\beta - R)(\alpha_t - \alpha_{t+1})}{1 + (V-1)\exp\{-\eta(\Pi_{max} - \Pi_{min})\}} \quad \forall t = 1, ..., \tilde{t}. \tag{54}$$

Observe, that by choosing sufficiently large value of η, we can make this marginal reward being very close to its maximal possible value $b_{max} - \frac{R}{L}$ for each unmasked token.

Now, the total constraint violation:

$$-\sum_{t=1}^{T}\mathbb{E}_{p_\theta}[r_t] = -\sum_{t=\tilde{t}+1}^{T}\mathbb{E}_{p_\theta}[r_t] - \sum_{t=1}^{\tilde{t}}\mathbb{E}_{p_\theta}[r_t] \leq$$

$$R(\alpha_{\tilde{t}+1} - \alpha_T) - \frac{(L\beta - R)(\alpha_1 - \alpha_{\tilde{t}+1})}{1 + (V-1)\exp\{-\eta(\Pi_{max} - \Pi_{min})\}} \leq$$

$$R(\alpha_{\tilde{t}+1} - \alpha_T) - (L\beta - R)(\alpha_1 - \alpha_{\tilde{t}+1}) \leq$$

$$R(\alpha_1 - \alpha_T) - L\beta(\alpha_1 - \alpha_{\tilde{t}+1}).$$

Notice, that for Algorithm 1 to reach $\tilde{\lambda} = (\eta+1) \cdot \frac{\Pi_{max}-\Pi_{min}}{b_{max}-\beta}$, it must accumulate the total expected marginal reward of at least $-R(\alpha_{\tilde{t}} - \alpha_T)$ at which point

$$\alpha_{\tilde{t}} = \frac{\log\tilde{\lambda}}{\eta R}. \tag{55}$$

Consequently,

$$\alpha_{\tilde{t}+1} \leq \alpha_{\tilde{t}} = \frac{1}{\eta R}\log\Big\{(\eta+1) \cdot \frac{\Pi_{max} - \Pi_{min}}{b_{max} - \beta}\Big\}. \tag{56}$$

Thus, the total constraint violation must be bounded by

$$-\sum_{t=1}^{T}\mathbb{E}_{p_\theta}[r_t] \leq R - L\beta(1 - \frac{1}{\eta R}\log\Big\{(\eta+1) \cdot \frac{\Pi_{max} - \Pi_{min}}{b_{max} - \beta}\Big\}). \tag{57}$$

We set $\beta = R/L$ to obtain:

$$-\sum_{t=1}^{T}\mathbb{E}_{p_\theta}[r_t] \leq \frac{1}{\eta}\log\Big\{(\eta+1) \cdot \frac{\Pi_{max} - \Pi_{min}}{b_{max} - \frac{R}{L}}\Big\}. \tag{58}$$

Q.E.D. □

B Extended Results

Sampling schedule. The reported MDLM evaluations use a uniform grid of T reverse-time steps. The backbone's log-linear noise schedule gives mask probability $m(t) = (1 - \epsilon)t$ ($\epsilon = 10^{-3}$); at each step, each still-masked position is independently unmasked with probability $[m(t_k) - m(t_{k+1})]/m(t_k)$. Thus the number of newly unmasked tokens is random, not a fixed quota. Before the final greedy decode, its expectation is approximately L/T per step over L initially masked positions. The main text, molecular, and playlist evaluations use $(L, T) = (256, 256)$, $(64, 64)$, and $(50, 128)$, respectively; in playlist completion only the 30 unclamped positions are generated, giving about $30/128$ expected new tokens per step. Any positions still masked after the T stochastic steps are filled by the final greedy decode. The diagnostic plots in the appendix specify their own L and T where they differ from these evaluation settings.

B.1 Text Generation Details

Table 2: TinyStories ocean steering, N=1,000. SPDD: accumulated slack, η=2.0. LLaDA-8B transfer in Appendix B.2.

Method	Target (avg)	Pass%	PPL↓	Dist-2↑	KL↓
Unconstrained	0.3	0.5 ± 0.2%	24.7	0.187	0.000
Static bias (α=2.0)	19.2	81.9 ± 1.2%	24.8	0.169	0.706
CDD (τ=0.5)	8.1	38.0 ± 1.5%	24.6	0.318	0.221
D-CBG (γ=5)	0.1	0.0 ± 0.0%	25.2	0.336	0.127
GPT-4.1-mini	31.5	100 %	27.1	0.082	2.852
SPDD (Ours)	11.0	92.1 ± 0.9%	25.3	0.175	0.420

Table 3: Direct repetition diagnostics for N=1,000 TinyStories outputs at T=256. The baseline uses saved unconstrained samples associated with the TinyStories checkpoint; SPDD is a fresh run from that checkpoint with accumulated slack, η=2.0, and the documented sampling configuration. Repeat-4 is the mean fraction of four-gram positions recurring within each story; Adjacent is the fraction of neighboring tokens that are identical.

Method	Dist-2↑	Repeat-4↓	Adjacent↓
Unconstrained	0.1859	1.619%	0.073%
SPDD	0.1765	1.606%	0.149%

In this diagnostic, SPDD lowers pooled Dist-2 while within-story four-gram reuse stays near 1.6% for both methods. Identical adjacent tokens become more frequent but remain below 0.2% of neighboring pairs. Repeat-4 and adjacent repetition measure behaviors distinct from Dist-2; the complementary quantity $1 -$ Dist-2 would not provide an independent check.

Two simultaneous text constraints. We also require at least ten ocean tokens and five nature tokens in each story. Each topic uses 18 exact single-token words with unit scores; the two multipliers share η=1 and λ_0=0.5 and are updated from their respective accumulated slack. Table 4 reports all 100 attempted outputs per setting with T=256 and no multiplier noise. The dual setting raises joint success to 62%, while 64% of outputs meet the ocean target and 94% meet the nature target. PPL and pooled Dist-2 are descriptive at this sample size.

Table 4: Ocean (R=10) and nature (R=5) constraints on TinyStories (N=100). Pass percentages use all attempted outputs. PPL uses GPT-2; Dist-2 pools bigrams over the 100 outputs.

Method	Ocean%	Nature%	Joint%	PPL↓	Dist-2↑
Unconstrained	2	2	0	23.97	0.389
SPDD, ocean only	71	2	2	24.55	0.374
SPDD, both	64	94	62	24.32	0.369

Standard sampling controls. We tested two inference-time filters on the same $N{=}100$, $T{=}256$ TinyStories setup, with accumulated-slack SPDD at $\eta{=}1$ and no multiplier noise. The frequency penalty subtracts 0.5 times each token's count among already revealed positions from its next-step log score. Top-k keeps the 100 highest-scoring tokens at each masked position. Each filter is applied after SPDD's logit bias when guidance is enabled; neither changes the model or noise schedule. Table 5 reports the matched outcomes. Top-k raises SPDD pass rate from 66% to 82% and lowers PPL, but also lowers Dist-2 and increases repeated four-grams. The frequency penalty lowers repeated four-grams while increasing PPL substantially. These quality comparisons are descriptive at $N{=}100$.

Table 5: Standard sampling controls for the TinyStories ocean target ($R{=}10$, $N{=}100$). Repeat-4 is the within-story repeated four-gram fraction, as in Table 3.

Method	Pass%	PPL↓	Dist-2↑	Repeat-4↓
Unconstrained	2	23.97	0.389	1.585%
Unconstrained + frequency penalty	0	32.85	0.447	0.696%
Unconstrained + top-k	1	19.49	0.333	2.000%
SPDD	66	25.78	0.375	1.474%
SPDD + frequency penalty	64	35.80	0.439	0.704%
SPDD + top-k	82	20.29	0.312	2.162%

Table 6: Slack mode ablation, per-position λ, $R{=}10$, $N{=}1{,}000$ (same setup as Table 2). Ocean reported as mean $\pm$ across-output SD; Ocean$/R$ is the overshoot ratio.

Slack mode	η	Ocean	Ocean$/R$	Pass%	PPL↓	Dist-2↑	KL↓
Accumulated	0.1	6.5 ± 4.0	0.65	20.5 ± 1.3%	25.5	0.185	**0.281**
Accumulated	0.5	9.5 ± 2.6	0.95	46.9 ± 1.6%	25.6	0.181	**0.365**
Accumulated	1.0	10.5 ± 2.0	1.05	67.6 ± 1.5%	25.7	0.180	**0.402**
Accumulated	2.0	11.0 ± 1.7	1.10	92.1 ± 0.9%	25.3	0.175	**0.420**
Instantaneous	0.1	10.9 ± 5.5	1.09	55.2 ± 1.6%	24.8	0.178	0.412
Instantaneous	0.5	21.4 ± 4.4	2.14	100 %	**23.0**	0.159	0.890
Instantaneous	1.0	21.5 ± 4.3	2.15	100 %	24.5	0.161	0.916
Instantaneous	2.0	21.8 ± 4.8	2.18	100 %	24.8	0.164	0.924
Early	0.1	8.5 ± 5.5	0.85	37.0 ± 1.5%	25.2	0.182	0.342
Early	0.5	19.6 ± 4.4	1.96	99.6 ± 0.2%	23.7	0.160	0.828
Early	1.0	20.5 ± 4.5	2.05	100 %	24.0	0.161	0.879
Early	2.0	21.0 ± 4.6	2.10	100 %	24.6	0.164	0.892
Optimistic (OMD)	0.1	10.6 ± 5.4	1.06	53.1 ± 1.6%	24.8	0.176	0.399
Optimistic (OMD)	0.5	21.3 ± 4.3	2.13	100 %	23.2	0.158	0.898
Optimistic (OMD)	1.0	21.4 ± 4.6	2.14	100 %	23.7	0.161	0.930
Optimistic (OMD)	2.0	21.5 ± 4.6	2.15	100 %	24.7	0.166	0.919

Slack Mode Ablation. Slack computation strongly affects both satisfaction and how far the constrained distribution drifts from the unconstrained model. *Accumulated* stays closest to the target ($\bar{c} \approx R$) and is uniformly the lowest-KL mode at every η (KL ≤ 0.42), at the cost of slower satisfaction ramp (21% $\to$ 92% across $\eta \in [0.1, 2]$) and elevated PPL. *Instantaneous*, *early*, and *optimistic* all saturate pass rate at $\eta \geq 0.5$ but at $2{-}3\times$ higher KL ($0.83{-}0.93$): λ becomes large in the first steps (bang-bang control [Bryson and Ho, 1975]), irreversibly committing early tokens and pulling the output distribution toward a thin slice of ocean-heavy text. Within those three saturating modes, instantaneous achieves the lowest PPL at $\eta = 0.5$ (23.0), with optimistic at 23.2 and early at 23.7 PPL. At $\eta{=}0.1$, optimistic reaches 53.1% Pass at KL 0.399, compared with 55.2% at KL 0.412 for instantaneous and 37.0% at KL 0.342 for early slack.

