CONVERGEFLOW: LANGUAGE FLOW WITH PROVABLE CONVERGENCE TO TOKEN EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in continuous diffusion and flow-based language models (LMs) have achieved performance competitive with discrete LMs. However, existing continuous frameworks still rely on decoders supervised with cross entropy (CE) because the flow trajectories are not guaranteed to terminate at valid token embeddings. Motivated by this limitation, we introduce **ConvergeFlow**, an embedding-space flow-based LM, which constrains the data predictor to the convex hull of token embeddings and trains it solely with the mean squared error objective induced by flow matching. Under suitable regularity conditions, we prove that the resulting flow converges to valid token embeddings despite errors in the data predictor, enabling direct token prediction without a CE-supervised decoder. We further develop three sampling mechanisms for controlling the trade-off between the generative perplexity and entropy. Experiments on OpenWebText demonstrate that ConvergeFlow achieves performance competitive with existing continuous and discrete diffusion LMs. These findings demonstrate the potential of the flow-based paradigm for language modeling. Our code is available at `https://anonymous.4open.science/r/ConvergeFlow-3C12`.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) and flow matching (FM) (Lipman et al., 2022; Liu et al., 2022) have become the backbone for generative modeling in continuous data domains, with applications spanning image synthesis (Dhariwal & Nichol, 2021; Rombach et al., 2022; Esser et al., 2024), video generation (Ho et al., 2022; Wan et al., 2025), and protein design (Trippe et al., 2022). At a high level, these models learn a transport from a simple noise distribution to the data distribution. Diffusion models construct this transformation by learning to reverse a progressive noise corruption process, while FM learns the velocity field of a prescribed probability path. Once learned, the resulting generative dynamics iteratively transform fresh noise into new samples from the data distribution.

Language modeling, a central task in modern generative modeling, has long been dominated by autoregressive (AR) language models (LMs) (Radford et al., 2019; Brown et al., 2020). Despite their remarkable success in practice, AR models have two inherent drawbacks. First, the left-to-right generation order prevents earlier tokens from being revised using later context, limiting bidirectional reasoning and controllable generation. Second, one-by-one sequential generation inherently restricts parallelism and creates a fundamental bottleneck in sampling speed.

To overcome these limitations, substantial effort has recently been devoted to diffusion and flow-based language modeling (Li et al., 2022; Sahoo et al., 2024). These models offer a fundamentally different generation principle—they iteratively refine all token positions using bidirectional context. This formulation permits parallel token updates and allows global planning, controllable generation, and iterative revision, offering the potential for faster and more flexible generation.

Existing diffusion language models (DLMs) can be broadly categorized into continuous and discrete approaches. Continuous diffusion/flow-based models (Li et al., 2022; Han et al., 2023; Lovelace et al., 2023) map discrete tokens to continuous representations and apply Gaussian diffusion in the

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.

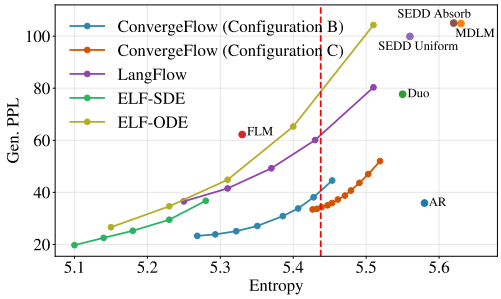


Figure 1: Gen. PPL-entropy trade-off across different models on the OWT dataset (Raffel et al., 2020). Curves show the trade-offs achieved by ConvergeFlow, LangFlow, ELF-SDE, and ELF-ODE, while individual markers indicate the reported results of the remaining baselines. Schedules B and C denote two time-adaptive guidance schedules for ConvergeFlow, defined in Section 3.2. The red dashed line marks the entropy of the OWT dataset, 5.44.

resulting continuous space.[1] Discrete DLMs (Sahoo et al., 2024; Shi et al., 2024) tailor the diffusion framework to the discrete nature of text by leveraging discrete diffusion models (Hoogeboom et al., 2021; Austin et al., 2021; Campbell et al., 2022), which define categorical corruption processes directly in token space. Recent scaling efforts have shown that discrete DLMs can achieve performance competitive with AR models (Nie et al., 2025; You et al., 2025; Ye et al., 2025; Song et al., 2025; Labs et al., 2025). Despite these substantial advances, operating in categorical state spaces makes it difficult to apply the extensive toolkit developed for continuous diffusion, including classifier-free guidance (CFG) (Ho & Salimans, 2022), self-conditioning (Chen et al., 2022b), few-step ODE solvers (Lu et al., 2022a;b), and distillation (Song & Dhariwal, 2024; Yin et al., 2024). Moreover, their reliance on discrete token states may also limit their ability to exploit the rich latent geometry underlying language.

