MINIMAL TRAVEL SAMPLING FOR METRIC-PATH DISCRETE FLOW MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete flow models built on metric-induced probability paths have emerged as a strong approach to image and video synthesis, yet their fast sampling remains underexplored. In this paper, we propose the travel-optimal (TO) sampler, a training-free sampler for fast generation with pretrained discrete flow models. Our key observation is that a sampler simulating the underlying continuous-time Markov chain on a finite time grid can represent at most one jump per token within each step, so how much probability moves within a step largely determines the generation quality under a limited number of function evaluations (NFE). Motivated by this, we seek the *probability flux* with minimal total travel cost, defined as the expected distance traveled by probability mass. Since infinitely many rates generate the same probability path, this flux is a free design choice. We show that this problem reduces to optimal transport and propose row-Sinkhorn to solve its entropic OT relaxation efficiently for large vocabularies. We further reduce the remaining scheduler error through temporal integration, and our error analysis shows that the sampling error is bounded by a term proportional to the total travel cost. Extensive experiments show that TO improves image generation quality over the default sampler by up to 37% in GenEval at low NFE and video generation quality by up to 3.7 points in VBench, consistently across different models and against various baselines.

1 INTRODUCTION

Discrete flow models generate data by simulating a continuous-time Markov chain (CTMC) over token sequences (Campbell et al., 2022; 2024; Gat et al., 2024), and combined with discrete visual tokenizers, have recently emerged as a strong alternative for image and video synthesis (Wang et al., 2026; Deng et al., 2026). Particularly promising are metric-induced probability paths (Shaul et al., 2025), which move each token progressively toward its target along a metric on the codebook embeddings instead of masking it. Every token can thus be revised throughout generation, enabling iterative global refinement with self-correction, unlike autoregressive or masked diffusion models that fix each token once it is generated (Chen et al., 2025; Swerdlow et al., 2025; Liu et al., 2024).

Fast sampling for these models, however, has received comparatively little attention. While a large body of work has developed samplers that produce high-quality samples of continuous diffusion models with a smaller number of function evaluations (NFE) (Song et al., 2020; Karras et al., 2022; Lu et al., 2022), only a handful exist for discrete flow models. Tau-leaping (Gillespie, 2001; Campbell et al., 2022) accumulates all transitions within a step under frozen rates, and Euler-type samplers (Campbell et al., 2022; Gat et al., 2024), used by metric-path models with the kinetic-optimal (KO) rate (Shaul et al., 2025), allow each token at most one jump per step. More recently, Wan et al. (2026) correct the temporal and location errors of Euler steps, the latter with additional NFE, and TR-CIE (Fu & Fan, 2026) extrapolates the cumulative jump intensity under a time reparameterization. All of these methods take the generating rate as given and aim to follow the resulting CTMC more faithfully.

Yet the error of all such samplers is governed by *how much the true process moves within a step*. Each step approximates the transition kernel of a whole interval from a single network evaluation, so whenever a token jumps several times or the rates change substantially after a jump, any update that caps the number of jumps or freezes the rates becomes inaccurate. We therefore take the opposite

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

view. Instead of making the sampler follow a given rate more faithfully, we choose the rate so that it is easy for the sampler to follow.

To this end, we propose the *travel-optimal (TO) sampler*, which exploits the freedom in the generating rate to make finite-step sampling easier. Since the marginal path constrains only the divergence of the probability flux, infinitely many rates can generate the same marginals. Among them, the ideal TO rate minimizes probability-mass travel, which we show reduces to a Kantorovich optimal transport (OT) problem between the instantaneous net outflow and inflow, while also attaining the smallest possible jump intensity at every state. This directly addresses the one-jump bottleneck: a finite-step Euler sampler can realize at most one jump per coordinate, making frequent or long moves harder to approximate. Since exact OT is prohibitively expensive for large vocabularies, we propose *row-Sinkhorn*, an $O(S)$ relaxation that preserves the minimal jump intensity, routes departing mass locally through a metric kernel, and incurs only a bounded deviation from the prescribed path (Proposition 1). Finally, we adopt analytic temporal integration from Wan et al. (2026) at no additional NFE. Our analysis (Section 5) bounds the sampling error by temporal, jump-intensity, and travel-intensity terms, directly linking the TO objective to finite-step sampling accuracy.

Extensive experiments show that our method improves sample quality under the same NFE budget across models and modalities. Specifically, on the large-scale metric-path models URSA (Deng et al., 2026) and FUDOKI (Wang et al., 2026), our TO sampler raises the GenEval score of URSA from 0.46 to 0.63 and its VBench score from 68.88 to 72.57 at $K{=}4$, and also outperforms recently published discrete flow samplers (Fu & Fan, 2026; Wan et al., 2026).

2 Preliminaries

2.1 Continuous-time Markov chains

We consider sequences $x = (x^1, \dots, x^N) \in \mathcal{D} = [S]^N$ over a vocabulary $[S] = \{1, \dots, S\}$ of size S, and write $x^{\bar{\imath}}$ for all coordinates except the i-th. A CTMC on $\mathcal{D}$ is specified by a time-dependent rate matrix R_t, where $R_t(x, y) \geq 0$ for $y \neq x$ is the instantaneous rate of a jump $x \to y$ and $R_t(x, x) = -\sum_{y \neq x} R_t(x, y)$. Its total jump rate is $\Lambda_t(x) = \sum_{y \neq x} R_t(x, y)$. In contrast, the exact transition kernel $P_{s,t}(x, \cdot) = \mathcal{L}(X_t \mid X_s = x)$ describes transitions over a finite interval. Locally,

$$P_{t,t+h}(x, y) = \delta_x(y) + hR_t(x, y) + o(h), \quad R_t(x, y) = \partial_h P_{t,t+h}(x, y)\big|_{h=0}. \tag{1}$$

Thus, for a sufficiently small interval h, the transition probability from x to $y \neq x$ is approximately $hR_t(x, y)$, while the probability of leaving x is approximately $h\Lambda_t(x)$.

Let p_t denote the marginal law of X_t. We define the probability flux $J_t(x, y) = p_t(x)R_t(x, y)$ for $x \neq y$, the probability mass moving from state x to state y per unit time, and the discrete divergence of a flux J as $(\operatorname{div} J)(y) = \sum_{x \neq y}[J(y, x) - J(x, y)]$, the net outflow at y. The Kolmogorov forward equation then takes the form of a continuity equation,

$$\dot{p}_t(y) + (\operatorname{div} J_t)(y) = 0, \qquad y \in \mathcal{D}, \tag{2}$$

which states that the probability of each state changes only through the net flow of mass into it, so that the total probability is conserved. We say that R_t generates p_t if (2) holds.

Following standard discrete flow models (Campbell et al., 2022; 2024; Gat et al., 2024), we use factorized rates under which each jump changes a single coordinate,

$$R_t(x, y) = \sum_{i=1}^{N} \delta_{x^{\bar{\imath}}}(y^{\bar{\imath}})\, R_t^i(y^i \mid x), \qquad y \neq x, \tag{3}$$

where $\delta_{x^{\bar{\imath}}}(y^{\bar{\imath}})$ indicates that all coordinates other than i remain unchanged, and $R_t^i(v \mid x)$ is the rate at which coordinate i moves from x^i to token $v \neq x^i$. The per-coordinate jump rate is $\Lambda_t^i(x) = \sum_{v \neq x^i} R_t^i(v \mid x)$, so that $\Lambda_t(x) = \sum_i \Lambda_t^i(x)$, and we set $R_t^i(x^i \mid x) = -\Lambda_t^i(x)$.

2.2 Metric-Path Discrete Flow Models

Discrete flow models construct the marginal path from per-token conditional paths $p_t(x^i \mid x_1^i)$ toward a clean endpoint x_1, averaged over the data distribution. Correspondingly, if a conditional rate

$R_t^i(y^i \mid x^i, x_1^i)$ generates $p_t(\cdot \mid x_1^i)$, the marginal path is generated by its posterior average (Gat et al., 2024, Thm. 2),

$$p_t(x) = \sum_{x_1} p_1(x_1) \prod_{i=1}^{N} p_t(x^i \mid x_1^i), \qquad R_t^i(y^i \mid x) = \sum_{x_1^i \in [S]} R_t^i(y^i \mid x^i, x_1^i)\, p_{1|t}(x_1^i \mid x). \quad (4)$$

The posterior $p_{1|t}(x_1^i \mid x)$ is unknown at sampling time and is approximated by a neural network $p_{1|t}^\theta$ that predicts the clean tokens from the current state x. In practice, the conditional rate is evaluated at a clean-token estimate $\hat{a}$ drawn from $p_{1|t}^\theta(\cdot \mid x)$ for each coordinate, which yields an unbiased estimate of the marginal rate (Wang et al., 2026; Deng et al., 2026).

Metric paths. In this paper, we focus on metric-induced paths (Shaul et al., 2025), which are mostly employed in recent large-scale visual generative models (Wang et al., 2026; Deng et al., 2026),

$$p_t(x^i \mid x_1^i) = \frac{\exp[-\beta_t d(x^i, x_1^i)]}{\sum_{b \in [S]} \exp[-\beta_t d(b, x_1^i)]}, \quad (5)$$

where d is a squared euclidean distance on the codebook embeddings, and β_t is an increasing scheduler with $\beta_0 = 0$ and $\beta_t \to \infty$ as $t \to 1$ (in practice, sampling typically stops early before $t = 1$). The conditional distribution thus evolves from a uniform source toward the clean token x_1^i, progressively concentrating probability on tokens that are closer to it in the metric. Unlike in mixture paths, a token can therefore travel through intermediate tokens that are increasingly similar to its target, so that where each jump lands becomes a meaningful degree of freedom.

3 Problem and Motivation

3.1 Sampling under a finite NFE budget

Exact simulation of a CTMC must account both for the time dependence of the rates and for their change after each jump. It determines when the next jump occurs and to which state, and then re-evaluates the rates at the new state before simulating the following jump. Since the rate depends on $p_{1|t}^\theta(\cdot \mid x)$, every re-evaluation costs a network evaluation, which makes exact simulation expensive when many jumps occur.