Optimistic Mirror Descent Slack Mode. Standard mirror descent updates λ using only the current constraint violation $h_k = R - c_k$, where c_k is the current count and R is the target. Optimistic Mirror Descent (OMD) [Rakhlin and Sridharan, 2013] improves convergence by incorporating a prediction

M_k of the *next* violation, yielding the combined update:

$$h_k^{\text{opt}} = h_k + \tilde{M}_k - \tilde{M}_{k-1}, \quad \lambda_{k+1} = \lambda_0 \cdot \exp\big(\eta \cdot h_k^{\text{opt}}\big) \tag{59}$$

where $\tilde{M}_k = (1 - \bar{m}_k) \cdot M_k$ anneals the prediction by the unmasked fraction $\bar{m}_k$, and $M_k = R - (c_k + \hat{E}_k)$ with $\hat{E}_k = \sum_{\ell \in \text{masked}} \sum_j p_\theta(j \mid x_k) \cdot s_j$ being the expected future contribution from masked positions.

This is a natural fit for discrete diffusion: early in denoising, most tokens are masked and predictions are unreliable (annealed to zero, reducing to standard mirror descent); late in denoising, few tokens remain masked and predictions are highly accurate, giving near-optimal lambda adjustment. The OMD regret bound replaces $\sum_k \|g_k\|^2$ with $\sum_k \|g_k - M_k\|^2$, which is small precisely when it matters most (late steps).

Table 6 reports the OMD row alongside accumulated/instantaneous/early slack at matched η. At η=0.5, OMD and instantaneous both reach 100% Pass; OMD has PPL 23.2 and KL 0.898, compared with PPL 23.0 and KL 0.890 for instantaneous. At η=1.0, both again reach 100% Pass; OMD has PPL 23.7 versus 24.5 for instantaneous, with KL 0.930 versus 0.916. These rows show comparable operating points rather than a uniform fluency improvement.

Table 7: Guidance-target sweep, per-position λ, $\eta = 0.5$, $N = 1{,}000$, $R \in \{5, 10, 15, 20\}$. Ocean is mean $\pm$ across-output SD. Pass@10 is the fraction with at least 10 ocean tokens, evaluated at the same threshold for every guidance target; its uncertainty is binomial SE. Complements Figure 3.

R	Slack mode	Ocean	Pass@10	PPL↓	Dist-2↑	KL↓
5	Optimistic	13.7 ± 4.3	$83.1 \pm 1.2\%$	23.9	0.166	0.558
5	Instantaneous	13.9 ± 4.4	$83.7 \pm 1.2\%$	23.9	0.165	0.583
5	Accumulated	6.3 ± 2.8	$11.5 \pm 1.0\%$	25.8	0.187	**0.260**
10	Optimistic	21.3 ± 4.4	$99.9 \pm 0.1\%$	23.8	0.163	0.906
10	Instantaneous	21.2 ± 4.4	100 %	23.4	0.160	0.903
10	Accumulated	9.6 ± 2.6	$45.1 \pm 1.6\%$	25.6	0.179	**0.372**
15	Optimistic	26.9 ± 4.7	100 %	23.4	0.155	1.111
15	Instantaneous	27.5 ± 4.8	100 %	**23.2**	0.156	1.135
15	Accumulated	13.1 ± 2.5	$93.1 \pm 0.8\%$	25.6	0.176	**0.475**
20	Optimistic	32.7 ± 4.7	100 %	23.9	0.155	1.354
20	Instantaneous	32.9 ± 4.8	100 %	23.4	0.152	1.340
20	Accumulated	16.9 ± 2.4	$99.5 \pm 0.2\%$	25.8	0.172	**0.625**

Guidance-Target (R) Sweep. For every slack mode, raising the guidance target increases the mean ocean-token count, while Pass@10 rises or saturates. The difference in Pass@10 between accumulated and the other modes narrows with R: accumulated rises from 11.5% at $R = 5$ to 99.5% at $R = 20$, while optimistic and instantaneous are already above 83% at $R = 5$ and reach 100% at $R = 20$. Accumulated produces fewer ocean tokens and has the lowest KL at every R (0.26–0.63 versus 0.56–1.35 for the other modes). Pass@10 is a fixed benchmark, not the fraction meeting each changing target: for accumulated, the latter is 73.0%, 45.1%, 24.2%, and 12.0% at $R = 5, 10, 15, 20$, respectively.

KL–pass rate Pareto. Figure 4 visualises the cross-method trade-off on text from Table 1: KL divergence against the unconstrained distribution on the x-axis, Pass% on the y-axis. SPDD attains the highest Pass% (92.1%) at substantially lower KL than the next-best high-Pass baseline (Static at 81.9% Pass / KL 0.71); CDD trades much lower Pass for moderately lower KL; D-CBG cannot satisfy the sparse 18-token constraint at any tested γ. The dashed line shows the empirical Pareto frontier across these methods.

Beyond aggregate metrics, the next two paragraphs illustrate qualitatively how SPDD shapes generation, both as a step-by-step denoising trajectory and as completed sample stories.

Denoising Progression by Slack Mode. Table 8 shows the denoising progression for the TinyStories ocean-guided generation experiment in the main text. For each of three slack modes (accumulated,

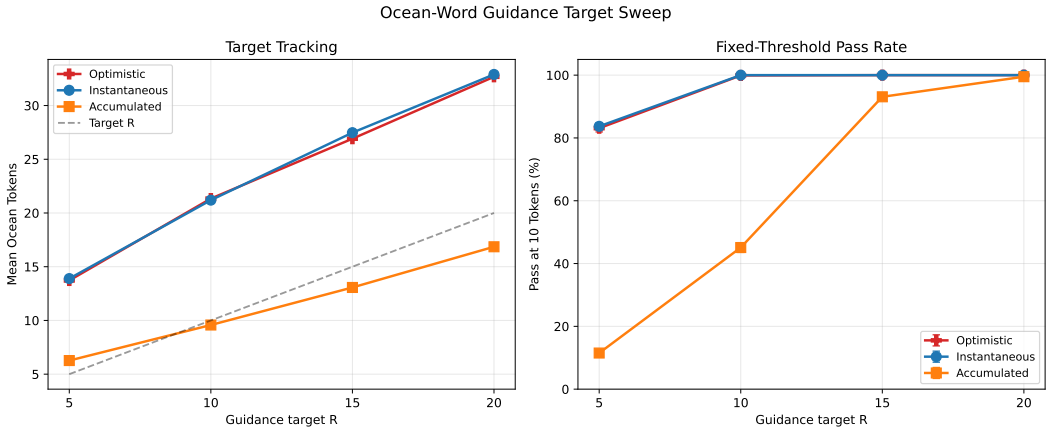

Figure 3: Guidance-target sweep ($R \in \{5, 10, 15, 20\}$, $\eta = 0.5$, $N = 1{,}000$). Mean ocean-token count increases with R for each slack mode; pass rate at the fixed 10-token benchmark rises or saturates. Error bars on Pass@10 show binomial SE.

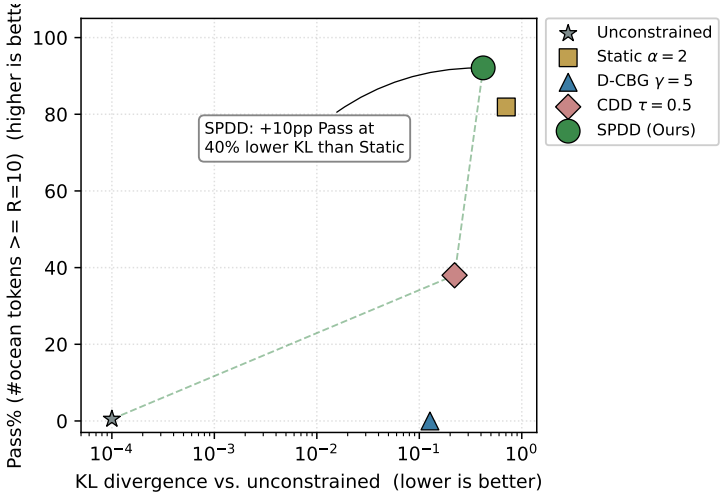

Figure 4: Text Pareto front: unigram KL vs. unconstrained against Pass% (#ocean tokens $\geq R$=10), at the operating points reported in Table 1 (N=1,000, 142M-MDLM TinyStories backbone). SPDD occupies the high-Pass, lower-KL part of the frontier: highest Pass%, with strictly lower KL than Static bias. The dashed line traces the empirical Pareto frontier (Unconstrained $\rightarrow$ CDD $\rightarrow$ SPDD).

instantaneous, optimistic), we display the partial token sequence at four stages of denoising (25%, 50%, 75%, 100% of $K = 256$ steps), using per-position λ with $\eta = 0.5$ and target $R = 10$. Underscores denote masked (uncommitted) positions; ocean words are in **bold**.