These considerations have motivated renewed interest in continuous flow-based LMs. Notably, recent embedding-space flow-based LMs, including LangFlow (Chen et al., 2026), ELF (Hu et al., 2026), and FLM (Lee et al., 2026), have achieved performance competitive with discrete DLMs. However, these continuous models still rely on token-level cross entropy (CE) supervision during training—LangFlow and FLM apply the CE objective along the flow trajectory, whereas ELF combines the FM objective at intermediate denoising steps with the CE objective at the final decoding step. Crucially, their learned flow trajectories are not guaranteed to terminate at valid token embeddings, because errors in the learned data predictor or velocity can leave the terminal state between token embeddings. Hence, these models require a CE-trained decoding mechanism to map such off-embedding states to discrete tokens. Although effective, this reintroduced discrete supervision is inconsistent with the continuous nature of flow-based models, and may limit their full potential.

Consequently, an important question remains unresolved:

> *Can the sampling trajectories of a flow-based LM converge directly to valid token embeddings, enabling discrete token prediction without a CE-supervised decoder?*

1.1 Contributions

We provide an affirmative answer by introducing **ConvergeFlow**, an embedding-space flow-based LM that retains a fully continuous formulation while incorporating the discrete structure of text.

We parameterize the data predictor as a weighted average of vocabulary embeddings, first introduced in Plaid (Gulrajani & Hashimoto, 2023) and later adopted by LangFlow. Each coefficient is parameterized using a learnable weight and an exact Gaussian kernel induced by the corruption process. The resulting data predictor is trained using the mean squared error (MSE) loss induced by the FM objective. We emphasize that LangFlow directly learns the convex-combination coefficients as

[1] Because continuous diffusion models are equivalent to FM with linear Gaussian interpolation, we use the terms interchangeably throughout this paper; see Section 2.

token posteriors using CE supervision and subsequently maps them to a continuous data predictor. Instead, ConvergeFlow uses the factorization solely as an architectural parameterization and trains the resulting continuous data predictor directly with the FM objective.

Theoretically, under suitable regularity conditions, we prove that this parameterization ensures the learned flow converges to a valid token embedding despite errors in the learned data predictor. The same data predictor can therefore drive the intermediate continuous flow updates and produce the final discrete token prediction, thereby eliminating the need for a CE-supervised token decoder. We further validate this theoretical guarantee by showing that discrete tokens can be recovered accurately and directly from the terminal flow states.

Our contributions can be summarized as twofold:

1. Flow-based LM with provable convergence to token embeddings. We introduce **ConvergeFlow**, an embedding-space flow-based LM. We prove that the resulting flow provably converges to valid token embeddings, enabling direct token prediction without a CE-supervised decoder. To our knowledge, ConvergeFlow is the first flow-based LM with provable convergence to token embeddings.

2. Quality–diversity control and strong empirical performance. We propose three sampling mechanisms that provide explicit control over the trade-off between generation quality, measured by generative perplexity (Gen. PPL), and diversity, measured by entropy. Combined with these mechanisms, ConvergeFlow achieves performance competitive with continuous baselines, including LangFlow and ELF, as well as discrete baselines such as Duo (Sahoo et al., 2025). In particular, on the OpenWebText (OWT) dataset (Raffel et al., 2020), ConvergeFlow achieves a Gen. PPL of 34.19 while maintaining an entropy of 5.44; see Figure 1 and Table 2 for details.

2 Background

2.1 Flow matching and diffusion models

FM (Lipman et al., 2022) is a continuous-time generative modeling framework that transports a sample from a source distribution p_0 (typically the standard Gaussian) to a target distribution $p_1 = p_{\mathsf{data}}$. The framework consists of two steps. First, one specifies a probability path $(p_t)_{t\in(0,1)}$ interpolating between the source p_0 and target p_1. A common choice defines p_t as the marginal distribution of

$$x_t = \alpha_t x_\star + \sigma_t z, \quad t \in [0,1], \tag{1}$$

where $x_\star \sim p_{\mathsf{data}}$ and $z \sim \mathcal{N}(0, I)$ is independent Gaussian noise. The differentiable schedules $(\alpha_t, \sigma_t)_{t\in(0,1)}$ are chosen such that $\lim_{t\to 0} \alpha_t/\sigma_t = 0$ and $\lim_{t\to 1} \alpha_t/\sigma_t = \infty$. Under the standard conditions $(\alpha_0, \sigma_0) = (0, 1)$ and $(\alpha_1, \sigma_1) = (1, 0)$, one has $x_1 = x_\star \sim p_1$ and $x_0 = z \sim p_0$.

