A2D2: FINETUNING ANY-LENGTH DISCRETE DIFFUSION FOR ADAPTIVE DECODING

Sophia Tang,[1] **Yuchen Zhu,**[2] **Molei Tao,**[2] **Pranam Chatterjee**[1,3]

[1]Department of Computer and Information Science, University of Pennsylvania
[2]School of Mathematics, Georgia Institute of Technology
[3]Department of Bioengineering, University of Pennsylvania

{sophtang, pranam}@seas.upenn.edu, {yzhu738, mtao}@gatech.edu

ABSTRACT

Masked discrete diffusion models (MDMs) offer a simple and stable likelihood-based framework for sequence generation and have recently been extended to any-length settings via token insertion. However, principled reward-guided fine-tuning for any-length discrete diffusion remains largely unexplored. We introduce Finetuning **A**ny-Length **D**iscrete **D**iffusion for **A**daptive Decoding (**A2D2**), a unified framework for reward-guided fine-tuning of any-length MDMs. A2D2 formulates generation as a controlled continuous-time Markov chain and jointly optimizes insertion and unmasking policies to learn a reward-tilted path measure without requiring target samples. We derive the Radon–Nikodym derivative for the joint insertion–unmasking process and introduce the Adaptive Joint Decoding (AJD) loss, which provably minimizes trajectory-induced error while preserving the target distribution. Empirically, A2D2 improves reward optimization, generation accuracy, and flexibility over prior fixed-length and inference-time guidance methods.

1 INTRODUCTION

Discrete diffusion models (Lou et al., 2023; Austin et al., 2021) have emerged as a leading paradigm for sequence generation, addressing key limitations of autoregressive models by enabling bidirectional context dependencies, any-order generation (Shi et al., 2024), flexible reward guidance (Schiff et al., 2024; Nisonoff et al., 2024; Tang et al., 2025a), and parallel decoding (Christopher et al., 2025; Ren et al., 2025). These properties have led to strong empirical performance across diverse domains, including biological sequence design (Wang et al., 2024b; Gruver et al., 2023; Tang et al., 2025a), reasoning (Ye et al., 2024; Zhu et al., 2025b), and efficient sampling (Holderrieth et al., 2025; Zhu et al., 2025a). Among discrete diffusion approaches, **masked discrete diffusion models** (MDMs) (Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2024; Zheng et al., 2024) have demonstrated particularly strong performance due to their simple design and stable, likelihood-based training objective. Recent work has further extended MDMs to any-length generation by allowing the insertion of masked tokens at arbitrary positions during the diffusion process (Kim et al., 2025a). Despite this progress, how to effectively scale any-length discrete diffusion models for reward optimization and fine-tuning remains largely unexplored.

Recently, Tang et al. (2025b) introduced a principled framework for robust fine-tuning of fixed-length discrete diffusion models by combining off-policy learning with optimized buffer generation to learn a controlled continuous-time Markov chain (CTMC) path measure $\mathbb{P}^\star$ that samples from an intractable reward-tilted target distribution $p_{\text{target}}(\boldsymbol{x})$, without requiring explicit target samples. While this framework is effective for fixed-length MDMs—where the action space is limited to token unmasking—any-length MDMs introduce a substantially larger, more complex action space that includes both unmasking and variable-length insertions. As a result, model performance becomes highly sensitive to the chosen insertion and unmasking trajectory. This motivates our **key question**: *Can we efficiently fine-tune any-length discrete diffusion models to sample from an intractable reward-tilted distribution while preserving high generation quality?*

Contributions To answer this, we introduce Finetuning **A**ny-Length **D**iscrete **D**iffusion for **A**daptive Decoding (**A2D2**), a unified framework for reward-guided fine-tuning of any-length discrete diffusion models via joint optimization of the insertion and unmasking policies and quality-based inference schedule. We derive the Radon-Nikodym derivative for the joint insertion and unmasking path measures, which enables theoretically-guaranteed convergence to an intractable reward-tilted sequence distribution. We establish unmasking and insertion quality as tractable methods of minimizing compounding parallelization error (CPE) and introduce the **Adaptive Joint Decoding (AJD)** loss, which provably yields the optimal path measure that *minimizes error* and *generates the reward-tilted distribution*. We demonstrate that A2D2 simultaneously optimizes rewards while enhancing generation flexibility and accuracy over prior fixed-length fine-tuning and inference-time guidance approaches.

Related Works We provide a comprehensive discussion of related works in App A.

2 PRELIMINARIES

Notation We denote the CTMC of a diffusion trajectory as $\boldsymbol{X}_{0:1} := (\boldsymbol{X}_t)_{t\in[0,1]}$ which lies in a discrete state space $\mathcal{X} \in \{1,\dots,V\}^L$ of sequences of length L and vocabulary size V. A any-length discrete diffusion model is characterized by a path measure $\mathbb{P} \in \mathcal{P}(\mathcal{X})$ over trajectories that follow an joint generator $\boldsymbol{A}_t = \boldsymbol{Q}_t + \boldsymbol{R}_t$, where $\boldsymbol{Q}_t$ is the unmasking rate and $\boldsymbol{R}_t$ is the insertion rate. Specifically, we denote the pre-trained discrete diffusion model as $\mathbb{P}^{\text{pre}}$ and the optimal target model as $\mathbb{P}^\star$. For a corrupted sequence $\boldsymbol{x}_t$ at time t, the unmasking posterior is given by $f_\theta(\boldsymbol{x}, t)$ with indices $[\ell, \boldsymbol{v}]$ denoting the probability of a single token at position ℓ and the insertion rate is given by $g_\theta(\boldsymbol{x}_t, t)[\ell]$ with indices $[\ell]$ indicating the insertion expectation between positions $\ell-1$ and ℓ of the sequence.

Masked Discrete Diffusion Models The discrete diffusion paradigm that models sequence generation as a continuous-time Markov chain (CTMC) which starts at a uninformative prior distribution p_{prior} and makes discrete jumps over a finite time horizon $t \in [0,1]$ to generate a sample from the data distribution p_{data}. **Masked discrete diffusion models** (MDMs) have a unique prior distribution p_{prior} consisting of fully masked sequences $(\boldsymbol{M})^L$ (Shi et al., 2024; Sahoo et al., 2024; Ou et al., 2024; Zheng et al., 2024). Given a clean data distribution p_{data}, the MDM's forward process progressively converts clean tokens to masked tokens according to a time schedule. Then, the training process aims to reconstruct the clean sequence $\boldsymbol{x}_1 \sim p_{\text{data}}$ from intermediate partially masked sequences $\boldsymbol{x}_t$ by minimizing the **denoising cross-entropy (DCE)** loss:

$$\mathcal{L}_{\text{DCE}}(\theta; \boldsymbol{x}_1) := \mathop{\mathbb{E}}_{t\sim\mathcal{U}(0,1)} \mathop{\mathbb{E}}_{\boldsymbol{x}_t\sim p_t(\cdot|\boldsymbol{x}_1)} \left[-\frac{\dot{\alpha}_t}{1-\alpha_t} \sum_{\ell:\boldsymbol{x}_t^\ell=\boldsymbol{M}} \log f_\theta(\boldsymbol{x}_t, t)[\ell, \boldsymbol{x}_1^\ell] \right] \tag{1}$$

where $\alpha_t : [0,1] \to [0,1]$ is the unmasking schedule bounded by $\alpha_0 = 0$ and $\alpha_1 = 1$ where $1-\alpha_t$ determines the probability of a token remaining *unmasked* at time t in the forward masking process and the unmasking occurs in the reverse process if the time t is greater than the unmasking time $t_u^\ell \sim \dot{\alpha}_t dt$, i.e. $t \geq t_u^\ell$. Under the standard linear schedule, $\alpha_t = t$.

Any-Length Discrete Diffusion To overcome the limitation of requiring a fixed-length initialization of the masked sequence, any-length discrete diffusion models (Havasi et al., 2025; Kim et al., 2025a) have enabled the ability to **insert** tokens along the generation process. In this work, we extend the **flexible-length masked diffusion model** framework introduced in (Kim et al., 2025a), which defines a *joint stochastic interpolant* for both the insertion process and the unmasking process. In contrast to fixed-length MDMs, Kim et al. (2025a) defines the forward corruption process by gradually masking *and* removing tokens by drawing an insertion time $t_i^\ell \sim \dot{\alpha}_t dt$ and unmasking time $t_u^\ell \sim \mathbf{1}[t \geq t_i^\ell] \frac{\dot{\beta}_t}{1-\beta_{t_i^\ell}} dt$, which enforces that that the token is unmasked only after it is inserted, i.e. $t_u^\ell > t_i^\ell$. To track the position of the clean token as tokens are removed, we set $s_t[\ell]$ as the position of token $\boldsymbol{x}_1^\ell$ in the intermediate state $(\boldsymbol{x}_t, s_t) \sim p_t(\cdot|\boldsymbol{x}_1)$, such that $\boldsymbol{x}_t^{s_t[\ell]} = \boldsymbol{x}_1^\ell$. To train the generative any-length MDM, we parameterize both the **unmasking posterior** $f_\theta(\boldsymbol{x}_t, t)[\ell] \in \Delta^V$ and

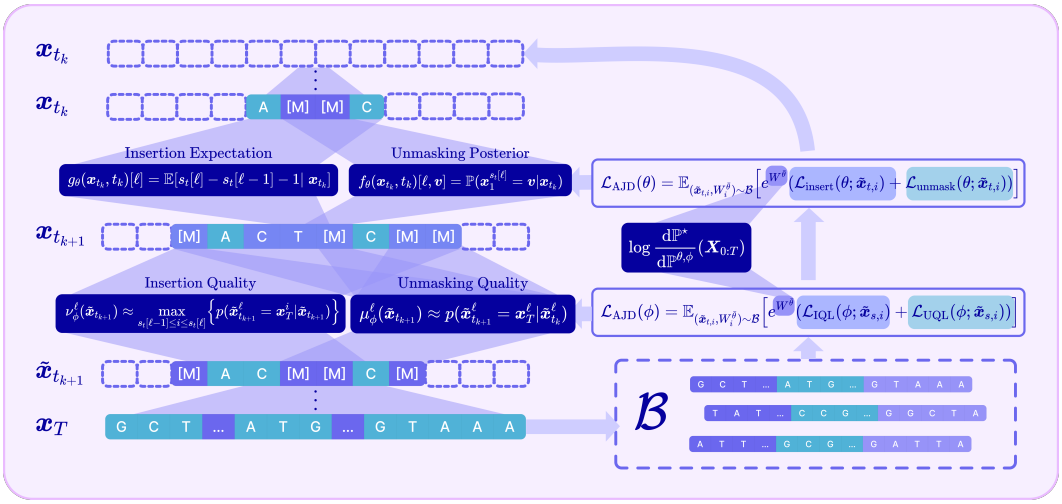

Figure 1: **Adaptive Any-Length Discrete Diffusion.** Our framework introduces a framework for joint fine-tuning of the insertion and unmasking policies and quality classifier to generate sequences from the optimal reward-tilted distribution $\mathbb{P}^\star$. First, we generate sequences using adaptive inference and store them in a replay buffer. Then, we compute the Adaptive Joint Decoding ($\mathcal{L}_{\text{AJD}}$), which weights each sequence's loss contribution with an importance weight $W^{\bar{\theta},\bar{\phi}}$.

an **insertion expectation** $g_\theta(\boldsymbol{x}_t, t)[\ell] \in \mathbb{R}_{\geq 0}$ that returns the predicted number of tokens to insert at each gap and minimize the following losses:

$$\mathcal{L}_{\text{insert}}(\theta; \boldsymbol{x}_1) = \mathop{\mathbb{E}}_{t\sim\mathcal{U}(0,1)} \mathop{\mathbb{E}}_{\boldsymbol{x}_t\sim p_t(\cdot|\boldsymbol{x}_1)} \left[-\frac{\dot{\alpha}_t}{1-\alpha_t} \sum_{\ell=0}^{\text{len}(\boldsymbol{x}_t)} \phi\left(s_t[\ell] - s_t[\ell-1], g_\theta(\boldsymbol{x}_t, t)[\ell]\right) \right] \tag{2}$$

$$\mathcal{L}_{\text{unmask}}(\theta; \boldsymbol{x}_1) = \mathop{\mathbb{E}}_{t\sim\mathcal{U}(0,1)} \mathop{\mathbb{E}}_{\boldsymbol{x}_t\sim p_t(\cdot|\boldsymbol{x}_1)} \left[-\frac{\dot{\beta}_t}{1-\beta_t} \sum_{\ell:\boldsymbol{x}_t^\ell=\boldsymbol{M}} \log f_\theta(\boldsymbol{x}_t, t)[\ell, \boldsymbol{x}_1^\ell] \right] \tag{3}$$

where $\phi(x, y) = y - x \log y$ is the scalar Bregman divergence. A detailed background on any-length MDMs is provided in App B.1.

3 Defining the Unmasking and Insertion Quality

In this section, we formally define the quality of a step taken in the decoding process of any-length MDM. First, we define the *unmasking quality* and derive its relationship to the error accumulated in an unmasking step in Sec 3.1. Then, we introduce a theoretically-grounded definition of the *insertion quality* and in Sec 3.2.

3.1 Unmasking Quality

Definition We define the **unmasking quality** as the probability that the unmasked token is sampled from the unmasking posterior given the context from the rest of the sequence. Concretely, given a clean sequence from the target $\boldsymbol{x}_1 \sim p_{\text{target}}$, a sample along the interpolant $\tilde{\boldsymbol{x}}_t$, a masked position $\tilde{\boldsymbol{x}}_t^\ell = \boldsymbol{M}$, a set of masked tokens $\mathcal{M}_t$ that are unmasked to obtain $\boldsymbol{y}$ where $\boldsymbol{y}^\ell \sim f_\theta(\tilde{\boldsymbol{x}}_t, t)[\ell]$, and the unmasking posterior at position ℓ given by $p(\boldsymbol{y}^\ell = \cdot|\tilde{\boldsymbol{x}}_t) = f_\theta(\tilde{\boldsymbol{x}}_t, t)[\ell]$, we define the unmasking quality as:

$$\mu_\star^\ell(\boldsymbol{y}) := p(\boldsymbol{y}^\ell = \boldsymbol{x}_1^\ell|\boldsymbol{y}) = f_\theta(\tilde{\boldsymbol{x}}_t, t)[\ell, \boldsymbol{x}_1^\ell] \tag{4}$$

which aligns with the definition of the *per-token quality* introduced in (Kim et al., 2025b). When the probability of the true token $p(\boldsymbol{y}^\ell = \boldsymbol{x}_1^\ell|\boldsymbol{y})$ for $\tilde{\boldsymbol{x}}_t^\ell = \boldsymbol{M}$ is high given the unmasked context $\tilde{\boldsymbol{x}}_t^{\text{UM}}$, then the quality is high, and if the probability is low, then the quality is low. To predict the quality of an already unmasked token $\boldsymbol{y}^\ell$ such that it estimates the unmasking posterior defined on its masked

state $\boldsymbol{y}^{\triangleleft_{\mathcal{M}_t}\boldsymbol{M}} = \tilde{\boldsymbol{x}}_t$, we train a parameterized model $\mu_\phi : \mathcal{X} \to [0,1]$ to approximate $\mu^\ell_\star$ given its unmasked state $\boldsymbol{y}$ as:

$$\mu^\ell_\phi(\boldsymbol{y}) \approx p(\boldsymbol{y}^\ell = \boldsymbol{x}_1^\ell | \boldsymbol{y}) \tag{5}$$

During inference, when we do not have access to the ground-truth sequence $\boldsymbol{x}_1$, the unmasking quality determines which tokens are inconsistent and should be re-masked.

Training the Unmasking Quality Predictor To train $\mu^\ell_\phi : \mathcal{X} \to [0,1]$, we start with a clean sequence from the target distribution $\boldsymbol{x}_1 \sim p_{\text{target}}$ and sample an intermediate state from the joint interpolant $\tilde{\boldsymbol{x}}_t$ by partially masking and removing tokens in $\boldsymbol{x}_1$. Then, we take a single unmasking step to obtain $\boldsymbol{y}$ by replacing a subset $\mathcal{M}$ of tokens in $\tilde{\boldsymbol{x}}_t$ with the clean token sampled from the unmasking posterior $\boldsymbol{y}^\ell \sim f_\theta(\tilde{\boldsymbol{x}}_t, t)[\ell]$. Finally, we minimize an **Unmasking Quality Loss** (UQL) defined as:

$$\mathcal{L}_{\text{UQL}}(\phi; \boldsymbol{x}_1) := \mathop{\mathbb{E}}_{t\sim\mathcal{U}(0,1)} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}_t\sim p_t(\cdot|\boldsymbol{x}_1)} \mathop{\mathbb{E}}_{\boldsymbol{y}\sim p_{s|t}(\cdot|\tilde{\boldsymbol{x}}_t)} \left[\sum_{\ell\in\mathcal{M}} \text{BCE}\left(\mathbf{1}[\boldsymbol{y}^\ell = \boldsymbol{x}_1^\ell], \mu^\ell_\phi(\boldsymbol{y}) \right) \right] \tag{6}$$

where BCE denotes the **binary cross-entropy** loss defined as $\text{BCE}(b,c) = -b\log c - (1-b)\log(1-c)$ for $b \in \{0,1\}$ and $c \in [0,1]$. We note that this aligns with the loss defined in Kim et al. (2025b) for fixed-length MDMs, where it is shown that the unmasking quality is the *unique minimizer* of $\mathcal{L}_{\text{UQL}}(\phi)$.

Proposition 3.1 (Unique Minimizer of Unmasking Quality Loss)**.** *The unique minimizer of the unmsaking quality loss $\mathcal{L}_{UQL}(\phi)$ is the true unmasking quality:*

$$\mu_\star = \mu_{\phi^\star}, \quad \textit{where} \quad \phi^\star = \arg\min_\phi \mathcal{L}_{UQL}(\phi) \tag{7}$$

The proof is restated for clarity in App C.2.

Compounding Parallelization Error of Unmasking Steps For **unmasking**, the compounding parallelization error is defined as the error that arises from sampling the *factorized distribution* over unmasked tokens sampled in parallel and the *true joint distribution* (Park et al., 2024). Concretely, it is the KL divergence between the true joint distribution sampled by unmasking only a single token per step, and the product of the marginals, which is the approximation made when unmasking multiple tokens in parallel.

Definition 3.1 (Unmasking Compounding Parallelization Error)**.** *Consider a unmasking step from state $\boldsymbol{x}_s$ to $\boldsymbol{x}_t$, where $\{\boldsymbol{x}_t^{\ell_k}\}_{k=1}^K$ denotes the set of K tokens unmasked in parallel at time t. The unmasking CPE over the transition $s \to t$ is defined as:*

$$\mathcal{E}^{unmsk}_{CPE}(s \to t) = \mathbb{E}_{\boldsymbol{x}_s\sim\mathbb{P}_s}\left[D_{KL}\left(p(\boldsymbol{x}_t^{\ell_1}, \ldots, \boldsymbol{x}_t^{\ell_K} | \boldsymbol{x}_s) \middle\| \prod_{k=1}^K p(\boldsymbol{x}_t^{\ell_k} | \boldsymbol{x}_s) \right) \right] \tag{8}$$

where $\mathbb{P}_s$ is the marginal distribution of sequences at time s.

Optimal Parallel Unmasking as Maximizing the Unmasking Quality We now show that unmasking quality provides an upper bound for the probability of successful parallel unmasking.

Proposition 3.2 (Relationship Between Unmasking Quality and Parallel Unmasking)**.** *Assume that in a parallel unmasking step on indices $\mathcal{M}_t = \{\ell_k\}_{k=1}^K$, the unmasked tokens are conditionally independent given the unchanged context $\bar{\boldsymbol{x}}_s$ and the current state $\boldsymbol{x}_s$, i.e. $p(\boldsymbol{x}_t^{\mathcal{M}_t} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) = \prod_{k=1}^K p(\boldsymbol{x}_t^{\ell_k} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s)$. If we define the unmasking quality $\mu^{\ell_k}_\star(\boldsymbol{x}_t) := p(\boldsymbol{x}_t^{\ell_k} = \boldsymbol{x}_1^{\ell_k} | \boldsymbol{x}_t^{\ell_k \leftarrow \boldsymbol{M}})$ and note that $p(\boldsymbol{x}_t^{\ell_k} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) = p(\boldsymbol{x}_t^{\ell_k} | \boldsymbol{x}_t^{\ell_k \leftarrow \boldsymbol{M}})$, then*

$$p(\boldsymbol{x}_t^{\mathcal{M}_t} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) = \prod_{k=1}^K \nu^{\ell_k}_\star(\boldsymbol{x}_t) \tag{9}$$

The proof is deferred to App C.4. This establishes the theoretical grounding for our adaptive unmasking strategy, which remasks low-quality tokens in $\boldsymbol{x}_t$ to ensure only mutually high-quality tokens are unmasked in parallel, which effectively maximizes the probability of optimal parallel unmasking.

3.2 Insertion Quality

Definition It is natural to define the **insertion quality** as the probability that the insertion is likely to be decoded into a true token in the corresponding gap of the target sequence. Concretely, given a clean sequence from the target distribution $\boldsymbol{x}_1 \sim p_{\text{target}}$ and a sample along the interpolant $\tilde{\boldsymbol{x}}_t$, we define quality of a mask inserted between positions $\ell - 1$ and ℓ, denoted $\boldsymbol{y} := \tilde{\boldsymbol{x}}_t^{\triangleleft_\ell \boldsymbol{M}}$, as:

$$\nu_\star(\boldsymbol{y}) := \max_{s_t[\ell-1] \le i \le s_t[\ell]} \left\{ p(\boldsymbol{y}^\ell = \boldsymbol{x}_1^i | \boldsymbol{y}) \right\} \tag{10}$$

which returns the highest probability of any ground truth token between positions $s_t[\ell-1]$ and $s_t[\ell]$ in the clean sequence $\boldsymbol{x}_1$ being predicted by the unmasking posterior $p(\boldsymbol{y}^i = \cdot | \boldsymbol{y}) = f_\theta(\boldsymbol{y}, t)[i]$ at the newly inserted mask token $\boldsymbol{y}^i = (\tilde{\boldsymbol{x}}_t^{\triangleleft_\ell \boldsymbol{M}})^i = \boldsymbol{M}$. Alternatively, we can define the insertion quality as the sum of the probabilities of all possible tokens that exist within the gap in the ground truth sequence, given by:

$$\nu_\star^\ell(\boldsymbol{y}) := \sum_{i=s_t[\ell-1]}^{s_t[\ell]} p(\boldsymbol{y}^\ell = \boldsymbol{x}_1^i | \boldsymbol{y}) \tag{11}$$

which is high when several potential tokens have a **high potential** to be unmasked to one of the tokens within the gap. Note that when there is **no token** between positions $s_t[\ell-1]$ and $s_t[\ell]$ in $\boldsymbol{x}_1$, the quality is zero. Since we have no access to the ground truth sequence at inference, we aim to train a parameterized model ν_ϕ that takes only the sequence *after the insertion step* $\boldsymbol{y}$ as input and approximates the quality of the insertion:

$$\nu_\phi^i(\boldsymbol{y}) \approx \sum_{i=s_t[\ell-1]}^{s_t[\ell]} p(\boldsymbol{y}^\ell = \boldsymbol{x}_1^i | \boldsymbol{y}) \in [0, 1] \tag{12}$$

This allows us to evaluate the quality of an insertion and remove low-quality insertions that are likely to result in mistakes in unmasking.