In practice, one instead fixes a grid $0 = t_0 < t_1 < \cdots < t_K = 1$, evaluates the network once at each t_k, and approximates the entire transition kernel $P_{t_k, t_{k+1}}(x, \cdot)$ of the interval $[t_k, t_{k+1}]$ from that single evaluation. The most widely used approximation replaces the infinitesimal step (1) by a finite one and applies it independently to each coordinate. Writing $h_k = t_{k+1} - t_k$ and $z = X_{t_k}$, the *Euler* kernel for coordinate i is

$$P_k^i(y^i \mid z) = \delta_{z^i}(y^i) + h_k\, R_{t_k}^i(y^i \mid z). \quad (6)$$

This is a valid probability mass function only if $h_k \Lambda_{t_k}^i(z) \le 1$, since otherwise the self-transition probability $1 - h_k \Lambda_{t_k}^i(z)$ becomes negative. In practice, one therefore uses the exponential update

$$P_k^i(y^i \mid z) = \begin{cases} 1 - \rho_k^i, & y^i = z^i, \\ \rho_k^i \dfrac{R_{t_k}^i(y^i \mid z)}{\Lambda_{t_k}^i(z)}, & y^i \neq z^i, \end{cases} \qquad \rho_k^i = 1 - \exp\big(-h_k \Lambda_{t_k}^i(z)\big), \quad (7)$$

with $P_k^i(\cdot \mid z) = \delta_{z^i}$ when $\Lambda_{t_k}^i(z) = 0$, referred to as the *always-valid Euler* (AVE) sampler (Shaul et al., 2025; Deng et al., 2026; Wan et al., 2026). It agrees with (6) to first order in h_k and remains a valid probability kernel for any step size.

3.2 The one-jump bottleneck

The kernel (7) has a structural limitation. Within one interval, *each coordinate jumps at most once*, and the destination is drawn from the rate at the *initial* state z, while the true CTMC may jump several times in coordinate i, and the rates of every coordinate keep changing as the scheduler advances and the posterior $p_{1|t}(\cdot \mid X)$ is reconditioned after each jump on the updated sequence.

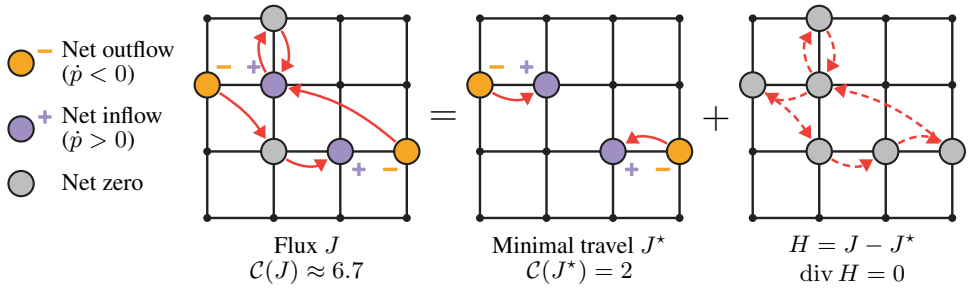

Figure 1: J and $J^\star$ have the same net flow at every token and thus generate the same marginals. Their difference H is invisible to the marginals but adds travel that a sampler should approximate.

To see how large these effects are, consider a single coordinate whose jump rate stays close to Λ over an interval of length h. The number of its jumps is then approximately Poisson with mean $h\Lambda$, so

$$\Pr(\text{jumps} \geq 2) \approx 1 - e^{-h\Lambda}(1 + h\Lambda) \approx \tfrac{1}{2}(h\Lambda)^2, \tag{8}$$

an event to which (7) assigns probability zero. This inability to represent multiple jumps also causes a second problem, which we call *state staleness*. After the first jump, the true process evolves from a new state, and its subsequent rates are conditioned on that state. The sampler, however, remains tied to the rates computed at the beginning of the interval. The mismatch is more severe when the jump carries the process *farther* from its initial state, since nearby states tend to induce similar subsequent dynamics while distant ones need not.

Existing methods try to address this from the sampler side, but each still has limitations. Tau-leaping allows several jumps under frozen rates, which is inaccurate when jumps interact and ill-defined for categorical tokens. The location-corrected sampler of Wan et al. (2026) re-evaluates the network after a jump, which buys accuracy at the price of additional NFE. **All of these methods take the rate R_t as given and ask how to follow it more faithfully**. In contrast, we ask whether the rate itself can be chosen to make sampling easier.

4 Minimal Travel Sampler

To address the sampling errors discussed above, we directly optimize the **generating flux** so that probability mass moves as little as possible between states, which makes it easier for a one-jump sampler to approximate. The key observation is that the flux generating a prescribed marginal path is not unique. The continuity equation (2) constrains only the *divergence* of the flux, so any two fluxes with the same divergence generate the same marginal path.

Lemma 1 (Divergence-free freedom)**.** *Let J_t generate p_t. A flux $J'_t \geq 0$ generates p_t if and only if $H_t = J'_t - J_t$ is divergence-free, $\operatorname{div} H_t = 0$. In particular, for any H_t with $\operatorname{div} H_t = 0$ and $J_t + H_t \geq 0$, the rate $R'_t(x, y) = (J_t(x, y) + H_t(x, y))/p_t(x)$ generates p_t.*

As shown in Figure 1, the same net flow can be realized by many different fluxes. If each edge carries a movement cost, fluxes that route mass through additional detours incur unnecessary movement. Such movement may act as self-correction, but at low NFE it causes larger approximation errors for a one-jump sampler. Our main motivation is to reduce this unnecessary travel.

4.1 Total travel cost minimization

For simplicity, we assume a single-coordinate process ($N = 1$) on the vocabulary $[S]$. Let $c : [S] \times [S] \to [0, \infty)$ be a metric. We define the **travel cost** of a flux J as $\mathcal{C}(J) = \sum_{x \neq y} c(x, y)J(x, y)$. We generally take $c = \sqrt{d}$ for the metric path (5), so the travel cost measures the expected distance traveled in the embedding. Among all fluxes that generate the prescribed marginal path $(p_t)_{t\in[0,1]}$, we seek the one with minimal total travel over the entire path,

$$\{J^\star_t\}_{t\in[0,1]} \in \operatorname*{arg\,min}_{\{J_t\}_{t\in[0,1]}} \int_0^1 \mathcal{C}(J_t)\, dt \quad \text{s.t.} \quad J_t \geq 0, \qquad \dot{p}_t + \operatorname{div} J_t = 0 \quad \text{for all } t \in [0, 1]. \tag{9}$$

Since the marginal path p_t is fixed, both the objective and the continuity constraint are separable across time, and the global problem (9) can be reduced to independently solving, at each time t,

$$J^\star_t \in \operatorname*{arg\,min}_{J \geq 0} \ \mathcal{C}(J) \quad \text{s.t.} \quad \dot{p}_t + \operatorname{div} J = 0. \tag{10}$$

Any collection of such pointwise minimizers minimizes the total travel over the entire path. Moreover, each $J_t^\star$ generates p_t by construction, so $R_t^\star = J_t^\star / p_t$ is a legitimate generating rate that can be plugged into any sampler.

4.2 Minimal travel flux as optimal transport

It remains to solve the instantaneous problem (10) at each time t. We now show that it is exactly a direct optimal transport problem from states losing probability to states gaining it, since routing mass through intermediate states cannot reduce the cost.

Theorem 1 (Minimal travel is optimal transport)**.** *Let c be a metric on $[S]$, and define the instantaneous net outflow μ_t and net inflow ν_t by*

$$\mu_t(x) = \underbrace{\big[-\dot{p}_t(x)\big]_+}_{\textit{outflow}}, \qquad \nu_t(x) = \underbrace{\big[\dot{p}_t(x)\big]_+}_{\textit{inflow}}, \tag{11}$$

where $[a]_+ = \max(a, 0)$. Since $\sum_x \dot{p}_t(x) = 0$, both have the same total mass $m_t := \sum_x \mu_t(x) = \sum_x \nu_t(x) = \frac{1}{2}\|\dot{p}_t\|_1$. Then the minimum in (10) equals

$$\mathrm{OT}_c(\mu_t, \nu_t) = \min_{\gamma \in \Gamma(\mu_t, \nu_t)} \sum_{x,y} c(x, y)\, \gamma(x, y), \tag{12}$$

where $\Gamma(\mu_t, \nu_t)$ is the set of couplings of μ_t and ν_t, that is, of nonnegative γ with $\sum_y \gamma(x, y) = \mu_t(x)$ and $\sum_x \gamma(x, y) = \nu_t(y)$. Moreover, every optimal coupling $\gamma_t^\star$ is a minimizer of (10).

Notably, the jump intensity can be computed directly without solving the optimal transport problem, since μ_t already determines how much mass must leave each state, and an OT solver is needed only to decide where that mass goes. The following corollary makes this decomposition explicit.

Corollary 1 (Closed-form jump intensity)**.** *Let $\gamma_t^\star \in \Gamma(\mu_t, \nu_t)$ be an optimal coupling, and let $R_t^\star(x, y) = \gamma_t^\star(x, y)/p_t(x)$ be the induced rate for $p_t(x) > 0$. Its jump intensity is*

$$\Lambda_t^\star(x) = \sum_{y \neq x} R_t^\star(x, y) = \frac{[-\dot{p}_t(x)]_+}{p_t(x)}, \tag{13}$$

and is minimal among all rates realizing the same marginal path p_t. For the metric path (5) conditioned on a clean token a, with $\dot{\beta}_t \geq 0$, the jump intensity is

$$\Lambda_t^\star(b \mid a) = \dot{\beta}_t \left[d(b, a) - \bar{d}_t(a)\right]_+, \qquad \bar{d}_t(a) = \sum_{b' \in [S]} p_t(b' \mid a)\, d(b', a). \tag{14}$$

The minimal jump intensity is thus fixed by the probability path alone, and optimal transport determines only the conditional destination distribution. For metric paths, the intensity has a transparent form. Only tokens farther from the target than the current mean distance $\bar{d}_t(a)$ carry outgoing mass, and they leave at a rate proportional to their excess distance.

4.3 Solving the transport problem

Entropic transport. The transport problem (12) is a linear program with S^2 variables, which is prohibitively expensive to solve directly for large vocabularies. Following Cuturi (2013), we add an entropic regularizer and obtain a problem that Sinkhorn iterations solve efficiently,

$$\gamma_\varepsilon = \operatorname*{arg\,min}_{\gamma \in \Gamma(\mu, \nu)} \sum_{b, b'} c(b, b')\, \gamma(b, b') + \varepsilon \sum_{b, b'} \gamma(b, b') \log \gamma(b, b'), \tag{15}$$

where μ and ν stand for $\mu_t(\cdot \mid \hat{a})$ and $\nu_t(\cdot \mid \hat{a})$. The solution has the form $\gamma_\varepsilon(b, b') = u(b)\, K(b, b')\, v(b')$ with the kernel $K(b, b') = \exp(-c(b, b')/\varepsilon)$, where the scaling vectors u, v are obtained by alternating the Sinkhorn updates $u \leftarrow \mu/(Kv)$ and $v \leftarrow \nu/(K^\top u)$ until convergence.

Even so, solving the entropic problem still incurs substantial overhead at the vocabulary sizes used by modern visual tokenizers. Each Sinkhorn iteration requires the two dense kernel-vector products Kv and $K^\top u$ with $K \in \mathbb{R}^{S \times S}$. For $S = 131{,}072$, explicitly materializing K would require $S^2 \approx 1.7 \times 10^{10}$ entries (68.7 GB in FP32). Even without materializing K, every Sinkhorn normalization requires a dense $S \times S$ kernel action, for every coordinate i and every time step t. Repeated Sinkhorn iterations are therefore too expensive for the inner loop of a low-NFE sampler.