Second, one learns a time-dependent velocity field $v : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ whose induced flow realizes the prescribed probability path. Specifically, the velocity $v(x,t)$ generates the probability path p_t if the solution to the following ordinary differential equation (ODE) satisfies $x_t \sim p_t$ for all $t \in [0,1]$:

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = v(x_t, t), \quad t \in (0,1); \qquad x_0 \sim p_0, \tag{2}$$

Training. A natural objective for learning the velocity v_t is the *flow matching loss*

$$\ell_{\mathsf{FM}}(\theta) := \mathbb{E}_{t,x_t}\Big[\big\|v_\theta(x_t,t) - v(x_t,t)\big\|_2^2\Big], \tag{3}$$

where $t \sim \mathsf{Unif}(0,1)$ and $x_t \sim p_t$. However, this objective cannot be evaluated directly because the velocity v_t is generally unavailable. Fortunately, one can obtain a tractable objective by conditioning on the target $x_\star$. Under the probability path (1), consider the conditional distribution $p_t(\cdot \mid x_\star)$ of x_t given $x_\star$. By the path construction in (1) and the ODE in (2), the conditional velocity field is given:

$$v(x_t, t \mid x_\star) = \alpha_t' x_\star + \sigma_t' z = \frac{\sigma_t'}{\sigma_t} x_t + \Big(\alpha_t' - \frac{\sigma_t'}{\sigma_t}\alpha_t\Big) x_\star = \frac{\alpha_t'}{\alpha_t} x_t + \Big(\sigma_t' - \frac{\alpha_t'}{\alpha_t}\sigma_t\Big) z. \tag{4}$$

The key observation underlying conditional flow matching (CFM) is that under mild regularity conditions, the posterior expectation of the conditional velocity,

$$v(x,t) = \mathbb{E}_{x_\star}[v(x_t, t \mid x_\star) \mid x_t = x], \tag{5}$$

yields the marginal velocity v_t that generates the probability path p_t (Lipman et al., 2022). This leads to the *conditional flow matching loss*:

$$\ell_{\mathsf{CFM}}(\theta) \coloneqq \mathbb{E}_{t,x_\star,z}\Big[\big\|v_\theta(x_t,t) - v(x_t,t \mid x_\star)\big\|_2^2\Big], \tag{6}$$

where $t \sim \mathsf{Unif}(0,1)$, $x_\star \sim p_1$, $z \sim \mathcal{N}(0,I)$, and $x_t = \alpha_t x_\star + \sigma_t z$. The minimizer of the objective (6) is given by the conditional expectation $v_{\theta^\star}(x,t) = \mathbb{E}_{x_\star}\big[v(x_t,t \mid x_\star) \mid x_t = x\big] = v(x,t)$. Therefore, the CFM loss (6) shares the same minimizer as the FM loss (3) and thus provides a tractable objective for learning the velocity v_t. For simplicity of presentation, we will refer to the CFM loss as the FM loss in the rest of the paper.

In addition, the velocity can also be expressed via either a data predictor or a noise predictor. Define

$$\mu(x,t) \coloneqq \mathbb{E}[x_\star \mid x_t = x] \quad \text{and} \quad \varepsilon(x,t) \coloneqq \mathbb{E}[z \mid x_t = x]. \tag{7}$$

Combining (4) and (5), we obtain

$$v(x,t) = \frac{\sigma_t'}{\sigma_t}x + \Big(\alpha_t' - \frac{\sigma_t'}{\sigma_t}\alpha_t\Big)\mu(x,t) = \frac{\alpha_t'}{\alpha_t}x + \Big(\sigma_t' - \frac{\alpha_t'}{\alpha_t}\sigma_t\Big)\varepsilon(x,t). \tag{8}$$

This identity shows that learning the velocity is equivalent to predicting the clean data or noise. Therefore, FM is essentially the probability flow ODE in diffusion models (Song et al., 2020). Since we use the linear Gaussian interpolation, we use FM and diffusion models interchangeably.

Finally, for the linear schedule $\alpha_t = t$ and $\sigma_t = 1 - t$, the FM objective under the data prediction parameterization simplifies to $\mathbb{E}_{t,x_\star,z}[(1-t)^{-2}\|x_\star - \mu_\theta(x_t,t)\|_2^2]$. For general schedules, the corresponding FM objective has a schedule-dependent weighting that is not invariant across schedules at a fixed signal-to-noise ratio (SNR). To remove this dependence, we instead use the following objective: $\mathbb{E}_{t,x_\star,z}[(1+\alpha_t/\sigma_t)^2\|x_\star - \mu_\theta(x_t,t)\|_2^2]$.

Inference. After learning a velocity v_θ, approximate samples from the target distribution p_1 can be generated by drawing $x_0 \sim p_0$ from the source distribution p_0 and numerically solving the ODE $\frac{\mathrm{d}x_t}{\mathrm{d}t} = v_\theta(x_t,t)$, $t \in [0,1]$. This ODE can be reparameterized using either a data predictor μ_θ or a noise predictor ε_θ. Replacing μ with μ_θ in the data-prediction parameterization of the velocity in (8) gives $\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{x_t}{\sigma_t}\right) = \mu_\theta(x_t,t)\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\alpha_t}{\sigma_t}\right)$. This leads to the data prediction-based inference procedure:

$$\frac{x_s}{\sigma_s} - \frac{x_t}{\sigma_t} = \int_t^s \mu_\theta(x_\tau,\tau)\,\mathrm{d}\left(\frac{\alpha_\tau}{\sigma_\tau}\right) \approx \mu_\theta(x_t,t)\left(\frac{\alpha_s}{\sigma_s} - \frac{\alpha_t}{\sigma_t}\right). \tag{9}$$