The timing of ocean token placement differs substantially across slack modes. With instantaneous slack, 11 of 21 final ocean tokens are already committed at 25% of denoising: λ becomes large in the first steps, forcing every early token to be ocean-themed. With optimistic slack, ocean tokens appear more gradually (7 at 25%, 10 at 50%), because the prediction correction $\hat{E}_k$ tempers the guidance as context builds. Accumulated slack is the slowest, with only 4 ocean tokens at 25%, reflecting its conservative target prorating.

Table 8: Denoising progression for a single sample ($\eta = 0.5$, $R = 10$). First $\sim$40 tokens shown at each stage. Underscores are masked positions; ocean words in **bold**.

Mode	Stage	Ocean	Token sequence (first $\sim$40 tokens)
Accumulated	25%	4	_ _ _ _ _ _ the **beach** near _ _ _ _ _ a big _ with _ _ _ _ _ splash _ the _ . " _ ' _ _ _ **wave** _ the _ _ _ _ _ _
	50%	9	_ and _ _ _ _ the **beach** near the **ocean** _ They _ a big _ with some seagulls _ They liked _ _ the waves splash _ the waves . " Let ' s _ _ **wave** to the **shore**
	75%	12	_ and Mia _ _ _ the **beach** near the **ocean** . They had a big truck with some seagulls . They liked to watch the waves splash _ the waves . " Let ' s _ _ **wave** to the **shore**
	100%	15	Tom and Mia were playing on the **beach** near the **ocean** . They had a big truck with some seagulls . They liked to watch the waves splash and the waves . " Let ' s send a **wave** to the **shore**
Instantaneous	25%	11	One _ _ _ and _ _ to _ _ with _ mom _ dad _ They _ _ boat _ _ **dolphin** jumped out _ _ **dolphin** was _ _ of _ _ _ _ _ _ **dolphin** _ closer
	50%	15	One _ , Ben and _ went to _ _ with _ mom _ dad _ They saw a big boat _ _ **dolphin** jumped out _ _ **dolphin** was _ _ of _ _ _ . Ben _ _ **dolphin** _ closer
	75%	19	One _ , Ben and Lily went to the **beach** with their mom _ dad _ They saw a big boat and a **dolphin** jumped out . _ **dolphin** was a _ of a _ tail . Ben _ the **dolphin** _ closer
	100%	21	One day , Ben and Lily went to the **beach** with their mom and dad . They saw a big boat and a **dolphin** jumped out . The **dolphin** was a **fish** of a shiny tail . Ben and the **dolphin** came closer
Optimistic	25%	7	_ _ and _ _ to go _ _ **beach** _ _ _ _ _ slow _ _ _ loved _ play _ _ _ _ that _ _ _ _ in _ _ _ _ _ down _ the _ _ _
	50%	10	_ _ and Mia _ to go _ _ **beach** with their _ _ _ slow turtle _ _ loved to play _ _ _ _ that they kicked _ _ in _ _ _ _ _ down _ the **shore**
	75%	15	Sam and Mia liked to go _ the **beach** with their mom _ _ slow turtle . _ loved to play with the _ **whale** that they kicked and _ in _ _ _ _ came down _ the **shore**
	100%	18	Sam and Mia liked to go to the **beach** with their mom and the slow turtle . They loved to play with the big **whale** that they kicked and splashed in . But the **whale** came down from the **shore**

Qualitative Examples. Statement 2 rests on Assumption 3: that logit increments factor as a slow drift plus near-zero-covariance Gaussian residuals. The remaining paragraphs verify this empirically on text and molecular MDLMs, and then check whether the downstream bound holds in practice.

Empirical Verification of Weak Temporal Consistency. Assumption 3 requires: (i) the logit increments $\varepsilon_{tj\ell}$ are Gaussian with bounded variance σ^2, and (ii) the cross-position covariance $\mathrm{Cov}(\varepsilon_{tj\ell}, \varepsilon_{tjk})$ is at most a small constant ρ. We test these empirically on the 142M MDLM trained on TinyStories (N=10 samples, L=256, T=128 denoising steps). Increments are computed only at positions that remain masked at both consecutive steps (once unmasked, logits are frozen and ε trivially vanishes). After decomposing $\varepsilon_{tj\ell} = \mu_{tj\ell} + \eta_{tj\ell}$ via the per-(t, j, ℓ) sample mean $\hat{\mu}_{tj\ell}$, we inspect the residual η.

Figure 5 (top) shows the raw ε distribution and Q–Q plot; Figure 5 (bottom) shows the centred residual η. The centred distribution is visibly closer to Gaussian in the bulk, confirming that a large fraction of the deviation from Gaussianity is absorbed by the drift term $\mu_{tj\ell}$. A heavy left tail remains

Table 9: Generated story excerpts (truncated to ∼80 words). Ocean words in **bold**.

Method	Story excerpt
Unconstrained (0 ocean)	Lily and Ben were playing in the backyard. They had a big bowl of water. They wanted to make yummy cakes and cookies. "Can you mix too, Lily?" Ben asked his sister. "Yes, yes, yes!" Lily said. She helped him mix some flowers and mud shapes. She made a tasty cake also with two sprinkles...
Optimistic $\eta = 0.5$ (10 ocean)	Tim and Lily are friends who like to play at the **beach**. They have a ball and a net. They like to kick the ball and chase each other. One day, they see a big dog at the **beach**. It looks mighty. Tim and Lily think it is friendly to play. They run to the dog and try to grab the ball. The dog sees Tim and Lily are afraid and **wave** their ball...
Instantaneous $\eta = 1.0$ (10 ocean)	Lily and Ben liked to play at the **beach**. They liked to kick the ball and dig holes and look for shells. One day, they saw a **dolphin** in the water. "Look, look!" Lily said, watching the **dolphin**. "It is so pretty and shiny." "Go away, Lily!" Ben said. "No, **shark**s are mean and rude"...

at large unmasking events; this is consistent with the assumption holding in expectation but admitting occasional outliers.

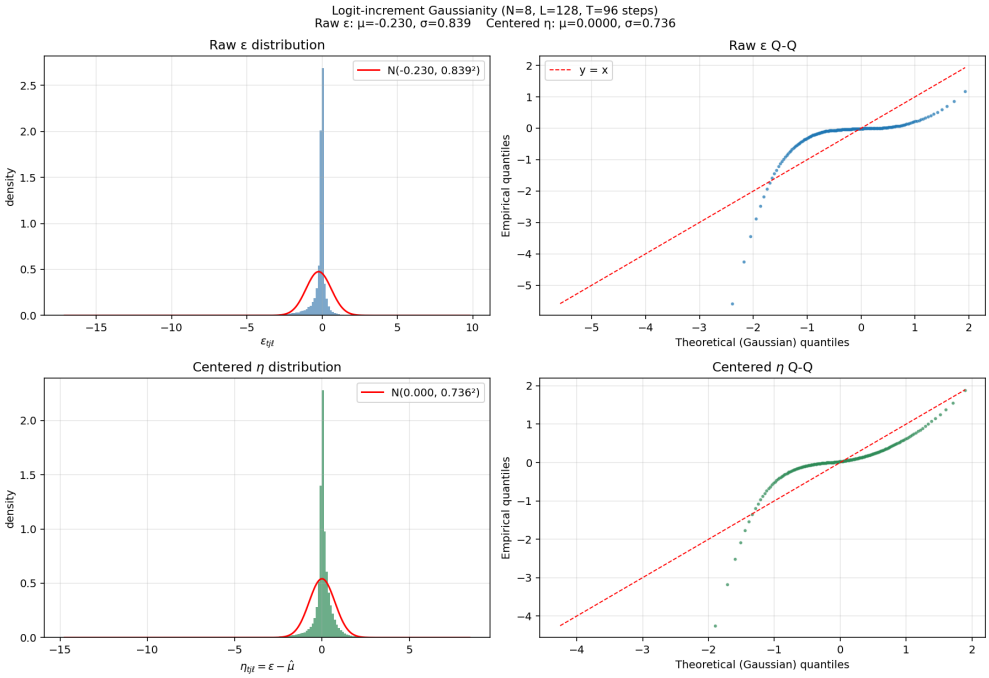

Figure 5: Logit-increment distribution on TinyStories MDLM. Top row: raw $\varepsilon_{tj\ell}$. Bottom row: residual $\eta_{tj\ell} = \varepsilon - \hat{\mu}_{tj\ell}$ after subtracting the per-(t, j, ℓ) sample mean. Centring concentrates the bulk near a Gaussian; the remaining heavy left tail corresponds to unmasking events. Overall: the factorisation required by the theory holds directionally; cross-position covariance is empirically near zero ($\bar{\rho} \approx 10^{-5}$), and the central mass of η is near-Gaussian with time-varying (but bounded) σ_t.

Across the 128 denoising steps, σ_t fluctuates between ∼0.2 and ∼1.6 with mean 0.71 (not constant, but bounded). The cross-position covariance is the strongest empirical result: for every token tested, the empirical $\bar{\rho}$ of the residual η is on the order of 10^{-5}, i.e., essentially zero relative to its diagonal variance. This is the component of the assumption that drives the per-position factorisation in Section 4, and it holds very cleanly on our model.