Row-Sinkhorn. To avoid this cost, we exploit an asymmetry specific to sampling. At a given step and coordinate, the sampler needs only the row of the transport plan that leaves the *current* token b. Enforcing the exact target marginal $\sum_b \gamma(b, b') = \nu_t(b' \mid \hat{a})$, in contrast, requires coordinating the outgoing transport of every other source token, and this coordination is precisely what introduces the dense $S \times S$ computation. We therefore retain the source-marginal constraint and relax the target-marginal constraint. For $m_t = \sum_b \mu_t(b \mid \hat{a}) > 0$, we consider

$$\begin{aligned} \widehat{\gamma}_t(\cdot, \cdot \mid \hat{a}) = \arg\min_{\gamma \geq 0} \; & \sum_{b,b'} c(b, b')\, \gamma(b, b') + \varepsilon\, \mathrm{KL}\Big(\gamma \,\Big\|\, \tfrac{1}{m_t}\, \mu_t(\cdot \mid \hat{a}) \otimes \nu_t(\cdot \mid \hat{a})\Big) \\ \text{s.t.} \; & \sum_{b'} \gamma(b, b') = \mu_t(b \mid \hat{a}) \quad \text{for all } b. \end{aligned} \tag{16}$$

This formulation arises naturally from (15). Under both marginal constraints, the entropic objective equals the transport cost plus $\varepsilon\, \mathrm{KL}(\gamma \,\|\, \mu_t \otimes \nu_t / m_t)$ up to a constant independent of γ, and (16) simply drops the target-marginal constraint from this equivalent form. Unlike the full transport problem, (16) decouples across source tokens b, and, since the objective is strictly convex, each row has a unique closed-form solution

$$\widehat{\gamma}_t(b, b' \mid \hat{a}) = \mu_t(b \mid \hat{a})\, \frac{\nu_t(b' \mid \hat{a})\, K(b, b')}{\sum_{b''} \nu_t(b'' \mid \hat{a})\, K(b, b'')}, \qquad K(b, b') = \exp\!\left(-\frac{c(b, b')}{\varepsilon}\right). \tag{17}$$

Dividing by the prescribed row mass gives the conditional destination distribution

$$\widehat{Q}_t(b' \mid b, \hat{a}) = \frac{\widehat{\gamma}_t(b, b' \mid \hat{a})}{\mu_t(b \mid \hat{a})} \;\propto\; \nu_t(b' \mid \hat{a})\, \exp\!\left(-\frac{c(b, b')}{\varepsilon}\right). \tag{18}$$

Row-Sinkhorn can thus be implemented simply by reweighting the original inflow distribution $\nu_t(\cdot \mid \hat{a})$ by a distance kernel centered at the current token b. This distinguishes our rate from the KO rate used by existing metric-path models (Shaul et al., 2025), whose destination weights depend on the current token b only through the scalar $d(b, \hat{a})$. Our kernel instead depends on the source–destination distance $d(b, b')$ itself, so that each jump favors destinations close to the current token in codebook space. Since sampling requires only the single kernel row $K(b, \cdot)$, this reweighting costs $O(S)$ per coordinate and can be computed on the fly from the codebook embeddings. Interestingly, (17) is exactly the first row-normalization half-step of Sinkhorn initialized with $v = \nu_t(\cdot \mid \hat{a})$, which motivates the name *row-Sinkhorn*. In practice, additional Sinkhorn iterations toward the exact target marginal do not improve sample quality at low NFE (Table 5).

Relaxing the target marginal changes where the departing mass lands, but not how much of it leaves the current state. Since $\nu_t(b \mid \hat{a}) = 0$ whenever $\mu_t(b \mid \hat{a}) > 0$, the plan $\widehat{\gamma}_t$ has no diagonal mass, and the rate $\widehat{R}_t(b' \mid b, \hat{a}) = \widehat{\gamma}_t(b, b' \mid \hat{a}) / p_t(b \mid \hat{a})$ has jump rate

$$\widehat{\Lambda}_t(b \mid \hat{a}) := \sum_{b' \neq b} \widehat{R}_t(b' \mid b, \hat{a}) = \frac{1}{p_t(b \mid \hat{a})} \sum_{b' \neq b} \widehat{\gamma}_t(b, b' \mid \hat{a}) = \frac{\mu_t(b \mid \hat{a})}{p_t(b \mid \hat{a})} = \Lambda^\star_t(b \mid \hat{a}). \tag{19}$$

Row-Sinkhorn therefore preserves the minimal jump intensity exactly and keeps transport local through the metric kernel, relaxing only where the departing mass lands. The resulting deviation from the model's path is controlled by the inflow mismatch and remains bounded as,

Proposition 1 (Landing error). *Let $\widehat{\nu}_t(b' \mid a) = \sum_b \widehat{\gamma}_t(b, b' \mid a)$ denote the inflow realized by row-Sinkhorn. Then $\|\widehat{\nu}_t(\cdot \mid a) - \nu_t(\cdot \mid a)\|_1 \leq 2m_t$, so it never exceeds the path speed and $\widehat{\nu}_t(\cdot \mid a) \to \nu_t(\cdot \mid a)$ as $\varepsilon \to \infty$. Moreover, the CTMC started from p_0 using $\widehat{R}_t(\cdot \mid \cdot, a)$ has marginals $\widetilde{p}_t$ satisfying $\|\widetilde{p}_t - p_t\|_1 \leq \int_0^t \Delta_s\, ds$, where $\Delta_s = \sum_{i=1}^N \mathbb{E}_{x_1 \sim p_1} \big\|\widehat{\nu}_s(\cdot \mid x_1^i) - \nu_s(\cdot \mid x_1^i)\big\|_1$.*

4.4 Temporal integration

Our transport construction changes the spatial routing of the rate, but AVE still incurs the usual temporal error from evaluating the scheduler at the left endpoint. Following Wan et al. (2026), we freeze only the state- and network-dependent terms and integrate the scalar scheduler factor exactly. Whenever the frozen jump rate takes the form $\Lambda^i_s(z) = \dot{\beta}_s\, A^i_{t_k}(z)$ for $s \in [t_k, t_{k+1}]$, we can safely replace the left-endpoint approximation $h_k \dot{\beta}_{t_k} A^i_{t_k}(z)$ by

$$\int_{t_k}^{t_{k+1}} \Lambda^i_s(z)\, ds = A^i_{t_k}(z) \int_{t_k}^{t_{k+1}} \dot{\beta}_s\, ds = \big(\beta_{t_{k+1}} - \beta_{t_k}\big)\, A^i_{t_k}(z), \tag{20}$$

and use this integrated jump rate in the AVE jump probability, $\rho_k^i = 1 - \exp\Big(-\int_{t_k}^{t_{k+1}} \Lambda_s^i(z)\, ds\Big)$. Temporal integration thus corrects the scheduler discretization while leaving the network evaluation unchanged, and requires no additional NFE. We present the full algorithm of the *travel-optimal sampler* with temporal integration for metric-path discrete flow models in Algorithm 1.

5 Theoretical Analysis

We now formalize the intuition of Section 3.2 that a rate with shorter within-step travel is easier for a one-jump sampler to follow. For the analysis, we extend c additively to sequences, $c(x, y) = \sum_{i=1}^N c(x^i, y^i)$, and define the instantaneous travel intensity $C_t(x)$ and its path integral $\mathcal{T}(R)$ by

$$C_t(x) = \sum_{i=1}^N \sum_{v \neq x^i} R_t^i(v \mid x)\, c(x^i, v), \qquad \mathcal{T}(R) = \int_0^1 \mathbb{E}_{X_t \sim p_t}\left[C_t(X_t)\right] dt. \tag{21}$$

Here $C_t(x)$ measures the expected distance traveled per unit time from state x. Since $\mathbb{E}_{p_t}[C_t(X_t)] = \mathcal{C}(J_t)$, the path integral $\mathcal{T}(R)$ recovers the total travel objective (9) of Section 4.1.

Theorem 2 (One-step AVE error)**.** *Consider an interval $I_k = [t_k, t_{k+1}]$ of length h_k, and let $P_k = P_{t_k, t_{k+1}}$ and P_k^{AVE} denote the one-step kernels of the true CTMC and of the AVE sampler, respectively. Assume that the rate is L_k^t-Lipschitz in time, $\sum_i \sum_{v \neq x^i} |R_t^i(v \mid x) - R_s^i(v \mid x)| \leq L_k^t |t - s|$, that the jump rates are bounded, $\sum_{v \neq x^i} R_t^i(v \mid x^i, x_1^i) \leq H_k$, and that the posterior of each coordinate is ℓ_k-Lipschitz in the other coordinates, $\|p_{1|t}^i(\cdot \mid x) - p_{1|t}^i(\cdot \mid y)\|_1 \leq \ell_k\, c(x^j, y^j)$ whenever x and y differ only in a coordinate $j \neq i$, for all $s, t \in I_k$. Let $c_k = (N-1) H_k \ell_k$. Then the total variation distance between them satisfies*

$$\mathrm{TV}\big(P_k(x, \cdot), P_k^{\mathrm{AVE}}(x, \cdot)\big) \leq \frac{1}{2} h_k^2 L_k^t + \frac{1}{2} h_k^2 \big(H_k \Lambda_{t_k}(x) + c_k C_{t_k}(x)\big). \tag{22}$$

See Appendix C for the proof. For the metric path rate we mainly compare (Shaul et al. (2025)), both constants are explicit and independent of the choice of rate: $H_k = \sup_{t \in I_k} \dot{\beta}_t\, d_{\max}$ and $\ell_k = \sqrt{d_{\max}}\, \sup_{t \in I_k} \beta_t$, and thus governed by the jump intensity Λ_{t_k} and the travel intensity C_{t_k}. This bound can also be understood intuitively. By the triangle inequality, the rate change satisfies

$$|R_s^i(v \mid X_s) - R_{t_k}^i(v \mid x)| \leq |R_s^i(v \mid X_s) - R_{t_k}^i(v \mid X_s)| + |R_{t_k}^i(v \mid X_s) - R_{t_k}^i(v \mid x)|.$$

The first term compares the rate at the same state but at different times, and therefore captures *temporal staleness*, which is controlled by L_k^t (hence the first term in (22)), and our temporal integration aims to reduce this error. The second term captures *state staleness*, which arises only after the process has moved within the step. A coordinate that has already jumped is frozen by AVE and misses at most H_k of jump rate, while the remaining coordinates see stale posteriors, off by at most $H_k \ell_k$ per unit distance traveled by the others. By time s into the step, the expected number of jumped coordinates and the distance traveled are at most $s\Lambda_{t_k}(x)$ and $sC_{t_k}(x)$, so integrating the mismatch over the step gives the h_k^2 jump-rate and travel terms. Since H_k and ℓ_k do not depend on the choice of rate for the metric path, the exact minimal-travel rate reduces both terms by attaining the minimal jump rate and the minimal travel. Similarly, we show in the following corollary that this bound naturally extends to the final sample distribution, whose error is governed by the total travel cost and the jump intensity.