Self-conditioning. Self-conditioning (Chen et al., 2022b) is a technique that adds an additional input c to the predictor. During training, the conditional flow matching loss in (6) is modified to

$$\mathbb{E}_{t,x_\star,z,c}\Big[\big\|v_\theta(x_t,t,c) - v(x_t,t \mid x_\star)\big\|_2^2\Big],$$

where the input c is constructed by

$$c = \begin{cases} \varnothing, & \text{with probability } 1-p, \\ \mathsf{stopgrad}\big(v_\theta(x_t,t,\varnothing)\big), & \text{with probability } p. \end{cases} \tag{10}$$

During training, the model makes an ordinary prediction without self-conditioning with a certain probability. Otherwise, it first produces an ordinary prediction and then uses the resulting prediction as an additional input in a second forward pass. In this way, the model learns to refine a prediction previously produced by itself. During inference, self-conditioning is applied at every step. The self-conditioning input is initialized as empty at the first step, when no previous prediction is available, and is set to the model's prediction from the preceding step thereafter.

3 ConvergeFlow

3.1 Framework and convergence theory

Let $s = (s^{(1)}, \ldots, s^{(L)})$ be a token sequence of length L drawn from the data distribution p_{data}, where each token $s^{(i)}$ belongs to a vocabulary of size V. Without loss of generality, we assume the

vocabulary is $[V] \coloneqq \{1, \dots, V\}$. We map tokens to continuous representations using an embedding matrix $E \in \mathbb{R}^{V \times d}$, where d is the embedding dimension. For each $j \in [V]$, let $e_j^\top \coloneqq E_{j,:} \in \mathbb{R}^d$ denote the j-th row of the embedding matrix E, representing the embedding of the j-th token in the vocabulary. The target $x_\star$ is the continuous representation of the token sequence, given by $x_\star = [e_{s^{(1)}}, \dots, e_{s^{(L)}}]^\top \in \mathbb{R}^{L \times d}$.

We consider FM with general interpolation schedules (α_t, σ_t): $x_t = \alpha_t x_\star + \sigma_t z$, where $z \in \mathbb{R}^{L \times d}$ is a standard Gaussian random matrix with i.i.d. entries $z_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. Using the data-prediction parameterization, we train a data predictor $\mu_\theta : \mathbb{R}^{L \times d} \times [0,1] \to \mathbb{R}^{L \times d}$ using the MSE loss by:

$$\mathbb{E}_{t, x_\star, z}\left[\Big(1 + \alpha_t/\sigma_t\Big)^2 \big\|x_\star - \mu_\theta(x_t, t)\big\|_{\mathrm{F}}^2\right]. \tag{11}$$

This objective, also used by ELF, provides purely continuous supervision and does not involve a token-level CE loss. An important caveat is that the FM objective alone does not sufficiently supervise the joint learning of the embedding matrix and the data predictor. Because the target $x_\star$ itself is defined by the embedding matrix, the FM objective admits degenerate embedding-collapse solutions. We therefore use the pre-trained embedding matrix from LangFlow and keep it fixed throughout training and inference.

We note that for language data, each row of the target $x_\star$ is supported on a finite collection of token embeddings rather than an unrestricted Euclidean space. The MSE loss in (11), however, treats the data predictor $\mu_\theta(x_t, t)$ as an unconstrained regressor and therefore fails to exploit this discrete support. This observation motivates the structured parameterization introduced next.

Embedding-weighted data predictor. Fix a token position $i \in [L]$. The clean embedding of the token at position i is $x_\star^{(i)} = e_{s^{(i)}}$. Observe that its conditional expectation given x_t is a weighted average of all token embeddings according to the posterior token distribution:

$$\mathbb{E}[x_\star^{(i)} \mid x_t] = \sum\nolimits_{j=1}^{V} \mathbb{P}\{s^{(i)} = j \mid x_t\}\, e_j. \tag{12}$$

Consequently, the Bayes-optimal data predictor under the MSE loss (11) lies in the convex hull of the vocabulary embeddings. Moreover, the following proposition reveals a useful multiplicative structure of the posterior distribution: it can be factored into a context-only posterior and an exact Gaussian kernel. The proof is deferred to Appendix D.2.

Proposition 1. *For $x = [x^{(1)}, \dots, x^{(L)}]^\top \in \mathbb{R}^{L \times d}$, denote*

$$x^{(-i)} \coloneqq \big[x^{(1)}, \dots, x^{(i-1)}, x^{(i+1)}, \dots, x^{(L)}\big]^\top \in \mathbb{R}^{(L-1) \times d}. \tag{13}$$

The posterior token distribution satisfies

$$\mathbb{P}\{s^{(i)} = j \mid x_t\} \propto \mathbb{P}\{s^{(i)} = j \mid x_t^{(-i)}\} \exp\!\big(-\|x_t^{(i)} - \alpha_t e_j\|_2^2/(2\sigma_t^2)\big). \tag{14}$$

The identity in (12) establishes an existence result—the posterior probabilities constitute one set of convex weights whose embedding-space barycenter equals the conditional mean of the clean embedding. However, these weights are generally not unique. Because the vocabulary size V is typically much larger than the embedding dimension d, distinct convex weights can produce exactly the same data prediction. Nevertheless, the convex structure in (12) suggests a useful parameterization.