Logit trajectories. To visualise how individual logits evolve under the reverse process, we plot $\log p_\theta(x_t^\ell = j \mid x_{t+1})$ directly from the backbone (before the SUBS parameterisation rewrites unmasked positions to identity, which would otherwise produce spurious step-changes at unmasking events). Figure 6 shows the top-3 tokens per position across all $L = 32$ positions of a single sample, colour-coded by position. The trajectories are continuous and smoothly varying: most tokens stabilise into distinct log-probability bands by step $\sim$20–40 and then drift slowly for the remainder of the denoising process. This is precisely the picture Assumption 3 encodes: a slow per-(t, j, ℓ) drift $\mu_{tj\ell}$ plus small noise $\eta_{tj\ell}$.

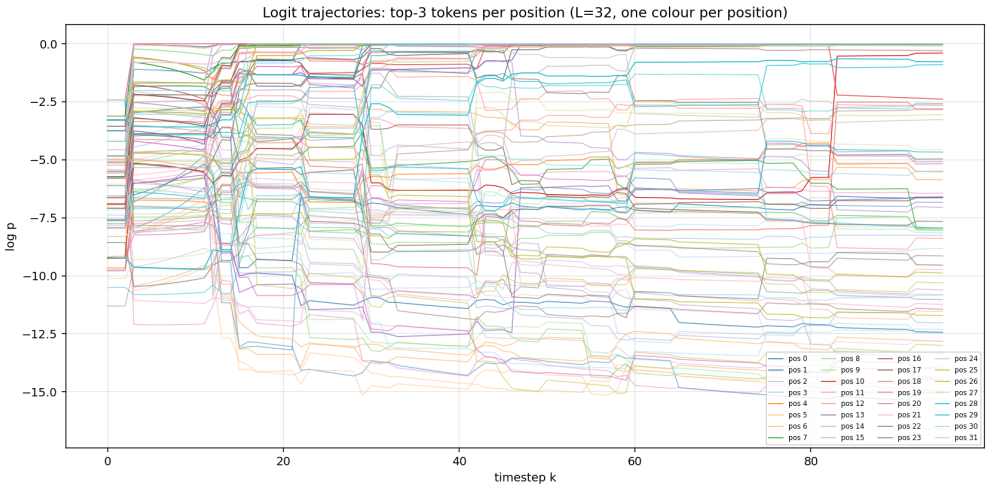

Figure 6: Raw backbone log-probabilities $\log p_\theta(x_t^\ell = j \mid x_{t+1})$ for the top-3 tokens at each of $L = 32$ positions, across $T = 96$ denoising steps (one colour per position; transparency encodes rank within the position). Logits evolve smoothly, with no sharp transitions; we plot pre-parameterisation values, bypassing the SUBS rewrite that forces unmasked positions to identity. Different positions cluster at different log-probability bands but do not co-move strongly, consistent with the low-cross-position-covariance finding.

Figure 7 complements this with a vocabulary-wide view: for eight selected positions, we show heatmaps of $\log p_\theta(x_t^\ell = j \mid x_{t+1})$ with the vocabulary sorted by final-step probability. The top tokens (top rows) brighten smoothly as t decreases; lower-rank tokens fade continuously. No sharp bands appear: the model's beliefs concentrate gradually rather than via discrete commitments at specific steps.

Taken together, Figures 6–7 illustrate that (i) logits evolve continuously without large jumps within the reverse process, (ii) different positions follow similar *shape* dynamics but with uncorrelated residuals (consistent with $\bar{\rho} \approx 0$), and (iii) the per-position factorisation used in our derivation is not merely a convenient idealisation but reflects actual behaviour of a trained MDLM on text.

Cross-domain comparison. To test whether the factorisation extends beyond text, we repeated the analysis on the SMILES MDLM (21.9M params, vocab 59, $L = 72$) used for the molecular experiments. Figure 8 shows the cross-position covariance of the centred residual η for four tokens. In contrast to the near-zero values observed on text ($\bar{\rho} \approx 10^{-5}$), molecular covariance is substantial and spatially structured: $\bar{\rho}$ ranges from 0.03 for the top-1 token up to 0.39 for sub-top tokens, with a clear banded diagonal indicating that nearby positions have correlated η. This reflects the strong local constraints of SMILES syntax (valence, ring closures, bracket pairing) which induce inter-position dependence that text does not exhibit. Kurtosis of the centred η also rises from 29 (text) to 80 (molecular), indicating substantially heavier tails.

This does not invalidate our empirical results on molecular (Table 11), since SPDD still improves constraint satisfaction over unconstrained generation at competitive KL, but it does indicate that the parametric form of Assumption 3 is an approximation, and the theoretical bound in Statement 2 is likely loose for strongly-structured discrete domains. A principled sharpening of the assumption (e.g., allowing a block-diagonal covariance reflecting local structure) is left for future work.

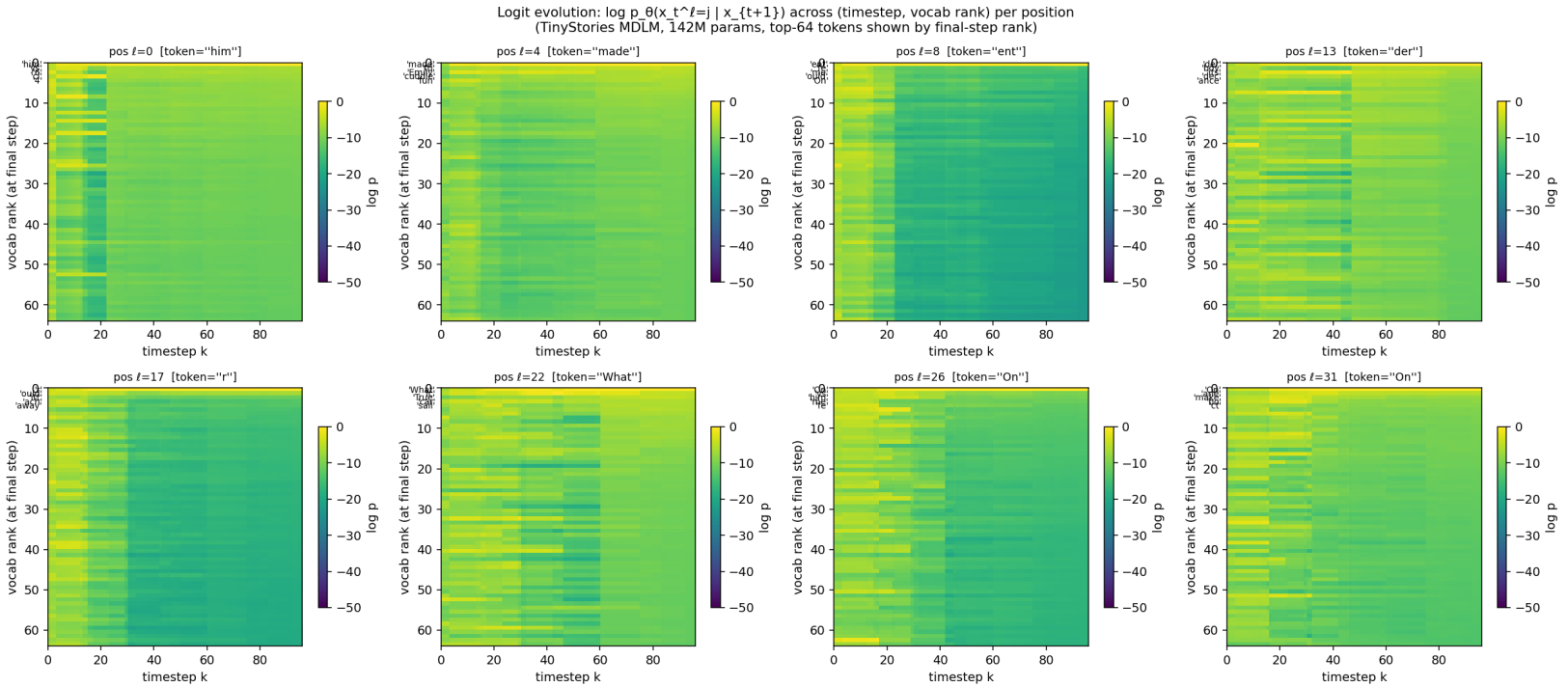

Figure 7: Vocabulary-wide logit evolution at eight selected positions. Horizontal axis: denoising step k; vertical axis: vocabulary rank by final-step log-probability, with the top-64 tokens shown. Colour denotes $\log p_\theta$ on the same $[-50, 0]$ scale in all panels. Concentration is gradual, supporting the smooth-drift picture of Assumption 3.