Corollary 2 (Final sampling error)**.** *Let $\widehat{p}_{t_K}$ denote the distribution of the AVE sampler at the final time t_K, initialized with p_{t_0}, and define $\mathcal{E}_h^{\mathrm{time}} = \sum_{k=0}^{K-1} h_k^2 L_k^t$, $\mathcal{J}_h(R) = \sum_{k=0}^{K-1} h_k\, \mathbb{E}_{X_{t_k} \sim p_{t_k}}[\Lambda_{t_k}(X_{t_k})]$, and $\mathcal{T}_h(R) = \sum_{k=0}^{K-1} h_k\, \mathbb{E}_{X_{t_k} \sim p_{t_k}}[C_{t_k}(X_{t_k})]$. Then*

$$\|\widehat{p}_{t_K} - p_{t_K}\|_1 \leq \mathcal{E}_h^{\mathrm{time}} + \max_k (h_k H_k)\, \mathcal{J}_h(R) + \max_k (h_k c_k)\, \mathcal{T}_h(R). \tag{23}$$

For row-Sinkhorn, the error with respect to the model's path p_{t_K} is at most the RHS of (23) plus the landing error of Proposition 1. The RHS of (23) carries a factor of the maximal step size, whereas the landing error does not depend on the time grid. Hence, at low NFE, reducing travel with row-Sinkhorn can be worth the price of a small landing error.

Table 1: Text-to-image generation with URSA at K sampling steps. We report PSNR (↑), GenEval (↑), and HPSv2 (↑). Best results per sampling budget are shown in bold. We use $\varepsilon = 0.1$ for TO.

	K=4			K=8			K=16			K=32	
Sampler	PSNR	GenEval	HPSv2	PSNR	GenEval	HPSv2	PSNR	GenEval	HPSv2	GenEval	HPSv2
AVE	10.21	0.46	0.198	10.46	0.64	0.267	11.09	0.74	0.288	0.786	**0.298**
TR-CIE	10.63	0.54	0.248	10.64	0.68	0.277	11.29	0.76	0.291	0.780	0.293
Corr-T	10.43	0.59	0.232	10.57	0.67	0.275	11.37	0.75	0.291	0.785	0.293
Corr-L	10.73	0.60	0.250	11.19	0.71	0.280	11.45	0.77	0.293	0.789	0.294
TO (ours)	**10.89**	**0.63**	**0.253**	**11.24**	**0.75**	**0.284**	**11.53**	**0.79**	**0.295**	**0.792**	**0.298**

Table 2: Generation quality with other pretrained models and modalities. Higher is better for all metrics.

Model / Task	Metric	Sampler	K=4	K=8	K=16	K=32
FUDOKI (T2I)	GenEval	AVE	0.46	0.66	0.74	0.76
		TO (ours)	**0.62**	**0.73**	**0.76**	**0.76**
	PSNR	AVE	14.25	14.61	14.53	-
		TO (ours)	**14.67**	**14.81**	**14.84**	-
URSA (T2V)	VBench	AVE	68.88	78.48	82.02	83.01
		TO (ours)	**72.57**	**80.81**	**82.96**	82.90

Table 3: Average token flips per generated token on URSA T2I.

K	Sampler	FLIPS/token ↓
4	AVE	2.25
	TO (ours)	**1.90**
8	AVE	2.69
	TO (ours)	**2.35**
16	AVE	3.29
	TO (ours)	**2.69**

6 Experiments

Setup. We use two pretrained large-scale metric-path discrete flow models, URSA and FUDOKI, and mainly evaluate them on text-to-image and text-to-video generation. URSA (Deng et al., 2026) is a 1.7B-parameter model for text-to-image and text-to-video generation, using an IBQ tokenizer with $S = 131{,}072$ codes and $N = 4096$ tokens per image, and a 64K-entry FSQ codebook. FUDOKI (Wang et al., 2026) is a 1.5B-parameter unified model that generates 384×384 images. For text-to-image generation, we evaluate semantic and compositional quality on GenEval (Ghosh et al., 2023) and additionally report HPSv2 and PSNR with AVE sampler 32 step results. For text-to-video generation, we evaluate generation quality on VBench (Huang et al., 2024). We mainly compare with the model's default sampler, *always-valid* Euler (AVE) with the Kinetic Optimal rate (Shaul et al., 2025). We further compare with the recently proposed sampler, TR-CIE (Fu & Fan, 2026), and the corrected samplers of Wan et al. (2026), Corr-T and Corr-L. Corr-T corresponds to the temporal integration component in our method, while Corr-L uses additional NFEs to better approximate the jump process. More detailed settings and hyperparameters are provided in the Appendix.

Text-to-image generation. As shown in Table 1, our sampler consistently achieves higher generation quality than the default URSA sampler across all sampling budgets, with particularly large improvements in the low-step regime. This supports our motivation that using a rate that is easier to approximate within each sampling step leads to more accurate sampling when the number of NFEs is limited. Moreover, our method consistently outperforms Corr-T, which corresponds to applying only the temporal integration component of our method. Since the two share the same temporal integration, this comparison isolates the effect of the rate. Section 6.1 further separates this effect into minimal jump intensity and locality.

Transfer to other models. We further evaluate our sampler beyond URSA image generation to examine its transferability across models and modalities. As shown in Table 2, our method consistently improves generation quality on both FUDOKI image and URSA video generation over the AVE baselines. These results suggest that the benefit of our sampling strategy is not specific to a particular architecture or modality, but transfers to general discrete flow models and generation tasks.

6.1 Further Analysis

Latency analysis. Table 4 compares the wall-clock latency of different samplers. Our sampler introduces only a small overhead over the default AVE sampler, while remaining more efficient than methods that require additional state caching or network evaluations. In particular, the temporal correction is handled by a simple analytic integration without extra NFE, and the row-Sinkhorn

Table 4: Wall-clock latency (s) of URSA text-to-image generation on RTX-Pro 6000. Per-step latency is measured at K=32.

Sampler	Sampling steps K				Per step
	4	8	16	32	
AVE	0.87	1.46	2.66	5.09	0.159
TR-CIE	0.90	1.55	2.87	5.49	0.172
Corr-L	1.36	2.60	5.07	10.00	0.313
TO (ours)	0.94	1.56	2.80	5.30	0.165

Table 5: Row-Sinkhorn (Ours) versus k full Sinkhorn iterations on URSA text-to-image generation with K=8.

Sinkhorn Iter.	PSNR	HPSv2	Latency (s)
AVE	10.46	0.267	1.46
ours	11.24	0.284	**1.56**
1	11.25	0.284	17.5
2	11.17	0.288	33.5
4	11.06	0.281	65.0

reweighting only requires an additional lightweight vector computation. As a result, TO remains close to the latency of the default sampler, while others are not.

Jump-rate analysis. Table 3 reports the average number of token flips per generated token. TO substantially reduces FLIPS/token compared with AVE across all NFE budgets, consistent with our theoretical result that the minimal-travel rate also attains the minimum jump intensity among rates generating the same conditional path. This provides direct empirical evidence that our sampler follows a process with fewer unnecessary jumps, thereby making the finite-step approximation easier.

Row-Sinkhorn versus full Sinkhorn. Table 5 compares row-Sinkhorn with $k \in \{1, 2, 4\}$ full Sinkhorn iterations on the entropic OT problem (15) from the same initialization. As shown, all variants improve over AVE, but matching the target marginal more closely has little or no effect, while every full iteration requires a dense $S \times S$ kernel action and latency grows from 1.56 s to 65.0 s (42×). Relaxing the target marginal lets each row favor nearby destinations, whereas enforcing it forces some mass to travel farther, and at K=8 the shorter travel outweighs the resulting landing error. Row-Sinkhorn is therefore not only a cheap approximation of the minimal-travel rate but also a favorable one at low NFE.

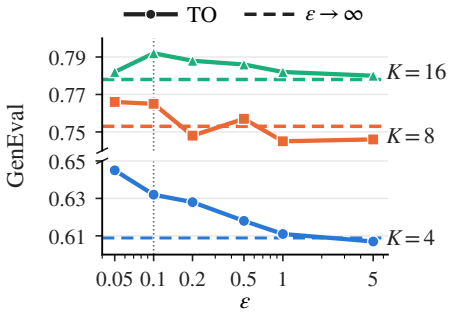

Figure 2: GenEval of TO versus ε on URSA text-to-image generation. Dashed lines show the $\varepsilon \to \infty$ limit without locality, and the dotted line marks the default $\varepsilon = 0.1$.

Sensitivity to ε. Figure 2 shows the performance of TO as ε varies. Since $\widehat{\Lambda}_t = \Lambda_t^\star$ for every ε by (19), this sweep keeps the jump intensity of every state fixed and varies only where tokens land, and thereby isolates the two components of TO. For all K, TO performs best at a small ε of around 0.05–0.1, and its performance generally degrades as ε grows and the kernel loses locality. The $\varepsilon \to \infty$ limit, where row-Sinkhorn reduces to the exact product coupling $\mu_t \otimes \nu_t / m_t$ that retains the minimal jump intensity but ignores locality, still outperforms Corr-T, which uses the same temporal integration. This shows that minimizing the jump intensity alone already improves sampling, and that the locality kernel adds a further gain at the same jump intensity.

7 Conclusion

We have presented the travel-optimal (TO) sampler, a training-free sampler for fast generation with pretrained metric-path discrete flow models. We showed that the error of a finite-step sampler is tied to how much probability moves within a single step, and proposed to control this movement through the total travel of probability mass, selecting among all rates that preserve the prescribed marginals the one with minimal total travel. We further showed that this reduces to an optimal transport problem, and proposed row-Sinkhorn to solve its entropic relaxation efficiently at every sampling step for large vocabularies. Our analysis and experiments show that TO consistently improves generation quality across large-scale metric-path models for both image and video synthesis, with the largest gains in the low-NFE regime where the movement within each step matters most.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing of drafts written by the authors, and an AI coding agent assisted with implementing the experimental code and running the experiments. Generative AI tools were also used to help draft the mathematical proofs, all of which were manually checked by the authors.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure that our results are reproducible. The travel-optimal sampler is fully specified in Section 4, with all theoretical claims stated in Sections 4 and 5 under explicit assumptions and complete proofs are given in Appendix C, together with the full pseudocode of the sampler (Algorithm 1). Our method is training-free and relies only on publicly released pretrained models; the exact checkpoints, schedulers, guidance settings, and the regularization ε used for each model are listed in Appendix B, and the evaluation protocols for GenEval, HPSv2, PSNR, and VBench, including prompt sets, and hardware, are described in Appendix B.1. We will release the source code upon publication.