Training the Insertion Quality Predictor To train $\nu_\star^\ell(\boldsymbol{y})$, we start with a clean target sequence $\boldsymbol{x}_1 \sim p_{\text{target}}$, we apply the standard joint interpolant to sample $\tilde{\boldsymbol{x}}_t$ by partially masking and removing tokens in $\boldsymbol{x}_1$. Then, we insert the set of masks $\mathcal{I} = \{\boldsymbol{y}^i \mid \boldsymbol{x}_t^{\triangleleft_\ell \boldsymbol{M}}; g_\theta(\boldsymbol{x}_t, t)[\ell]\}$ predicted by the insertion expectation $g_\theta(\boldsymbol{x}_t, t)$ at each gap to get $\boldsymbol{y}$ and predict the unmasking posterior over the newly inserted masked positions $f_\theta(\boldsymbol{y}, t)[i]$. Finally, we minimize the **Insertion Quality Loss** (IQL), defined as:

$$\mathcal{L}_{\text{IQL}}(\phi; \boldsymbol{x}_1) := \mathop{\mathbb{E}}_{t \sim \mathcal{U}(0,1)} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}_t \sim p_t(\cdot|\boldsymbol{x}_1)} \mathop{\mathbb{E}}_{\boldsymbol{y} \sim p_{s|t}(\cdot|\tilde{\boldsymbol{x}}_t)} \left[\sum_{i \in \mathcal{I}} \text{BCE}\big(\nu_\star^i(\boldsymbol{y}), \nu_\phi^i(\boldsymbol{y})\big) \right] \tag{13}$$

where BCE denotes the **Bernoulli cross-entropy** loss for two values in defined as $\text{BCE}(b, c) = -b \log c - (1-b)\log(1-c)$ for two values $b, c \in [0, 1]$. If there are multiple masks $(\boldsymbol{y}^{i_1}, \ldots, \boldsymbol{y}^{i_D})$ inserted in a gap $(\boldsymbol{x}_1^{s_t[\ell-1]+1}, \ldots, \boldsymbol{x}_1^{s_t[\ell]-1})$, then for each i_d, we sum over the probabilities over the range $s_t[\ell-1] + d$ to $s_t[\ell] - 1 - (D - d)$. We show that the minimizer of the insertion quality loss $\mathcal{L}_{\text{IQL}}(\phi)$ is the true insertion quality $\nu_\star$ defined in (11).

Proposition 3.3 (Minimizer of Insertion Quality Loss)**.** *The unique minimizer of the insertion quality loss $\mathcal{L}_{IQL}(\phi)$ is the true insertion quality:*

$$\nu_\star = \nu_{\phi^\star}, \quad \textit{where} \quad \phi^\star = \arg\min_\phi \mathcal{L}_{IQL}(\phi) \tag{14}$$

The proof is provided in App C.3. To formalize insertion quality as a measure of error accumulation, we now introduce the **insertion compounding parallelization error** (CPE) and derive its relationship to insertion quality.

Optimizing Reconstruction Likelihood as Maximizing Insertion Quality To theoretically justify maximizing insertion quality, we prove that it provides an upper bound for the probability of reconstructing a clean sequence $\boldsymbol{x}_1 \sim p_{\text{target}}$.

Proposition 3.4 (Insertion Quality as an Upper Bound on Reconstruction via Insertions)**.** *Consider a clean target sequence $\boldsymbol{x}_1 \sim p_{target}$ and an intermediate sequence $\boldsymbol{x}_s \sim p_s(\cdot|\boldsymbol{x}_1)$. Given a set of indices of inserted masks at each gap $\mathcal{I}_t$ which yields $\boldsymbol{x}_t^{\mathcal{I}_t}$, the probability of reconstructing $\boldsymbol{x}_1$ is upper bounded by the product of insertion qualities:*

$$p(\boldsymbol{x}_t^{\mathcal{I}_t} = \boldsymbol{x}_1^{\mathcal{I}_t}|\boldsymbol{x}_t) \leq \prod_{i\in\mathcal{I}_t} \mu_\star^i(\boldsymbol{x}_t) \approx \prod_{i\in\mathcal{I}_t} \mu_\phi^i(\boldsymbol{x}_t) \tag{15}$$

assuming that the unmasking posterior for each inserted mask is conditionally independent given $\boldsymbol{x}_t$.

We defer the proof to App C.5. This justifies maximizing insertion quality as optimizing reconstruction quality.

4 A2D2: Finetuning Any-Length Discrete Diffusion for Adaptive Decoding

In this section, we introduce Finetuning **A**ny-Length **D**iscrete **D**iffusion for **A**daptive Decoding (**A2D2**), a novel framework tailored for any-length masked diffusion models which jointly optimizes the parameterized quality predictors and policy to sample from the reward-tilted data distribution.

4.1 Adaptive Joint Decoding Loss

Optimal Reward-Tilted Path Measure Since the path measure $\mathbb{P}^v$ of any-length MDMs consists of both insertion steps *and* unmasking steps which form the joint CTMC with rates $\boldsymbol{R}_t^v$ and $\boldsymbol{Q}_t^v$, respectively, we aim to define the tilted path measure $\mathbb{P}^\star$ that optimally samples from the reward-tilted distribution p_{target}. Concretely, we define the tilted path measure as:

$$\mathbb{P}^\star(\boldsymbol{X}_{0:1}) := \frac{1}{Z}\mathbb{P}^{\text{pre}}(\boldsymbol{X}_{0:1}) \exp\left(\frac{r(\boldsymbol{X}_1)}{\alpha}\right), \quad \mathbb{P}_1^\star(\boldsymbol{x}) = \frac{1}{Z} p_{\text{data}}(\boldsymbol{x}) \exp\left(\frac{r(\boldsymbol{x})}{\alpha}\right) =: p_{\text{target}}(\boldsymbol{x}) \tag{16}$$

where $\boldsymbol{X}_{0:1} = (\boldsymbol{X}_t)_{t\in[0,1]}$ is a joint CTMC induced by a path measure and $\mathbb{P}^{\text{pre}}$ is the reference path measure induced by the pre-trained insertion rate $\boldsymbol{R}_t^{\text{pre}}$ and unmasking rate $\boldsymbol{Q}_t^{\text{pre}}$. Given parameterized insertion and unmasking rates $\boldsymbol{R}_t^\theta$ and $\boldsymbol{Q}_t^\theta$, we can optimize θ such that the path measure $\mathbb{P}^\theta$ matches $\mathbb{P}^\star$ by minimizing the following **entropy-regularized reward optimization** problem (Uehara et al., 2024):

$$\min_\theta \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}^\theta}\left[r(\boldsymbol{X}_1)\right] - \alpha D_{\text{KL}}(\mathbb{P}^\theta \| \mathbb{P}^{\text{pre}}) \tag{17}$$

which is uniquely minimized when $\mathbb{P}^\theta = \mathbb{P}^\star$. For any-length joint CTMCs, the KL-divergence takes a unique form, as derived in App 1.

Weighted Cross-Entropy Objective To derive a loss that provably converges to $\mathbb{P}^\star$, let's consider directly minimizing the KL objective $D_{\text{KL}}(\mathbb{P}^{\theta,\phi}, \mathbb{P}^\star)$. While it's minimizer indeed matches the optimal measure $\mathbb{P}^\star$, it is difficult to optimize due to the expectation over $\mathbb{P}^{\theta,\phi}$ which changes as the model is updated. Instead, we use the **cross-entropy loss** which takes the expectation over the **fixed** target path measure $\mathbb{P}^\star$:

$$\mathcal{F}_{\text{CE}}(\mathbb{P}^{\theta,\phi}, \mathbb{P}^\star) := D_{\text{KL}}(\mathbb{P}^\star \| \mathbb{P}^{\theta,\phi}) = \mathbb{E}_{\mathbb{P}^\star}\left[\log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^{\theta,\phi}}\right] = \mathbb{E}_{\mathbb{P}^v}\left[\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v} \log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^{\theta,\phi}}\right] \tag{18}$$

where the final equality allows us to sample trajectories from any arbitrary path measure $\mathbb{P}^v$ and **reweight** them such that they approximate the target path measure $\mathbb{P}^\star$ at the infinite sampling limit. The reweighting term $\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v}$ is the Radon-Nikodym derivative (RND) between the two path measures. We derive the full tractable form of the RND in Prop 4.1.

Proposition 4.1 (Radon-Nikodym Derivative of Parameterized Rates)**.** *Let the fine-tuned unmasking rate be $f^v(\boldsymbol{x}_t, t)[\ell] \in \Delta^{V-1}$ and the insertion rate be $g^v(\boldsymbol{x}_t, t)[\ell] \in \mathbb{R}_{\geq 0}$ that generates the path measure $\mathbb{P}^v$. Then, given optimal rates $f^{pre}(\boldsymbol{x}_t, t)$ and $g^{pre}(\boldsymbol{x}_t, t)$ and reward function $r : \mathcal{X} \to \mathbb{R}$, the log RND between the optimal joint CTMC and the fine-tuned CTMC over the trajectory $\boldsymbol{X}_{0:1} = (\boldsymbol{X}_t)_{t\in[0,1]}$ is defined as:*

$$\log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1}) = \frac{r(\boldsymbol{X}_1)}{\alpha} - \log Z + \sum_{t_u : \boldsymbol{X}_{t_u^-} \neq \boldsymbol{X}_{t_u}} \sum_{\ell : \boldsymbol{X}^\ell_{t_u^-} \neq \boldsymbol{X}^\ell_{t_u}} \log \frac{f^{pre}(\boldsymbol{X}_{t_u}, t)[\ell, \boldsymbol{d}]}{f^v(\boldsymbol{X}_{t_u}, t)[\ell, \boldsymbol{d}]}$$

$$+ \sum_{t_i : \boldsymbol{X}_{t_i^-} \neq \boldsymbol{X}_{t_i}} \sum_{\ell : \boldsymbol{X}_{t_u^-} \neq \boldsymbol{X}^\ell} \log \frac{g^{pre}(\boldsymbol{X}_{t_i}, t_i)[\ell]}{g^v(\boldsymbol{X}_{t_i}, t_i)[\ell]} + \int_0^1 \frac{\dot{\alpha}_t}{1 - \alpha_t} \left(\sum_\ell (g^{pre} - g^v)(\boldsymbol{X}_{t_i}, t_i)[\ell] \right) dt \quad (19)$$

where $t_i \in [0, 1]$ denotes the times of insertion events and $t_u \in [0, 1]$ denotes the times of unmasking events.

The proof is given in App C.8. Then, defining $W^v(\boldsymbol{X}_1) := \log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^{\theta,\phi}}(\boldsymbol{X}_{0:1})$, we can write the cross-entropy loss as $\mathcal{F}_{\text{CE}}(\mathbb{P}^{\theta,\phi}, \mathbb{P}^\star) = \mathbb{E}_{\mathbb{P}^v}[\frac{1}{Z} e^{W^v} \log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^{\theta,\phi}}]$.

Adaptive Joint Decoding Loss Now, we derive our **Adaptive Joint Decoding (AJD)** loss, which allows us to optimize the cross-entropy loss without computing the inner RND $\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^{\theta,\phi}}$ over trajectories after each update to the model parameters.

$$\min_{\theta,\phi} \mathbb{E}_{\mathbb{P}^v} \left[\frac{1}{Z} e^{W^v} \log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^{\theta,\phi}} \right] = \min_{\theta,\phi} \mathbb{E}_{\mathbb{P}^v} \left[\frac{1}{Z} e^{W^v} [-\log \mathbb{P}^{\theta,\phi}] \right]$$

$$= \min_{\theta,\phi} \mathbb{E}_{\mathbb{P}^v} \left[\frac{1}{Z} e^{W^v} \left[\sum_{t_i : \boldsymbol{X}_{t_i^-} \neq \boldsymbol{X}_{t_i}} - \log \mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{t_i^-} | \boldsymbol{X}_{t_i},) + \sum_{t_u : \boldsymbol{X}_{t_u^-} \neq \boldsymbol{X}_{t_u}} - \log \mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{t_u^-} | \boldsymbol{X}_{t_u}) \right] \right] \quad (20)$$

where the first equality follows from dropping the $\log \mathbb{P}^\star$ term independent of θ, ϕ, and the second equality follows from decomposing the probability path into the sum of probabilities of each insertion and unmasking step at times t_i and t_u, respectively. Rather than explicitly computing the sum over the full trajectory, we observe that each inner term can be reframed as an expectation over samples from the interpolant $\tilde{\boldsymbol{x}}_t \sim p_t(\cdot|\boldsymbol{X}_1)$ given a clean sample $\boldsymbol{X}_1$ from $\boldsymbol{X}_{0:1} \sim \mathbb{P}^v$. Given the clean sample $\boldsymbol{x}_1$, the optimal f_θ, g_θ, μ_ϕ, and ν_ϕ can be obtained by minimizing $\mathcal{L}_{\text{unmask}}(\theta; \boldsymbol{x}_1)$ (3), $\mathcal{L}_{\text{insert}}(\theta; \boldsymbol{x}_1)$ (2), $\mathcal{L}_{\text{UQL}}(\phi; \boldsymbol{x}_1)$ (6), and $\mathcal{L}_{\text{IQL}}(\phi; \boldsymbol{x}_1)$ (13), respectively. This yields our **Adaptive Joint Decoding (AJD)** loss, defined as:

$$\mathcal{L}_{\text{AJD}}(\theta, \phi) := \mathop{\mathbb{E}}_{\boldsymbol{X}_{0:1} \sim \mathbb{P}^v} \left[\frac{1}{Z} \mathrm{e}^{W^v} \left[\mathcal{L}_{\text{unmask}}(\theta; \boldsymbol{X}_1) + \mathcal{L}_{\text{insert}}(\theta; \boldsymbol{x}_1) + \mathcal{L}_{\text{UQL}}(\phi; \boldsymbol{X}_1) + \mathcal{L}_{\text{IQL}}(\phi; \boldsymbol{X}_1) \right] \right] \quad (21)$$

where the normalizing constant Z is approximated as $Z \approx \mathbb{E}_{\boldsymbol{X}_{0:1} \sim \mathbb{P}^v} \left[\mathrm{e}^{W^v} \right]$. This loss can be seen as an extension of the weighted denoising cross-entropy loss used for fixed-length discrete diffusion sampling (Zhu et al., 2025a) and fine-tuning (Tang et al., 2025b). This unique minimizer of the AJD loss is the optimal unmasking generator $\boldsymbol{Q}^\star$ and insertion generator $\boldsymbol{R}^\star$ of the reward-tilted path measure $\mathbb{P}^\star$, with proof deferred to App C.8.

4.2 Fine-Tuning Any-Order Discrete Diffusion

Off-Policy Reinforcement Learning To optimize the AJD loss defined in Sec 4.1, we leverage an **off-policy reinforcement learning** strategy, where we define the path measure $\mathbb{P}^v := \mathbb{P}^{\bar{\theta},\bar{\phi}}$ as the fine-tuned policy and planner models with detached gradients $\bar{\theta} := \texttt{stopgrad}(\theta)$ and $\bar{\phi} := \texttt{stopgrad}(\phi)$. With this definition, the fine-tuning procedure proceeds as follows: **(1)** we sample a batch of B sequences $\boldsymbol{x}_1$ while computing their log RND $W^{\bar{\theta},\bar{\phi}}$ and store them in a replay buffer $\mathcal{B} = \{(\boldsymbol{x}_{1,i}, W^{\bar{\theta},\bar{\phi}})\}_{i=1}^B$, **(2)** for each $\boldsymbol{x}_{1,i}$, we sample R intermediate sequences from the interpolant $\tilde{\boldsymbol{x}}_{t,j} \sim p_t(\cdot|\boldsymbol{x}_{1,i})$, and **(3)** we compute the AJD loss (21) and update the parameters θ, ϕ.

Alternating Optimization of Policy and Quality Predictors Since the number of parameters θ in the insertion and unmasking policy is orders of magnitude larger than the number of parameters ϕ

Table 1: **Multi-objective peptide design results.** All values are averaged over 100 generated peptides. Best values are **bolded** and second best values are underlined. **Pre-trained** denotes unconditional sampling with the pre-trained peptide SMILES model from PepTune (Tang et al., 2025a) and our pre-trained any-length model. **PepTune** denotes samples from 100 iterations of inference-time multi-objective guidance with the fixed-length model. **TR2-D2 w/o search** denotes sampling from a fixed-length model fine-tuned with TR2-D2 for multiple objectives without search (Tang et al., 2025b). **A2D2 w/o quality** denotes sampling from a fine-tuned any-length model using the AJD loss without training or sampling with the quality predictors. **A2D2** denotes sampling from a fine-tuned any-length model using the AJD loss and quality predictor optimization and sampling. † indicates values taken from (Tang et al., 2025b).

Target Protein	Method	Binding Affinity (↑)	Solubility (↑)	Non-hemolysis (↑)	Non-fouling (↑)	Permeability (↑)
TfR	Pre-trained (Fixed Length) †	$8.008_{\pm 0.673}$	$0.742_{\pm 0.166}$	$0.874_{\pm 0.063}$	$0.102_{\pm 0.083}$	$-7.470_{\pm 0.120}$
	Pre-trained (Any Length)	$7.788_{\pm 0.798}$	$0.773_{\pm 0.202}$	$0.875_{\pm 0.084}$	$0.172_{\pm 0.163}$	$-7.248_{\pm 0.314}$
	PepTune †	$8.216_{\pm 0.703}$	$0.789_{\pm 0.144}$	$\mathbf{0.902_{\pm 0.051}}$	$0.121_{\pm 0.081}$	$-7.389_{\pm 0.119}$
	TR2-D2 w/o search	$8.518_{\pm 0.667}$	$0.664_{\pm 0.143}$	$0.876_{\pm 0.048}$	$0.067_{\pm 0.055}$	$-7.296_{\pm 0.140}$
	A2D2 w/o quality	$8.057_{\pm 0.681}$	$0.648_{\pm 0.271}$	$0.862_{\pm 0.095}$	$0.135_{\pm 0.167}$	$-7.252_{\pm 0.320}$
	A2D2 (Ours)	$\mathbf{11.283_{\pm 0.295}}$	$\mathbf{0.820_{\pm 0.095}}$	$0.754_{\pm 0.058}$	$\mathbf{0.214_{\pm 0.048}}$	$\mathbf{-6.628_{\pm 0.110}}$
GLP-1R	Pre-trained (Fixed Length) †	$8.233_{\pm 0.367}$	$0.742_{\pm 0.166}$	$0.874_{\pm 0.063}$	$0.102_{\pm 0.083}$	$-7.470_{\pm 0.120}$
	Pre-trained (Any Length)	$7.788_{\pm 0.798}$	$0.773_{\pm 0.202}$	$0.875_{\pm 0.084}$	$0.172_{\pm 0.163}$	$-7.248_{\pm 0.314}$
	PepTune †	$8.403_{\pm 0.365}$	$0.774_{\pm 0.170}$	$\mathbf{0.907_{\pm 0.057}}$	$0.125_{\pm .082}$	$-7.388_{\pm 0.128}$
	TR2-D2 w/o search	$8.698_{\pm 0.266}$	$0.692_{\pm 0.118}$	$0.864_{\pm 0.048}$	$0.243_{\pm 0.088}$	$-7.332_{\pm 0.059}$
	A2D2 w/o quality	$8.104_{\pm 0.769}$	$0.647_{\pm 0.273}$	$0.863_{\pm 0.088}$	$0.112_{\pm 0.131}$	$-7.228_{\pm 0.336}$
	A2D2 (Ours)	$\mathbf{9.724_{\pm 0.628}}$	$\mathbf{0.795_{\pm 0.101}}$	$0.621_{\pm 0.071}$	$\mathbf{0.323_{\pm 0.073}}$	$\mathbf{-6.689_{\pm 0.074}}$
GLAST	Pre-trained (Fixed Length) †	$7.830_{\pm 0.420}$	$0.742_{\pm 0.166}$	$0.874_{\pm 0.063}$	$0.102_{\pm 0.083}$	$-7.470_{\pm 0.120}$
	Pre-trained (Any Length)	$7.100_{\pm 1.274}$	$0.742_{\pm 0.166}$	$0.874_{\pm 0.063}$	$0.102_{\pm 0.083}$	$-7.470_{\pm 0.120}$
	PepTune †	$8.400_{\pm 0.353}$	$0.815_{\pm 0.139}$	$\mathbf{0.937_{\pm 0.029}}$	$0.137_{\pm 0.086}$	$-7.311_{\pm 0.106}$
	TR2-D2 w/o search	$8.579_{\pm 0.591}$	$0.709_{\pm 0.144}$	$0.913_{\pm 0.029}$	$0.119_{\pm 0.059}$	$-7.327_{\pm 0.063}$
	A2D2 w/o quality	$7.545_{\pm 1.259}$	$0.691_{\pm 0.233}$	$0.860_{\pm 0.084}$	$0.134_{\pm 0.153}$	$-7.239_{\pm 0.322}$
	A2D2 (Ours)	$\mathbf{11.265_{\pm 0.345}}$	$\mathbf{0.827_{\pm 0.103}}$	$0.703_{\pm 0.063}$	$\mathbf{0.183_{\pm 0.055}}$	$\mathbf{-6.460_{\pm 0.099}}$

in the quality prediction heads, optimizing both models simultaneously results in unstable training and sub-optimal convergence of the planner model, especially since the planner is being trained from scratch. Inspired by recent works (Zhou et al., 2025), we leverage an alternating optimization strategy which updates θ for N_m iterations with ϕ frozen, followed by updating ϕ for N_p iterations with θ frozen. Since the planner is initialized from scratch, we sample the replay buffer using only the policy model θ for N_{warmup} iterations. The full pseudo-code for fine-tuning with A2D2 is provided in Algorithm 1.

Adaptive Inference with A2D2 We describe the adaptive inference with the model parameters f_θ, g_θ and quality predictors μ_ϕ, ν_ϕ which starts from an empty sequence of pad tokens $\boldsymbol{x}_0$. At each intermediate discrete time step $[t_k, t_{k+1}]$, we perform the following steps:

(i) **Adaptive Unmasking**: We sample a subset of mask tokens $\mathcal{M}$ to unmask via the parameterized unmasking posterior $f_\theta(\boldsymbol{x}_t, t)[\ell]$. Then, we predict the unmasking quality for each token $\mu_\phi(\boldsymbol{x}_t, t)[\ell]$ and re-mask low-quality tokens that either fall below a threshold or until the expected number of masked tokens at time t is reached.

(ii) **Adaptive Insertion**: We insert a set of masks $\mathcal{I}$ according by sampling insertion counts $I_t^\ell \sim \text{Poisson}(g_\theta(\boldsymbol{x}_t, t)[\ell] \cdot \Delta t)$ for each gap. Then, we predict the insertion quality $\nu_\phi(\boldsymbol{x}_t, t)[i]$ for each inserted mask and subsequently remove low-quality masks.