Consequently, our goal is not to learn the posterior distribution itself, but to learn a valid set of convex weights whose embedding-space barycenter accurately predicts the conditional mean. Inspired by the multiplicative form in Proposition 1, we parameterize these convex coefficients using a learned base weight function and the known Gaussian corruption kernel. Specifically, we learn a base weight function $f_\theta^{(i)} : \mathbb{R}^{L \times d} \times [0,1] \mapsto \Delta([V])$, which is optimized using the MSE loss in (11). Importantly, $f_\theta^{(i)}$ is neither supervised nor interpreted as a token posterior. In particular, it is not intended to estimate the distribution $\mathbb{P}\{s^{(i)} = \cdot \mid x_t^{(-i)}\}$ appearing in Proposition 1. Rather, the proposition motivates only the form of the parameterization. We then define the convex weights

$$w_\theta^{(i)}(j \mid x_t, t) = \frac{f_\theta^{(i)}(j \mid x_t, t) \exp\!\big(-\|x_t^{(i)} - \alpha_t e_j\|_2^2/(2\sigma_t^2)\big)}{\sum_{j' \in [V]} f_\theta^{(i)}(j' \mid x_t, t) \exp\!\big(-\|x_t^{(i)} - \alpha_t e_{j'}\|_2^2/(2\sigma_t^2)\big)}, \quad j \in [V]. \tag{15}$$

The resulting data predictor for the token at position i is given by

$$\mu_\theta^{(i)}(x_t, t) = \sum_{j=1}^{V} w_\theta^{(i)}(j \mid x_t, t) e_j = E^\top w_\theta^{(i)}(\cdot \mid x_t, t). \tag{16}$$

Applying (16) to each token position i and stacking the outputs yields the full data predictor $\mu_\theta(x_t, t)$.

In summary, our proposed parameterization preserves the target of unconstrained MSE data prediction while explicitly incorporating both the discrete token structure and known Gaussian corruption.

Provable convergence to token embeddings. Notably, our proposed parameterization for the data predictor guarantees convergence of the sampling trajectory to a valid token embedding. This is formalized below, with the proof deferred to Appendix D.1.

Theorem 1 (Flow convergence to token embeddings)**.** *Assume that, for every token position $i \in [L]$, the learned base weight function satisfies $f_\theta^{(i)}(j \mid x_t, t) > 0$ for any $j \in [V]$, state x_t, and time $t \in (0, 1)$. Moreover, assume that the log-weight is Lipschitz continuous along the sampling trajectory: there exists a constant $\widetilde{L}$ such that*

$$\max_{j \in [V]} \left| \log f_\theta^{(i)}(j \mid x_t, t) - \log f_\theta^{(i)}(j \mid x_\tau, \tau) \right| \le \widetilde{L} \, |t - \tau|, \quad \forall\, 0 < t, \tau < 1.$$

Finally, assume that the grid $0 = t_0 < t_1 < \ldots < t_{N-1} < 1$ satisfies $t_{N-1} \to 1$ as $N \to \infty$ and $(1 + \alpha_{t_k}/\sigma_{t_k})|\alpha_{t_k}/\sigma_{t_k} - \alpha_{t_{k+1}}/\sigma_{t_{k+1}}| < \delta$ for a sufficiently small $\delta > 0$. Then, for each position $i \in [L]$, there exists some $j_i \in [V]$ such that

$$x_{t_{N-1}}^{(i)} \to e_{j_i} \ \text{ in probability } \quad \text{as } \ N \to \infty. \tag{17}$$

Theorem 1 shows that every token-level state converges to a valid token embedding. Consequently, provided that the vocabulary embeddings are distinct, nearest-neighbor decoding naturally yields the token prediction. In particular, no separately trained terminal decoder is required. We empirically validate this convergence behavior in Section 4.

We next explain why data prediction accuracy alone does not guarantee convergence to token embeddings, and why additional structure, such as our convex-structured parameterization, is necessary. In particular, an unconstrained data predictor may be asymptotically accurate along any corruption path while its induced flow fails to converge to any token embedding.

The proposition below provides a concrete counterexample; with proof deferred to Appendix D.3.

Proposition 2. *There exists a smooth, unconstrained data predictor μ_θ such that $\mu_\theta(x_t, t) \to x_\star$ in probability as $t \to 1$, where $x_t = \alpha_t x_\star + \sigma_t z$ and $z_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. Let $x_{t_{N-1}}$ denote the flow output induced by this data predictor on a time grid $0 = t_0 < t_1 < \ldots < t_{N-1} < 1$ with $t_{N-1} \to 1$ as $N \to \infty$. Then there exists a constant $c_{\mathsf{lb}} > 0$, independent of N, such that, for every token position $i \in [L]$ and all sufficiently large N,*

$$\mathbb{P}\left\{ \min_{j \in [V]} \left\| \alpha_{t_{N-1}}^{-1} x_{t_{N-1}}^{(i)} - e_j \right\|_2 \ge 1 \right\} \ge c_{\mathsf{lb}}. \tag{18}$$

This counterexample demonstrates that a smooth, asymptotically accurate data predictor does not by itself ensure convergence to the discrete vocabulary. The convex-structured parameterization provides sufficient structure to guarantee convergence to a valid token embedding.