REFERENCES

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

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

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Haoge Deng, Ting Pan, Fan Zhang, Yang Liu, Zhuoyan Luo, Yufeng Cui, Wenxuan Wang, Chunhua Shen, Shiguang Shan, Zhaoxiang Zhang, et al. Uniform discrete diffusion with metric path for video generation. In *International Conference on Learning Representations*, volume 2026, pp. 135312–135327, 2026.

Feiyang Fu and Hehe Fan. A time-reparameterized cumulative intensity extrapolation sampler for discrete flow matching. *arXiv preprint arXiv:2606.24140*, 2026.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

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

Daniel T Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of chemical physics*, 115(4):1716–1733, 2001.

Etrit Haxholli, Yeti Z Gurbuz, Ogul Can, and Eli Waxman. Minibatch optimal transport and perplexity bound estimation in discrete flow matching. *arXiv preprint arXiv:2411.00759*, 2024.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Yuchen Liang, Yingbin Liang, Lifeng Lai, and Ness Shroff. Discrete diffusion models: Novel analysis and new sampler guarantees. *Advances in Neural Information Processing Systems*, 38: 165511–165548, 2026.

Dongyang Liu, Shitian Zhao, Le Zhuo, Weifeng Lin, Yi Xin, Xinyue Li, Qi Qin, Yu Qiao, Hongsheng Li, and Peng Gao. Lumina-mgpt: Illuminate flexible photorealistic text-to-image generation with multimodal generative pretraining. *arXiv preprint arXiv:2408.02657*, 2024.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.

Svetlozar Todorov Rachev et al. Duality theorems for kantorovich-rubinstein and wasserstein functionals. 1990.

Yinuo Ren, Haoxuan Chen, Yuchen Zhu, Wei Guo, Yongxin Chen, Grant Rotskoff, Molei Tao, and Lexing Ying. Fast solvers for discrete diffusion models: Theory and applications of high-order algorithms. *Advances in Neural Information Processing Systems*, 38:167228–167282, 2026.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky TQ Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. In *International Conference on Learning Representations*, volume 2025, pp. 16397–16429, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Alexander Swerdlow, Mihir Prabhudesai, Siddharth Gandhi, Deepak Pathak, and Katerina Fragkiadaki. Unified multimodal discrete diffusion. *arXiv preprint arXiv:2503.20853*, 2025.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Zhengyan Wan, Yidong Ouyang, Liyan Xie, Hongyuan Zha, Fang Fang, and Guang Cheng. Corrected samplers for discrete flow models. *arXiv preprint arXiv:2601.22519*, 2026.

Jin Wang, Yao Lai, Aoxue Li, Shifeng Zhang, Jiacheng Sun, Ning Kang, Chengyue Wu, Zhenguo Li, and Ping Luo. Fudoki: Discrete flow-based unified understanding and generation via kinetic-optimal velocities. *Advances in Neural Information Processing Systems*, 38:81402–81440, 2026.

A Related Work

Discrete flows and diffusions. Continuous-time discrete diffusion was formalized as a CTMC by Campbell et al. (2022), who introduced tau-leaping and predictor-corrector sampling and derived the first error bound for the sampled distribution. Discrete flow matching (Campbell et al., 2024; Gat et al., 2024) rephrased the framework in terms of probability paths and generating velocities expressed through a learned posterior, and Shaul et al. (2025) extended it to arbitrary paths, including metric-induced ones, which Wang et al. (2026) later adopted for unified visual generation. Similar to our work, Campbell et al. (2024) observed that many rate matrices generate the same probability path, but exploited this freedom in the opposite direction, adding a detailed-balance term that leaves the net flux unchanged while increasing stochasticity to improve sample quality when many sampling steps are available. However, they did not analyze how this additional movement affects a finite-step sampler at low NFE. Shaul et al. (2025) instead select the velocity that minimizes a kinetic energy quadratic in the rate, but this objective is motivated mainly by analogy with the kinetic energy of continuous flows rather than by its effect on sampling, and its weighting is chosen largely for computational tractability. In contrast, we minimize the total travel cost, which we show directly bounds the error of a finite-step sampler. Moreover, whereas the destinations of the KO rate depend on the current token only through its distance to the target, ours depend on the distance from the current token itself, which favors destinations near the current token in codebook space and keeps transport local.

Samplers for discrete models. Fast sampling for discrete diffusion and flow models has been approached through higher-order integrators (Ren et al., 2026), uniformization-based exact simulation, adaptive re-evaluation (Wan et al., 2026), and time reparameterization with cumulative-intensity extrapolation (Fu & Fan, 2026). All of these keep the rate fixed and change the integrator. Closest to ours, Wan et al. (2026) address the temporal error with a time-corrected sampler and state staleness with a location-corrected sampler, but since they take the rate as given, the latter re-evaluates the network after the first jump within a step at the cost of additional NFE. Theoretical analyses of tau-leaping and Euler-type samplers (Campbell et al., 2022; Liang et al., 2026; Wan et al., 2026) bound the error in terms of the maximal jump rate and the dimension, which is consistent with our observation that reducing the jump rate at fixed marginals reduces the error.

Optimal transport in generative modeling. Optimal transport has been widely used across generative modeling to design shorter and more efficient generative trajectories. Pooladian et al. (2023) and Tong et al. (2023) use minibatch OT couplings between noise and data to straighten continuous flows, and Haxholli et al. (2024) extend minibatch couplings to discrete flow matching to reduce the number of transitions. While these methods aim to shorten the generative trajectory, they still apply OT only to the data coupling $\pi(x_0, x_1)$ at training time, which requires retraining the model and leaves the rate generating a given path unchanged. In contrast, we apply OT to the rate when constructing the sampler, fixing the coupling and the path p_t and solving transport at each time to route probability mass from states losing probability to states gaining it, so that it applies to any pretrained model without retraining.

B Experimental details

We use three publicly released metric-path discrete flow models without modification, and follows standard settings from them. URSA-1.7B-IBQ1024 generates 1024^2 images as $N = 4096$ tokens over an IBQ codebook with $S = 131{,}072$ entries. URSA-1.7B-FSQ320 generates 49-frame 512×320 videos as 13 latent frames of 2560 tokens over a 64K-entry FSQ codebook. FUDOKI (Janus-1.5B backbone) generates 384×384 images. All three models use the metric path of Eq. (5). URSA uses $\beta_t = c\,(t/(1-t))^{\alpha}$ with $c = 6$, $\alpha = 0.5$, with released timestep grid with shift 4, and FUDOKI uses $\beta_t = 3\,(t/(1-t))^{0.9}$. For the video model the row kernel is evaluated on the raw FSQ grid coordinates. We use CFG(Classifier-free guidance) with strength 7 for URSA images and videos and 5 for FUDOKI, in one batch of the conditional and unconditional sequences. Video generation uses the released negative prompt, `motion=9.0`, and 12 fps output. For our TO sampler, we use $\varepsilon = 0.1$ on URSA images, $\varepsilon = 0.5$ on URSA video (FSQ grid units), and $\epsilon = 0.05$ for FUDOKI .

B.1 Evaluation protocols

For GenEval, We use all 553 prompts and generate two images per prompt, and noise seeds are shared by every sampler and NFE. Scoring uses the official GenEval Mask2Former detector; our port to mmdet 3.3 loads the official checkpoint and reproduces the published URSA and FUDOKI scores at their default step counts. We report image-level accuracy; category-macro accuracy moves identically in every comparison. For Hps v2.1, Human Preference Score v2.1 on the first 100 GenEval prompts (URSA) or on all 553 images (FUDOKI), computed once over the same images for every cell of a table. For PSNR, we compute the PSNR between a sampler's image at NFE k and the *released sampler's* image at NFE 32 for the same prompt and seed, and average over the 100 HPS prompts. Finally, for VBench, we measure with the 28 videos using prompts sampled from MovieGenVideoBench and evaluate Vbench score with seven quality dimensions (temporal flickering, subject consistency, background consistency, motion smoothness, dynamic degree, imaging quality, aesthetic quality) with the official VBench code and aggregate them with the official quality-score normalization and weights. All generation, scoring and timing in Tables 1–3 ran on a single node with eight NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs (96 GB each), PyTorch 2.7.1 with CUDA 12.8 environment.

C Proofs

C.1 Proof of Lemma 1

By the continuity equation (2), a flux $J'_t \geq 0$ generates p_t if and only if $\dot{p}_t + \operatorname{div} J'_t = 0$. Since div is linear and $\dot{p}_t + \operatorname{div} J_t = 0$ by assumption, this holds if and only if $\operatorname{div}(J'_t - J_t) = 0$. For the second statement, R'_t has nonnegative off-diagonal entries because $J_t + H_t \geq 0$, and its rows sum to zero by the choice of the diagonal. □

C.2 Proof of Theorem 1

Write $\sigma = \dot{p}_t$, $\mu = [-\sigma]_+$, and $\nu = [\sigma]_+$, so that $\sigma = \nu - \mu$. We use the Kantorovich–Rubinstein duality (Rachev et al., 1990), which expresses the transport cost under a metric c as the largest gain of a potential φ that changes by at most $c(x, y)$ between any two states,

$$\mathrm{OT}_c(\mu, \nu) = \max_{\varphi} \sum_x \varphi(x)\sigma(x),$$

where the maximum is over all functions $\varphi : [S] \to \mathbb{R}$ that are 1-Lipschitz with respect to c, that is, $|\varphi(x) - \varphi(y)| \leq c(x, y)$ for all x, y.

Lower bound. Let $J \geq 0$ satisfy $\dot{p}_t + \operatorname{div} J = 0$, and fix any such φ. Since $J \geq 0$ and $c(x, y) \geq \varphi(y) - \varphi(x)$,

$$\mathcal{C}(J) = \sum_{x \neq y} c(x, y)J(x, y) \;\geq\; \sum_{x \neq y} \big[\varphi(y) - \varphi(x)\big] J(x, y) = -\sum_x \varphi(x)(\operatorname{div} J)(x) = \sum_x \varphi(x)\sigma(x),$$

where the first equality regroups the sum by the state at which φ is evaluated. The total gain of φ is thus fixed by the marginals, while each unit of mass moving along an edge raises it by at most the cost of that edge, so routing mass through intermediate states cannot reduce the cost. Taking the maximum over φ gives $\mathcal{C}(J) \geq \mathrm{OT}_c(\mu, \nu)$.