We stop when no masks remain and the insertion expectation falls below 0.5 or when the total number of time steps is reached. The full inference procedure is detailed in Algorithm 2.

5 Experiments

5.1 Multi-Objective Therapeutic Peptide Generation

Setup In this experiment, we pre-train an any-length MDM on a dataset of 11 million peptide SMILES containing 7451 sequences from the CycPeptMPDB database (Li et al., 2023), $825,632$ unique peptides from SmProt (Li et al., 2021), and approximately 10 million modified peptides generated from CycloPs (Duffy et al., 2011; Feller & Wilke, 2025). Then, we use A2D2 to fine-tune

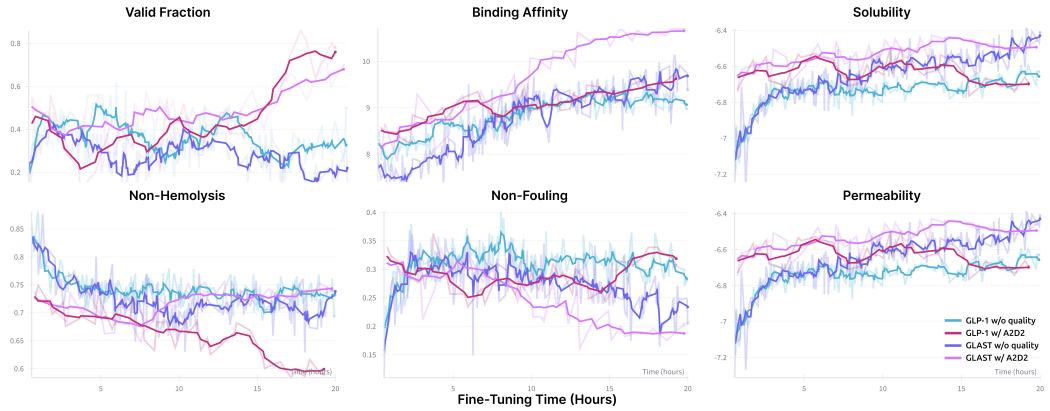

Figure 2: **Multi-objective fine-tuning with A2D2 for peptide generation with and without quality optimization.** Fine-tuning of any-length peptide SMILES generator with A2D2 was performed by optimizing the sum of five reward functions, including either binding affinity to GLP-1R or GLAST protein, solubility, non-hemolysis, non-fouling, and membrane permeability. Fraction of valid peptides over fine-tuning epochs with quality optimization (**blue and purple**) and without quality optimization (**red and magenta**).

the pre-trained model on multiple therapeutic properties, including binding affinity to a protein target, solubility, non-hemolysis, non-fouling, and permeability. Since not all SMILES sequences can be decoded into peptides, we evaluate the validity fraction with the SMILES2PEPTIDE decoder from Tang et al. (2025a).

Baselines To the best of our knowledge, there have been no prior works that train any-length discrete diffusion models for peptide SMILES generation or that introduce a method for inference-time guidance or fine-tuning of any-length discrete diffusion. Therefore, we compare against fixed-length masked diffusion model baselines, including **PepTune** (Tang et al., 2025a), which leverages Monte Carlo Tree Guidance (MCTG) for inference-time multi-objective guidance, and **TR2-D2** (Tang et al., 2025b), which leverages off-policy RL for fine-tuning discrete diffusion models. We also compare with the unconditional generative performance of the pre-trained fixed-length MDM from Tang et al. (2025a) and our pre-trained any-length MDM. Finally, to determine the effects of inference-time planning with our unmasking quality and insertion quality metrics, we compare against A2D2 fine-tuned without the remasking and deletion operation (**A2D2 w/o quality**).

Results First, we show that with and without quality-based adaptive inference, fine-tuning any-length MDMs with our AJD weighted loss yields significant increases in reward across multiple objectives as shown by the consistently increasing trend of the evaluation curves (Fig 2). Compared to both the pre-trained fixed-length and any-length baseline, **A2D2** produces higher scoring sequences across *all objectives* with the same inference cost (Table 1). Furthermore, we show that **A2D2** produces higher rewards across *most* properties compared to fixed-length discrete diffusion fine-tuning with **TR2-D2** (Tang et al., 2025b). Notably, when optimizing the same reward functions, **A2D2** with our quality-based adaptive inference increases the fraction of valid peptide sequences, indicating that maximizing quality translates empirically to higher-quality and more accurate generation (Fig 2).

6 Conclusion

In this work, we introduce Finetuning **A**ny-Length **D**iscrete **D**iffusion for **A**daptive Decoding (**A2D2**), a novel framework that unlocks reward-guided fine-tuning for any-length masked diffusion models with adaptive quality-based inference optimization. Driven by the importance of inference-schedule in any-length decoding, we define the notion of unmasking quality and insertion quality and establish a theoretical connection to compounding parallelization error. To enable adaptive inference and sampling from an intractable reward-tilted distribution, we introduce a unified fine-tuning method that co-optimizes the policy and quality predictors to obtain the model that provably generates the optimal any-length path measure. Our approach is a significant step toward any-length generation for diverse reward-based tasks.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 843–852, 2023.

Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018.

Haoxuan Chen, Yinuo Ren, Martin Renqiang Min, Lexing Ying, and Zachary Izzo. Solving inverse problems via diffusion-based priors: An approximation-free ensemble sampling approach. *arXiv preprint arXiv:2506.03979*, 2025.

Jacob K Christopher, Brian R Bartoldson, Tal Ben-Nun, Michael Cardei, Bhavya Kailkhura, and Ferdinando Fioretto. Speculative diffusion decoding: Accelerating language generation through diffusion. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 12042–12059, 2025.

Riccardo De Santi, Marin Vlastelica, Ya-Ping Hsieh, Zebang Shen, Niao He, and Andreas Krause. Provable maximum entropy manifold exploration via diffusion models. *arXiv preprint arXiv:2506.15385*, 2025.

Fergal J Duffy, Mélanie Verniere, Marc Devocelle, Elise Bernard, Denis C Shields, and Anthony J Chubb. Cyclops: generating virtual libraries of cyclized and constrained peptides including nonnatural amino acids. *Journal of chemical information and modeling*, 51(4):829–836, 2011.

Aaron L Feller and Claus O Wilke. Peptide-aware chemical language model successfully predicts membrane diffusion of cyclic peptides. *Journal of Chemical Information and Modeling*, 65(2): 571–579, 2025.

Nic Fishman, Leo Klarner, Valentin De Bortoli, Emile Mathieu, and Michael Hutchinson. Diffusion models for constrained domains. *arXiv preprint arXiv:2304.05364*, 2023.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36:12489–12517, 2023.

Wei Guo, Yuchen Zhu, Molei Tao, and Yongxin Chen. Plug-and-play controllable generation for discrete masked models. *arXiv preprint arXiv:2410.02143*, 2024.

Marton Havasi, Brian Karrer, Itai Gat, and Ricky TQ Chen. Edit flows: Flow matching with edit operations. *arXiv preprint arXiv:2506.09018*, 2025.

Peter Holderrieth, Michael S Albergo, and Tommi Jaakkola. Leaps: A discrete neural sampler via locally equivariant networks. *arXiv preprint arXiv:2502.10843*, 2025.

Jaeyeon Kim, Lee Cheuk-Kit, Carles Domingo-Enrich, Yilun Du, Sham Kakade, Timothy Ngotiaoco, Sitan Chen, and Michael Albergo. Any-order flexible length masked diffusion. *arXiv preprint arXiv:2509.01025*, 2025a.

Jaeyeon Kim, Seunggeun Kim, Taekyun Lee, David Z Pan, Hyeji Kim, Sham Kakade, and Sitan Chen. Fine-tuning masked diffusion for provable self-correction. *arXiv preprint arXiv:2510.01384*, 2025b.

Jianan Li, Keisuke Yanagisawa, Masatake Sugita, Takuya Fujie, Masahito Ohue, and Yutaka Akiyama. Cycpeptmpdb: a comprehensive database of membrane permeability of cyclic peptides. *Journal of Chemical Information and Modeling*, 63(7):2240–2250, 2023.

Xiner Li, Masatoshi Uehara, Xingyu Su, Gabriele Scalia, Tommaso Biancalani, Aviv Regev, Sergey Levine, and Shuiwang Ji. Dynamic search for inference-time alignment in diffusion models. *arXiv preprint arXiv:2503.02039*, 2025.

Yanyan Li, Honghong Zhou, Xiaomin Chen, Yu Zheng, Quan Kang, Di Hao, Lili Zhang, Tingrui Song, Huaxia Luo, Yajing Hao, et al. Smprot: a reliable repository with comprehensive annotation of small proteins identified from ribosome profiling. *Genomics, proteomics & bioinformatics*, 19 (4):602–610, 2021.

Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. *arXiv preprint arXiv:2410.06264*, 2024.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

Yong-Hyun Park, Chieh-Hsin Lai, Satoshi Hayakawa, Yuhta Takida, and Yuki Mitsufuji. Jump your steps: Optimizing sampling schedule of discrete diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2024.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Avishek Joey Bose, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025a.

Fred Zhangzhi Peng, Zachary Bezemek, Jarrid Rector-Brooks, Shuibai Zhang, Anru R Zhang, Michael Bronstein, Avishek Joey Bose, and Alexander Tong. Planner aware path learning in diffusion language models training. *arXiv preprint arXiv:2509.23405*, 2025b.

Vignav Ramesh and Morteza Mardani. Test-time scaling of diffusion models via noise trajectory search. *arXiv preprint arXiv:2506.03164*, 2025.

Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Cheng-Hao Liu, Sarthak Mittal, Nouha Dziri, Michael M. Bronstein, Pranam Chatterjee, Alexander Tong, and Joey Bose. Steering masked discrete diffusion models via discrete denoising posterior prediction. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yinuo Ren, Haoxuan Chen, Yuchen Zhu, Wei Guo, Yongxin Chen, Grant M Rotskoff, Molei Tao, and Lexing Ying. Fast solvers for discrete diffusion models: Theory and applications of high-order algorithms. *arXiv preprint arXiv:2502.00234*, 2025.

Kevin Rojas, Ye He, Chieh-Hsin Lai, Yuta Takida, Yuki Mitsufuji, and Molei Tao. Theory-informed improvements to classifier-free guidance for discrete diffusion models. *arXiv preprint arXiv:2507.08965*, 2025.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 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. *arXiv preprint arXiv:2412.10193*, 2024.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P de Almeida, Alexander M Rush, Thomas PIERROT, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. *arXiv preprint arXiv:2503.02819*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Peptune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. *42nd International Conference on Machine Learning*, 2025a.

Sophia Tang, Yuchen Zhu, Molei Tao, and Pranam Chatterjee. Tr2-d2: Tree search guided trajectory-aware fine-tuning for discrete diffusion. *arXiv preprint arXiv:2509.25171*, 2025b.

Runchu Tian, Junxia Cui, Xueqiang Xu, Feng Yao, and Jingbo Shang. Finish first, perfect later: Test-time token-level cross-validation for diffusion large language models. *arXiv preprint arXiv:2510.05090*, 2025.

Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.

Xinyou Wang, Zaixiang Zheng, Fei YE, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. In *Forty-first International Conference on Machine Learning*, 2024a.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Dplm-2: A multimodal diffusion protein language model. *arXiv preprint arXiv:2410.13782*, 2024b.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*, 2024.

Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. *arXiv preprint arXiv:2407.21243*, 2024.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

Renping Zhou, Zanlin Ni, Tianyi Chen, Zeyu Liu, Yang Yue, Yulin Wang, Yuxuan Wang, Jingshu Liu, and Gao Huang. Co-grpo: Co-optimized group relative policy optimization for masked diffusion model. *arXiv preprint arXiv:2512.22288*, 2025.

Yuchen Zhu, Wei Guo, Jaemoo Choi, Guan-Horng Liu, Yongxin Chen, and Molei Tao. Mdns: Masked diffusion neural sampler via stochastic optimal control. *arXiv preprint arXiv:2508.10684*, 2025a.

Yuchen Zhu, Wei Guo, Jaemoo Choi, Petr Molodyk, Bo Yuan, Molei Tao, and Yongxin Chen. Enhancing reasoning for diffusion llms via distribution matching policy optimization. *arXiv preprint arXiv:2510.08233*, 2025b.

OUTLINE OF APPENDIX

In App A, we discuss related works in discrete diffusion, any-length generation, and fine-tuning and inference optimization strategies. In App B, we provide a comprehensive theoretical background on any-length masked diffusion, continuous-time Markov chains (CTMCs), and stochastic optimal control. App C provides rigorous theoretical proofs for our method. We provide the details for the multi-objective therapeutic peptide generation in App D. Finally, pseudo-code for training and inference with A2D2 are provided in E.

Notation We denote the CTMC of a diffusion trajectory as $\boldsymbol{X}_{0:1} := (\boldsymbol{X}_t)_{t\in[0,1]}$ which lies in a discrete state space $\mathcal{X} \in \{1, \dots, V\}^L$ of sequences of length L and vocabulary size V. A any-length discrete diffusion model is characterized by a path measure $\mathbb{P} \in \mathcal{P}(\mathcal{X})$ over trajectories that follow an joint generator $\boldsymbol{A}_t = \boldsymbol{Q}_t + \boldsymbol{R}_t$, where $\boldsymbol{Q}_t$ is the unmasking rate and $\boldsymbol{R}_t$ is the insertion rate. Specifically, we denote the pre-trained discrete diffusion model as $\mathbb{P}^{\text{pre}}$ and the optimal target model as $\mathbb{P}^\star$. For a corrupted sequence $\boldsymbol{x}_t$ at time t, the unmasking posterior is given by $f_\theta(\boldsymbol{x}, t)$ with indices $[\ell, \boldsymbol{v}]$ denoting the probability of a single token at position ℓ and the insertion rate is given by $g_\theta(\boldsymbol{x}_t, t)[\ell]$ with indices $[\ell]$ indicating the insertion expectation between positions $\ell - 1$ and ℓ of the sequence.

A RELATED WORK

Inference-Time Scaling of Discrete Diffusion Inference-time scaling of diffusion models seeks to maximize the capabilities of pre-trained diffusion models for specialized tasks during inference, such as sampling from a reward-tilted distribution (Skreta et al., 2025; Singhal et al., 2025; Chen et al., 2025), constrained sampling (Fishman et al., 2023), or entropy-maximization (De Santi et al., 2025). Specifically in the discrete state space, search-based methods (Tang et al., 2025a;b; Li et al., 2025; Ramesh & Mardani, 2025), importance-weighting techniques (Chatterjee & Diaconis, 2018), reward-gradient methods (Song et al., 2020; Bansal et al., 2023), and classifier-based and classifier-free guidance (Nisonoff et al., 2024; Rector-Brooks et al., 2025; Wang et al., 2024a; Schiff et al., 2025; Rojas et al., 2025; Guo et al., 2024) have been explored.

Optimization of Inference Schedule for Discrete Diffusion Closest to our work is PRISM (Kim et al., 2025b), which introduces *per-token quality* as a learned measure that determines whether to re-mask already unmasked tokens during inference. However, their framework remains limited to fixed-length masked diffusion for optimizing sampling of the data distribution. Co-GRPO (Zhou et al., 2025), which introduces a reward fine-tuning strategy for fixed-length masked diffusion that simultaneously optimizes the inference schedule and policy model with a shared reward signal with group-relative policy optimization (GRPO). Jump Your Steps (JYS) (Park et al., 2024) optimizes the times in which discrete jumps are made during inference by minimizing a KL upper bound on the compounding decoding error for fixed-length masked diffusion models. Related inference-time mechanisms include *planner*-guided decoding, which learns a planner to choose which tokens to reveal at each step (Liu et al., 2024; Peng et al., 2025a;b), as well as remasking and corrector-style methods that iteratively revise intermediate predictions to improve sample quality (Wang et al., 2025; Zhao et al., 2024; Tian et al., 2025).

B EXTENDED THEORETICAL BACKGROUND

B.1 ANY-LENGTH MASKED DIFFUSION

In this work, we build upon the any-length masked diffusion model framework introduced in Kim et al. (2025a), which leverages a pair of joint stochastic interpolants for the insertion and unmasking schedules. Here, we will provide a comprehensive background on the any-length MDM framework and highlight its distinction from standard fixed-length MDMs.

Forward Noising Process In standard MDM, the forward process involves applying mask tokens gradually across the sequence until the sequence is fully masked at time $t = 1$. For any-length

MDMs, the forward process consists not only of the masking step but also a **deletion** step, which serves as the inverse operation for the insertion step during inference. Therefore, we must define both the insertion time t_i^ℓ and the unmasking time t_u^ℓ for each token position $\boldsymbol{x}^\ell$ as:

$$t_i^\ell \sim \dot{\alpha}_t dt, \quad t_u^\ell \sim \mathbf{1}[t \geq t_u^\ell] \frac{\dot{\beta}}{1-\beta_{t_i}} dt, \quad \boldsymbol{x}_t^\ell = \begin{cases} (\text{empty}), & 0 < t < t_i^\ell \\ \boldsymbol{M}, & t_i^\ell \leq t \leq t_u^\ell \\ \boldsymbol{x}_1^\ell, & t_u^\ell \leq t \leq 1 \end{cases} \tag{22}$$

where (empty) indicates a position that has not been inserted yet. We denote the index of a token in the partial subsequence with s_t given by:

$$s_t := \{\ell \in \{1, \ldots, \text{len}(\boldsymbol{x}_1)\} |\ t_i^\ell \leq t\} \tag{23}$$

where $s_t[\ell]$ is the index of the token $\boldsymbol{x}_1^\ell$ in $\boldsymbol{x}_t$ at time t and we have $\text{len}(s_t) \leq \text{len}(\boldsymbol{x}_t)$.

Parameterization Just like fixed-length MDMs, the masking step in any-length MDMs has an inverse unmasking step that is parameterized with an **unmasking posterior** $f_\theta(\boldsymbol{x}_t, t)[\ell] : \mathcal{V}^L \times [0,1] \to \Delta^D$ that predicts the distribution over the token vocabulary for each masked position. This model is trained with the standard denoising cross-entropy loss defined as:

$$\mathcal{L}_{\text{unmask}}(\theta; \boldsymbol{x}_1) = \mathop{\mathbb{E}}_{t\sim\mathcal{U}(0,1)} \mathop{\mathbb{E}}_{\boldsymbol{x}_t\sim p_t(\cdot|\boldsymbol{x}_1)} \left[-\frac{\dot{\beta}_t}{1-\beta_t} \sum_{\ell:\boldsymbol{x}_t^\ell = \boldsymbol{M}} \log f_\theta(\boldsymbol{x}_t, t)[\ell, \boldsymbol{x}_1^\ell] \right] \tag{24}$$

Since any-length MDMs contain the additional deletion step, we parameterize the inverse **insertion** step with a **insertion expectation** $g_\theta(\boldsymbol{x}_t, t)[\ell] : \mathcal{V}^L \times [0,1] \to \mathbb{R}_{>0}$ which predicts the number of tokens that still need to be inserted between tokens $\boldsymbol{x}_t^{\ell-1}$ and $\boldsymbol{x}_t^\ell$. This model is trained by minimizing the scalar **Bregman divergence** $\phi(x, y) = y - x \log y$ between the predicted insertions and the true number of tokens that need to be inserted at each position:

$$\mathcal{L}_{\text{insert}}(\theta; \boldsymbol{x}_1) = \mathop{\mathbb{E}}_{t\sim\mathcal{U}(0,1)} \mathop{\mathbb{E}}_{\boldsymbol{x}_t\sim p_t(\cdot|\boldsymbol{x}_1)} \left[-\frac{\dot{\alpha}_t}{1-\alpha_t} \sum_{\ell=0}^{\text{len}(\boldsymbol{x}_t)} \phi\left(s_t[\ell] - s_t[\ell-1], g_\theta(\boldsymbol{x}_t, t)[\ell]\right) \right] \tag{25}$$

Minimizing both losses $\mathcal{L}_{\text{unmask}}(\theta)$ and $\mathcal{L}_{\text{insert}}(\theta)$ yields the *unique* optimal value which defines the rate matrices of the reverse process given by:

$$\boldsymbol{R}_t(\boldsymbol{x}, \boldsymbol{x}^{\triangleleft_\ell \boldsymbol{M}}) = \frac{\dot{\alpha}_t}{1-\alpha_t} \cdot \mathbb{E}[s_t[\ell] - s_t[\ell-1] - 1 |\ \boldsymbol{x}_t = \boldsymbol{x}] = \frac{\dot{\alpha}_t}{1-\alpha_t} \cdot g_\theta(\boldsymbol{x}, t)[\ell] \qquad \text{(Insertion Rate)}$$

$$\boldsymbol{Q}_t(\boldsymbol{x}, \boldsymbol{x}^{\ell \leftarrow \boldsymbol{d}}) = \frac{\dot{\beta}_t}{1-\beta_t} \cdot \mathbb{P}(\boldsymbol{x}_1^{s_t[\ell]} = \boldsymbol{d} | \boldsymbol{x}_t = \boldsymbol{x}) = \frac{\dot{\beta}_t}{1-\beta_t} \cdot f_\theta(\boldsymbol{x}, t)[\ell, \boldsymbol{d}] \qquad \text{(Unmasking Rate)}$$

Adaptive Inference A **crucial feature** of any-length MDMs is that the values of the rate matrix are *decoupled from the choice of masking schedule* β_t. This means that the unmasking posterior learned by $f_\theta(\boldsymbol{x}_t, t)$ and insertion expectation $g_\theta(\boldsymbol{x}_t, t)$ captures all possible unmasking transitions that interpolate between the empty prior p_0 and the target distribution p_{target}.

C Theoretical Proofs

C.1 Observations

Throughout the theoretical proofs, we make the following **key observations**:

(i) The unmasking rate $\boldsymbol{Q}_t$ and insertion rate $\boldsymbol{R}_t$ do not overlap, such that:

$$\boldsymbol{Q}_t(\boldsymbol{x}, \boldsymbol{y}) \boldsymbol{R}_t(\boldsymbol{x}, \boldsymbol{y}) = 0, \quad \forall t, \boldsymbol{x}, \boldsymbol{y} \tag{26}$$

which follows from the fact that insertion and unmasking actions are disjoint.

(ii) The total generator $\boldsymbol{A}_t$ decomposes additively:

$$\boldsymbol{A}_t(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{Q}_t(\boldsymbol{x}, \boldsymbol{y}) + \boldsymbol{R}_t(\boldsymbol{x}, \boldsymbol{y}), \quad \forall t, \boldsymbol{x}, \boldsymbol{y} \tag{27}$$

(iii) Both insertion and unmasking rates are *valid* such that:

$$\boldsymbol{Q}_t(\boldsymbol{x}, \boldsymbol{x}) = -\sum_{y \neq x} \boldsymbol{Q}_t(\boldsymbol{x}, \boldsymbol{y}), \quad \sum_{y \neq x} \boldsymbol{Q}_t(\boldsymbol{x}, \boldsymbol{y}) < \infty \tag{28}$$

$$\boldsymbol{R}_t(\boldsymbol{x}, \boldsymbol{x}) = -\sum_{y \neq x} \boldsymbol{R}_t(\boldsymbol{x}, \boldsymbol{y}), \quad \sum_{y \neq x} \boldsymbol{R}_t(\boldsymbol{x}, \boldsymbol{y}) < \infty \tag{29}$$

which ensures that the CTMC is well-defined such that reward tilting preserves the generator structure.