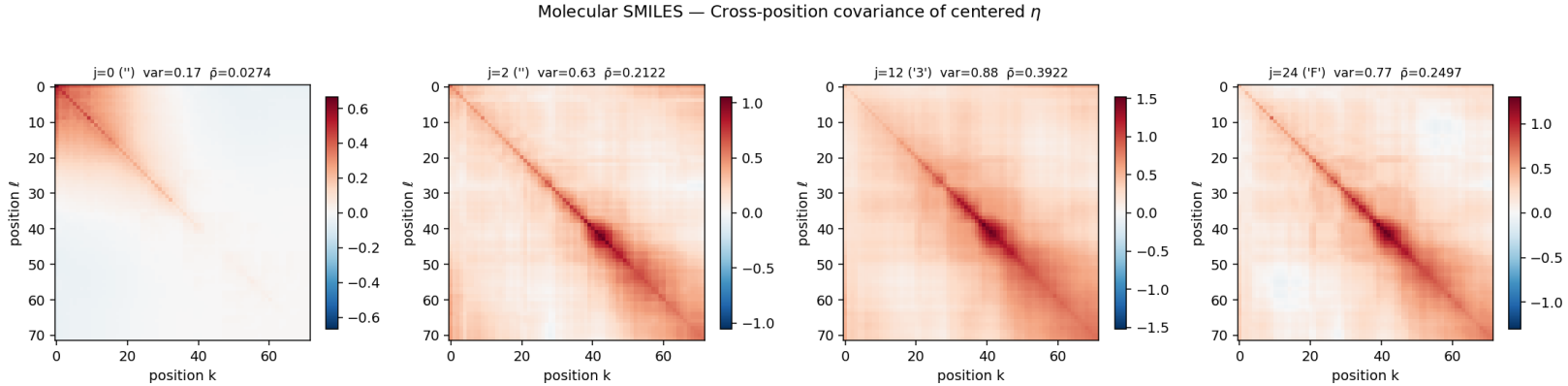

Figure 8: Molecular SMILES MDLM: cross-position covariance of centred η for four tokens. Unlike text, molecular shows strong local correlation structure ($\bar{\rho}$ up to 0.39) consistent with chemistry's valence and ring-closure constraints. The assumption's literal bound on ρ is violated in this domain, though constrained generation still succeeds empirically (Table 11), suggesting the Statement 2 bound is loose in practice for structured domains.

Does the bound actually hold in practice? The natural follow-up is whether the violations of the assumption's parametric form translate into violations of the downstream guarantee. Figure 9 compares the empirical per-step objective π_t attained by SPDD on a real ocean-steering run (LHS of Statement 2) against the theoretical lower bound (RHS), as a function of the denoising step. The bound holds empirically at **96.9% of steps**; the few small negative excursions near $t = 0$ are order -3 in magnitude compared to peak positive slack of $+60$. The bound is *loose but valid*: total empirical reward is -936 versus a lower bound of -1281 (27% slack). This matches the expected behaviour: assumption violations loosen the worst-case bound, but the empirical trajectory of a trained SPDD sampler still sits comfortably above the guarantee.

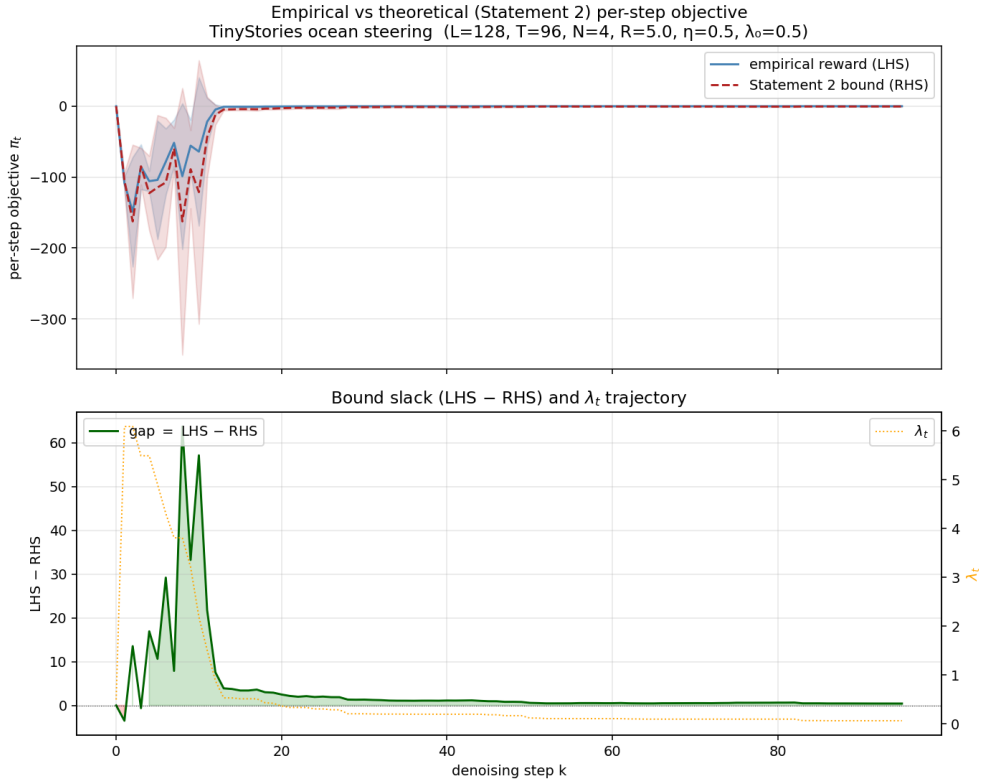

Figure 9: Per-step objective π_t under SPDD ocean steering (L=128, T=96, N=4, R=5, η=0.5, λ_0=0.5). Top: empirical reward (blue) and Statement 2 lower bound (red, dashed) over denoising steps. Bottom: slack (green = bound holds; red = bound violated) and λ_t trajectory (orange). The bound holds at 96.9% of steps; total empirical reward exceeds the bound by 27%. The dynamics concentrate in the first $\sim$15 steps; after constraint satisfaction, $\lambda_t \approx 0$ and both sides collapse to zero.

Taken together, the analyses in this subsection make three claims at different levels of strength. (i) The cross-position factorisation holds cleanly on text-like domains ($\bar{\rho} \approx 10^{-5}$) and is substantially weaker on structured domains like molecular ($\bar{\rho}$ up to 0.39). (ii) The Gaussian / bounded-σ^2 parametric form is an approximation: centred residuals have kurtosis 29 on text and 80 on molecular, and σ_t varies $8\times$ across denoising steps. (iii) Despite (i)–(ii), the Statement 2 bound is empirically valid on actual SPDD runs at >96% of denoising steps, with substantial slack. The theoretical guarantee is thus robust to the assumption-violations we observe.

B.2 Zero-Shot Transfer to LLaDA-8B

To test whether SPDD transfers to a substantially larger backbone without modification, we apply the same primal-dual update at decode time to LLaDA-8B [Nie et al., 2025], an absorbing-state diffusion model trained at the 8B scale. No fine-tuning or auxiliary training is performed; only the constraint-dependent logit bias is added at each denoising step. Table 10 reports the ocean-count target (R=10, N=1,000, sampling temperature 1.0). For this tokenizer, 22 of the 30 candidate ocean

words are single tokens and are used for the count and score. LLaDA uses a fixed story prefix and generates 128 tokens in 128 denoising steps.

Table 10: LLaDA-8B ocean-count steering (R=10, N=1,000, temperature 1.0). KL is unigram divergence from the unconstrained LLaDA-8B samples. PPL uses GPT-2, and Dist-2 pools bigrams across the 1,000 outputs.

Method	Target (avg)	Pass%	PPL↓	Dist-2↑	KL↓
Unconstrained LLaDA-8B	0.19	0.0%	8.74	0.183	—
SPDD on LLaDA-8B	9.71	41.7%	9.19	0.144	0.542

The same per-token scoring interface raises the LLaDA-8B pass rate from 0.0% to 41.7% without fine-tuning. The mean count remains just below the ten-token target, so this setting does not reliably satisfy the constraint. Guidance also increases GPT-2 PPL from 8.74 to 9.19 and reduces pooled Dist-2 from 0.183 to 0.144. These measurements show partial transfer together with a quality cost under this sampling protocol; standard repetition controls are a possible direction for improving that trade-off.

B.3 Molecular Generation Details

Table 11 reports the full MW $\geq$ 350 sweep that underlies the molecular column of Table 1. Each method is reported at a representative operating point; Static has three α values to expose its frontier directly. The SPDD point has 49.9% validity, so its satisfaction gain comes with a substantial validity cost. KL is the unigram KL between each method's samples and the unconstrained samples from the same model. Pass% (column "Above 350") is the fraction of generated samples whose RDKit-parsed heavy-atom MW is at or above the threshold.

Table 11: Molecular MW $\geq$ 350 guidance, N=1,000, 21.9M SMILES backbone. SPDD: accumulated slack, η=1.0, λ_0=0.1, per-position bias (best of a 7-config sweep).

Method	Valid	QED	MW (mean)	Above 350	KL↓
Unconstrained	59.9%	0.648	330	29.40%	—
Static (α=1)	60.9%	0.647	338	31.10%	0.005
Static (α=2)	59.1%	0.649	348	33.10%	0.008
Static (α=5)	59.6%	0.598	382	41.40%	0.019
D-CBG (γ=2)	57.4%	0.606	356	35.30%	0.014
D-CBG (γ=5)	54.8%	0.544	412	42.80%	0.048
CDD (τ=0.3, o=5, i=10)	57.9%	0.617	372	38.90%	0.019
CDD (τ=0.5, o=5, i=10)	59.7%	0.624	382	42.20%	0.025
SPDD (Ours)	49.9%	0.504	473	48.50%	0.096

Compute-matched rejection baseline. To make the best-of-N comparison explicit, we ran a separate matched SMILES experiment (N=1,000 outputs, sequence length 72, T=64). We generated 4,000 unconstrained candidates and grouped them by original sample index into 1,000 disjoint groups of four. For each N, the first N candidates in each group form one output: choose the valid molecule with MW $\geq$ 350 closest to the threshold if available, otherwise the valid molecule with highest MW, otherwise an invalid string. Empty decoded strings remain invalid attempted outputs. We separately generated 1,000 SPDD outputs with accumulated slack, η=1, λ_0=0.1, and per-position bias. All rows use the same checkpoint, denoising steps, and raw-t denoiser conditioning with the model's original noise transition schedule.