Achievability. Since μ and ν have disjoint supports, every $\gamma \in \Gamma(\mu, \nu)$ satisfies $\gamma(x, x) = 0$ and therefore defines a flux $J = \gamma$. Its net outflow at x is $\sum_y \gamma(x, y) - \sum_y \gamma(y, x) = \mu(x) - \nu(x) = -\sigma(x)$, so J is feasible with $\mathcal{C}(J) = \sum_{x,y} c(x, y)\gamma(x, y)$. Taking $\gamma = \gamma^\star$ attains $\mathrm{OT}_c(\mu, \nu)$, which also shows that $\gamma^\star$ is a minimizer of (10). □

C.3 Proof of Corollary 1

Since $\gamma_t^\star \in \Gamma(\mu_t, \nu_t)$, we have $\sum_y \gamma_t^\star(x, y) = \mu_t(x)$, and $\gamma_t^\star(x, x) = 0$ because the supports of μ_t and ν_t are disjoint. Therefore, for $p_t(x) > 0$,

$$\Lambda_t^\star(x) = \sum_{y \neq x} \frac{\gamma_t^\star(x, y)}{p_t(x)} = \frac{\mu_t(x)}{p_t(x)} = \frac{[-\dot{p}_t(x)]_+}{p_t(x)}.$$

For minimality, let R_t be any rate generating p_t, with flux J_t. Then

$$p_t(x)\Lambda_t(x) = \sum_{y\neq x} J_t(x,y) \;\geq\; \sum_{y\neq x} \big[J_t(x,y) - J_t(y,x)\big] = (\operatorname{div} J_t)(x) = -\dot{p}_t(x),$$

and since also $p_t(x)\Lambda_t(x) \geq 0$, we obtain $p_t(x)\Lambda_t(x) \geq [-\dot{p}_t(x)]_+ = p_t(x)\Lambda_t^\star(x)$.

For the metric path (5), differentiating $p_t(b \mid a) = \exp(-\beta_t d(b,a))/Z_t(a)$ with $Z_t(a) = \sum_{b'} \exp(-\beta_t d(b',a))$ gives

$$\frac{\dot{p}_t(b \mid a)}{p_t(b \mid a)} = -\dot{\beta}_t d(b,a) - \frac{\dot{Z}_t(a)}{Z_t(a)} = \dot{\beta}_t\big[\bar{d}_t(a) - d(b,a)\big].$$

Since $p_t(b \mid a) > 0$ for finite β_t and $\dot{\beta}_t \geq 0$, substituting into (13) yields

$$\Lambda_t^\star(b \mid a) = \left[-\frac{\dot{p}_t(b \mid a)}{p_t(b \mid a)}\right]_+ = \dot{\beta}_t\big[d(b,a) - \bar{d}_t(a)\big]_+,$$

which proves (14). □

C.4 Proof of Proposition 1

Summing (17) over b' shows that $\widehat{\gamma}_t(\cdot,\cdot \mid a)$ has row marginal $\mu_t(\cdot \mid a)$ and total mass m_t, so $\widehat{\nu}_t(\cdot \mid a)$ and $\nu_t(\cdot \mid a)$ are nonnegative with the same mass m_t, and $\|\widehat{\nu}_t(\cdot \mid a) - \nu_t(\cdot \mid a)\|_1 \leq 2m_t$. As $\varepsilon \to \infty$, $K \to \mathbf{1}\mathbf{1}^\top$, so $\widehat{\gamma}_t \to \mu_t \otimes \nu_t/m_t$, whose column marginal is ν_t.

For the path, we view rates as matrices acting on row vectors. Since $\widehat{\gamma}_t$ has no diagonal mass, the conditional rate $\widehat{R}_t(\cdot \mid \cdot, a)$ has flux $\widehat{\gamma}_t(\cdot,\cdot \mid a)$ and total outflow $\mu_t(b \mid a)$ at every token b. Using $\dot{p}_t(\cdot \mid a) = \nu_t(\cdot \mid a) - \mu_t(\cdot \mid a)$, we obtain

$$p_t(\cdot \mid a)\,\widehat{R}_t(\cdot \mid \cdot, a) = \widehat{\nu}_t(\cdot \mid a) - \mu_t(\cdot \mid a) = \dot{p}_t(\cdot \mid a) + \big[\widehat{\nu}_t(\cdot \mid a) - \nu_t(\cdot \mid a)\big].$$

On the sequence space, let $\widehat{R}_t$ denote the factorized rate built from these conditional rates through (4) with the exact posterior. The identity $p_t(x)\,p_{1|t}(x_1^i \mid x) = \sum_{x_1^{\bar{i}}} p_1(x_1) \prod_j p_t(x^j \mid x_1^j)$, which underlies Gat et al. (2024, Thm. 2), gives $p_t\widehat{R}_t = \dot{p}_t + \eta_t$ with

$$\eta_t(y) = \sum_{x_1} p_1(x_1) \sum_{i=1}^{N} \Big[\prod_{j\neq i} p_t(y^j \mid x_1^j)\Big]\big[\widehat{\nu}_t(y^i \mid x_1^i) - \nu_t(y^i \mid x_1^i)\big], \qquad \|\eta_t\|_1 \leq \Delta_t.$$

Let $e_t = \widetilde{p}_t - p_t$. Since $\dot{\widetilde{p}}_t = \widetilde{p}_t\widehat{R}_t$, we have $\dot{e}_t = e_t\widehat{R}_t + \eta_t$ with $e_0 = 0$, and hence $e_t = \int_0^t \eta_s \widetilde{P}_{s,t}\, ds$, where $\widetilde{P}_{s,t}$ is the transition kernel of $\widehat{R}$. A Markov kernel does not increase the ℓ_1 norm of a signed measure, so $\|e_t\|_1 \leq \int_0^t \|\eta_s\|_1\, ds \leq \int_0^t \Delta_s\, ds$. □

C.5 Proof of Theorem 2

Throughout, we write $t = t_k$, $h = h_k$, $H = H_k$, $\ell = \ell_k$, and fix the initial state x; $c = \sqrt{d}$ denotes the travel metric, extended additively to sequences as in Section 5. For a state y, let $y^{i\to v}$ denote y with its i-th coordinate replaced by v, and let $\mathcal{L}_s$ be the generator of the true CTMC at time s,

$$(\mathcal{L}_s f)(y) = \sum_{i=1}^{N} \sum_{v\neq y^i} R_s^i(v \mid y)\big[f(y^{i\to v}) - f(y)\big].$$

We use $\|\mu\|_1 = \sup_{\|f\|_\infty \leq 1} |\mu f|$ for signed measures.

Step 1: AVE as an auxiliary CTMC. Define the frozen one-jump generator

$$(B^x f)(y) = \sum_{i=1}^{N} \mathbf{1}\{y^i = x^i\} \sum_{v\neq x^i} R_t^i(v \mid x)\big[f(y^{i\to v}) - f(y)\big]. \tag{24}$$

Under B^x, each coordinate leaves x^i at the constant rate $\Lambda_t^i(x)$, lands on v with probability $R_t^i(v \mid x)/\Lambda_t^i(x)$, and never moves again, independently across coordinates. Hence the process Z with $Z_0 = x$ and generator B^x has, at time h, exactly the law of the AVE update (7), that is, $e^{hB^x}(x,\cdot) = P_k^{\mathrm{AVE}}(x,\cdot)$.

Step 2: Duhamel formula. Differentiating $s \mapsto e^{sB^x} P_{t+s,t+h}$ and using the backward equation $\partial_s P_{t+s,t+h} = -\mathcal{L}_{t+s} P_{t+s,t+h}$ gives

$$P_{t,t+h} - e^{hB^x} = \int_0^h e^{sB^x} \big(\mathcal{L}_{t+s} - B^x\big) P_{t+s,t+h}\, ds.$$

For $\|f\|_\infty \le 1$, the function $g_s = P_{t+s,t+h} f$ satisfies $\|g_s\|_\infty \le 1$, since $P_{t+s,t+h}$ is a Markov kernel. Therefore

$$\big\|P_k(x,\cdot) - P_k^{\mathrm{AVE}}(x,\cdot)\big\|_1 \le \int_0^h \mathbb{E}_x\Big[\sup_{\|g\|_\infty \le 1} \big|(\mathcal{L}_{t+s} - B^x) g(Z_s)\big|\Big] ds, \tag{25}$$

where the expectation is taken over the auxiliary process Z rather than over the true chain.

Step 3: generator mismatch. Fix a state y and $\|g\|_\infty \le 1$, so that $|g(y^{i\to v}) - g(y)| \le 2$. For a coordinate i with $y^i = x^i$, both generators move coordinate i. Since $y^i = x^i$, the conditional rates in (4) coincide at y and x, and only the posterior weights differ. Using the bound H on the conditional jump rates and applying the posterior Lipschitz condition one coordinate at a time, we obtain

$$\begin{aligned}
&\sum_{v \neq x^i} \big|R_{t+s}^i(v \mid y) - R_t^i(v \mid x)\big| \\
&\quad \le \sum_{v \neq x^i} \big|R_{t+s}^i(v \mid x) - R_t^i(v \mid x)\big| + H \big\|p_{1|t+s}^i(\cdot \mid y) - p_{1|t+s}^i(\cdot \mid x)\big\|_1 \\
&\quad \le \sum_{v \neq x^i} \big|R_{t+s}^i(v \mid x) - R_t^i(v \mid x)\big| + H\ell \sum_{j \neq i} c(x^j, y^j).
\end{aligned}$$

For a coordinate i with $y^i \neq x^i$, the generator B^x no longer moves coordinate i, and the missing jump rate is $\Lambda_{t+s}^i(y) \le H$. Summing over i, and using the time Lipschitz condition together with $\sum_i \sum_{j \neq i} c(x^j, y^j) = (N-1)\, c(x,y)$, we obtain

$$\big|(\mathcal{L}_{t+s} - B^x) g(y)\big| \le 2\big[L_k^t\, s + H\, n_x(y) + c_k\, c(x,y)\big], \qquad n_x(y) = \sum_{i=1}^N \mathbf{1}\{y^i \neq x^i\}. \tag{26}$$

Step 4: occupation of the auxiliary process. The coordinates of Z_s are independent. Coordinate i has left x^i with probability $1 - e^{-s\Lambda_t^i(x)} \le s\Lambda_t^i(x)$, and, given that it has left, it sits at v with probability $R_t^i(v \mid x)/\Lambda_t^i(x)$. Hence

$$\mathbb{E}_x\, n_x(Z_s) \le s\, \Lambda_t(x), \qquad \mathbb{E}_x\, c(x, Z_s) = \sum_{i=1}^N \frac{1 - e^{-s\Lambda_t^i(x)}}{\Lambda_t^i(x)} \sum_{v \neq x^i} R_t^i(v \mid x)\, c(x^i, v) \le s\, C_t(x),$$

where coordinates with $\Lambda_t^i(x) = 0$ contribute zero. Substituting (26) and these bounds into (25) gives

$$\big\|P_k(x,\cdot) - P_k^{\mathrm{AVE}}(x,\cdot)\big\|_1 \le 2\int_0^h s\big[L_k^t + H\Lambda_t(x) + c_k\, C_t(x)\big] ds = h^2 L_k^t + h^2\big(H\Lambda_t(x) + c_k\, C_t(x)\big),$$

which is (22).