(iv) The tilted generators $\boldsymbol{Q}_t^\star$ and $\boldsymbol{R}_t^\star$ and the reference generators $\boldsymbol{Q}_t^0$ and $\boldsymbol{R}_t^0$ have the same support, such that:

$$\boldsymbol{Q}_t^0(\boldsymbol{x}, \boldsymbol{y}) = 0 \implies \boldsymbol{Q}_t^\star(\boldsymbol{x}, \boldsymbol{y}) = 0 \tag{30}$$

$$\boldsymbol{R}_t^0(\boldsymbol{x}, \boldsymbol{y}) = 0 \implies \boldsymbol{R}_t^\star(\boldsymbol{x}, \boldsymbol{y}) = 0 \tag{31}$$

which ensures that the path measures $\mathbb{P}^0$ and $\mathbb{P}^\star$ are mutually absolutely continuous.

C.2 Proof of Proposition 3.1

Proposition 3.1 (Unique Minimizer of Unmasking Quality Loss)**.** *The unique minimizer of the unmsaking quality loss $\mathcal{L}_{UQL}(\phi)$ is the true unmasking quality:*

$$\mu_\star = \mu_{\phi^\star}, \quad \textit{where} \quad \phi^\star = \arg\min_\phi \mathcal{L}_{UQL}(\phi) \tag{7}$$

Proof. First, we recall the Unmasking Quality Loss $\mathcal{L}_{\text{UQL}}$ defined in (6) as:

$$\mathcal{L}_{\text{UQL}}(\phi; \boldsymbol{x}_1) := \mathop{\mathbb{E}}_{t \sim \mathcal{U}(0,1)} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}_t \sim p_t(\cdot|\boldsymbol{x}_1)} \mathop{\mathbb{E}}_{\boldsymbol{y} \sim p_{s|t}(\cdot|\tilde{\boldsymbol{x}}_t)} \left[\sum_{\ell \in \mathcal{M}} \text{BCE}\left(\mathbf{1}[\boldsymbol{y}^\ell = \boldsymbol{x}_1^\ell], \mu_\phi^\ell(\boldsymbol{y}) \right) \right] \tag{32}$$

Given a clean sequence $\boldsymbol{x}_1 \sim p_{\text{data}}$, let $\tilde{\boldsymbol{x}}_t$ be a sample from the posterior $\tilde{\boldsymbol{x}}_t \sim p_t(\cdot|\boldsymbol{x}_1)$. Then, we sample a subset $\mathcal{M}_t$ of masked indices in $\tilde{\boldsymbol{x}}_t$ to unmask to obtain $\boldsymbol{y}$ where $\forall \ell \in \mathcal{M}_t$, we sample $\boldsymbol{y}^\ell \sim f_\theta(\tilde{\boldsymbol{x}}_t, t)[\ell]$.

Fix a position $\ell \in \mathcal{M}_t$ and consider the random binary variable b_ℓ defined by categorically sampling from $f_\theta(\tilde{\boldsymbol{x}}_t, t)[\ell]$ and the model predicted token quality $c_\ell(\boldsymbol{y}) := \mu_\phi^\ell(\boldsymbol{y})$:

$$b^\ell := \mathbf{1}[\boldsymbol{y}^\ell = \boldsymbol{x}_1^\ell] \in \{0, 1\}, \quad c_\ell(\boldsymbol{y}) := \mu_\phi^\ell(\boldsymbol{y}) \in [0, 1] \tag{33}$$

For every position $\ell \in \mathcal{M}_t$, we minimize the following loss:

$$\mathcal{L}_{\text{UQL}}(\phi; \ell) := \mathbb{E}_{\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_t, \boldsymbol{y}} \left[\text{BCE}\left(b_\ell, c_\ell(\boldsymbol{y}) \right) \right] \tag{34}$$

Applying the law of total expectation, we have:

$$\mathbb{E}_{\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_t, \boldsymbol{y}} \left[\text{BCE}\left(b_\ell, c_\ell(\boldsymbol{y}) \right) \right] = \mathbb{E}_{\boldsymbol{y}} \underbrace{\left[\mathbb{E}_{\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_t} \left[\text{BCE}(b_\ell, c_\ell(\boldsymbol{y})) \middle| \boldsymbol{y} \right] \right]}_{:= D_\ell(\boldsymbol{y}, c_\ell)} \tag{35}$$

Let us define the conditional random variable:

$$q_\ell(\boldsymbol{y}) := p(b_\ell = 1 | \boldsymbol{y}) = p(\boldsymbol{y}^\ell = \boldsymbol{x}_1^\ell | \boldsymbol{y}) = \mathbb{E}_{\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_t} \left[b_\ell | \boldsymbol{y} \right] \tag{36}$$

Then, we can write the BCE loss $D_\ell(\boldsymbol{y}, c)$ as:

$$D_\ell(\boldsymbol{y}, c_\ell) = -q_\ell(\boldsymbol{y}) \log c_\ell - (1 - q_\ell(\boldsymbol{y})) \log(1 - c_\ell) \tag{37}$$

Differentiating with respect to the parameters ϕ denoted as c_ℓ and setting the derivative equal to 0, we get that the minimizer is:

$$\frac{d}{dc_\ell} D_\ell(\boldsymbol{y}, c_\ell) = -\frac{q_\ell(\boldsymbol{y})}{c_\ell} + \frac{1 - q_\ell(\boldsymbol{y})}{1 - c_\ell} \implies \frac{q_\ell(\boldsymbol{y})}{c_\ell^\star} = \frac{1 - q_\ell(\boldsymbol{y})}{1 - c_\ell^\star} \implies c_\ell^\star = q_\ell(\boldsymbol{y}) \tag{38}$$

Taking the second derivative with respect to c_ℓ, we show that this minimizer is unique:

$$\frac{d^2}{dc_\ell^2} D_\ell(\boldsymbol{y}, c_\ell) = \frac{q_\ell(\boldsymbol{y})}{c_\ell^2} + \frac{1 - q_\ell(\boldsymbol{y})}{(1 - c_\ell)^2} \geq 0 \tag{39}$$

Since we defined the true unmasking quality as $\mu_\star^\ell(\boldsymbol{y}) := p(\boldsymbol{y}^\ell = \boldsymbol{x}_1^\ell | \boldsymbol{y})$, the minimizer $c^\star$ exactly matches the true unmasking posterior:

$$c_\ell^\star = q_\ell(\boldsymbol{y}) = \mu_\star^\ell(\boldsymbol{y}) = \arg\min_{c_\ell \in [0,1]} \{D_\ell(\boldsymbol{y}, c_\ell)\} \tag{40}$$

Since for all pairs $(\boldsymbol{y}, \ell)$, we have shown that the BCE loss $D_\ell(\boldsymbol{y}, c_\ell)$ is minimized at $\mu_\phi^\ell(\boldsymbol{y}) = \mu_\star^\ell(\boldsymbol{y})$, so the function that minimizes the full objective $\mathcal{L}_{\text{UQL}}(\phi) := \mathbb{E}_{\boldsymbol{y}}[D_\ell(\boldsymbol{y}, c_\ell)]$ must be the true function $\mu_\star$. □

C.3 Proof of Proposition 3.3

Proposition 3.3 (Minimizer of Insertion Quality Loss)**.** *The unique minimizer of the insertion quality loss $\mathcal{L}_{IQL}(\phi)$ is the true insertion quality:*

$$\nu_\star = \nu_{\phi^\star}, \quad \textit{where} \quad \phi^\star = \arg\min_\phi \mathcal{L}_{IQL}(\phi) \tag{14}$$

Proof. First, we recall the Insertion Quality Loss $\mathcal{L}_{\text{IQL}}$ defined in (13) as:

$$\mathcal{L}_{\text{IQL}}(\phi) := \mathbb{E}_{\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_t, \boldsymbol{y}} \left[\sum_{i \in \mathcal{I}} \text{BCE}\big(\nu_\star^i(\boldsymbol{y}), \nu_\phi^i(\boldsymbol{y})\big) \right] \tag{41}$$

where the BCE is defined as:

$$\text{BCE}(b, c) = -b \log c - (1 - b) \log(1 - c), \quad b, c \in [0, 1] \tag{42}$$

Since the sum $\sum_{i \in \mathcal{I}} \text{BCE}\big(\nu_\star^i(\boldsymbol{y}), \nu_\phi^i(\boldsymbol{y})\big)$ is inside the expectation, and the loss is calculated independently for each position i given $\nu_\phi^i(\boldsymbol{y})$, minimizing $\mathcal{L}_{\text{IQL}}(\phi)$ is equivalent to minimizing the following for every inserted position i:

$$\mathcal{L}_{\text{IQL}}(\phi; i) := \mathbb{E}_{\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_t, \boldsymbol{y}} \left[\text{BCE}\big(\nu_\star^i(\boldsymbol{y}), \nu_\phi^i(\boldsymbol{y})\big) \right] \tag{43}$$

Applying the law of total expectation, we have:

$$\mathbb{E}_{\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_t, \boldsymbol{y}} \left[\text{BCE}\big(\nu_\star^i(\boldsymbol{y}), \nu_\phi^i(\boldsymbol{y})\big) \right] = \mathbb{E}_{\boldsymbol{y}} \underbrace{\left[\mathbb{E}_{\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_t} \left[\text{BCE}\big(\nu_\star^i(\boldsymbol{y}), \nu_\phi^i(\boldsymbol{y})\big) \middle| \boldsymbol{y} \right] \right]}_{:= B_i(\boldsymbol{y}, c)} \tag{44}$$

where we define $c_i := \nu_\phi^i(\boldsymbol{y}) \in [0, 1]$ as the scalar predicted insertion quality at position i and $B_i(\boldsymbol{y}, c_i)$ as the prediction error given the input sequence $\boldsymbol{y}$ and predicted quality c_i. Then, it suffices to prove that for all $(\boldsymbol{y}, i)$, the prediction error is *uniquely minimized* at $c_i = \nu_\star^i(\boldsymbol{y})$.

Fixing a pair $(\boldsymbol{y}, i)$, we can define:

$$b_i := \nu_\star^i(\boldsymbol{y}) \in [0, 1], \quad c_i := \nu_\phi^i(\boldsymbol{y}) \in [0, 1] \tag{45}$$

which yields the prediction error objective:

$$B_i(\boldsymbol{y}, c_i) = -b_i \log c_i - (1 - b_i) \log(1 - c_i) \tag{46}$$

Differentiating with respect to c_i (parameterized model), we have:

$$\frac{d}{dc_i} B_i(\boldsymbol{y}, c_i) = -\frac{b_i}{c_i} + \frac{1 - b_i}{1 - c_i} \tag{47}$$

and setting the derivative to zero, we get the minimizer:

$$-\frac{b_i}{c_i^\star} + \frac{1 - b_i}{1 - c_i^\star} = 0 \implies \frac{b_i}{c_i^\star} = \frac{1 - b_i}{1 - c_i^\star} \implies c_i^\star = b_i \tag{48}$$

To prove uniqueness, we take the second derivative:

$$\frac{d^2}{dc_i^2} B_i(\boldsymbol{y}, c_i) = \frac{b_i}{c_i^2} + \frac{1 - b_i}{(1 - c_i)^2} \geq 0 \tag{49}$$

So, the objective function is convex in c_i and the minimizer is unique. Therefore, we conclude that:

$$b_i = \nu_\star^i(\boldsymbol{y}) = \underset{c_i \in [0,1]}{\arg\min} \{B_i(\boldsymbol{y}, c_i)\} \tag{50}$$

We have shown that for all $(\boldsymbol{y}, i)$, the prediction error is uniquely minimized at $\nu_\phi^i(\boldsymbol{y}) = \nu_\star^i(\boldsymbol{y})$, the only function ν_ϕ that minimizes the outer expectation $\mathbb{E}_{\boldsymbol{y}}[B_i(\boldsymbol{y}, c_i)]$ is the true function $\nu_\star$. □

C.4 Proof of Proposition 3.2

Proposition C.1 (Optimal Parallel Unmasking as Maximizing the Unmasking Quality)**.** *Assume that in a parallel unmasking step on indices $\mathcal{M}_t = \{\ell_k\}_{k=1}^K$, the unmasked tokens are conditionally independent given the unchanged context $\bar{\boldsymbol{x}}_s$ and the current state $\boldsymbol{x}_s$, i.e. $p(\boldsymbol{x}_t^{\mathcal{M}_t} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) = \prod_{k=1}^K p(\boldsymbol{x}_t^{\ell_k} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s)$. If we define the unmasking quality $\mu_\star^{\ell_k}(\boldsymbol{x}_t) := p(\boldsymbol{x}_t^{\ell_k} = \boldsymbol{x}_1^{\ell_k} | \boldsymbol{x}_t^{\ell_k \leftarrow \boldsymbol{M}})$ and note that $p(\boldsymbol{x}_t^{\ell_k} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) = p(\boldsymbol{x}_t^{\ell_k} | \boldsymbol{x}_t^{\ell_k \leftarrow \boldsymbol{M}})$, then*

$$p(\boldsymbol{x}_t^{\mathcal{M}_t} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) = \prod_{k=1}^K \nu_\star^{\ell_k}(\boldsymbol{x}_t) \tag{51}$$

Furthermore, the compounding parallelization error can be written as the KL divergence

$$\mathcal{E}_{\text{unmask}}^{\text{CPE}}(s \to t | \boldsymbol{x}_s) := D_{\text{KL}} \left(p(\boldsymbol{x}_t^{\mathcal{M}_t} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) \,\middle\|\, \prod_{k=1}^K p(\boldsymbol{x}_t^{\ell_k} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) \right) \geq 0 \tag{52}$$

and admits the decomposition

$$\mathcal{E}_{\text{unmask}}^{\text{CPE}}(s \to t | \boldsymbol{x}_s) = \mathbb{E}_{p_t} \left[\log p(\boldsymbol{x}_t^{\mathcal{M}_t} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) \right] - \sum_{k=1}^K \mathbb{E}_{p_t} \left[\log \mu_\star^{\ell_k}(\boldsymbol{x}_t) \right] \tag{53}$$

with equality to 0 if and only if the KL divergence is 0 (i.e., optimal parallel decoding).

Proof. First, we recall the definition of token quality. Given a partially masked and deleted sequence $\boldsymbol{x}_t$, the unmasking quality of each token $\boldsymbol{x}_t^\ell$ is the probability that given the sequence masked at position ℓ, denoted $\tilde{\boldsymbol{x}}_s := \boldsymbol{x}_t^{\ell \leftarrow \boldsymbol{M}}$, we sample $\tilde{\boldsymbol{x}}_s^\ell = \boldsymbol{x}_t^\ell$ via the unmasking posterior $f_\theta(\tilde{\boldsymbol{x}}_s, s)$. This is written as:

$$\mu_\star^\ell(\tilde{\boldsymbol{x}}_s) = p(\tilde{\boldsymbol{x}}_s^\ell = \boldsymbol{x}_t^\ell | \boldsymbol{x}_t^{\ell \leftarrow \boldsymbol{M}}) \tag{54}$$

Consider a unmasking step $\boldsymbol{x}_s \to \boldsymbol{x}_t$ where a subset of K tokens with indices $\mathcal{M}_t := \{\ell_k\}_{k=1}^K$ are unmasked where $\bar{\boldsymbol{x}}_s$ is the subset of the sequence that remains unchanged from $s \to t$. Let the joint conditional and the product of marginals conditioned on $\boldsymbol{x}_s$ be denoted as:

$$p_{\mathcal{M}_t}(\cdot) := p(\boldsymbol{x}_t^{\mathcal{M}_t} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s), \quad q_{\mathcal{M}_t}(\cdot) := \prod_{k=1}^K p(\boldsymbol{x}_t^{\ell_k} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) \tag{55}$$

Then, the compounding parallelization error (CPE) of the step is given by:

$$\mathcal{E}_{\text{unmask}}^{\text{CPE}}(s \to t | \boldsymbol{x}_s) = D_{\text{KL}} \left(p_{\mathcal{M}_t} \| q_{\mathcal{M}_t} \right) \geq 0 \tag{56}$$

where $\mathcal{E}_{\text{unmask}}^{\text{CPE}}(s \to t | \boldsymbol{x}_s) = 0$ if and only if $p_{\mathcal{M}_t} = q_{\mathcal{M}_t}$ and all the unmasked tokens are *conditionally independent* such that the product of the marginal probabilities and the joint probability is equal given $(\bar{\boldsymbol{x}}_s, \boldsymbol{x}_s)$. Expanding the KL divergence yields:

$$\mathcal{E}_{\text{unmask}}^{\text{CPE}}(s \to t | \boldsymbol{x}_s) = \mathbb{E}_{\boldsymbol{x}_t^{\mathcal{M}_t} \sim p_{\mathcal{M}_t}} \left[\log p(\boldsymbol{x}_t^{\mathcal{M}_t} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) - \sum_{k=1}^K \log p(\boldsymbol{x}_t^{\ell_k} | \bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) \right] \geq 0 \tag{57}$$

In the case where all unmasked tokens are conditionally independent, we have:

$$p(\boldsymbol{x}_t^{\mathcal{M}_t}|\bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) = \prod_{k=1}^{K} p(\boldsymbol{x}_t^{\ell_k}|\bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) \tag{58}$$

In addition, under conditional independence of $\mathcal{M}_t$, the probability of a particular unmasked token $\boldsymbol{x}_t^{\ell_k}$ should be the same given the context $\boldsymbol{x}_s$ and given the context with only ℓ_k masked $\boldsymbol{x}_t^{\ell_k \leftarrow \boldsymbol{M}}$, which aligns with the definition of the true unmasking quality. Therefore,

$$p(\boldsymbol{x}_t^{\ell_k}|\bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) = p(\boldsymbol{x}_t^{\ell_k}|\boldsymbol{x}_t^{\ell_k \leftarrow \boldsymbol{M}}) =: \mu_\star^{\ell}(\boldsymbol{x}_t) \tag{59}$$

Substituting (59) into the optimal parallel unmasking condition in (58), we have:

$$p(\boldsymbol{x}_t^{\mathcal{M}_t}|\bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) = \prod_{k=1}^{K} p(\boldsymbol{x}_t^{\ell_k}|\bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) = \prod_{k=1}^{K} \mu_\star^{\ell}(\boldsymbol{x}_t) \tag{60}$$

which means that maximizing the product of token qualities maximizes the probability of optimal parallel unmasking. We can also derive a tractable Monte Carlo estimator for the CPE in (58) as:

$$\widehat{\mathcal{E}}^{\text{CPE}}_{\text{unmask}}(s \to t|\boldsymbol{x}_s) = \log p(\boldsymbol{x}_t^{\mathcal{M}_t}|\bar{\boldsymbol{x}}_s, \boldsymbol{x}_s) - \sum_{k=1}^{K} \log p(\boldsymbol{x}_t^{\ell_k}|\bar{\boldsymbol{x}}_s, \boldsymbol{x}_s), \quad \mathcal{E}^{\text{CPE}}_{\text{unmask}} = \mathbb{E}_{p_t}\left[\widehat{\mathcal{E}}^{\text{CPE}}_{\text{unmask}}\right] \geq 0 \tag{61}$$

which concludes our proof. □

C.5 Proof of Proposition 3.4

Proposition 3.4 (Insertion Quality as an Upper Bound on Reconstruction via Insertions)**.** *Consider a clean target sequence $\boldsymbol{x}_1 \sim p_{target}$ and an intermediate sequence $\boldsymbol{x}_s \sim p_s(\cdot|\boldsymbol{x}_1)$. Given a set of indices of inserted masks at each gap $\mathcal{I}_t$ which yields $\boldsymbol{x}_t^{\mathcal{I}_t}$, the probability of reconstructing $\boldsymbol{x}_1$ is upper bounded by the product of insertion qualities:*

$$p(\boldsymbol{x}_t^{\mathcal{I}_t} = \boldsymbol{x}_1^{\mathcal{I}_t}|\boldsymbol{x}_t) \leq \prod_{i \in \mathcal{I}_t} \mu_\star^i(\boldsymbol{x}_t) \approx \prod_{i \in \mathcal{I}_t} \mu_\phi^i(\boldsymbol{x}_t) \tag{15}$$

assuming that the unmasking posterior for each inserted mask is conditionally independent given $\boldsymbol{x}_t$.

Proof. First, we recall the definition of insertion quality as the sum of the probabilities of all possible tokens that exist within the gap in the ground truth sequence, defined as:

$$\nu_\star(\boldsymbol{y}) := \max_{s_t[\ell-1] \leq i \leq s_t[\ell]} \left\{p(\boldsymbol{y}^\ell = \boldsymbol{x}_1^i|\boldsymbol{y})\right\} \tag{62}$$

Consider a target sequence $\boldsymbol{x}_1 \sim p_{\text{target}}$. After an insertion step $(\mathcal{I}_t, \boldsymbol{x}_t) \sim g_\theta(\boldsymbol{x}_s, s)$, we unmask the masked tokens to reconstruct $\boldsymbol{x}_1$. For this proof, we assume that the unmasking posterior factorizes across all masked tokens, which gives the equality:

$$p(\boldsymbol{x}_t^{\mathcal{I}_t} = \boldsymbol{x}_1^{\mathcal{I}_t}|\boldsymbol{x}_t) = \prod_{i \in \mathcal{I}} p(\boldsymbol{x}_t^i = \boldsymbol{x}_1^i|\boldsymbol{x}_t) \tag{63}$$

$$\log p(\boldsymbol{x}_t^{\mathcal{I}_t} = \boldsymbol{x}_1^{\mathcal{I}_t}|\boldsymbol{x}_t) = \sum_{i \in \mathcal{I}} \log p(\boldsymbol{x}_t^i = \boldsymbol{x}_1^i|\boldsymbol{x}_t) \tag{64}$$

Since we define the **insertion quality** $\mu_\star^i$ at an inserted mask position i as the largest probability mass assigned by the unmasking posterior to any of the ground truth tokens at the gap $\mathcal{G}_i := [s_t[i-1], s_t[i]]$, we can write:

$$p(\boldsymbol{x}_t^i = \boldsymbol{x}_1^{\mathcal{G}_i}|\boldsymbol{x}_t) \leq \mu_\star^i(\boldsymbol{x}_t) \implies \log p(\boldsymbol{x}_t^i = \boldsymbol{x}_1^{\mathcal{G}_i}|\boldsymbol{x}_t) \leq \log \mu_\star^i(\boldsymbol{x}_t) \tag{65}$$

Assuming conditional independence across all inserted masks given $\boldsymbol{x}_t$,we get:

$$p(\boldsymbol{x}_t^{\mathcal{I}_t} = \boldsymbol{x}_1^{\mathcal{I}_t}|\boldsymbol{x}_t) = \prod_{i \in \mathcal{I}} p(\boldsymbol{x}_t^i = \boldsymbol{x}_1^{\mathcal{G}_i}|\boldsymbol{x}_t) \leq \prod_{i \in \mathcal{I}} \mu_\star^i(\boldsymbol{x}_t) \tag{66}$$

where $\boldsymbol{x}_t^{\mathcal{I}_t} = \boldsymbol{x}_1^{\mathcal{I}_t}$ means that target tokens are aligned to inserted slots. Taking the expectation over $(\mathcal{I}_t, \boldsymbol{x}_t) \sim p(\mathcal{I}_t, \boldsymbol{x}_t|\boldsymbol{x}_s)$, we have:

$$\mathbb{E}_{p(\mathcal{I}_t, \boldsymbol{x}_t|\boldsymbol{x}_s)}\left[p(\boldsymbol{x}_t^{\mathcal{I}_t} = \boldsymbol{x}_1^{\mathcal{I}_t}|\boldsymbol{x}_t)\right] \leq \mathbb{E}_{p(\mathcal{I}_t, \boldsymbol{x}_t|\boldsymbol{x}_s)}\left[\prod_{i\in\mathcal{I}} \mu_\star^i(\boldsymbol{x}_t)\right] \tag{67}$$

Therefore, maximizing $\prod_{i\in\mathcal{I}} \mu_\star^i(\boldsymbol{x}_t)$ maximizes a tractable upper bound on the probability that all inserted slots are reconstructed correctly, providing a principled surrogate objective for insertion scheduling. □

C.6 Compounding Parallelization Error for Insertion and Unmasking

First, we establish the following lemma that decomposes the KL divergence between joint distributions.