Comparison with embedding-space flow-based LMs. *Comparison with ELF (Hu et al., 2026).* Both ELF and our framework use the MSE objective to train a data predictor. ELF directly learns the data predictor as an unconstrained regressor. At the final step, it invokes a distinct trained decoder. In contrast, we impose additional structure on the data predictor through (16). This guarantees that the flow automatically converges to a token embedding, so the final decoding does not require a separate decoder.

Comparison with LangFlow (Chen et al., 2026). Both LangFlow and our framework produce a data predictor through a convex combination of vocabulary embeddings. LangFlow directly learns the posterior distribution and trains it using the discrete CE objective. In contrast, we parameterize each convex weight using a learned base weight and an exact Gaussian likelihood, and train the resulting data predictor using the continuous MSE objective (11).

3.2 SAMPLING

Given a trained data predictor μ_θ, we generate samples by solving the data prediction-based ODE in (9), initialized with a standard Gaussian random matrix x_{t_0} with i.i.d. $\mathcal{N}(0,1)$ entries.

Given N sampling steps and a time grid $t_0 < t_1 < \ldots < t_{N-1}$, the ODE can be solved using the first-order Euler method, yielding the following update rule:

$$\frac{x_{t_{i+1}}}{\sigma_{t_{i+1}}} - \frac{x_{t_i}}{\sigma_{t_i}} = \mu_{t_i}\left(\frac{\alpha_{t_{i+1}}}{\sigma_{t_{i+1}}} - \frac{\alpha_{t_i}}{\sigma_{t_i}}\right), \tag{19}$$

where $\mu_{t_i} = \mu_\theta(x_{t_i}, t_i)$ denotes the data prediction used at step i. As we will see momentarily, the data prediction μ_{t_i} can be constructed in various ways, leading to different trade-offs between Gen. PPL and entropy.

After the last step, we convert the generated embedding $x_{t_{N-1}}$ into a token sequence $\widehat{s} = (\widehat{s}^{(1)}, \ldots, \widehat{s}^{(L)})$ by taking the nearest neighbor in the embedding space for each token position:

$$\widehat{s}^{(i)} = \arg\min_{j\in[V]} \|x^{(i)}_{t_{N-1}} - e_j\|_2, \quad i \in [L]. \tag{20}$$

Also, we can use the trained weights $w_\theta(x_{t_{N-1}}, t_{N-1})$ as the token distribution for decoding, i.e.,

$$\widehat{s}^{(i)} = \arg\max_{j\in[V]} w^{(i)}_\theta(j \mid x_{t_{N-1}}, t_{N-1}), \quad i \in [L]. \tag{21}$$

Both token prediction rules are parameter-free and require no separately trained terminal decoder.

Next, we describe sampling with self-conditioning. The data prediction in the ideal two-pass implementation of self-conditioning is given by $\mu_{t_i} = \mu_\theta\big(x_{t_i}, t_i, \mu_\theta(x_{t_i}, t_i, \varnothing)\big)$. To reduce the computational burden, it is common to construct the data prediction μ_{t_i} at step i using that from the preceding step $i-1$ as the self-conditioning input, in place of the same-step unconditional prediction $\mu_\theta(x_{t_i}, t_i, \varnothing)$, i.e.,

$$\mu_{t_0} = \mu_\theta\big(x_{t_0}, t_0, \varnothing\big) \quad \text{and} \quad \mu_{t_i} = \mu_\theta\big(x_{t_i}, t_i, \mu_{t_{i-1}}\big), \;\; i \geq 1. \tag{22}$$

The data prediction μ_{t_i} is then used in the sampling update (19). One can expect that $\mu_{t_{i-1}} \approx \mu_\theta(x_{t_i}, t_i, \varnothing)$ because t_{i-1} and t_i, as well as $x_{t_{i-1}}$ and x_{t_i} are close when the solver uses sufficiently many steps.

In addition, we introduce three inference mechanisms for controlling the trade-off between Gen. PPL and entropy. Due to the space limitation, we defer the detailed discussion to Appendix C.2.