Constants for the metric path. For the metric path (5) with $\dot\beta_t \ge 0$, the minimal jump ratesatisfies $\Lambda_t^\star(b \mid a) \le \dot\beta_t d_{\max}$, and the same bound holds for row-Sinkhorn and for the KO rate of Shaul et al. (2025), whose jump rate is bounded by $\dot\beta_t\, d(b,a) \le \dot\beta_t d_{\max}$ by direct computation. Hence $H_k = \sup_{t \in I_k} \dot\beta_t d_{\max}$ for all these rates. For the posterior, changing x^j to y^j multiplies the posterior weight of each x_1 by $e^{g(x_1)}$ with $g(x_1) = -\beta_t\big[d(y^j, x_1^j) - d(x^j, x_1^j)\big]$, since the normalizer of $p_t(\cdot \mid x_1^j)$ does not depend on the token. Next, we bound the range of g through the embeddings. Writing e_b for

Algorithm 1 TO sampler (metric-path discrete flow sampling with row-Sinkhorn)

Require: grid $t_0 < \cdots < t_K$; denoiser $p^{\theta}_{1|t}$; metric d; regularization ε

$X \sim p_0$
for $k = 0, \dots, K-1$ **do**
  $\{\widehat{p}^i\}_{i=1}^N \leftarrow p^{\theta}_{1|t_k}(\cdot \mid X)$; $\Delta\beta_k \leftarrow \beta_{t_{k+1}} - \beta_{t_k}$ ▷ one NFE
  for $i = 1, \dots, N$ **in parallel do**
    $b \leftarrow X^i$
    $\widehat{a} \leftarrow \textsc{Select}(\widehat{p}^i)$ ▷ same clean-token estimate as the base sampler
    $d_{\mathrm{tgt}}(y) \leftarrow d(y, \widehat{a})$ for all $y \in [S]$ ▷ already used by the metric-path sampler
    $p_{\widehat{a}}(y) \propto \exp\big(-\beta_{t_k} d_{\mathrm{tgt}}(y)\big)$
    $\overline{d}_{\widehat{a}} \leftarrow \sum_y p_{\widehat{a}}(y) d_{\mathrm{tgt}}(y)$
    $g \leftarrow \big[d(b, \widehat{a}) - \overline{d}_{\widehat{a}}\big]_+$ ▷ $\Lambda^\star/\dot{\beta}_{t_k}$, (14)
    $d_{\mathrm{cur}}(y) \leftarrow d(b, y)$ for all $y \in [S]$ ▷ one additional distance row
    $Q(y) \propto p_{\widehat{a}}(y) \big[\overline{d}_{\widehat{a}} - d_{\mathrm{tgt}}(y)\big]_+ \exp\left(-\frac{\sqrt{d_{\mathrm{cur}}(y)}}{\varepsilon}\right)$ ▷ row-Sinkhorn, (18)
    $\rho \leftarrow 1 - \exp(-\Delta\beta_k\, g)$ ▷ temporal integration
    $X^i \sim (1-\rho)\delta_b + \rho Q$ ▷ AVE
return X

the codebook embedding of token b, so that $d(b, b') = \|e_b - e_{b'}\|_2^2$ and $c(b, b') = \|e_b - e_{b'}\|_2$, and expanding the squares, for any two clean sequences x_1, x_1' with $a = x_1^j$ and $a' = x_1'^j$,

$$g(x_1) - g(x_1') = 2\beta_t \left\langle e_{y^j} - e_{x^j},\, e_a - e_{a'} \right\rangle,$$

so by the Cauchy–Schwarz inequality $|g(x_1) - g(x_1')| \le 2\beta_t\, c(x^j, y^j)\, c(a, a') \le 2\beta_t \sqrt{d_{\max}}\, c(x^j, y^j)$. Hence g takes values in an interval of length at most $2\beta_t \sqrt{d_{\max}}\, c(x^j, y^j)$. Interpolating the weights $w_\theta \propto w_0 e^{\theta g}$ for $\theta \in [0, 1]$ gives $\sum_{x_1} |\partial_\theta w_\theta(x_1)| = \mathbb{E}_{w_\theta}|g - \mathbb{E}_{w_\theta} g| \le \beta_t \sqrt{d_{\max}}\, c(x^j, y^j)$, since the mean absolute deviation of a variable confined to an interval is at most half its length, and marginalizing onto coordinate i does not increase the ℓ_1 distance. Hence $\ell_k = \sqrt{d_{\max}}\, \sup_{t \in I_k} \beta_t$. □

C.6 Proof of Corollary 2

Write $p_k = p_{t_k}$ and let $\widehat{p}_k$ be the law of the AVE state at t_k, with $\widehat{p}_0 = p_0$. Since R_t generates p_t, we have $p_{k+1} = p_k P_k$, while $\widehat{p}_{k+1} = \widehat{p}_k P_k^{\mathrm{AVE}}$. Hence

$$\widehat{p}_{k+1} - p_{k+1} = (\widehat{p}_k - p_k) P_k^{\mathrm{AVE}} + p_k \big(P_k^{\mathrm{AVE}} - P_k\big).$$

A Markov kernel does not increase the ℓ_1 norm of a signed measure, so

$$\|\widehat{p}_{k+1} - p_{k+1}\|_1 \le \|\widehat{p}_k - p_k\|_1 + \sum_x p_k(x) \left\| P_k^{\mathrm{AVE}}(x, \cdot) - P_k(x, \cdot) \right\|_1.$$

We apply Theorem 2 to each x, sum over k, and use

$$h_k^2 H_k\, \mathbb{E}_{p_k}[\Lambda_{t_k}(X_{t_k})] \le \max_j (h_j H_j)\, h_k\, \mathbb{E}_{p_k}[\Lambda_{t_k}(X_{t_k})],$$
$$h_k^2 c_k\, \mathbb{E}_{p_k}[C_{t_k}(X_{t_k})] \le \max_j (h_j c_j)\, h_k\, \mathbb{E}_{p_k}[C_{t_k}(X_{t_k})],$$

which yields (23). □

D Visualization Results

We qualitatively compare TO with the default sampler of each model (AVE with the KO rate). All prompts are taken from GenEval, and in each pair of rows the top row shows the default sampler and the bottom row shows TO.

D.1 Qualitative Comparison

Figures 3 and 4 show samples across NFE budgets on FUDOKI and URSA text-to-image generation, respectively. The difference is most visible at low NFE. On FUDOKI, the default sampler omits the horse at NFE 4–8, merges the two frisbees into a single object at NFE 4–8, and renders the stop sign as an irregular red shape at NFE 4, whereas TO generates every object with the correct shape and count from NFE 4. Similarly on URSA at NFE 3–4, the default sampler often fails to form the objects at all, producing color blobs for "a green bus and a purple microwave" and a purple smear for "a purple hair drier", while TO already produces recognizable objects with the requested attributes. As NFE grows, the samples of the two samplers become increasingly similar, consistent with the narrowing gap in Table 1. We also depicts text-to-video generation results on FIg. 5.

D.2 Denoising Trajectory Comparison

Figure 6 visualizes the intermediate states of URSA text-to-image sampling with NFE 32, together with the tokens changed at each displayed step. As shown, TO changes fewer tokens than the default sampler at every displayed step for both prompts, while reaching a final image of comparable quality. This agrees with the lower FLIPS/token in Table 3 and with Corollary 1, since the minimal-travel rate attains the minimal jump intensity and thus requires fewer tokens to jump within each step.

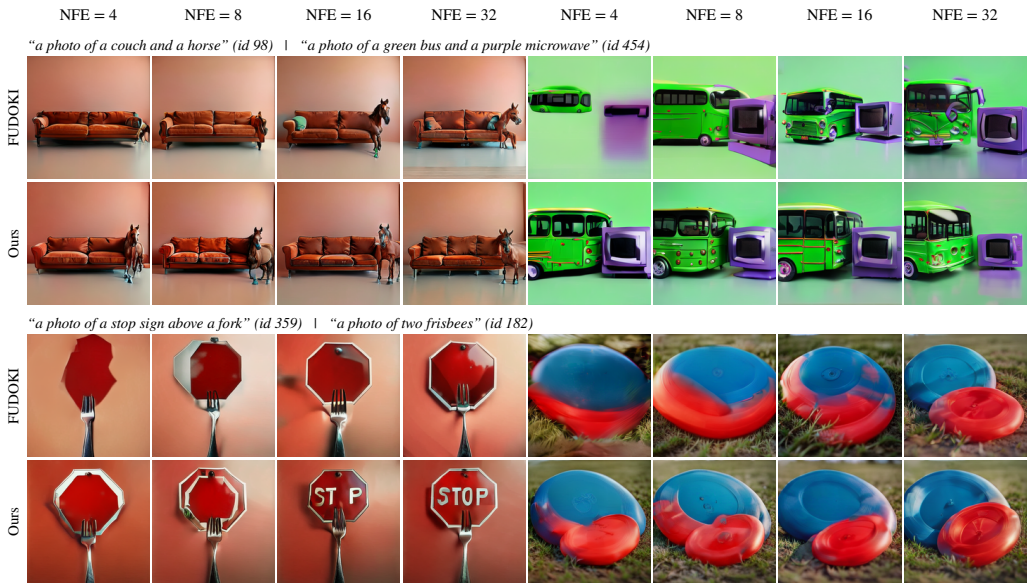

Figure 3: Qualitative comparison on FUDOKI text-to-image generation across NFE budgets. Each block shows two GenEval prompts side by side, with the default sampler in the top row and TO in the bottom row.