Lemma 1 (KL Divergence Chain Rule)**.** *The KL divergence between the joint distributions $\mathbb{P}_{s,t}(\boldsymbol{X}_s, \boldsymbol{X}_t)$ and $\mathbb{P}'_{s,t}(\boldsymbol{X}_s, \boldsymbol{X}_t)$ on the joint space $\mathcal{X} \times \mathcal{X}$ where $\mathcal{X}$ is a finite discrete state space, satisfies:*

$$D_{KL}(\mathbb{P}_{s,t}\|\mathbb{P}'_{s,t}) = D_{KL}(\mathbb{P}_s\|\mathbb{P}'_s) + \mathbb{E}_{\boldsymbol{x}_s\sim\mathbb{P}_s}\left[D_{KL}(\mathbb{P}_{t|s}(\cdot|\boldsymbol{x}_s)\|\mathbb{P}'_{t|s}(\cdot|\boldsymbol{x}_s)\right] \tag{68}$$

where $\mathbb{P}_s, \mathbb{P}'_s$ denote marginal distributions and $\mathbb{P}_{t|s}, \mathbb{P}'_{t|s}$ denote conditional distributions.

Proof. By expanding the definition of KL divergence, we have:

$$\begin{aligned}
D_{\mathrm{KL}}(\mathbb{P}_{s,t}\|\mathbb{P}'_{s,t}) &= \sum_{\boldsymbol{x}_s,\boldsymbol{x}_t} \mathbb{P}_{s,t}(\boldsymbol{x}_s, \boldsymbol{x}_t) \log \frac{\mathbb{P}_{s,t}(\boldsymbol{x}_s, \boldsymbol{x}_t)}{\mathbb{P}'_{s,t}(\boldsymbol{x}_s, \boldsymbol{x}_t)} \\
&= \sum_{\boldsymbol{x}_s,\boldsymbol{x}_t} \mathbb{P}_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s)\mathbb{P}_s(\boldsymbol{x}_s) \log \frac{\mathbb{P}_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s)\mathbb{P}_s(\boldsymbol{x}_s)}{\mathbb{P}'_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s)\mathbb{P}'_s(\boldsymbol{x}_s)} \\
&= \underbrace{\sum_{\boldsymbol{x}_s,\boldsymbol{x}_t} \mathbb{P}_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s)}_{=1}\, \mathbb{P}_s(\boldsymbol{x}_s) \log \frac{\mathbb{P}_s(\boldsymbol{x}_s)}{\mathbb{P}'_s(\boldsymbol{x}_s)} + \sum_{\boldsymbol{x}_s,\boldsymbol{x}_t} \mathbb{P}_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s)\mathbb{P}_s(\boldsymbol{x}_s) \log \frac{\mathbb{P}_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s)}{\mathbb{P}'_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s)} \\
&= \underbrace{\sum_{\boldsymbol{x}_s} \mathbb{P}_s(\boldsymbol{x}_s) \log \frac{\mathbb{P}_s(\boldsymbol{x}_s)}{\mathbb{P}'_s(\boldsymbol{x}_s)}}_{D_{\mathrm{KL}}(\mathbb{P}_s\|\mathbb{P}_s)} + \sum_{\boldsymbol{x}_s} \mathbb{P}_s(\boldsymbol{x}_s) \underbrace{\sum_{\boldsymbol{x}_s,\boldsymbol{x}_t} \mathbb{P}_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s) \log \frac{\mathbb{P}_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s)}{\mathbb{P}'_{t|s}(\boldsymbol{x}_t|\boldsymbol{x}_s)}}_{D_{\mathrm{KL}}(\mathbb{P}_{t|s}\|\mathbb{P}_{t|s})} \\
&= D_{\mathrm{KL}}(\mathbb{P}_s\|\mathbb{P}_s) + \mathbb{E}_{\boldsymbol{x}_s\sim\mathbb{P}_s}[D_{\mathrm{KL}}(\mathbb{P}_{t|s}\|\mathbb{P}_{t|s})]
\end{aligned} \tag{69}$$

which concludes the proof. □

Proposition C.2 (CPE as Upper Bound on KL Divergence)**.** *Given a sampling schedule $\{T \to \cdots \to t_{K-1} \to 0\}$, the KL divergence between the sampled distribution $\mathbb{P}_0^v$ and true distribution $\mathbb{P}_1^\star$ is upper bounded by the total CPE:*

$$D_{KL}(\mathbb{P}_1^\star\|\mathbb{P}_1^v) \leq \sum_{k=0}^{K-1}\left[\mathcal{E}_{CPE}^{ins}(t_k \to t_{k+1}) + \mathcal{E}_{CPE}^{unmsk}(t_k \to t_{k+1})\right] \tag{70}$$

Proof. First, we recall our definition for the CPE for insertion as:

$$\mathcal{E}_{\mathrm{CPE}}(s \to t) := \mathbb{E}_{\boldsymbol{x}_s\sim\mathbb{P}_s}\left[D_{\mathrm{KL}}\left(p(\mathcal{I}_t, \boldsymbol{X}_t|\boldsymbol{X}_s)\middle\|p(\mathcal{I}_t|\boldsymbol{X}_s)p(\boldsymbol{X}_t|\boldsymbol{X}_s)\right)\right] \tag{71}$$

and the CPE for unmasking as:

$$\mathcal{E}_{\mathrm{CPE}}^{\mathrm{unmsk}}(s \to t) = \mathbb{E}_{\boldsymbol{x}_s\sim\mathbb{P}_s}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{X}_t^{\ell_1}, \ldots, \boldsymbol{X}_t^{\ell_n}|\boldsymbol{X}_s)\middle\|\prod_{i=1}^n p(\boldsymbol{X}_t^{\ell_i}|\boldsymbol{X}_s)\right)\right] \tag{72}$$

Denoting the joint distribution under the true path measure $\mathbb{P}^\star$ as $\mathbb{P}^\star_{t_{k+1}|t_k}(\cdot|\boldsymbol{x}_{t_k}) := p(\boldsymbol{X}^{\ell_1}_{t_{k+1}}, \dots, \boldsymbol{X}^{\ell_n}_{t_{k+1}}|\boldsymbol{X}_{t_k})$ and the parameterized marginal distribution as $\mathbb{P}^v_{t_{k+1}|t_k}(\cdot|\boldsymbol{x}_{t_k}) := \prod_{i=1}^n p(\boldsymbol{X}^{\ell_i}_{t_{k+1}}|\boldsymbol{X}_{t_k})$, we can write the terminal distribution generated by both path measures as:

$$\mathbb{P}^\star_0 = \mathbb{P}^\star_{t_K|t_{K-1}} \dots \mathbb{P}^\star_{t_1|t_0} \mathbb{P}^\star_{t_0}, \quad \mathbb{P}^v_0 = \mathbb{P}^v_{t_K|t_{K-1}} \dots \mathbb{P}^v_{t_1|t_0} \mathbb{P}^v_{t_0} \tag{73}$$

Now, using Lemma 1, we can write the KL divergence between the marginals at time t where $s > t$ as:

$$\begin{aligned}
D_{\mathrm{KL}}(\mathbb{P}^\star_t \| \mathbb{P}^v_t) &\leq D_{\mathrm{KL}}(\mathbb{P}^\star_{s,t} \| \mathbb{P}^v_{s,t}) \\
&= D_{\mathrm{KL}}(\mathbb{P}^\star_s \| \mathbb{P}^v_s) + \mathbb{E}_{\boldsymbol{x}_s \sim \mathbb{P}_s}\left[D_{\mathrm{KL}}(\mathbb{P}_{t|s}(\cdot|\boldsymbol{x}_s) \| \mathbb{P}'_{t|s}(\cdot|\boldsymbol{x}_s)) \right] \\
&= D_{\mathrm{KL}}(\mathbb{P}^\star_s \| \mathbb{P}^v_s) + \mathbb{E}_{\boldsymbol{x}_s \sim \mathbb{P}_s}\left[\mathcal{E}^{\text{insert}}_{\text{CPE}}(s \to t|\boldsymbol{x}_s) + \mathcal{E}^{\text{unmask}}_{\text{CPE}}(s \to t|\boldsymbol{x}_s) \right] \\
&= D_{\mathrm{KL}}(\mathbb{P}^\star_s \| \mathbb{P}^v_s) + \mathcal{E}^{\text{insert}}_{\text{CPE}}(s \to t) + \mathcal{E}^{\text{unmask}}_{\text{CPE}}(s \to t)
\end{aligned} \tag{74}$$

where the equality holds if and only if $D_{\mathrm{KL}}(\mathbb{P}^\star_{t|s} \| \mathbb{P}^v_{t|s}) = 0$. Then, applying this inequality over all K time steps from $t_0 \to \dots \to t_{K-1} \to t_K = 0$, we have:

$$\begin{aligned}
D_{\mathrm{KL}}(\mathbb{P}^\star_0 \| \mathbb{P}^v_0) &= D_{\mathrm{KL}}(\mathbb{P}^\star_{t_K} \| \mathbb{P}^v_{t_K}) \\
&\leq D_{\mathrm{KL}}(\mathbb{P}^\star_{t_{K-1}} \| \mathbb{P}^v_{t_{K-1}}) + \mathcal{E}^{\text{insert}}_{\text{CPE}}(t_{K-1} \to t_K) + \mathcal{E}^{\text{unmask}}_{\text{CPE}}(t_{K-1} \to t_K) \\
&\vdots \\
&\leq \underbrace{D_{\mathrm{KL}}(\mathbb{P}^\star_{t_0} \| \mathbb{P}^v_{t_0})}_{=0} + \sum_{k=0}^{K-1} \left[\mathcal{E}^{\text{insert}}_{\text{CPE}}(t_k \to t_{k+1}) + \mathcal{E}^{\text{unmask}}_{\text{CPE}}(t_k \to t_{k+1}) \right]
\end{aligned} \tag{75}$$

Since the distribution at time t_0 is the fully empty sequence in any-length MDMs (i.e. $\mathbb{P}^\star_{t_0} = \mathbb{P}^v_{t_0} = \pi_{t_0}$), we have finished the proof. This means that the KL divergence of the generated distribution and the true distribution is zero if and only if the compounding parallelization error over all time steps is zero, i.e. $\mathcal{E}^{\text{insert}}_{\text{CPE}}(t_k \to t_{k+1}) = 0$ and $\mathcal{E}^{\text{unmask}}_{\text{CPE}}(t_k \to t_{k+1}) = 0$ for all $k \in \{0, \dots, K-1\}$. □

C.7 Radon-Nikodym Derivative Between Joint CTMCs

Proposition C.3 (Radon-Nikodym Derivative of Joint Any-Length CTMCs)**.** *Consider two joint any-length path measures $\mathbb{P}$ and $\mathbb{P}'$ defined by the unmasking rate matrices $\boldsymbol{Q}$ and $\boldsymbol{Q}'$ and the insertion rate matrices $\boldsymbol{R}$ and $\boldsymbol{R}'$. Then, the Radon-Nikodym derivative over the trajectory $\boldsymbol{X}_{0:1} = (\boldsymbol{X}_t)_{t\in[0,1]}$ is defined as:*

$$\begin{aligned}
\log \frac{\mathrm{d}\mathbb{P}'}{\mathrm{d}\mathbb{P}}(\boldsymbol{X}_{0:1}) &= \log \frac{\mathrm{d}\pi'_0}{\mathrm{d}\pi_0}(\boldsymbol{X}_0) \\
&+ \sum_{t_u : \boldsymbol{X}_{t_u-} \neq \boldsymbol{X}_{t_u}} \log \frac{\boldsymbol{Q}'_{t_u}(\boldsymbol{X}_{t_u-}, \boldsymbol{X}_{t_u})}{\boldsymbol{Q}_{t_u}(\boldsymbol{X}_{t_u-}, \boldsymbol{X}_{t_u})} + \int_0^1 \sum_{z \neq \boldsymbol{X}_{t_u}} (\boldsymbol{Q}_{t_u} - \boldsymbol{Q}'_{t_u})(\boldsymbol{X}_{t_u}, z) dt \\
&+ \sum_{t_i : \boldsymbol{X}_{t_i} \neq \boldsymbol{X}_{t_i}} \log \frac{\boldsymbol{R}'_{t_i}(\boldsymbol{X}_{t_i-}, \boldsymbol{X}_{t_i})}{\boldsymbol{R}_{t_i}(\boldsymbol{X}_{t_i-}, \boldsymbol{X}_{t_i})} + \int_0^1 \sum_{y \neq \boldsymbol{X}_{t_u}} (\boldsymbol{R}_{t_i} - \boldsymbol{R}'_{t_i})(\boldsymbol{X}_{t_i}, z) dt
\end{aligned} \tag{76}$$

where $t_i \in [0,1]$ denotes the times of insertion events and $t_u \in [0,1]$ denotes the times of unmasking events.

Proof. The discrete time log RND between the CTMC path measures $\mathbb{P}$ and $\mathbb{P}'$ is defined as:

$$\log \frac{\mathrm{d}\mathbb{P}'}{\mathrm{d}\mathbb{P}}(\boldsymbol{X}_{0:1}) = \log \frac{\mathrm{d}\pi'_0}{\mathrm{d}\pi_0}(\boldsymbol{X}_0) + \sum_{n=0}^{N-1} \log \frac{\mathrm{d}\mathbb{P}'(\boldsymbol{X}_{t_{n+1}}|\boldsymbol{X}_{t_n})}{\mathrm{d}\mathbb{P}(\boldsymbol{X}_{t_{n+1}}|\boldsymbol{X}_{t_n})} + \mathcal{O}(\Delta t) \tag{77}$$

where $\mathbb{P}_0 = \pi_0$ and $\mathbb{P}'_0 = \pi'_0$ are the initial distributions. Now, we have both unmasking and insertion rates given by $\boldsymbol{Q}_t(\boldsymbol{x}, \boldsymbol{y})$, which denotes the rate of unmasking from state $\boldsymbol{x} \to \boldsymbol{y}$, and $\boldsymbol{R}_t(\boldsymbol{x}, \boldsymbol{y})$, which denotes the rate of insertion from state $\boldsymbol{x} \to \boldsymbol{y}$. Therefore, the total probability of a single

jump under the joint path measure $\mathbb{P}$ can be decomposed into the probability of remaining the same state ($\boldsymbol{y} = \boldsymbol{x}$) and the probability of jumping to a different state ($\boldsymbol{y} \neq \boldsymbol{x}$):

$$\mathbb{P}(\boldsymbol{X}_{t_{n+1}} = \boldsymbol{y} | \boldsymbol{X}_{t_n} = \boldsymbol{x}) = \begin{cases} 1 - \Delta t \sum_{\boldsymbol{z} \neq \boldsymbol{x}} (\boldsymbol{Q}_t(\boldsymbol{x}, \boldsymbol{y}) + \boldsymbol{R}_t(\boldsymbol{x}, \boldsymbol{y})) + \mathcal{O}(\Delta t^2) & \boldsymbol{y} = \boldsymbol{x} \\ \Delta t (\boldsymbol{Q}_t(\boldsymbol{x}, \boldsymbol{y}) + \boldsymbol{R}_t(\boldsymbol{x}, \boldsymbol{y})) + \mathcal{O}(\Delta t^2) & \boldsymbol{y} \neq \boldsymbol{x} \end{cases} \tag{78}$$

For both cases, we will derive the log ratio of the two CTMCs. When the state remains the same over the interval $[t_n, t_{n+1}]$, the log-ratio expands to:

$$\begin{aligned} \log \frac{\mathrm{d}\mathbb{P}'(\boldsymbol{X}_{t_{n+1}} | \boldsymbol{X}_{t_n})}{\mathrm{d}\mathbb{P}(\boldsymbol{X}_{t_{n+1}} | \boldsymbol{X}_{t_n})} &= \log \frac{1 - \Delta t \sum_{\boldsymbol{z} \neq \boldsymbol{x}} \left(\boldsymbol{Q}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) + \boldsymbol{R}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) \right) + \mathcal{O}(\Delta t^2)}{1 - \Delta t \sum_{\boldsymbol{z} \neq \boldsymbol{x}} \left(\boldsymbol{Q}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) + \boldsymbol{R}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) \right) + \mathcal{O}(\Delta t^2)} \\ &= \Delta t \sum_{z \neq \boldsymbol{X}_{t_n}} \big(\boldsymbol{Q}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) + \boldsymbol{R}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) \\ &\quad - \boldsymbol{Q}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) - \boldsymbol{R}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) \big) + \mathcal{O}(\Delta t^2) \\ &= \Delta t \sum_{z \neq \boldsymbol{X}_{t_n}} \big(\boldsymbol{Q}_{t_n}(\boldsymbol{X}_{t_n}, z) - \boldsymbol{Q}'_{t_n}(\boldsymbol{X}_{t_n}, z) \big) \\ &\quad + \Delta t \sum_{z \neq \boldsymbol{X}_{t_n}} \big(\boldsymbol{R}_{t_n}(\boldsymbol{X}_{t_n}, z) - \boldsymbol{R}'_{t_n}(\boldsymbol{X}_{t_n}, z) \big) + \mathcal{O}(\Delta t^2) \end{aligned} \tag{79}$$

When the state changes over $[t_n, t_{n+1}]$, the log-ratio expands to:

$$\begin{aligned} \log \frac{\mathrm{d}\mathbb{P}'(\boldsymbol{X}_{t_{n+1}} | \boldsymbol{X}_{t_n})}{\mathrm{d}\mathbb{P}(\boldsymbol{X}_{t_{n+1}} | \boldsymbol{X}_{t_n})} &= \log \frac{\Delta t (\boldsymbol{Q}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) + \boldsymbol{R}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}})) + \mathcal{O}(\Delta t^2)}{\Delta t (\boldsymbol{Q}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) + \boldsymbol{R}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}})) + \mathcal{O}(\Delta t^2)} \\ &= \log \frac{\boldsymbol{Q}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}})}{\boldsymbol{Q}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}})} + \log \frac{\boldsymbol{R}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}})}{\boldsymbol{R}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}})} + \mathcal{O}(\Delta t) \end{aligned} \tag{80}$$

Substituting (79) and (80) into (77) and taking the limit as $\Delta t \to 0$, we get:

$$\begin{aligned} \log \frac{\mathrm{d}\mathbb{P}'}{\mathrm{d}\mathbb{P}}(\boldsymbol{X}_{0:1}) = \lim_{\Delta t \to 0} \Bigg\{ & \log \frac{\mathrm{d}\pi'_0}{\mathrm{d}\pi_0}(\boldsymbol{X}_0) + \sum_{n=0}^{N-1} \log \frac{\boldsymbol{Q}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}})}{\boldsymbol{Q}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}})} + \sum_{n=0}^{N-1} \log \frac{\boldsymbol{R}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}})}{\boldsymbol{R}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}})} \\ & + \Delta t \sum_{z \neq \boldsymbol{X}_{t_n}} \big(\boldsymbol{Q}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) - \boldsymbol{Q}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) \big) \\ & + \Delta t \sum_{z \neq \boldsymbol{X}_{t_n}} \big(\boldsymbol{R}_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) - \boldsymbol{R}'_{t_n}(\boldsymbol{X}_{t_n}, \boldsymbol{X}_{t_{n+1}}) \big) + \mathcal{O}(\Delta t) \Bigg\} \\ = & \log \frac{\mathrm{d}\pi'_0}{\mathrm{d}\pi_0}(\boldsymbol{X}_0) + \sum_{t_u : \boldsymbol{X}_{t_u -} \neq \boldsymbol{X}_{t_u}} \log \frac{\boldsymbol{Q}'_{t_u}(\boldsymbol{X}_{t_u -}, \boldsymbol{X}_{t_u})}{\boldsymbol{Q}_{t_u}(\boldsymbol{X}_{t_u -}, \boldsymbol{X}_{t_u})} + \int_0^1 \sum_{z \neq \boldsymbol{X}_{t_u}} (\boldsymbol{Q}_{t_u} - \boldsymbol{Q}'_{t_u})(\boldsymbol{X}_{t_u}, z) dt \\ & + \sum_{t_i : \boldsymbol{X}_{t_i} \neq \boldsymbol{X}_{t_i}} \log \frac{\boldsymbol{R}'_{t_i}(\boldsymbol{X}_{t_i -}, \boldsymbol{X}_{t_i})}{\boldsymbol{R}_{t_i}(\boldsymbol{X}_{t_i -}, \boldsymbol{X}_{t_i})} + \int_0^1 \sum_{y \neq \boldsymbol{X}_{t_u}} (\boldsymbol{R}_{t_i} - \boldsymbol{R}'_{t_i})(\boldsymbol{X}_{t_i}, z) dt \end{aligned} \tag{81}$$

which concludes our proof. □

Corollary 1 (KL Divergence Between Any-Length Joint CTMCs)**.** *The KL divergence between two joint CTMCs $\mathbb{P}, \mathbb{P}'$ with unmasking rates $\boldsymbol{Q}, \boldsymbol{Q}'$ and insertion rates $\boldsymbol{R}, \boldsymbol{R}'$ is given by:*

$$\begin{aligned} D_{KL}(\mathbb{P} \| \mathbb{P}') = & D_{KL}(\mathbb{P}_1 \| \mathbb{P}'_1) \\ & + \mathbb{E}_{\boldsymbol{X}_{0:1} \sim \mathbb{P}} \Bigg[\sum_{t_u : \boldsymbol{X}_{t_u^-} \neq \boldsymbol{X}_{t_u}} \log \frac{\boldsymbol{Q}'_{t_u}(\boldsymbol{X}_{t_u^-}, \boldsymbol{X}_{t_u})}{\boldsymbol{Q}_{t_u}(\boldsymbol{X}_{t_u^-}, \boldsymbol{X}_{t_u})} + \int_0^1 \sum_{z \neq \boldsymbol{X}_{t_u}} (\boldsymbol{Q}_{t_u} - \boldsymbol{Q}'_{t_u})(\boldsymbol{X}_{t_u}, z) dt \\ & + \sum_{t_i : \boldsymbol{X}_{t_i} \neq \boldsymbol{X}_{t_i}} \log \frac{\boldsymbol{R}'_{t_i}(\boldsymbol{X}_{t_i^-}, \boldsymbol{X}_{t_i})}{\boldsymbol{R}_{t_i}(\boldsymbol{X}_{t_i^-}, \boldsymbol{X}_{t_i})} + \int_0^1 \sum_{y \neq \boldsymbol{X}_{t_u}} (\boldsymbol{R}_{t_i} - \boldsymbol{R}'_{t_i})(\boldsymbol{X}_{t_i}, y) dt \Bigg] \end{aligned} \tag{82}$$