Best-of-4 has the highest success probability per selected output and retains the highest validity and QED; it uses four generations per output. SPDD produces about three times as many passing molecules per generation-cost unit as best-of-4 at these settings, with lower validity and QED. The selected-output success probability must therefore be distinguished from compute-normalized yield. For the 16.2% single-draw success rate measured here, independent best-of-4 selection gives $1-(1-0.162)^4 \approx 50.7\%$ in expectation, close to the observed 49.2%.

Table 12: Selected-output MW $\geq$ 350 success and quality in the matched SMILES experiment. Success% and Valid% use all 1,000 attempted outputs per row; QED averages valid outputs. Yield is successful outputs per 1,000 unconstrained-generation cost units; KL is unigram divergence from the 4,000-draw unconstrained pool. SPDD cost $1.03\times$ uses the relative inference-cost estimate reported in Table 1; wall-clock time varies with hardware contention.

Method	Success%	Valid%	QED	KL	Cost×	Yield / 1k
Unconstrained (one draw)	16.2	53.0	0.624	0.0003	1.00	162
Best-of-2	29.1	78.9	0.632	0.0018	2.00	146
Best-of-4	49.2	96.2	0.638	0.0030	4.00	123
SPDD	38.0	39.5	0.471	0.124	1.03	369

Multi-constraint composition. A key structural advantage of the primal-dual formulation is composition of multiple soft above-target objectives: each constraint has its own multiplier λ_i, updated from its own slack, and Equation 14 sums their logit biases. The text dual-constraint experiment uses the same η and λ_0 for both binary token-score vectors. In the molecular experiments of Tables 13 and 14, the reported η and λ_0 are shared *base* values; the MW objective uses them directly, while the binary #(N+O) objective uses both multiplied by $\max_j b_{\mathrm{MW}}(j)$. This deterministic rescaling balances the unequal raw score ranges; the per-objective settings are not tuned independently. We compare against three classes of multi-constraint baselines: *Static bias* adds a scalar over the sum of the two per-token score vectors; *D-CBG (joint)* trains a single classifier on the joint binary label $\big[\mathrm{MW} \geq T_{\mathrm{MW}} \wedge \#(N+O) \geq T_{N+O}\big]$; *D-CBG (MW only)* and *CDD* optimise the MW objective alone (the latter has no native multi-constraint extension).

Table 13: Dual above-target guidance: MW $\geq$ 350 AND #(N+O) $\geq$ 5, N=1,000, 21.9M SMILES backbone. Joint Pass% requires both constraints. SPDD: best of an 11-config sweep over $\eta \in \{0.1, 0.2, 0.5, 1.0, 2.0, 5.0\}$ and $\lambda_0 \in \{0.05, 0.1, 0.5\}$, per-objective rescaled by score range; the picked config is η=0.2, λ_0=0.05, accumulated slack, per-position bias.

Method	Valid	Joint Pass%	MW Pass%	#(N+O) Pass%	mean #(N+O)	KL↓
Unconstrained	61.3%	20.2%	21.4%	40.3%	5.42	0.005
Static (α=2)	54.0%	27.8%	27.8%	48.7%	8.98	0.038
D-CBG (γ=5, MW only)	55.2%	36.0%	38.3%	40.9%	5.91	0.038
D-CBG (γ=5, joint)	53.4%	36.5%	39.0%	39.2%	5.96	0.041
CDD (τ=0.5, MW only)	56.1%	34.2%	36.4%	46.7%	6.49	0.033
SPDD (Ours)	38.7%	38.1%	38.1%	38.7%	7.32	0.073

Table 14: Dual above-target guidance with the tighter #(N+O) $\geq$ 8 second threshold (same MW threshold), N=1,000, 21.9M SMILES backbone. The tight second threshold breaks the natural MW↔#(N+O) correlation: MW-only baselines plateau well below SPDD, and only the *joint-label* D-CBG variant or Static α=2 remain competitive. SPDD: best of the same 11-config sweep (η=0.1, λ_0=0.05, accumulated, per-position).

Method	Valid	Joint Pass%	MW Pass%	#(N+O) Pass%	mean #(N+O)	KL↓
Unconstrained	61.0%	10.4%	21.8%	13.0%	5.41	0.004
Static (α=2)	54.9%	27.0%	28.7%	39.9%	8.83	0.035
D-CBG (γ=5, MW only)	52.8%	17.1%	40.3%	17.2%	6.16	0.044
D-CBG (γ=5, joint)	53.7%	25.1%	34.3%	27.9%	7.12	0.029
CDD (τ=0.5, MW only)	56.7%	16.8%	35.7%	18.8%	6.44	0.030
SPDD (Ours)	35.9%	34.4%	35.4%	34.9%	8.83	0.070

At the looser #(N+O) $\geq$ 5 threshold (Table 13) the natural correlation between MW and #(N+O) in ZINC20 lets MW-only baselines satisfy both constraints passively; SPDD reaches the highest joint Pass% (38.1%) by a small $\sim$ 1.5-point margin over the strongest baseline (D-CBG joint at 36.5%), at the cost of lower validity (38.7% vs. 53.4%) since SPDD pushes the distribution further into both constraint regions.

The tighter #(N+O) ≥ 8 regime (Table 14) breaks the natural correlation. The MW-only baselines that ignored the heteroatom signal collapse to $\sim 17\%$ joint Pass%, while SPDD continues to lift both axes (joint 34.4%, mean #(N+O) ≈ 8.8). The *joint-label* D-CBG variant, given the conjunction as its training signal, recovers some of the gap (joint 25.1% at higher validity than SPDD), making it the strongest of the multi-constraint baselines, but SPDD still wins joint Pass% by ~ 9 points. The validity penalty for SPDD is real: pushing both axes simultaneously strains chemical syntax. Lowering η trades joint Pass% for KL and validity along the same controllable frontier as the single-constraint case; even at the smallest η in our sweep (η=0.1), SPDD's KL (~ 0.07) sits roughly $2\times$ above the baselines' KL (0.03–0.04), reflecting the cost of steering both axes through the model's distribution rather than only one.

The structural advantage of SPDD over D-CBG-joint is composability rather than absolute Pass%: SPDD adds one Lagrangian multiplier per constraint and uses the existing per-token scoring functions; D-CBG-joint instead requires re-training a classifier each time the constraint set changes, which does not scale gracefully with the number of constraints. CDD has no native joint extension and remains a single-constraint baseline. The deterministic score-range rescaling is needed because atomic-mass scores ($\sim$ 12–127) and binary heteroatom scores have different scales; using identical numerical multipliers would let the MW bias dominate. Automatic score normalization and tuning across less-correlated objectives (e.g., aromatic-ring counts) are future work.

Three simultaneous molecular constraints. Using the SMILES setup from Table 12, we additionally target MW ≥ 300, #(N+O) ≥ 5, and at least one halogen (N=200, T=256). Each target has its own multiplier, updated from its respective slack. The MW objective uses base η and λ_0; the two binary objectives use those same base values multiplied by the maximum MW token score (126.9), as in the dual-objective setup above. All 200 attempted strings, including empty invalid decodes, remain in the denominator for validity and joint success. Table 15 shows that three-objective guidance increases unconditional joint success from 9.5% to 33.0%, while validity falls from 63.5% to 33.0% and QED among valid molecules falls from 0.644 to 0.450. All 66 valid outputs from the accumulated η=1 triple setting satisfy all three targets; this conditional 100% should be read together with its 33.0% validity.

Table 15: MW ≥ 300, #(N+O) ≥ 5, and halogen ≥ 1 in the SMILES diagnostic. Joint% and Valid% use all N=200 attempted outputs; QED averages valid outputs. The dual row steers MW and N+O, while halogen remains an unsteered evaluation target.

Method	Joint / 200	Joint%	Valid%	QED
Unconstrained	19	9.5	63.5	0.644
SPDD dual, accumulated η=1	50	25.0	44.0	0.537
SPDD triple, accumulated η=1	66	33.0	33.0	0.450
SPDD triple, accumulated η=2	64	32.0	33.0	0.452
SPDD triple, instantaneous η=1	6	3.0	3.0	0.203

Structural Constraints. Table 16 evaluates soft multi-token subsequence guidance using **SubsequenceScorer**, which biases toward generating a target subsequence (benzene ring `c1ccccc1`) at an unknown position via context-aware per-position bias scores, unlike hard inpainting which requires known positions. SPDD uses η=2.0, λ_0=0.5, instantaneous slack.

Table 16: Soft substructure guidance via SubsequenceScorer (N=1,000, SMILES, 21.9M backbone). Constraint satisfaction measured as fraction of generated SMILES containing the target subsequence (benzene ring `c1ccccc1`).

Method	Validity	Benzene rate	Cost
Unconstrained	64.4%	29.9%	1×
SPDD SubsequenceScorer	56.9%	40.3%	1×

SPDD lifts benzene containment by 10 points (29.9%→40.3%, $1.35\times$ lift) with modest validity degradation (64.4%→56.9%), extending the framework from single-token constraints to multi-token structural motifs at unknown positions.

MW validity trade-off and qualitative examples. Sweeping η and the slack mode traces a Pareto frontier in Pass% versus chemical validity: weaker guidance keeps validity near the unconstrained level but undershoots the threshold, while more aggressive guidance lifts Pass% but degrades validity through malformed SMILES. Table 11 summarises the operating points along this frontier; the corresponding sweep configurations are saved alongside the released results. Figure 10 shows the qualitative effect on individual molecules: row (i) spans the natural MW distribution of ZINC20, row (ii) shifts mass into the upper tail while preserving drug-like scaffolds, and row (iii) additionally enforces nitrogen/oxygen richness without visibly collapsing structural diversity.