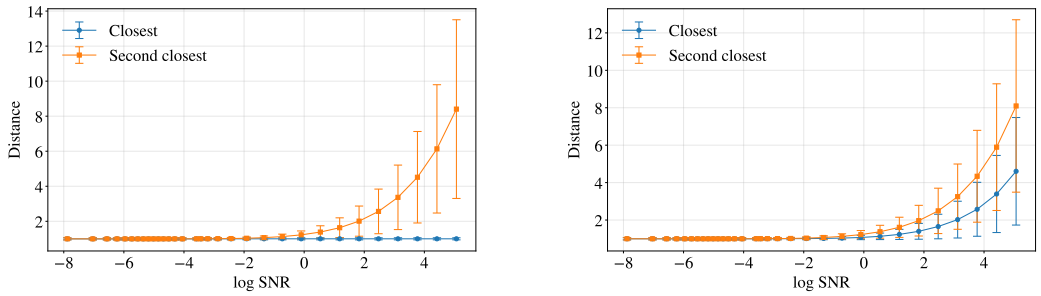
4 EXPERIMENTS

Dataset. We follow the experimental setup used in the literature on DLMs (Chen et al., 2026; Hu et al., 2026; Sahoo et al., 2025). We conduct all experiments on the OpenWebText (OWT) dataset (Raffel et al., 2020), which contains approximately 9B tokens, and pack the text into sequences of length $L = 1024$.

Training. We follow the architecture and setup of LangFlow (Chen et al., 2026). We use the same DiT-style Transformer architecture (Peebles & Xie, 2023) as LangFlow, which consists of 12 layers, a hidden dimension of 768, and 12 attention heads, totaling approximately 130M parameters. Self-conditioning is applied during training with probability 0.25.

Because jointly learning the token embeddings and data predictor under the MSE objective admits degenerate embedding-collapse solutions, we use the embedding matrix from the LangFlow checkpoint and keep it fixed throughout training. For all experiments except the convergence study, the remaining trainable parameters are also initialized from the same checkpoint.

For comparison, we continue training our model using the MSE objective and the LangFlow baseline using its token-level CE objective for additional 200K steps. Both models are trained using AdamW (Loshchilov & Hutter, 2017) with a batch size of 480 and a learning rate of 10^{-5}. Training is distributed across four or eight NVIDIA A100 40 GB GPUs, depending on availability.



(a) Embedding-weighted data predictor

(b) Unconstrained direct data predictor

Figure 2: Comparison of flow convergence between the embedding-weighted and direct data predictors. The two models use the same fixed LangFlow embeddings, architecture, and training configuration, differing only in their output parameterization. We plot the smallest and second-smallest normalized distances $\|x_t^{(i)} - \alpha_t e_j\|_2/(\sigma_t\sqrt{d})$ from the flow state associated with each token position $x_t^{(i)}$ to the scaled token embeddings as functions of log-SNR.

Evaluation. For each sampling configuration, we generate 1024 samples with length $L = 1024$. We measure quality using Gen. PPL, evaluated by GPT-2 Large (Radford et al., 2019), and quantify diversity using unigram entropy. We compare sampling methods based on their Gen. PPL–entropy trade-off, where lower Gen. PPL indicates higher quality and higher entropy means greater diversity.

We observe that the sampling grid used by LangFlow is highly nonuniform near the two endpoints. Therefore, we use the uniform grid $t_i = (i + 0.5)/N$ for $0 \le i < N$, where N denotes the number of sampling steps. This grid slightly outperforms the original one; see Figure 4 in Appendix E.

Unless otherwise specified, our default sampling configuration is the standard sampler with one-step self-conditioning, as defined in (19) and (22), without additional techniques introduced in Section 3.2; see Appendix E.5 for generated examples at an entropy of 5.44. This corresponds to $w_{\mathsf{scg}} = 1$, $w_{\mathsf{ug}} = 0$, and $K = 1$. We compare all sampling methods under the same number of function evaluations (NFEs). Every evaluation of the data predictor is counted, including additional evaluations introduced by self-conditioning guidance or iterative self-conditioning refinement.

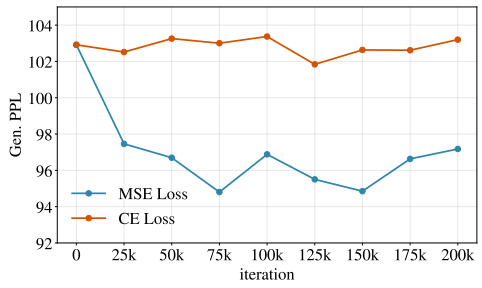
Unless otherwise stated, the results reported below use the checkpoint obtained after 175K additional training steps, which achieved the best performance among the evaluated checkpoints.

4.1 Empirical flow convergence to token embeddings

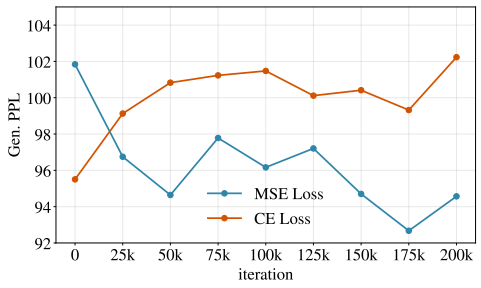
We empirically examine how the parameterization of the data predictor affects the convergence of the learned flow. We train two models from random initialization using the same pre-trained LangFlow embedding matrix, network architecture, and training configuration. The models differ only in their output parameterization. The embedding-weighted data predictor expresses its output as a weighted combination of token embeddings, as defined in (16), whereas the direct data predictor outputs an unconstrained vector in the embedding space. This controlled comparison isolates the effect of the structured parameterization: Theorem 1 guarantees the learned flow driven by the embedding-weighted data predictor converges to valid token embeddings, while Proposition 2 shows that such a convergence guarantee may not hold for an unconstrained data predictor.