Proof. The KL divergence is defined as the expectation of the log RND derived in Lemma C.3 as:

$$\begin{aligned}
D_{\mathrm{KL}}(\mathbb{P}'\|\mathbb{P}) &= \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\log\frac{\mathrm{d}\mathbb{P}'}{\mathrm{d}\mathbb{P}}(\boldsymbol{X}_{0:1})\right] \\
&= \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\log\frac{\mathrm{d}\pi_0'}{\mathrm{d}\pi_0}(\boldsymbol{X}_0)\right] \\
&+ \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\sum_{t_u:\boldsymbol{X}_{t_u-}\neq\boldsymbol{X}_{t_u}}\log\frac{\boldsymbol{Q}'_{t_u}(\boldsymbol{X}_{t_u-},\boldsymbol{X}_{t_u})}{\boldsymbol{Q}_{t_u}(\boldsymbol{X}_{t_u-},\boldsymbol{X}_{t_u})}+\int_0^1\sum_{z\neq\boldsymbol{X}_{t_u}}(\boldsymbol{Q}_{t_u}-\boldsymbol{Q}'_{t_u})(\boldsymbol{X}_{t_u},z)dt\right] \\
&+ \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\sum_{t_i:\boldsymbol{X}_{t_i}\neq\boldsymbol{X}_{t_i}}\log\frac{\boldsymbol{R}'_{t_i}(\boldsymbol{X}_{t_i-},\boldsymbol{X}_{t_i})}{\boldsymbol{R}_{t_i}(\boldsymbol{X}_{t_i-},\boldsymbol{X}_{t_i})}+\int_0^1\sum_{y\neq\boldsymbol{X}_{t_u}}(\boldsymbol{R}_{t_i}-\boldsymbol{R}'_{t_i})(\boldsymbol{X}_{t_i},z)dt\right] \qquad (83)
\end{aligned}$$

We can decompose the first term as:

$$\mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}}\left[\log\frac{\mathrm{d}\pi_0'}{\mathrm{d}\pi_0}(\boldsymbol{X}_0)\right] = \mathbb{E}_{\boldsymbol{X}_0\sim\pi_0'}\left[\log\frac{\mathrm{d}\pi_0'}{\mathrm{d}\pi_0}(\boldsymbol{X}_0)\right] = D_{\mathrm{KL}}(\pi_0'\|\pi_0) \qquad (84)$$

The second term can be decomposed as:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\sum_{t_u:\boldsymbol{X}_{t_u-}\neq\boldsymbol{X}_{t_u}}\log\frac{\boldsymbol{Q}'_{t_u}(\boldsymbol{X}_{t_u-},\boldsymbol{X}_{t_u})}{\boldsymbol{Q}_{t_u}(\boldsymbol{X}_{t_u-},\boldsymbol{X}_{t_u})}+\int_0^1\sum_{z\neq\boldsymbol{X}_{t_u}}(\boldsymbol{Q}_{t_u}-\boldsymbol{Q}'_{t_u})(\boldsymbol{X}_{t_u},z)dt\right] \\
&= \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\sum_{t_u:\boldsymbol{X}_{t_u-}\neq\boldsymbol{X}_{t_u}}\log\frac{\boldsymbol{Q}'_{t_u}(\boldsymbol{X}_{t_u-},\boldsymbol{X}_{t_u})}{\boldsymbol{Q}_{t_u}(\boldsymbol{X}_{t_u-},\boldsymbol{X}_{t_u})}\right]+\mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\int_0^1\sum_{z\neq\boldsymbol{X}_{t_u}}(\boldsymbol{Q}_{t_u}-\boldsymbol{Q}'_{t_u})(\boldsymbol{X}_{t_u},z)dt\right] \qquad (85)
\end{aligned}$$

For the first component, we can consider the time discretization and take the limit for $\Delta t\to 0$ to get:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\sum_{k=0}^{K-1}\mathbf{1}[\boldsymbol{X}_{t_{k+1}}\neq\boldsymbol{X}_{t_k}]\log\frac{\boldsymbol{Q}'_{t_u}(\boldsymbol{X}_{t_k},\boldsymbol{X}_{t_{k+1}})}{\boldsymbol{Q}_{t_u}(\boldsymbol{X}_{t_k},\boldsymbol{X}_{t_{k+1}})}\right] \\
&= \sum_{k=0}^{K-1}\mathbb{E}_{\mathbb{P}'(\boldsymbol{X}_{t_k}),\mathbb{P}'(\boldsymbol{X}_{t_{k+1}}|\boldsymbol{X}_{t_k})}\left[\mathbf{1}[\boldsymbol{X}_{t_{k+1}}\neq\boldsymbol{X}_{t_k}]\log\frac{\boldsymbol{Q}'_{t_u}(\boldsymbol{X}_{t_k},\boldsymbol{X}_{t_{k+1}})}{\boldsymbol{Q}_{t_u}(\boldsymbol{X}_{t_k},\boldsymbol{X}_{t_{k+1}})}\right] \\
&= \sum_{k=0}^{K-1}\mathbb{E}_{\mathbb{P}'(\boldsymbol{X}_{t_k})}\sum_{z\neq\boldsymbol{X}_{t_k}}\left[\mathbb{P}'(z|\boldsymbol{X}_{t_k})\log\frac{\boldsymbol{Q}'_{t_u}(\boldsymbol{X}_{t_k},z)}{\boldsymbol{Q}_{t_u}(\boldsymbol{X}_{t_k},z)}\right] \\
&= \sum_{k=0}^{K-1}\mathbb{E}_{\mathbb{P}'(\boldsymbol{X}_{t_k})}\sum_{z\neq\boldsymbol{X}_{t_k}}\left[\Delta t\boldsymbol{Q}'_{t_k}(\boldsymbol{X}_{t_k},z)\log\frac{\boldsymbol{Q}'_{t_k}(\boldsymbol{X}_{t_k},z)}{\boldsymbol{Q}_{t_k}(\boldsymbol{X}_{t_k},z)}+\mathcal{O}(\Delta t^2)\right] \\
&\underset{\Delta t\to 0}{=} \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\int_0^1\sum_{z\neq\boldsymbol{X}_{t_u}}\boldsymbol{Q}'_{t_u}\log\frac{\boldsymbol{Q}'_{t_u}}{\boldsymbol{Q}_{t_u}}(\boldsymbol{X}_t,z)dt\right] \qquad (86)
\end{aligned}$$

Then, summing with the second integral, we get:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\int_0^1\sum_{z\neq\boldsymbol{X}_{t_u}}\boldsymbol{Q}'_{t_u}\log\frac{\boldsymbol{Q}'_{t_u}}{\boldsymbol{Q}_{t_u}}(\boldsymbol{X}_t,z)dt\right]+\mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\int_0^1\sum_{z\neq\boldsymbol{X}_{t_u}}(\boldsymbol{Q}_{t_u}-\boldsymbol{Q}'_{t_u})(\boldsymbol{X}_{t_u},z)dt\right] \\
&= \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\int_0^1\sum_{z\neq\boldsymbol{X}_{t_u}}\left[\left(\boldsymbol{Q}'_{t_u}\log\frac{\boldsymbol{Q}'_{t_u}}{\boldsymbol{Q}_{t_u}}+\boldsymbol{Q}_{t_u}-\boldsymbol{Q}'_{t_u}\right)(\boldsymbol{X}_{t_u},z)\right]dt \qquad (87)
\end{aligned}$$

Applying the same procedure from lines (85) to (87) to the insertion rates, we get:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\left[\sum_{t_i:\boldsymbol{X}_{t_i}\neq\boldsymbol{X}_{t_i}}\log\frac{\boldsymbol{R}'_{t_i}(\boldsymbol{X}_{t_i-},\boldsymbol{X}_{t_i})}{\boldsymbol{R}_{t_i}(\boldsymbol{X}_{t_i-},\boldsymbol{X}_{t_i})}+\int_0^1\sum_{y\neq\boldsymbol{X}_{t_u}}(\boldsymbol{R}_{t_i}-\boldsymbol{R}'_{t_i})(\boldsymbol{X}_{t_i},z)dt\right] \\
&= \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}'}\int_0^1\sum_{z\neq\boldsymbol{X}_{t_u}}\left[\left(\boldsymbol{R}'_{t_u}\log\frac{\boldsymbol{R}'_{t_u}}{\boldsymbol{R}_{t_u}}+\boldsymbol{R}_{t_u}-\boldsymbol{R}'_{t_u}\right)(\boldsymbol{X}_{t_u},z)\right]dt \qquad (88)
\end{aligned}$$

Putting these terms together, we get that the KL divergence between the joint CTMCs $\mathbb{P}$ and $\mathbb{P}'$ is given by:

$$
\begin{aligned}
D_{\mathrm{KL}}(\mathbb{P}\|\mathbb{P}') &= D_{\mathrm{KL}}(\mathbb{P}_1\|\mathbb{P}'_1) \\
&+ \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}}\Bigg[\sum_{t_u:\boldsymbol{X}_{t_u^-}\neq\boldsymbol{X}_{t_u}} \log\frac{\boldsymbol{Q}'_{t_u}(\boldsymbol{X}_{t_u^-},\boldsymbol{X}_{t_u})}{\boldsymbol{Q}_{t_u}(\boldsymbol{X}_{t_u^-},\boldsymbol{X}_{t_u})} + \int_0^1 \sum_{z\neq\boldsymbol{X}_{t_u}} (\boldsymbol{Q}_{t_u}-\boldsymbol{Q}'_{t_u})(\boldsymbol{X}_{t_u},z)dt \\
&+ \sum_{t_i:\boldsymbol{X}_{t_i}\neq\boldsymbol{X}_{t_i}} \log\frac{\boldsymbol{R}'_{t_i}(\boldsymbol{X}_{t_i^-},\boldsymbol{X}_{t_i})}{\boldsymbol{R}_{t_i}(\boldsymbol{X}_{t_i^-},\boldsymbol{X}_{t_i})} + \int_0^1 \sum_{y\neq\boldsymbol{X}_{t_u}} (\boldsymbol{R}_{t_i}-\boldsymbol{R}'_{t_i})(\boldsymbol{X}_{t_i},y)dt\Bigg]
\end{aligned} \tag{89}
$$

which concludes our proof. □

C.8 Adaptive Joint Decoding Loss

In this section, we provide theoretical justifications for our **Adaptive Joint Decoding (AJD)** loss, which can be used to jointly fine-tune the insertion and unmasking policies, as well as an adaptive inference schedule, to optimize sampling from the reward-tilted distribution. Throughout the proofs, we denote the full any-length generator as $\boldsymbol{A}_t := \boldsymbol{Q}_t + \boldsymbol{R}_t$ where $\boldsymbol{Q}_t$ denotes the unmasking rate and $\boldsymbol{R}_t$ denotes the insertion rate.

We first derive the form of the optimal any-length generator $\boldsymbol{A}^\star$, which follows seamlessly from the theory of fixed-length MDM generators (Zhu et al., 2025a; Tang et al., 2025b) but is defined on a larger state space of any-length sequences $\boldsymbol{x}\in\mathcal{X}$. Then, we show that our Adaptive Joint Decoding loss provably converges to the optimal generator.

For an intermediate state $\boldsymbol{x}\in\mathcal{X}$, we define the *total cost* $J_t(\boldsymbol{x},v)$ of insertions and unmasking steps required to reach a final state $\boldsymbol{X}_1$ under a tilted path measure $\mathbb{P}^v$. Given a terminal reward $r(\boldsymbol{X}_1)$, we define the cost-minimization objective as:

$$
J_t(\boldsymbol{x},v) = \mathop{\mathbb{E}}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}^v}\left[\int_0^1 \sum_{y\neq\boldsymbol{X}_s} C_s(\boldsymbol{X}_s,y)ds - r(\boldsymbol{X}_1)\middle|\boldsymbol{X}_t=\boldsymbol{x}\right] \tag{90}
$$

where the cost is defined as the KL divergence between generators derived in (1) as $C_s(x,y) := \left(\boldsymbol{A}_t^v\log\frac{\boldsymbol{A}^v}{\boldsymbol{A}^0} - \boldsymbol{A}^v + \boldsymbol{A}^0\right)(x,y)$. We can minimize $J_t(\boldsymbol{x},v)$ by maximizing a **value function** defined as the *negative optimal cost-to-go* $V_t(\boldsymbol{x}) := -J_t^\star(\boldsymbol{x}) = -\inf_v J_t(\boldsymbol{x},v)$. Using the recursive nature of the cost-to-go where the total cost is equal to the immediate and future cost, we can decompose the value function using Bellman's principle as:

$$
\begin{aligned}
-V_t(\boldsymbol{x}) = J_t^\star(\boldsymbol{x}) &= \inf_v \mathop{\mathbb{E}}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}^v}\left[\left(\int_t^{t+\Delta t} + \int_{t+\Delta t}^1\right)\sum_{y\neq\boldsymbol{X}_s} C_s(\boldsymbol{X}_s,y)ds - r(\boldsymbol{X}_1)\middle|\boldsymbol{X}_t=\boldsymbol{x}\right] \\
&= \left[\Delta t\inf_v\sum_{y\neq x} C_s(\boldsymbol{x},\boldsymbol{y}) + \mathcal{O}(\Delta t^2)\right] + \inf_v \mathop{\mathbb{E}}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}^v}[-V_{t+\Delta t}(\boldsymbol{X}_{t+\Delta t})|\boldsymbol{X}_t=\boldsymbol{x}]
\end{aligned} \tag{91}
$$

Now, we derive the full form of the optimal any-length generator $\boldsymbol{A}^\star$ using our definition of the value function.

Lemma 2 (Optimal Generator)**.** *Given a reference any-length generator $\boldsymbol{A}^0 := \boldsymbol{Q}^0 + \boldsymbol{R}^0$ a value function V_t, the optimal generator takes the form:*

$$
\begin{aligned}
\boldsymbol{A}_t^\star(\boldsymbol{x},\boldsymbol{y}) &= \boldsymbol{A}_t^0(\boldsymbol{x},\boldsymbol{y})\exp(V_t(\boldsymbol{y}) - V_t(\boldsymbol{x})) \\
&= \boldsymbol{Q}_t^0(\boldsymbol{x},\boldsymbol{y}_u)\exp(V_t(\boldsymbol{y}_u) - V_t(\boldsymbol{x})) + \boldsymbol{R}_t^0(\boldsymbol{x},\boldsymbol{y}^i)\exp(V_t(\boldsymbol{y}^i) - V_t(\boldsymbol{x}))
\end{aligned} \tag{92}
$$

where $\boldsymbol{y}_u$ denotes the state after an unmasking transition and $\boldsymbol{y}_i$ denotes the state after an insertion transition

Proof. Starting with the expanded value function in (91), we expand the recursive term by defining the next state $\boldsymbol{X}_{t+\Delta t} := \boldsymbol{y}$ and applying the CTMC transition probability as:

$$\begin{aligned}
&\inf_v \mathbb{E}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}^v}[-V_{t+\Delta t}(\boldsymbol{X}_{t+\Delta t})|\boldsymbol{X}_t = \boldsymbol{x}] \\
&= \inf_v \left[-\sum_{\boldsymbol{y}} V_{t+\Delta t}(\boldsymbol{y}) \left(\mathbf{1}_{x=y} + \Delta t \boldsymbol{A}_t^v(\boldsymbol{x},\boldsymbol{y}) + \mathcal{O}(\Delta t^2)\right)\right] \\
&= \inf_v \left[-V_{t+\Delta t}(\boldsymbol{x}) - \Delta t \sum_{x\neq y} V_{t+\Delta t}(\boldsymbol{y}) \boldsymbol{A}_t^v(\boldsymbol{x},\boldsymbol{y}) + \Delta t \sum_{x\neq y} V_{t+\Delta t}(\boldsymbol{x}) \boldsymbol{A}_t^v(\boldsymbol{x},\boldsymbol{y}) + \mathcal{O}(\Delta t^2)\right] \\
&= -V_{t+\Delta t}(\boldsymbol{x}) + \Delta t \inf_u \left[\sum_{x\neq y} \boldsymbol{A}_t^v(\boldsymbol{x},\boldsymbol{y}) \left(V_{t+\Delta t}(\boldsymbol{x}) - V_{t+\Delta t}(\boldsymbol{y})\right)\right] + \mathcal{O}(\Delta t^2) \qquad (93)
\end{aligned}$$

Substituting (93) back into the Bellman recursion in (91), we get:

$$\begin{aligned}
-V_t(\boldsymbol{x}) = &\left[\Delta t \inf_v \sum_{y\neq x} C_s(\boldsymbol{x},\boldsymbol{y}) + \mathcal{O}(\Delta t^2)\right] \\
&- V_{t+\Delta t}(\boldsymbol{x}) + \Delta t \inf_u \left[\sum_{x\neq y} \boldsymbol{A}_t^v(\boldsymbol{x},\boldsymbol{y}) \left(V_{t+\Delta t}(\boldsymbol{x}) - V_{t+\Delta t}(\boldsymbol{y})\right)\right] + \mathcal{O}(\Delta t^2) \\
V_{t+\Delta t}(\boldsymbol{x}) - V_t(\boldsymbol{x}) = \Delta t \inf_v &\sum_{y\neq x} \left[C_s(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{A}_t^v(\boldsymbol{x},\boldsymbol{y}) \left(V_{t+\Delta t}(\boldsymbol{x}) - V_{t+\Delta t}(\boldsymbol{y})\right)\right] \qquad (94)
\end{aligned}$$

Then, dividing by Δt, taking the limit $\Delta t \to 0$, and substituting $C_s(x,y) = \left(\boldsymbol{A}_t^v \log \frac{\boldsymbol{A}^v}{\boldsymbol{A}^0} - \boldsymbol{A}^v + \boldsymbol{A}^0\right)(x,y)$, we get:

$$\begin{aligned}
\partial_t V_t(\boldsymbol{x}) &= \lim_{\Delta t\to 0} \frac{V_{t+\Delta t}(\boldsymbol{x}) - V_t(\boldsymbol{x})}{\Delta t} = \inf_v \sum_{y\neq x} \left[C_s(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{A}_t^v(\boldsymbol{x},\boldsymbol{y}) \left(V_t(\boldsymbol{x}) - V_t(\boldsymbol{y})\right)\right] \\
&= \inf_v \sum_{y\neq x} \left[\left(\boldsymbol{A}_t^v \log \frac{\boldsymbol{A}^v}{\boldsymbol{A}^0} - \boldsymbol{A}^v + \boldsymbol{A}^0\right)(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{A}_t^v(\boldsymbol{x},\boldsymbol{y}) \left(V_t(\boldsymbol{x}) - V_t(\boldsymbol{y})\right)\right] \qquad (95)
\end{aligned}$$

The infimum is achieved by minimizing the following for every pair $(\boldsymbol{x},\boldsymbol{y})$:

$$f(\boldsymbol{A}^v) = \boldsymbol{A}_t^v \log \frac{\boldsymbol{A}^v}{\boldsymbol{A}^0} - \boldsymbol{A}^v + \boldsymbol{A}^0 + \boldsymbol{A}_t^v \left(V_t(\boldsymbol{x}) - V_t(\boldsymbol{y})\right) \qquad (96)$$

$$f'(\boldsymbol{A}^v) = \log \frac{\boldsymbol{A}^v}{\boldsymbol{A}^0} + \left(V_t(\boldsymbol{x}) - V_t(\boldsymbol{y})\right) \qquad (97)$$

The minimizer $\boldsymbol{A}^\star$ satisfying $f'(\boldsymbol{A}^v) = 0$ is defined as:

$$\log \frac{\boldsymbol{A}^\star}{\boldsymbol{A}^0} = V_t(\boldsymbol{y}) - V_t(\boldsymbol{x}) \implies \boldsymbol{A}_t^\star(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{A}_t^0(\boldsymbol{x},\boldsymbol{y}) \exp\left(V_t(\boldsymbol{y}) - V_t(\boldsymbol{x})\right) \qquad (98)$$

Since the total generator factorizes into the sum of insertion and unmasking rates and rates do not overlap as defined in App C.1, we have:

$$\boldsymbol{A}_t^\star(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{Q}_t^\star(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{R}_t^\star(\boldsymbol{x},\boldsymbol{y}) \quad \text{s.t.} \quad \begin{cases} \boldsymbol{Q}_t^\star(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{Q}_t^0(\boldsymbol{x},\boldsymbol{y}) \exp\left(V_t(\boldsymbol{y}) - V_t(\boldsymbol{x})\right) \\ \boldsymbol{R}_t^\star(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{R}_t^0(\boldsymbol{x},\boldsymbol{y}) \exp\left(V_t(\boldsymbol{y}) - V_t(\boldsymbol{x})\right) \end{cases} \qquad (99)$$

which proves our result. □

Now, we will derive the form of the optimal path measure $\mathbb{P}^\star$ corresponding to the optimal generator $\boldsymbol{A}^\star$.

Lemma 3 (Optimal Path Measure)**.** *The optimal path measure $\mathbb{P}^\star$ corresponding to the value function $V_t(\boldsymbol{x})$ is defined as:*

$$\mathbb{P}_t^\star(\boldsymbol{x}) = \frac{1}{Z}\mathbb{P}_t^0(\boldsymbol{x})\mathrm{e}^{V_t(\boldsymbol{x})}, \quad Z := \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_1^0}\left[\mathrm{e}^{r(\boldsymbol{x})}\right] \qquad (100)$$

for continuous $t \mapsto \boldsymbol{A}_t^0$.