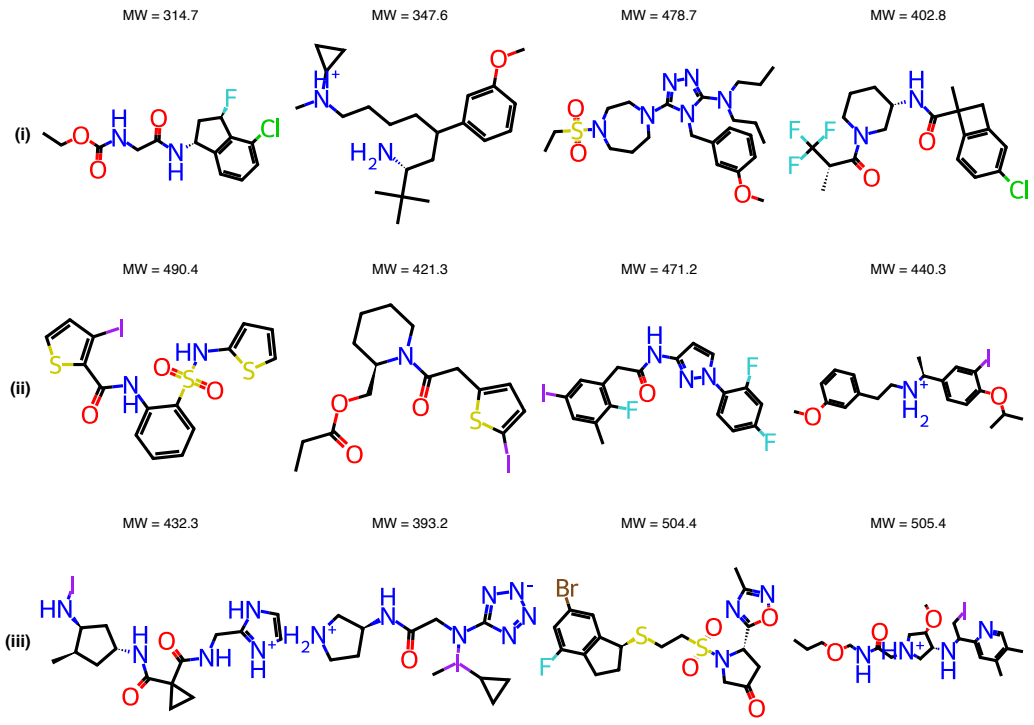

Figure 10: Example molecules from the 21.9M SMILES backbone. **(i)** Unconstrained generation. **(ii)** SPDD with MW ≥ 350 Da. **(iii)** SPDD with the dual constraint MW ≥ 350 Da and #(N+O) ≥ 5. Molecular weights are annotated above each structure.

Comparison with Best-of-N Baseline From Table 1 we see that the pass rate of the Unconstraned baseline in the molecular domain is 29.4%. This suggests that the running this baseline, for example, four times would allow us to reach $P(\text{at least 1 passing}) \approx 75\%$. While this significantly increases the validity of the generated sequences, this does not change the the per-sample pass rate: Generating N samples and keeping the best would still result in the effective yield around 29.4% of total compute.

Table 17 gives a cost-matched comparison using the numbers from Table 1, including KL divergence. It is important to note here that the positive KL divergence (0.096) incurred by SPDD does not result in a large degradation in sample quality. In particular, Table 11 shows that SPDD maintains validity (59.9%), QED (0.648 vs. 0.648 for the unconstrained model), and mean logP within the training distribution. Moreover, the KL divergence is small and controllable through the η parameter.

Critically, best-of-N cannot efficiently compose multiple constraints (e.g., molecular weight *and* heteroatom count) without exponential growth in computational cost, whereas SPDD handles constraint composition natively using additive per-constraint biases.

Table 17: Cost-matched comparison in the molecular domain.

Method	Pass (%)	KL	Cost	Passes per 1,000 cost-units
Unconstrained	29.4	—	1.00×	294
Best-of-4	29.4	0.000	4.00×	74
Static bias	41.4	0.019	1.03×	402
D-CBG	42.8	0.035	1.09×	393
CDD	42.2	0.034	7.28×	58
SPDD (Ours)	**48.5**	0.096	1.03×	**471**

Table 18: Playlist subgenre steering, 5-subgenre mean, N=1,000 sessions per subgenre. NDCG/Recall reported for completeness; see §5 for why they are not the primary metric.

Method	NDCG@30	Recall@30	Sat.%↑	Div.↑	IDist.↑	KL↓
Unconstrained	0.378	0.372	1.1%	0.497	0.987	0.000
Static (α=2.0)	0.504	0.372	1.7%	0.494	0.987	0.006
D-CBG (γ=5)	0.488	0.354	1.9%	0.481	0.979	0.118
CDD (τ=0.5)	0.439	0.299	9.2%	0.368	0.986	0.782
SPDD (Ours, η=0.2, frontload)	0.306	0.357	9.3%	0.513	0.969	0.355

B.4 Playlist Generation Details

Setup Details. Each token inherits genre metadata from its member tracks. For a target genre g, we compute per-token scores as the fraction of member tracks tagged with g (exact match):

$$b_g(j) = \frac{|\{t \in \text{members}(j) : g \in \text{genres}(t)\}|}{|\text{members}(j)|} \tag{60}$$

We target specific subgenres rather than broad categories. Because tracks carry multiple genre tags and LSH clusters mix tags, a single statistic does not capture how the constraint signal is distributed across the vocabulary. We therefore report two metrics per genre (Table 19): the *breadth* of the signal (the fraction of tokens with $b_g(j)>0$), and its *concentration* (the mean of $b_g(j)$ across the vocabulary, plus the fraction of tokens with $b_g(j)\geq 0.5$, i.e., clusters whose majority is tagged g). The two metrics decouple sharply: 19% of tokens carry *some* k-pop signal, but only 0.5% of tokens are majority-k-pop, and the average per-token score is 0.019 (i.e., a typical cluster has ∼1–2 k-pop tracks out of ∼50). Sparse genres are sparse on *both* axes: thrash metal has 2.4% breadth and 0.0022 mean. The breadth percentages overlap across genres and need not sum to $\leq 100\%$, but the concentration metrics do not have this property and remain meaningful as absolute discretisation statistics. Breadth captures how many tokens the constraint touches at all; concentration captures how strongly each cluster is dominated by the target subgenre. The two metrics decouple sharply because LSH clusters mix tags.

Table 19: Per-genre coverage statistics on the 3,887-token playlist vocabulary. *Breadth* ($b_g>0$) measures how widely the constraint signal is spread; *Concentration* (mean b_g, fraction of tokens with $b_g\geq 0.5$, max b_g) measures how strongly any single token leans toward the genre. The signal is broadly diffuse ($\geq 80\%$ of tokens have $b_g=0$ on every genre, and only 0.1–0.5% of tokens are majority-genre), reflecting the multi-tag nature of editorial track metadata under LSH clustering of audio embeddings.

Genre	$\lvert b_g>0\rvert$	$\mathbb{E}\, b_g$	$\lvert b_g\geq 0.1\rvert$	$\lvert b_g\geq 0.5\rvert$	$\max b_g$
reggaeton	18.9%	0.0169	5.7%	0.3%	1.00
k-pop	18.7%	0.0185	5.8%	0.5%	1.00
bossa nova	9.8%	0.0029	0.6%	0.1%	0.50
synthwave	5.3%	0.0023	0.6%	0.1%	0.85
thrash metal	2.4%	0.0022	0.6%	0.1%	1.00

We report *bucket-level* and *track-level* genre satisfaction (the latter after nearest-centroid track selection), embedding coherence, token diversity, and inter-playlist Jaccard distance.

Additional Subgenre Results. Table 20 reports appendix exact-match runs with an expanded metric set for k-pop (18.7% token coverage) and synthwave (5.3% coverage).

Table 20: Additional playlist subgenre results (N=200, exact match). Bucket and Track values are percentages; Coher. is embedding coherence, Div. is token diversity, and IDist. is inter-playlist Jaccard diversity.

Genre	Method	Bucket (%)↑	Track (%)↑	Coher.	Div.	IDist.
k-pop	Unconstrained	51.0	$5.2_{\pm 1.1}$	0.678	0.475	0.977
	Static (α=5)	67.1	$20.7_{\pm 3.1}$	0.691	0.452	0.967
	D-CBG (γ=5)	94.0	6.1	0.814	0.317	0.861
	CDD (τ=0.5)	97.8	50.3	0.930	0.090	0.910
	SPDD (η=5)	79.6	$36.7_{\pm 3.1}$	0.716	0.380	0.925
synthwave	Unconstrained	36.0	$1.0_{\pm 0.4}$	0.678	0.475	0.977
	Static (α=5)	48.5	$3.4_{\pm 0.8}$	0.672	0.387	0.971
	D-CBG (γ=5)	93.4	7.8	0.806	0.218	0.879
	CDD (τ=0.5)	100.0	24.8	0.998	0.021	0.028
	SPDD (η=5)	73.6	$20.9_{\pm 2.9}$	0.619	0.329	0.952

Requires training a separate classifier per genre (2× cost for D-CBG, 5–20× for CDD).

These appendix rows expose the same satisfaction–diversity tension with a wider metric set. On synthwave, CDD achieves 100% bucket satisfaction but diversity drops to 0.02 with inter-playlist Jaccard distance of 0.03: all playlists are nearly identical. SPDD reaches lower track satisfaction but maintains substantially higher diversity and inter-playlist distance.