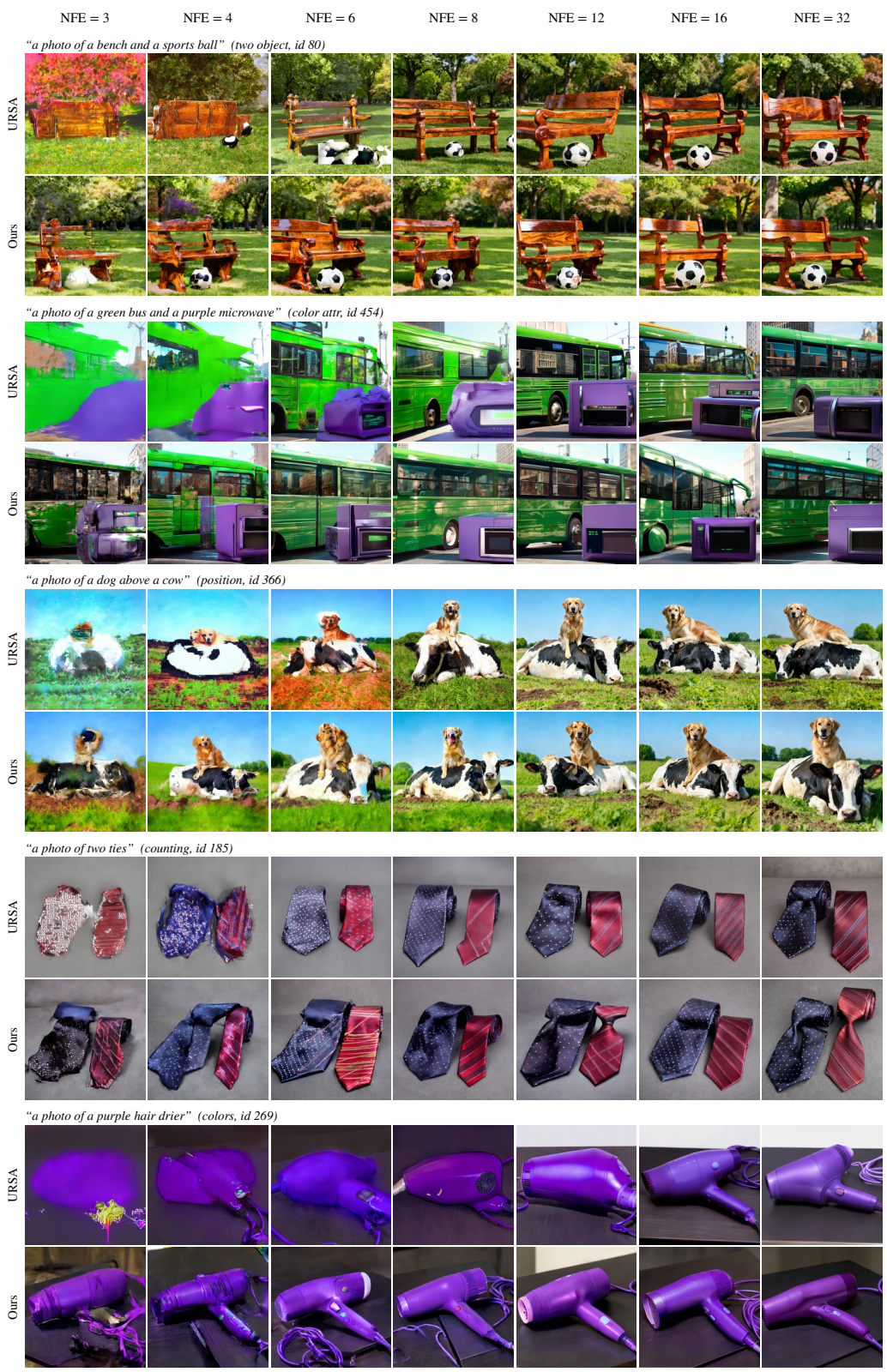

Figure 4: Qualitative comparison on URSA text-to-image generation across NFE budgets. For each GenEval prompt (task category and prompt id in parentheses), the top row shows the default sampler and the bottom row shows TO.

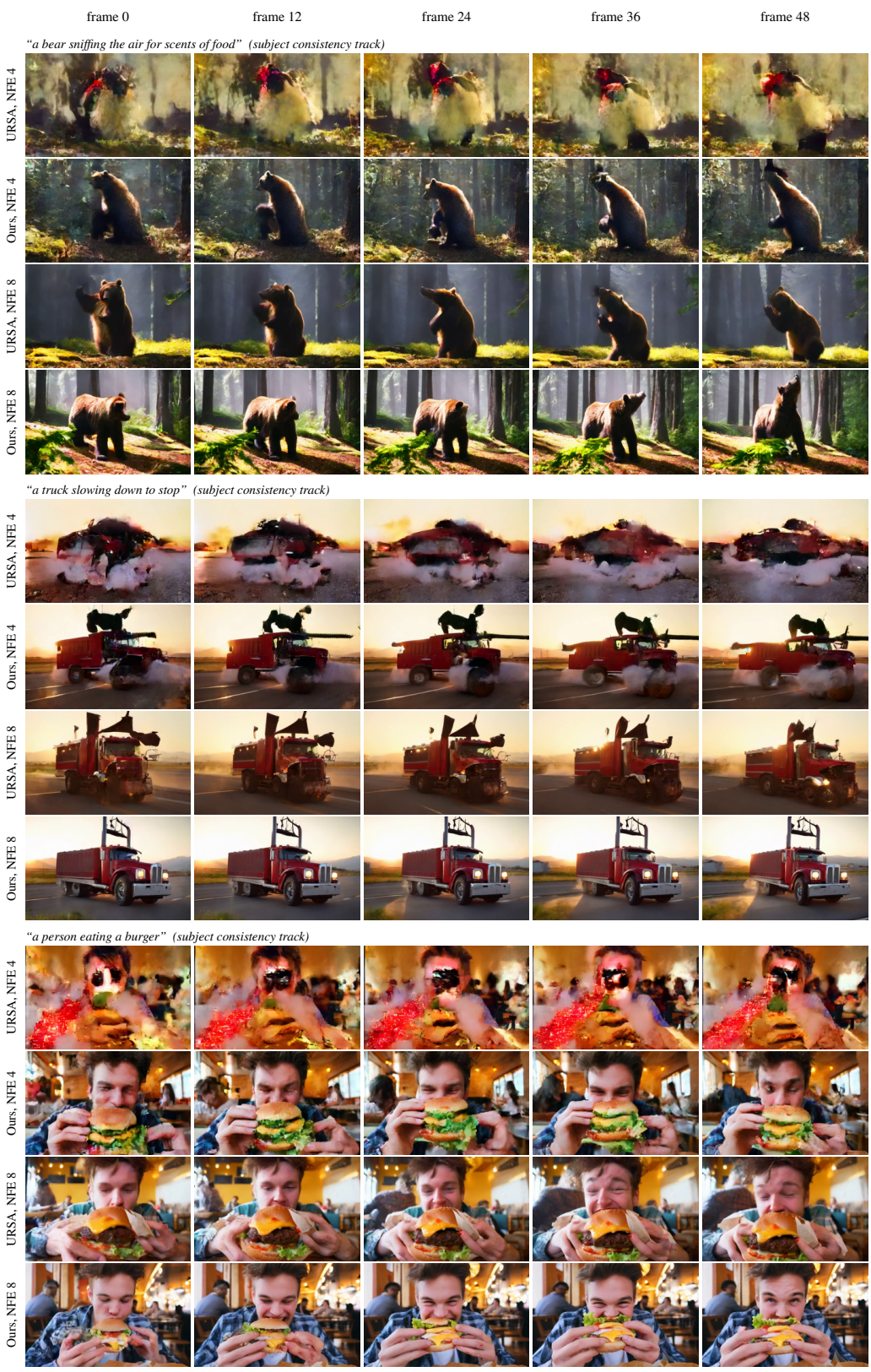

Figure 5: Qualitative comparison on URSA text-to-video generation across NFE budgets..

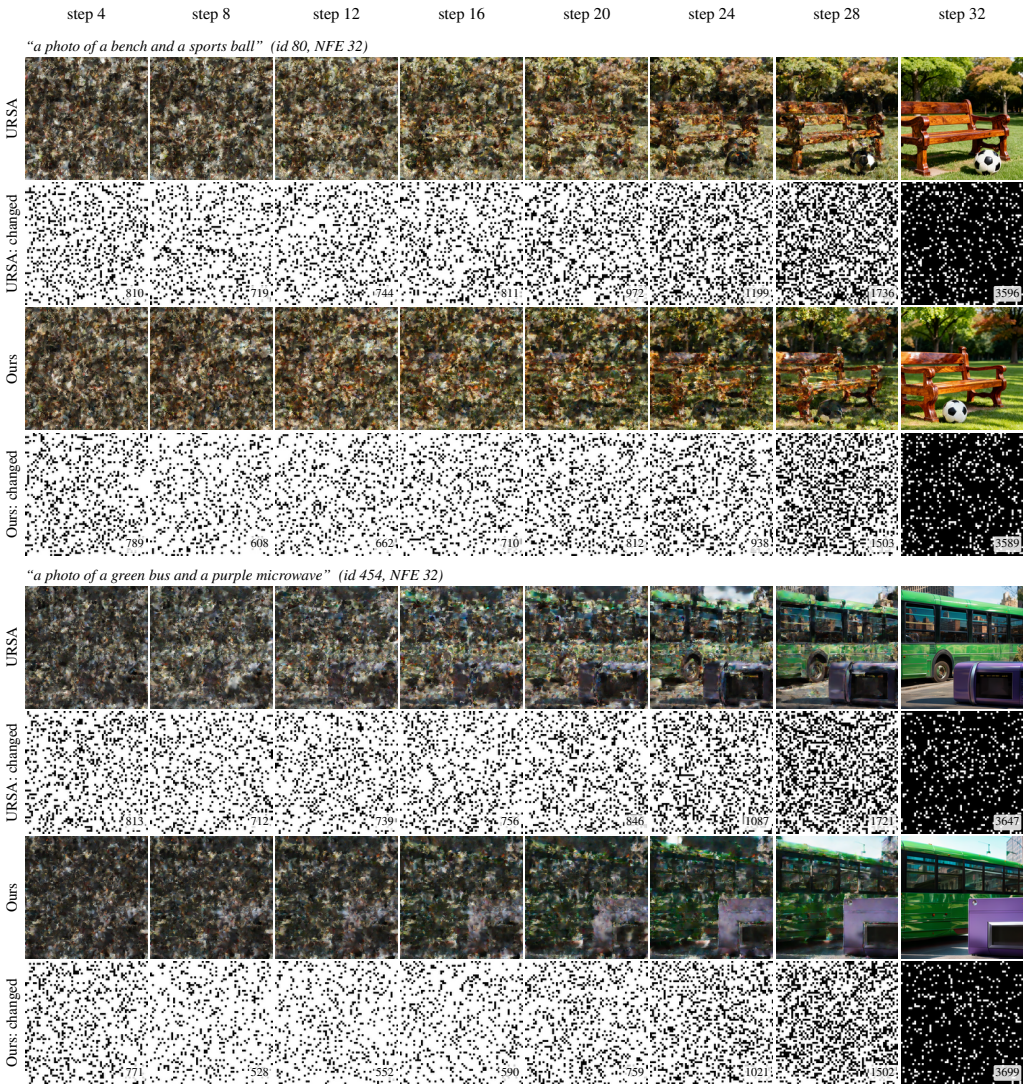

Figure 6: Denoising trajectories on URSA text-to-image generation with NFE 32. For each prompt, the intermediate states of the default sampler and TO are each followed by a map of the tokens changed at the displayed step (black), with their number out of $N{=}4096$ in the corner.