Proof. We define $h_t(\boldsymbol{x}) := \frac{1}{Z}\mathbb{P}_t^0(\boldsymbol{x})\mathrm{e}^{V_t(\boldsymbol{x})}$, which satisfies $h_1 = \mathbb{P}_1^\star$ by definition. To show that $\mathbb{P}^\star$ is the path measure generated by $\boldsymbol{A}^\star$, it suffices to show that $h_t(\boldsymbol{x})$ satisfies the Kolmogorov forward equation for $\boldsymbol{A}_t^\star$. First, the Kolmogorov forward equation for the reference generator $\boldsymbol{A}_t^0$ is given by:

$$\partial_t \mathbb{P}_t^0(\boldsymbol{x}) = \sum_{y \neq x} \left(\boldsymbol{A}_t^0(\boldsymbol{y}, \boldsymbol{x})\mathbb{P}_t^0(\boldsymbol{y}) - \boldsymbol{A}_t^0(\boldsymbol{x}, \boldsymbol{y})\mathbb{P}_t^0(\boldsymbol{x})\right) \tag{101}$$

Then, from Lemma 2, we have $\boldsymbol{A}_t^\star(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{A}_t^0(\boldsymbol{x}, \boldsymbol{y})\mathrm{e}^{V_t(\boldsymbol{y}) - V_t(\boldsymbol{x})}$ which can be substituted into (95) to get:

$$\partial_t V_t(\boldsymbol{x}) = \sum_{y \neq x} \boldsymbol{A}_t^0(\boldsymbol{x}, \boldsymbol{y}) \left(\mathrm{e}^{V_t(\boldsymbol{x})} - \mathrm{e}^{V_t(\boldsymbol{y})}\right) \tag{102}$$

Taking the partial derivative of $h_t(\boldsymbol{x})$ and substituting (102), we get:

$$\begin{aligned}
\partial_t h_t(\boldsymbol{x}) &= \frac{1}{Z}\left[\partial_t \mathbb{P}_t^0(\boldsymbol{x})\mathrm{e}^{V_t(\boldsymbol{x})} + \mathbb{P}_t^0 \partial_t \mathrm{e}^{V_t(\boldsymbol{x})}\right] \\
&= \frac{1}{Z}\left[\mathrm{e}^{V_t(\boldsymbol{x})} \sum_{y \neq x} \left(\boldsymbol{A}_t^0(\boldsymbol{y}, \boldsymbol{x})\mathbb{P}_t^0(y) - \boldsymbol{A}_t^0(\boldsymbol{x}, y)\mathbb{P}_t^0(\boldsymbol{x})\right) + \mathbb{P}_t^0(\boldsymbol{x}) \sum_{y \neq x} \boldsymbol{A}_t^0(\boldsymbol{x}, \boldsymbol{y}) \left(\mathrm{e}^{V_t(\boldsymbol{x})} - \mathrm{e}^{V_t(\boldsymbol{y})}\right)\right] \\
&= \sum_{y \neq x} \left(\boldsymbol{A}_t^0(\boldsymbol{y}, \boldsymbol{x}) \frac{1}{Z}\mathbb{P}_t^0(\boldsymbol{y})\mathrm{e}^{V_t(\boldsymbol{x})} - \boldsymbol{A}_t^0(\boldsymbol{x}, \boldsymbol{y}) \frac{1}{Z}\mathbb{P}_t^0(\boldsymbol{x})\mathrm{e}^{V_t(\boldsymbol{y})}\right) \\
&= \sum_{y \neq x} \left(\boldsymbol{A}_t^0(\boldsymbol{y}, \boldsymbol{x})\mathrm{e}^{V_t(\boldsymbol{x}) - V_t(\boldsymbol{y})} h_t(\boldsymbol{y}) - \boldsymbol{A}_t^0(\boldsymbol{x}, \boldsymbol{y})\mathrm{e}^{V_t(\boldsymbol{x}) - V_t(\boldsymbol{y})} h_t(\boldsymbol{x})\right) \\
&\overset{(102)}{=} \sum_{y \neq x} \left(\boldsymbol{A}_t^\star(\boldsymbol{y}, \boldsymbol{x}) h_t(\boldsymbol{y}) - \boldsymbol{A}_t^\star(\boldsymbol{x}, \boldsymbol{y}) h_t(\boldsymbol{x})\right)
\end{aligned} \tag{103}$$

The final equality is exactly Kolmogorov forward equation of the optimal any-length generator $\boldsymbol{A}_t^\star$ for the tilted path measure $\mathbb{P}^\star$. Since the Kolmogorov forward equation yields an unique solution for all t, we have shown that $\mathbb{P}_t^\star = \frac{1}{Z}\mathbb{P}_t^0(\boldsymbol{x})\mathrm{e}^{V_t(\boldsymbol{x})}$. □

Finally, we can derive the RND between the optimal any-length path measure $\mathbb{P}^\star$ with respect to the base measure $\mathbb{P}^0$ in the following Lemma.

Lemma 4 (Radon-Nikodym Derivative Between Optimal and Base Path Measures)**.** *The RND between the optimal path measure $\mathbb{P}^\star$ and its generator $\boldsymbol{Q}^\star$ and the reference path measure $\mathbb{P}^0$ and generator $\boldsymbol{Q}^0$ over any trajectory $\boldsymbol{X}_{0:1}$ can be expressed as:*

$$\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^0}(\boldsymbol{X}_{0:1}) = \frac{1}{Z} e^{r(\boldsymbol{X}_1)}, \quad \textit{where} \quad Z := \mathbb{E}_{\boldsymbol{X}_1 \sim \mathbb{P}_1^0}[e^{r(\boldsymbol{X}_1)}] \tag{104}$$

Proof. From Prop C.3, the RND between the path measures is defined as:

$$\log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^0}(\boldsymbol{X}_{0:1}) = \log \frac{\mathrm{d}\mathbb{P}_0^\star}{\mathrm{d}\mathbb{P}_0^0}(\boldsymbol{X}_0) + \sum_{t : \boldsymbol{X}_{t^-} \neq \boldsymbol{X}_t} \log \frac{\boldsymbol{A}_t^\star(\boldsymbol{X}_{t_u -}, \boldsymbol{X}_t)}{\boldsymbol{A}_t^0(\boldsymbol{X}_{t^-}, \boldsymbol{X}_t)} + \int_0^1 \sum_{z \neq \boldsymbol{X}_t} (\boldsymbol{A}_t^0 - \boldsymbol{A}_t^\star)(\boldsymbol{X}_t, z) dt$$

Then, applying Lemmas 2 and 3, we get:

$$\log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^0}(\boldsymbol{X}_{0:1}) = V_0(\boldsymbol{X}_0) - \log Z + \sum_{t : \boldsymbol{X}_{t^-} \neq \boldsymbol{X}_t} (V_t(\boldsymbol{X}_t) - V_t(\boldsymbol{X}_{t^-})) + \int_0^1 \sum_{y \neq \boldsymbol{X}_t} \boldsymbol{A}_t^0(\boldsymbol{X}_t, y) \left(1 - \mathrm{e}^{V_t(\boldsymbol{y}) - V_t(\boldsymbol{X}_t)}\right) dt \tag{105}$$

Now, we derive an expression for $V_0(\boldsymbol{X}_0)$ with respect to $V_1(\boldsymbol{X}_1) = r(\boldsymbol{X}_1)$ by recognizing that the CTMC is a piecewise càdlàg function where each discrete step $t_k \to t_{k+1}$ can be categorized as a time evolution at a fixed state $\boldsymbol{X}_{t_k}$ or a jump from state $\boldsymbol{X}_{t_k}$ to $\boldsymbol{X}_{t_k^-} \neq \boldsymbol{X}_{t_k}$. Given this, we

decompose the value difference as:

$$\begin{aligned} V_1(\boldsymbol{X}_1) - V_0(\boldsymbol{X}_0) &= \sum_{k=0}^{K-1} (V_{t_{k+1}}(\boldsymbol{X}_{t_k}) - V_{t_k}(\boldsymbol{X}_{t_k})) + \sum_{k=1}^{K-1} (V_{t_k}(\boldsymbol{X}_{t_k}) - V_{t_k}(\boldsymbol{X}_{t_{k-1}})) \\ &= \sum_{k=0}^{K-1} \int_{t_k}^{t_{k+1}} \partial_t V_t(\boldsymbol{X}_{t_k}) dt + \sum_{t:\boldsymbol{X}_{t^-} \neq \boldsymbol{X}_t} (V_t(\boldsymbol{X}_t) - V_t(\boldsymbol{X}_{t^-})) \\ &= \int_0^1 \partial_t V_t(\boldsymbol{X}_t) dt + \sum_{t:\boldsymbol{X}_{t^-} \neq \boldsymbol{X}_t} (V_t(\boldsymbol{X}_t) - V_t(\boldsymbol{X}_{t^-})) \end{aligned} \tag{106}$$

$$\begin{aligned} V_0(\boldsymbol{X}_0) &= V_1(\boldsymbol{X}_1) - \int_0^1 \partial_t V_t(\boldsymbol{X}_t) dt - \sum_{t:\boldsymbol{X}_{t^-} \neq \boldsymbol{X}_t} (V_t(\boldsymbol{X}_t) - V_t(\boldsymbol{X}_{t^-})) \\ &\overset{(2)}{=} V_1(\boldsymbol{X}_1) - \int_0^1 \sum_{y \neq \boldsymbol{X}_t} \boldsymbol{A}_t^0(\boldsymbol{X}_t, \boldsymbol{y}) \left(1 - \mathrm{e}^{V_t(\boldsymbol{y}) - V_t(\boldsymbol{X}_t)}\right) dt - \sum_{t:\boldsymbol{X}_{t^-} \neq \boldsymbol{X}_t} (V_t(\boldsymbol{X}_t) - V_t(\boldsymbol{X}_{t^-})) \end{aligned} \tag{107}$$

where the final equality follows from Lemma 2. Substituting this expression for $V_0(\boldsymbol{X}_0)$ into (105), terms cancel and the RND reduces to:

$$\log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^0}(\boldsymbol{X}_{0:1}) = V_1(\boldsymbol{X}_1) - \log Z \implies \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^0}(\boldsymbol{X}_{0:1}) = \frac{1}{Z} \mathrm{e}^{V_1(\boldsymbol{X}_1)} \tag{108}$$

where $V_1(\boldsymbol{X}_1) = r(\boldsymbol{X}_1)$, we conclude our proof. □

Finally, we are ready to derive the RND between the optimal path measure and the parameterized path measure.

Proposition 4.1 (Radon-Nikodym Derivative of Parameterized Rates)**.** *Let the fine-tuned unmasking rate be $f^v(\boldsymbol{x}_t, t)[\ell] \in \Delta^{V-1}$ and the insertion rate be $g^v(\boldsymbol{x}_t, t)[\ell] \in \mathbb{R}_{\geq 0}$ that generates the path measure $\mathbb{P}^v$. Then, given optimal rates $f^{pre}(\boldsymbol{x}_t, t)$ and $g^{pre}(\boldsymbol{x}_t, t)$ and reward function $r : \mathcal{X} \to \mathbb{R}$, the log RND between the optimal joint CTMC and the fine-tuned CTMC over the trajectory $\boldsymbol{X}_{0:1} = (\boldsymbol{X}_t)_{t \in [0,1]}$ is defined as:*

$$\begin{aligned} \log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1}) &= \frac{r(\boldsymbol{X}_1)}{\alpha} - \log Z + \sum_{t_u : \boldsymbol{X}_{t_u^-} \neq \boldsymbol{X}_{t_u}} \sum_{\ell : \boldsymbol{X}^\ell_{t_u^-} \neq \boldsymbol{X}^\ell_{t_u}} \log \frac{f^{pre}(\boldsymbol{X}_{t_u}, t)[\ell, \boldsymbol{d}]}{f^v(\boldsymbol{X}_{t_u}, t)[\ell, \boldsymbol{d}]} \\ &+ \sum_{t_i : \boldsymbol{X}_{t_i^-} \neq \boldsymbol{X}_{t_i}} \sum_{\ell : \boldsymbol{X}_{t_u^-} \neq \boldsymbol{X}^\ell} \log \frac{g^{pre}(\boldsymbol{X}_{t_i}, t_i)[\ell]}{g^v(\boldsymbol{X}_{t_i}, t_i)[\ell]} + \int_0^1 \frac{\dot{\alpha}_t}{1 - \alpha_t} \left(\sum_\ell (g^{pre} - g^v)(\boldsymbol{X}_{t_i}, t_i)[\ell] \right) dt \end{aligned} \tag{19}$$

where $t_i \in [0,1]$ denotes the times of insertion events and $t_u \in [0,1]$ denotes the times of unmasking events.

Proof. First, we decompose the RND as:

$$\log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1}) = \log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^0} \frac{\mathrm{d}\mathbb{P}^0}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1}) = \log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^0}(\boldsymbol{X}_{0:1}) + \log \frac{\mathrm{d}\mathbb{P}^0}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1}) \tag{109}$$

First, we derive the $\log \frac{\mathrm{d}\mathbb{P}^0}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1})$ where $\mathbb{P}^0$ is the path measure of the pretrained model $(f^{\text{pre}}, g^{\text{pre}})$ and $\mathbb{P}^v$ is the path measure generated from the tilted models (f^v, g^v). Let t_i denote the times of insertion events and t_u denote the times of unmasking events. Then, the log RND decomposes into:

$$\begin{aligned} \log \frac{\mathrm{d}\mathbb{P}^0}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1}) &= \log \frac{\mathrm{d}\mathbb{P}^0_0}{\mathrm{d}\mathbb{P}^v_0}(\boldsymbol{X}_0) + \sum_{t_u : \boldsymbol{X}_{t_u -} \neq \boldsymbol{X}_{t_u}} \log \frac{\boldsymbol{Q}^0_{t_u}(\boldsymbol{X}_{t_u -}, \boldsymbol{X}_{t_u})}{\boldsymbol{Q}^v_{t_u}(\boldsymbol{X}_{t_u -}, \boldsymbol{X}_{t_u})} + \int_0^1 \sum_{z \neq \boldsymbol{X}_{t_u}} (\boldsymbol{Q}^v_{t_u} - \boldsymbol{Q}^0_{t_u})(\boldsymbol{X}_{t_u}, z) dt \\ &+ \sum_{t_i : \boldsymbol{X}_{t_i} \neq \boldsymbol{X}_{t_i}} \log \frac{\boldsymbol{R}^0_{t_i}(\boldsymbol{X}_{t_i -}, \boldsymbol{X}_{t_i})}{\boldsymbol{R}^v_{t_i}(\boldsymbol{X}_{t_i -}, \boldsymbol{X}_{t_i})} + \int_0^1 \sum_{y \neq \boldsymbol{X}_{t_u}} (\boldsymbol{R}^v_{t_i} - \boldsymbol{R}^0_{t_i})(\boldsymbol{X}_{t_i}, z) dt \end{aligned} \tag{110}$$

Now, we recall the form of the any-length MDM unmasking rate:

$$\begin{aligned}\boldsymbol{Q}_t^v(\boldsymbol{x}, \boldsymbol{x}^{\ell \leftarrow \boldsymbol{d}}) &= \frac{\dot{\beta}_t}{1-\beta_t} \cdot \mathbb{P}(\boldsymbol{x}_1^{s_t[i]} = \boldsymbol{d} | \boldsymbol{x}_t = \boldsymbol{x}), \quad \boldsymbol{d} \in \mathcal{V}, \boldsymbol{x}^i = \boldsymbol{M} \\ &= \frac{\dot{\beta}_t}{1-\beta_t} \cdot f^v(\boldsymbol{x}, t)[i, \boldsymbol{d}] \end{aligned} \tag{111}$$

where the sum over the vocabulary $\sum_{d \in \mathcal{V}} f_\theta(\boldsymbol{x}, t)[i, \boldsymbol{d}] = 1$. For a state $\boldsymbol{x}$ with $|\{\ell : \boldsymbol{x}^\ell = \boldsymbol{M}\}|$ masked positions, we have that the **exit rate** from state $\boldsymbol{x}$ reduces to:

$$\sum_{\boldsymbol{y} \neq \boldsymbol{x}} \boldsymbol{Q}_t^v(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\ell : \boldsymbol{x}^\ell = \boldsymbol{M}} \sum_d \boldsymbol{Q}_t^v(\boldsymbol{x}, \boldsymbol{x}^{\ell \leftarrow \boldsymbol{d}}) = \frac{\dot{\beta}_t}{1-\beta_t} \sum_{\ell : \boldsymbol{x}^\ell = \boldsymbol{M}} 1 = \frac{\dot{\beta}_t}{1-\beta_t} \big| \{\ell : \boldsymbol{x}^\ell = \boldsymbol{M}\} \big| \tag{112}$$

Given the noise schedule β_t is equal for both the pretrained and fine-tuned model, the number of masked positions along the interpolant is equal. So we can write:

$$\sum_{\boldsymbol{y} \neq \boldsymbol{x}} \boldsymbol{Q}_t^0(\boldsymbol{x}, \boldsymbol{y}) = \frac{\dot{\beta}_t}{1-\beta_t} \big| \{\ell : \boldsymbol{x}^\ell = \boldsymbol{M}\} \big| \tag{113}$$

and the exit term vanishes:

$$\int_0^1 \sum_{y \neq \boldsymbol{X}_{t_u}} (\boldsymbol{Q}_{t_u}^v - \boldsymbol{Q}_{t_u}^0)(\boldsymbol{X}_{t_u}, y) \mathrm{d}t = 0 \tag{114}$$

The insertion rate is defined as:

$$\begin{aligned}\boldsymbol{R}_t^v(\boldsymbol{x}, \boldsymbol{x}^{\triangleleft_\ell \boldsymbol{M}}) &= \frac{\dot{\alpha}_t}{1-\alpha_t} \cdot \mathbb{E}[s_t[\ell] - s_t[\ell-1] - 1 | \boldsymbol{x}_t = \boldsymbol{x}] \\ &= \frac{\dot{\alpha}_t}{1-\alpha_t} g^v(\boldsymbol{x}, t)[\ell] \end{aligned} \tag{115}$$

Since the insertion rate depends on the model outputs, we have that $\boldsymbol{R}_t^0(\boldsymbol{x}, \boldsymbol{x}^{\triangleleft_\ell \boldsymbol{M}}) \neq \boldsymbol{R}_t^v(\boldsymbol{x}, \boldsymbol{x}^{\triangleleft_\ell \boldsymbol{M}})$. In this case, the exit term is defined as:

$$\int_0^1 \sum_{y \neq \boldsymbol{X}_{t_u}} (\boldsymbol{R}_{t_i}^v - \boldsymbol{R}_{t_i}^0)(\boldsymbol{X}_{t_i}, y) \mathrm{d}t = \int_0^1 \frac{\dot{\alpha}_t}{1-\alpha_t} \left(\sum_\ell g^{\text{pre}}(\boldsymbol{X}_{t_i}, t_i)[\ell] - \sum_\ell g_\theta^v(\boldsymbol{X}_{t_i}, t_i)[\ell] \right) dt \tag{116}$$

Putting these terms together, we get the final form of the log RND between any-length path measures:

$$\begin{aligned}\log \frac{\mathrm{d}\mathbb{P}^0}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1}) = &\sum_{t_u : \boldsymbol{X}_{t_u -} \neq \boldsymbol{X}_{t_u}} \sum_{\ell : \boldsymbol{X}_{t_u -}^\ell \neq \boldsymbol{X}_{t_u}^\ell} \log \frac{f^{\text{pre}}(\boldsymbol{X}_{t_u}, t)[\ell, \boldsymbol{d}]}{f^v(\boldsymbol{X}_{t_u}, t)[\ell, \boldsymbol{d}]} \\ + \sum_{t_i : \boldsymbol{X}_{t_i -} \neq \boldsymbol{X}_{t_i}} &\sum_{\ell : \boldsymbol{X}_{t_u -} = \boldsymbol{X}_{t_u}^{\triangleleft_\ell \boldsymbol{M}}} \log \frac{g_\theta^{\text{pre}}(\boldsymbol{X}_{t_i}, t_i)[\ell]}{g^v(\boldsymbol{X}_{t_i}, t_i)[\ell]} + \int_0^1 \frac{\dot{\alpha}_t}{1-\alpha_t} \left(\sum_\ell g^{\text{pre}}(\boldsymbol{X}_{t_i}, t_i)[\ell] - \sum_\ell g^v(\boldsymbol{X}_{t_i}, t_i)[\ell] \right) dt \end{aligned} \tag{117}$$

Finally, substituting the result from Lemma 4 and (117) into (109), we get:

$$\begin{aligned}\log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1}) &= \log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^0}(\boldsymbol{X}_{0:1}) + \log \frac{\mathrm{d}\mathbb{P}^0}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1}) \\ &= \frac{r(\boldsymbol{X}_1)}{\alpha} - \log Z + \sum_{t_u : \boldsymbol{X}_{t_u -} \neq \boldsymbol{X}_{t_u}} \sum_{\ell : \boldsymbol{X}_{t_u -}^\ell \neq \boldsymbol{X}_{t_u}^\ell} \log \frac{f^{\text{pre}}(\boldsymbol{X}_{t_u}, t)[\ell, \boldsymbol{d}]}{f^v(\boldsymbol{X}_{t_u}, t)[\ell, \boldsymbol{d}]} \\ &\quad + \sum_{t_i : \boldsymbol{X}_{t_i -} \neq \boldsymbol{X}_{t_i}} \sum_{\ell : \boldsymbol{X}_{t_u -} = \boldsymbol{X}_{t_u}^{\triangleleft_\ell \boldsymbol{M}}} \log \frac{g_\theta^{\text{pre}}(\boldsymbol{X}_{t_i}, t_i)[\ell]}{g^v(\boldsymbol{X}_{t_i}, t_i)[\ell]} \\ &\quad + \int_0^1 \frac{\dot{\alpha}_t}{1-\alpha_t} \left(\sum_\ell g^{\text{pre}}(\boldsymbol{X}_{t_i}, t_i)[\ell] - \sum_\ell g^v(\boldsymbol{X}_{t_i}, t_i)[\ell] \right) dt \end{aligned} \tag{118}$$

which concludes our proof. □

We use this form of the log RND to define the importance weight $W^v(\boldsymbol{X}_{0:1}) := \log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1})$ in the AJD loss, which yields the optimal any-length path measure $\mathbb{P}^\star$, as we will prove in Prop C.4 below.