Slack modes and λ dynamics. The slack-mode pattern observed on text generalises cleanly to playlists (Table 22). Instantaneous slack maximises track satisfaction (81.7% at η=5.0) but compresses diversity (0.33→0.21 as η rises); accumulated slack sacrifices peak satisfaction for monotonic controllability (7.6%→46.8%) while preserving diversity ($\sim$0.46); optimistic (OMD) slack tracks instantaneous at high η (63.4% at η=0.2, 81.0% at η=5.0). On the harder thrash-metal setting (2.4% coverage) the same ordering holds (instantaneous 24.2%, early 19.0%, optimistic 14.7%, accumulated 17.1% at η=5), confirming the pattern is genre-independent. Per-step dynamics (Figure 11) explain the trade-off: instantaneous slack produces a large early multiplier, committing all target tokens in the first $\sim$20 denoising steps and leaving no room for diversity, whereas optimistic slack rises gradually as the model's forecast $\hat{E}_k$ absorbs context.

Front-loaded positional bias. Table 21 reports a full η sweep for front-loaded positional bias across all five paper subgenres (N=1,000 held-out sessions per genre, mean across genres). The pattern is consistent across η: front-loading concentrates the genre signal at the top of the playlist (TrGNDCG 0.77–0.82 vs. uniform's 0.54–0.67, a $\sim$25pp lift) and slightly improves Recall@30 at η=0.5. The cost is concentrated at NDCG@5: the front-loaded variant places the catalogue's most-genre tokens at positions 1–5, which often differ from the held-out user's specific tracks (NDCG@5 drops from $\sim$0.10–0.19 to 0.03–0.05). NDCG@30 is somewhat lower at η=0.5 (0.252 vs. uniform's 0.295) but the gap narrows as η grows. The variant is therefore the right knob when genre purity at the playlist head matters more than personalised matching to a specific user's listening history (themed/radio playlists, mood-based generation).

Further observations. Embedding coherence follows a U-shape with increasing η: it dips at moderate guidance (0.65 at η=0.5) as the model explores genre-adjacent buckets, then recovers at strong guidance (0.73 at η=5) as it concentrates on high-scoring buckets; for bossa nova (9% token coverage), diversity drops to 0.16 at η=2, reflecting the tight token pool. Per-position λ outperforms scalar λ by +6–14% relative (at η=1.0, 31.2% vs. 27.3%; at η=5.0, 34.1% vs. 32.2%); the advantage is smaller than on text because playlist sequences are shorter (L=50 vs. L=256). Artist diversity degrades moderately under strong guidance: unconstrained playlists average 43.6% unique artists; SPDD (η=5) drops to 34.2% for reggaeton and 30.5% for thrash metal, reflecting the narrower candidate pool.

Table 21: Front-loaded vs. uniform SPDD bias, η sweep, mean across reggaeton, k-pop, synthwave, bossa nova, and thrash metal (N=1,000 held-out sessions per genre). *TrGNDCG* (track-level genre NDCG) measures whether target-genre tracks land at high-ranked positions. NDCG@5/@10/@30 measure recommendation relevance against the held-out user continuation at three cutoffs; Sat.% is bucket-level target-genre satisfaction.

Variant	η	NDCG@5↑	NDCG@10↑	NDCG@30↑	Recall@30↑	Sat.%↑	TrGNDCG↑
SPDD uniform	0.5	0.189	0.235	**0.295**	0.283	67.0%	0.543
SPDD uniform	1.0	0.134	0.165	0.194	0.230	76.4%	0.605
SPDD uniform	2.0	0.117	0.138	0.162	0.212	78.0%	0.632
SPDD uniform	5.0	0.098	0.117	0.141	0.203	77.9%	0.669
SPDD frontload	0.5	0.052	0.093	0.252	**0.296**	65.0%	**0.823**
SPDD frontload	1.0	0.025	0.054	0.152	0.230	75.4%	0.815
SPDD frontload	2.0	0.040	0.065	0.128	0.212	77.4%	0.795
SPDD frontload	5.0	0.054	0.078	0.125	0.203	77.7%	0.765

Table 22: Slack mode ablation for playlist generation (reggaeton, per-position λ, N=200).

Slack mode	η	Track (%)↑	Div.
Accumulated	0.2	7.6	0.476
Accumulated	1.0	39.5	0.447
Accumulated	5.0	46.8	0.455
Instantaneous	0.2	65.8	0.331
Instantaneous	1.0	80.7	0.210
Instantaneous	5.0	81.7	0.206
Early	0.2	51.9	0.392
Early	1.0	75.6	0.256
Early	5.0	76.8	0.238
Optimistic (OMD)	0.2	63.4	0.346
Optimistic (OMD)	1.0	79.8	0.217
Optimistic (OMD)	5.0	81.0	0.206

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract claims inference-time constrained generation at $1\times$ cost with formal bounds and cross-domain generality. These are supported by Statement 1 (theoretical bound), Algorithm 1 (method), and experiments across three domains (Section 5).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

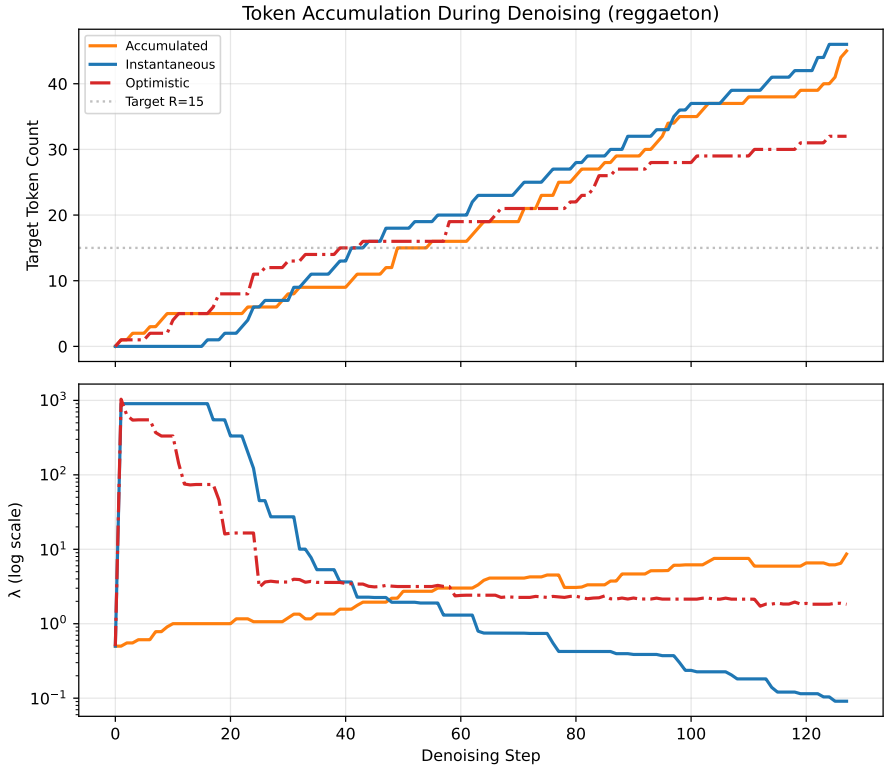

Figure 11: Per-step dynamics during playlist generation (reggaeton, η=0.5, single sample). Top: target token accumulation. Bottom: λ trajectory (log scale). Instantaneous slack produces a large early multiplier, front-loading all target tokens in the first $\sim$20 steps. Optimistic (OMD) slack rises gradually as the model's forecast $\hat{E}_k$ absorbs context.

Answer: [Yes]

Justification: The Limitations paragraph in the Discussion section addresses irrevocable token commitments in absorbing-state diffusion, noisy slack estimation for non-decomposable constraints, and the lack of hard satisfaction guarantees.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Statement 1 provides a constraint violation bound under explicitly stated assumptions (Assumptions 1 and 2). The full proof is in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Model architectures, hyperparameters (η, λ_0, slack modes), training details, and dataset descriptions are provided in the experiment sections and appendix. Algorithm 1 fully specifies the method.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The text (TinyStories) and molecular (ZINC20) datasets are publicly available. The playlist dataset is proprietary.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details (optimizer, learning rate, epochs, batch size, GPU type) are provided in each experiment section. Hyperparameter choices (η, λ_0) are reported in table captions and the appendix includes ablation studies.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Playlist and text experiments report standard errors ($\pm$ SE) computed across per-sample satisfaction. Statistical significance is reported via Mann-Whitney U tests ($p < 10^{-20}$) for playlist comparisons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were run on RTX Pro 6000 GPUs with DDP. The method runs at $1\times$ cost (same as unconstrained sampling). Baseline costs ($2\times$ for D-CBG, 5–$20\times$ for CDD) are reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available datasets (TinyStories, ZINC20) and a proprietary playlist dataset with appropriate internal approvals. No human subjects are involved.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is a foundational method for constrained discrete generation. The primary applications (molecular design, text generation, playlist curation) do not raise direct negative societal concerns beyond those inherent in generative models generally.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The method is an inference-time decoding strategy, not a new pretrained model. The models used are small (21M–142M parameters) and domain-specific, posing no meaningful misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: TinyStories Eldan and Li [2023] and ZINC20 Irwin et al. [2020] are cited with their original papers. Baseline methods (D-CBG Nisonoff et al. [2025], CDD Cardei et al. [2025]) are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets or pretrained models. The method is fully described in Algorithm 1.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects research was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects research was conducted.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: GPT-4.1-mini is used as a baseline for comparison in the text and molecular experiments (Tables 2 and 11). It is not part of the proposed method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.