Proposition C.4 (Adaptive Joint Decoding Loss)**.** *The unique minimizer of the **Adaptive Joint Decoding (AJD)** loss defined as:*

$$\mathcal{L}_{AJD}(\theta,\phi) := \mathop{\mathbb{E}}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}^v}\left[\frac{1}{Z}\mathrm{e}^{W^v}\left[\mathcal{L}_{unmask}(\theta;\boldsymbol{X}_1)+\mathcal{L}_{insert}(\theta;\boldsymbol{X}_1)+\mathcal{L}_{UQL}(\phi;\boldsymbol{X}_1)+\mathcal{L}_{IQL}(\phi;\boldsymbol{X}_1)\right]\right]$$

is the optimal unmasking generator $\boldsymbol{Q}^\star$ and insertion generator $\boldsymbol{R}^\star$ of the reward-tilted path measure $\mathbb{P}^\star$

Proof. First, we recall the form of the cross-entropy loss as:

$$\mathcal{F}_{\mathrm{CE}}(\mathbb{P}^{\theta,\phi},\mathbb{P}^\star) := D_{\mathrm{KL}}(\mathbb{P}^\star\|\mathbb{P}^{\theta,\phi}) = \mathbb{E}_{\mathbb{P}^\star}\left[\log\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^{\theta,\phi}}\right] = \mathbb{E}_{\mathbb{P}^v}\left[\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v}\log\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^{\theta,\phi}}\right] \tag{119}$$

Then, we substitute $W^v(\boldsymbol{X}_{0:1}) = \log\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v}(\boldsymbol{X}_{0:1})$. Since $\mathbb{P}^\star$ is fixed under (θ,ϕ), minimizing the CE loss is equivlalent to minimizing:

$$\min_{\theta,\phi}\mathbb{E}_{\mathbb{P}^v}\left[\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v}\log\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^{\theta,\phi}}\right] = \min_{\theta,\phi}\mathbb{E}_{\mathbb{P}^v}\left[\frac{1}{Z}\mathrm{e}^{W^v(\boldsymbol{X}_{0:1})}(-\log\mathrm{d}\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{0:1}))\right] \tag{120}$$

Since the path measure $\mathbb{P}^{\theta,\phi}$ is defined by disjoint insertion and unmasking decisions, the probability of a path $\boldsymbol{X}_{0:1}$ under $\mathbb{P}^{\theta,\phi}$ can be written as the product of insertion and unmasking rates:

$$\begin{aligned}\mathrm{d}\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{0:1}) &= \mathbb{P}_0^{\theta,\phi}(\boldsymbol{X}_0)\prod_{t:\boldsymbol{X}_{t^-}\neq\boldsymbol{X}_t}\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{t^-}|\boldsymbol{X}_t)\\ &= \mathbb{P}_0^{\theta,\phi}(\boldsymbol{X}_0)\prod_{t_i:\boldsymbol{X}_{t_i^-}\neq\boldsymbol{X}_{t_i}}\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{t_i^-}|\boldsymbol{X}_{t_i})\prod_{t_u:\boldsymbol{X}_{t_u^-}\neq\boldsymbol{X}_{t_u}}\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{t_u^-}|\boldsymbol{X}_{t_u})\end{aligned} \tag{121}$$

Taking logs, we have:

$$-\log\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{0:1}) = -\log\mathbb{P}_0^{\theta,\phi}(\boldsymbol{X}_0) + \sum_{t_i:\boldsymbol{X}_{t_i^-}\neq\boldsymbol{X}_{t_i}} -\log\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{t_i^-}|\boldsymbol{X}_{t_i}) + \sum_{t_u:\boldsymbol{X}_{t_u^-}\neq\boldsymbol{X}_{t_u}} -\log\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{t_u^-}|\boldsymbol{X}_{t_u})$$

Instead of defining the loss with the probability of generating $\boldsymbol{X}_1 \sim p_{\text{target}}$ via only the single trajectory $\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{0:1})$, we can define it with the expectation over many possible trajectories by taking an expectation over intermediate samples from the endpoint-conditioned interpolant $\tilde{\boldsymbol{x}}_t \sim p_t(\cdot|\boldsymbol{X}_1)$ and times $t\sim\mathcal{U}(0,1)$ following (Zhu et al., 2025a). Then, we have:

$$-\log\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{0:1}) = -\log\mathbb{P}_0^{\theta,\phi}(\boldsymbol{X}_0) + \mathop{\mathbb{E}}_{t\sim\mathcal{U}(0,1)}\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}_t\sim p_t(\cdot|\boldsymbol{X}_1)}\left[-\log\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_s|\tilde{\boldsymbol{x}}_t,\boldsymbol{X}_1) - \log\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_s|\tilde{\boldsymbol{x}}_t,\boldsymbol{X}_1)\right]$$

Since we decompose each step into unmasking via f_θ and remasking via μ_ϕ and insertion via g_θ and deletion under ν_ϕ and define the losses as a form of the negative log-likelihood of the step given the **ground-truth** sequence $\boldsymbol{X}_1 \sim p_{\text{target}}$, we can write:

$$-\log\mathbb{P}^{\theta,\phi}(\boldsymbol{X}_{0:1}) = \underbrace{\mathcal{L}_{\text{unmask}}(\theta;\boldsymbol{X}_1)}_{(3)} + \underbrace{\mathcal{L}_{\text{insert}}(\theta;\boldsymbol{X}_1)}_{(2)} + \underbrace{\mathcal{L}_{\text{UQL}}(\phi;\boldsymbol{X}_1)}_{(6)} + \underbrace{\mathcal{L}_{\text{IQL}}(\phi;\boldsymbol{X}_1)}_{(13)} + \text{const} \tag{122}$$

Therefore, the minimizer in (120) is equivalent to:

$$\begin{aligned}&\min_{\theta,\phi}\mathbb{E}_{\mathbb{P}^v}\left[\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^v}\log\frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^{\theta,\phi}}\right]\\ &\quad= \min_{\theta,\phi}\mathbb{E}_{\mathbb{P}^v}\left[\frac{1}{Z}\mathrm{e}^{W^v(\boldsymbol{X}_{0:1})}\left[\mathcal{L}_{\text{unmask}}(\theta;\boldsymbol{X}_1)+\mathcal{L}_{\text{insert}}(\theta;\boldsymbol{X}_1)+\mathcal{L}_{\text{UQL}}(\phi;\boldsymbol{X}_1)+\mathcal{L}_{\text{IQL}}(\phi;\boldsymbol{X}_1)\right]\right]\end{aligned} \tag{123}$$

and defining $\mathcal{L}_{\text{AJD}}(\theta,\phi)$ as:

$$\mathcal{L}_{\text{AJD}}(\theta,\phi) := \mathop{\mathbb{E}}_{\boldsymbol{X}_{0:1}\sim\mathbb{P}^v}\left[\frac{1}{Z}\mathrm{e}^{W^v}\left[\mathcal{L}_{\text{unmask}}(\theta;\boldsymbol{X}_1)+\mathcal{L}_{\text{insert}}(\theta;\boldsymbol{X}_1)+\mathcal{L}_{\text{UQL}}(\phi;\boldsymbol{X}_1)+\mathcal{L}_{\text{IQL}}(\phi;\boldsymbol{X}_1)\right]\right] \tag{124}$$

concludes our proof. □

D PEPTIDE EXPERIMENT DETAILS

D.1 HYPERPARAMETERS

We provide the specific hyperparameter values used in the peptide experiment in Table 2.

Table 2: **Hyperparameters for peptide design experiment.** The same base hyperparameters were used for **A2D2**, **A2D2 w/o quality**, and **TR2-D2** results.

Hyperparameter	**Value**
Number of Replicates R	8
Buffer Size B	50
Iterations Between Buffer Generation N_{resample}	10
Batch Size	10
Reward Scaling α	0.1

E ALGORITHMS

Here, we provide the complete pseudo-code for the algorithms used in the **A2D2** framework. Algorithm 1 describes the joint fine-tuning procedure that alternates between fine-tuning the policy model and the quality planner model to generate from a target reward-tilted distribution. Algorithm 2 describes the adaptive any-length inference procedure using our trained quality predictors. Algorithm 3 describes the method used to sample an intermediate state $\boldsymbol{x}_t$ from the interpolant given a clean sequence $\boldsymbol{x}_1 \sim p_{\text{target}}$. Algorithm 4 describes the procedure for adaptive remasking and Algorithm 5 describes the procedure for adaptive deletion.

Algorithm 1 `A2D2`: Adaptive Any-Length Discrete Diffusion

Input: Pretrained model $f^{\text{pre}}(\boldsymbol{x}_t, t), g^{\text{pre}}(\boldsymbol{x}_t, t)$, fine-tuned model $f_\theta(\boldsymbol{x}_t, t), g_\theta(\boldsymbol{x}_t, t)$, number of epochs N_{epochs}, buffer size B, number of repeats R
while not converged **do**
  $\{\boldsymbol{x}_i^\star, W^{\bar{\theta},\bar{\phi}}\}_{i=1}^B \leftarrow$ `BatchQualitySample`$(f^{\text{pre}}, g^{\text{pre}}, f_\theta, g_\theta)$
  $\mathcal{B} \leftarrow \{\boldsymbol{x}_{t,i}, W^{\bar{\theta},\bar{\phi}}\}_{i=1}^B$
  for `epoch` in $1, \ldots, N_{\text{epochs}}$ **do**
    ▷ *Fine-tune policy model with frozen planner* ◁
    $\{\tilde{\boldsymbol{x}}_{t,i}, W^{\bar{\theta},\bar{\phi}}\} \leftarrow$ `SampleInterpolant`$(\mathcal{B}; R)$
    Predict unmasking posterior $f_\theta(\tilde{\boldsymbol{x}}_{t,j}, t)$
    $\mathcal{L}_{\text{unmask}}(\theta; \tilde{\boldsymbol{x}}_{t,i}) \leftarrow -\frac{\dot{\beta}_t}{1-\beta_t} \sum_{\ell:\tilde{\boldsymbol{x}}_{t,j}=\boldsymbol{M}} \log f_\theta(\tilde{\boldsymbol{x}}_{t,j}, t)[\ell, \boldsymbol{x}_i^\star]$
    Predict insertion expectation $g_\theta(\tilde{\boldsymbol{x}}_{t,j}, t)$
    $\mathcal{L}_{\text{insert}}(\theta; \tilde{\boldsymbol{x}}_{t,i}) \leftarrow -\frac{\dot{\alpha}_t}{1-\alpha_t} \sum_{\ell=1}^{\text{len}(\tilde{\boldsymbol{x}}_{t,j})+1} \phi\left(s_t[\ell] - s_t[\ell-1], g_\theta(\tilde{\boldsymbol{x}}_{t,j}, t)[\ell]\right)$
    $\mathcal{L}_{\text{AJD}}(\theta) \leftarrow \frac{1}{BR} \sum_{(\tilde{\boldsymbol{x}}_{t,i}, W^{\bar{\theta},\bar{\phi}}) \sim \mathcal{B}} \left[e^{W^{\bar{\theta},\bar{\phi}}} (\mathcal{L}_{\text{unmask}}(\theta; \tilde{\boldsymbol{x}}_{t,i}) + \mathcal{L}_{\text{insert}}(\theta; \tilde{\boldsymbol{x}}_{t,i})) \right]$
    Update θ with $\nabla_\theta \mathcal{L}_{\text{AJD}}(\theta)$
  end for
  for `epoch` in $1, \ldots, N_{\text{epochs}}$ **do**
    ▷ *Fine-tune planner model with frozen policy* ◁
    $\{\tilde{\boldsymbol{x}}_{t,i}, W^{\bar{\theta},\bar{\phi}}\} \leftarrow$ `SampleInterpolant`$(\mathcal{B}; R)$
    Predict $f_\theta(\tilde{\boldsymbol{x}}_{t,j}, t), g_\theta(\tilde{\boldsymbol{x}}_{t,j}, t)$
    $\tilde{\boldsymbol{x}}_{s,i}^{\text{unmask}}, \mathcal{M}, \tilde{\boldsymbol{x}}_{s,i}^{\text{insert}}, \mathcal{I} \leftarrow$ `OneStepSampler`$(\tilde{\boldsymbol{x}}_{t,i}, t, f_\theta(\tilde{\boldsymbol{x}}_{t,j}, t), g_\theta(\tilde{\boldsymbol{x}}_{t,j}, t))$
    Predict insertion quality $\nu_\phi(\tilde{\boldsymbol{x}}_{t,i}^{\text{insert}}, s)$
    $\mathcal{L}_{\text{IQL}}(\phi; \tilde{\boldsymbol{x}}_{t,i}^{\text{insert}}) \leftarrow \sum_{i \in \mathcal{I}} \text{BCE}(\nu_\star^i, \nu_\phi^i(\tilde{\boldsymbol{x}}_{t,i}^{\text{insert}}))$
    Predict unmasking quality $\mu_\phi(\tilde{\boldsymbol{x}}_{t,i}^{\text{unmask}}, s)$
    $\mathcal{L}_{\text{UQL}}(\phi; \tilde{\boldsymbol{x}}_{t,i}^{\text{unmask}}) \leftarrow \sum_{\ell \in \mathcal{M}} \text{BCE}(\mathbf{1}[\tilde{\boldsymbol{x}}_{s,i}^{\text{unmask},\ell} = \boldsymbol{x}_i^{\star,\ell}], \mu_\phi^\ell(\tilde{\boldsymbol{x}}_{s,\ell}^{\text{unmask}}))$
    $\mathcal{L}_{\text{AJD}}(\phi) \leftarrow \frac{1}{BR} \sum_{(\tilde{\boldsymbol{x}}_{t,i}, W_i^{\bar{\theta},\bar{\phi}}) \sim \mathcal{B}} \left[e^{W^{\bar{\theta},\bar{\phi}}} (\mathcal{L}_{\text{IQL}}(\phi; \tilde{\boldsymbol{x}}_{s,i}^{\text{insert}}) + \mathcal{L}_{\text{UQL}}(\phi; \tilde{\boldsymbol{x}}_{s,i}^{\text{unmask}})) \right]$
    Update ϕ with $\nabla_\phi \mathcal{L}_{\text{AJD}}(\phi)$
  end for
end while

Algorithm 2 `BatchQualitySample`: Sample a batch of sequences from the fine-tuned model and compute Radon-Nikodym derivative importance weight

Input: Pretrained model $f^{\text{pre}}(\boldsymbol{x}_t, t), g^{\text{pre}}(\boldsymbol{x}_t, t)$, fine-tuned model $f_\theta(\boldsymbol{x}_t, t), g_\theta(\boldsymbol{x}_t, t)$, reward model $r : \mathcal{X} \to \mathbb{R}$, reward scaling α, sampling steps N_{steps}, max length L, tokens to remask N_{remask}
$\Delta t \leftarrow T / N_{\text{steps}}$
$t \leftarrow 0$
for $i = 1$ to N_{steps} **do**
  $\boldsymbol{R}_t^\theta(\boldsymbol{x}_t, \boldsymbol{x}_s) \leftarrow \frac{\dot\beta_t}{1-\beta_t} \cdot g_\theta(\boldsymbol{x}_t, t) \cdot \Delta t$ ▷ *Predict insertion expectation*
  $\boldsymbol{Q}_t^\theta(\boldsymbol{x}_t, \boldsymbol{x}_s) \leftarrow \frac{\dot\alpha_t}{1-\alpha_t} \cdot f_\theta(\boldsymbol{x}_t, t) \cdot \Delta t$ ▷ *Predict unmasking posterior*
  $\boldsymbol{R}_t^{\text{pre}}(\boldsymbol{x}_t, \boldsymbol{x}_s) \leftarrow \frac{\dot\beta_t}{1-\beta_t} \cdot g^{\text{pre}}(\boldsymbol{x}_t, t) \cdot \Delta t$ ▷ *Predict pretrained insertion expectation*
  $\boldsymbol{Q}_t^{\text{pre}}(\boldsymbol{x}_t, \boldsymbol{x}_s) \leftarrow \frac{\dot\alpha_t}{1-\alpha_t} \cdot f^{\text{pre}}(\boldsymbol{x}_t, t) \cdot \Delta t$ ▷ *Predict pretrained unmasking posterior*
  $\boldsymbol{x}_s \leftarrow \texttt{SampleCategorical}(\boldsymbol{Q}_t^\theta(\boldsymbol{x}_t, \boldsymbol{x}_s))$ ▷ *Unmasking step*
  if $i < N_{\text{steps}} - 1$ **then**
    $\boldsymbol{x}_s \leftarrow \texttt{SchedulAwareRemasking}(\boldsymbol{x}_s, \nu_\phi)$
  end if
  $\texttt{log\_rnd} \leftarrow \texttt{log\_rnd} + \sum_\ell \mathbf{1}[\boldsymbol{x}_s \neq \boldsymbol{x}_t] \cdot (\log \boldsymbol{Q}_t^{\text{pre}}(\boldsymbol{x}_s, \boldsymbol{x}_t)[\ell, \boldsymbol{x}_s^\ell] - \log \boldsymbol{Q}_t^\theta(\boldsymbol{x}_s, \boldsymbol{x}_t)[\ell, \boldsymbol{x}_s^\ell])$
  if $i < N_{\text{steps}} - 1$ **then**
    $L_t \leftarrow \text{len}(\boldsymbol{x}_t)$
    $I_t \sim \texttt{Poisson}(\boldsymbol{R}_t^\theta(\boldsymbol{x}_t, \boldsymbol{x}_s) \cdot \Delta t)$ ▷ *Sample number of tokens to insert at each gap*
    $\texttt{total\_ext} \leftarrow \sum_\ell L_t^\ell$
    if $\texttt{total\_ext} + L_t > L_{\max}$ **then**
      $L_t \leftarrow 0$ ▷ *Do not insert any masks*
    end if
    $L_s \leftarrow L_t + \texttt{total\_ext}$
    $\texttt{ext\_cum}[L_t + 1] \leftarrow \sum_{j=1}^{\ell} I_t^\ell$ ▷ *Compute cumulative insertions up to position ℓ*
    $\boldsymbol{x}'_s \leftarrow (\text{pad})^{L_{\max}}$ ▷ *Initialize sequence of pad tokens*
    $\boldsymbol{x}'_s[: L_s + 1] \leftarrow \boldsymbol{M}$ ▷ *Initialize L_s tokens to mask*
    for $\ell = 1$ to L_t **do**
      $\boldsymbol{x}'_s[\ell + \texttt{ext\_cum}[\ell]] \leftarrow \boldsymbol{x}_s[\ell]$ ▷ *Set unmasked tokens to their new position*
    end for
    $\boldsymbol{x}'_s, I_t^\star \leftarrow \texttt{ScheduleAwareDeletion}(\boldsymbol{x}'_s, \mu_\phi)$
    $\boldsymbol{x}_s \leftarrow \boldsymbol{x}'_s$
    ▷ *Compute log insertion rates under Poisson* ◁
    $\texttt{log\_policy\_insert} \leftarrow I_t^\star \log \boldsymbol{R}_t^\theta(\boldsymbol{x}_t, \boldsymbol{x}_s) - \boldsymbol{R}_t^\theta(\boldsymbol{x}_t, \boldsymbol{x}_s)$
    $\texttt{log\_pre\_insert} \leftarrow I_t^\star \log \boldsymbol{R}_t^{\text{pre}}(\boldsymbol{x}_t, \boldsymbol{x}_s) - \boldsymbol{R}_t^{\text{pre}}(\boldsymbol{x}_t, \boldsymbol{x}_s)$
    $\texttt{log\_rnd} \leftarrow \texttt{log\_rnd} + \sum_\ell (\texttt{log\_pre\_insert}[\ell] - \texttt{log\_policy\_insert}[\ell])$
  end if
  $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_s$
  $t \leftarrow t + \Delta t$
end for
$r(\boldsymbol{x}_1) \leftarrow \texttt{RewardFunc}(\texttt{decode}(\boldsymbol{x}_1))$
$\texttt{log\_rnd} \leftarrow \texttt{log\_rnd} + (r(\boldsymbol{x}_t)/\alpha)$
return $\boldsymbol{x}_1, \texttt{log\_rnd}, r(\boldsymbol{x}_1)$

Algorithm 3 `SampleInterpolant`: Partially remasks and deletes tokens in batch

Input: Batch of clean sequences $\{\boldsymbol{x}_{T,i}, W^{\bar{\theta},\bar{\phi}}\}_{i=1}^{B}$, number of replicates R
Repeat each sequence in batch $\{\boldsymbol{x}_{T,j}, W^{\bar{\theta},\bar{\phi}}\}_{j=1}^{B\times R}$
Initialize batch of corrupted sequences $\{\boldsymbol{x}_{t,j}\}_{j=1}^{B\times R}$
for $\boldsymbol{x}_{T,j}$ in $\{\boldsymbol{x}_{T,j}\}_{j=1}^{B\times R}$ **do**
  $t \sim \mathcal{U}(0,1)$
  for $\ell = 1$ to L **do**
    $t_i^\ell \sim \dot{\alpha}_t dt$ ▷ *Sample insertion time*
    $t_u^\ell \sim \mathbf{1}[t \geq t_i^\ell] \frac{\dot{\beta}_t}{1-\beta_{t_i^\ell}} dt$ ▷ *Sample unmasking time*
    if $t < t_i^\ell$ **then**
      $\boldsymbol{x}_{t,j}^\ell \leftarrow$ (empty)] ▷ *Deleted token at time t*
    else if $t_i^\ell \leq t < t_u^\ell$ **then**
      $\boldsymbol{x}_{t,j}^\ell \leftarrow \boldsymbol{M}$ ▷ *Masked token at time t*
    else
      $\boldsymbol{x}_{t,j}^\ell \leftarrow \boldsymbol{x}_{T,j}^\ell$ ▷ *Clean token at time t*
    end if
    Remove intermediate (empty) from $\boldsymbol{x}_{t,j}^\ell$ and replace as pads at end of sequence
  end for
end for
return $\{\boldsymbol{x}_{t,j}, W^{\bar{\theta},\bar{\phi}}\}_{j=1}^{B\times R}$

Algorithm 4 `ScheduleAwareRemasking`: Ensure the number of masked tokens after remasking matches the expected mask count by remasking low-quality tokens.

Input: Draft sequence after unmasking $\tilde{\boldsymbol{x}}_s$, time s, unmasking quality model μ_ϕ, unmasked indices $\mathcal{M}_t$
`exp_num_mask` $\leftarrow$ `interpolant.exp_mask_fraction`$(\tilde{\boldsymbol{x}}_s, s)$
`num_mask` $\leftarrow |\{\ell : \tilde{\boldsymbol{x}}_s^\ell = \boldsymbol{M}\}|$
`mask_to_add` $\leftarrow$ `exp_num_mask` $-$ `num_mask`
for $\ell \in \mathcal{M}_t$ **do**
  `remask_scores`$[\ell] \leftarrow -\mu_\phi^\ell(\tilde{\boldsymbol{x}}_s)$ ▷ *lower confidence has higher probability of remask*
end for
$\mathcal{R}_s \leftarrow$ `topk_indices(remask_scores,mask_to_add)` ▷ *set of indices to remask*
for $i \in \mathcal{R}_s$ **do**
  $\tilde{\boldsymbol{x}}_s^\ell \leftarrow \boldsymbol{M}$
end for
return $\tilde{\boldsymbol{x}}_s$

Algorithm 5 `ScheduleAwareDeletion`: Ensure the sequence length after insertion matches the expected length by deleting low-quality insertions

Input: Draft sequence after insertion $\tilde{\boldsymbol{x}}_s$, time s, insertion quality model ν_ϕ, inserted indices $\mathcal{I}_t$, threshold quality μ_{min}
`exp_length` $\leftarrow$ `interpolant.exp_length`$(\tilde{\boldsymbol{x}}_s, s)$
if `len`$(\tilde{\boldsymbol{x}}_s) \geq$ `exp_length` **then**
  `to_delete` $\leftarrow$ `len`$(\tilde{\boldsymbol{x}}_s)$ $-$ `exp_length`
  `insert_scores`$[i] \leftarrow -\nu_\phi^i(\tilde{\boldsymbol{x}}_s)$ ▷ *lower confidence has higher probability of deletion*
  $\mathcal{D}_s \leftarrow$ `topk_indices(insert_scores,to_delete)` ▷ *set of indices to delete*
  $\tilde{\boldsymbol{x}}_s^i \leftarrow$ (empty)
end if
return $\tilde{\boldsymbol{x}}_s$