For each position i, we compute the normalized distance between its state $x_t^{(i)}$ and every scaled token embedding $\alpha_t e_j$: $\frac{\|x_t^{(i)} - \alpha_t e_j\|_2}{\sigma_t\sqrt{d}}$, $j \in [V]$. For each log-SNR value, we aggregate the smallest and second-smallest distances over $16 \times 1024 = 16{,}384$ token positions from 16 generated sequences of length $L = 1024$, separately for each data predictor. For each distance statistic, we plot the midpoint of its empirical 1st–99th percentile interval, with error bars spanning the full interval.

As shown in Figure 2, the closest and second-closest distances are nearly indistinguishable in the low-SNR regime for both data predictors, but their behaviors diverge as the SNR increases. For the embedding-weighted data predictor in Figure 2(a), the closest normalized distance approaches



(a) Continued training from LangFlow checkpoint



(b) Swapping objectives after 100K training steps

Figure 3: Effect of the training objective on Gen. PPL. (a) Starting from the same pre-trained LangFlow checkpoint, MSE-based training yields an immediate and persistent improvement over CE-based training. (b) In the crossover experiment, the CE-trained checkpoint improves rapidly after switching to MSE, whereas the MSE-trained checkpoint degrades after switching to CE. The horizontal axis denotes additional training steps after the objective switch.

one and remains tightly concentrated, whereas the second-closest distance increases rapidly, producing a clear separation. The growing separation between the nearest and second-nearest normalized distances further indicates that the approaching token embedding becomes uniquely distinguishable. Together, these observations provide empirical evidence of convergence toward individual valid token embeddings. In contrast, for the unconstrained direct data predictor in Figure 2(b), both distances increase, exhibit substantially greater variation, and remain poorly separated. These empirical results support the geometric distinction between the structured and unconstrained predictors motivated by Theorem 1 and Proposition 2.

4.2 Effect of the training objective: MSE versus CE

Starting from the same pre-trained LangFlow checkpoint, we continue training two models for 200K iterations. The two branches differ only in the training objective: one uses the embedding-space MSE loss, while the other uses token-level CE. We evaluate Gen. PPL and entropy every 25K steps.

The entropy remains above 5.5 for both objectives. As for Gen. PPL, Figure 3(a) shows that training with the CE objective provides no consistent improvement over the initial checkpoint and its Gen. PPL remains near its initial value. In contrast, training with the MSE objective steadily reduces Gen. PPL and maintains a clear advantage, despite some fluctuations across checkpoints.

To further isolate the effect of the training objective, we conduct a crossover experiment using the checkpoints obtained after 100K steps. Specifically, we continue the CE-trained checkpoint using the MSE objective and, conversely, continue the MSE-trained checkpoint using the CE objective. As shown in Figure 3(b), despite starting from the worse-performing CE-trained checkpoint, switching to MSE reduces Gen. PPL consistently. Conversely, switching the better-performing MSE-trained checkpoint to CE degrades Gen. PPL. Together, these results consistently support the effectiveness of MSE relative to CE under the same initialization. In this setting, subsequent CE-based training can reverse gains previously obtained through MSE-based training.

5 Discussion

ConvergeFlow suggests several directions for future work. First, our formulation keeps the token embeddings fixed to avoid degenerate solutions. It would be interesting to develop fully continuous objectives that support joint learning of the embedding and data predictor. Second, extending the convergence analysis under weaker assumptions and developing non-asymptotic theoretical guarantees would provide a more complete account of practical generation. Third, scaling ConvergeFlow to larger models and evaluating it on conditional generation, instruction following, and reasoning tasks will clarify its broader applicability. Finally, the continuous formulation may facilitate the use of techniques such as distillation and higher-order acceleration, whose effectiveness for language generation remains to be explored.

AI USE STATEMENT

In preparing this paper, large language models (LLMs) were employed solely as assistive tools for minor language refinement and for literature retrieval and discovery (e.g., identifying related work). All technical contributions, results, and conclusions are entirely the work of the authors.

REFERENCES

Marianne Arriola, Aaron Gokaslan, Justin Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, volume 2025, pp. 50726–50753, 2025.

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.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *Advances in Neural Information Processing Systems*, 38:55981–56007, 2026.

Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. *arXiv preprint arXiv:2308.03686*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Changxiao Cai and Gen Li. Minimax optimality of the probability flow ode for diffusion models. *arXiv preprint arXiv:2503.09583*, 2025.

Changxiao Cai and Gen Li. Confidence-based decoding is provably efficient for diffusion language models. *arXiv preprint arXiv:2603.22248*, 2026.

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.

Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *International Conference on Machine Learning*, pp. 4735–4763. PMLR, 2023a.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022a.

Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ode is provably fast. *Advances in Neural Information Processing Systems*, 36:68552–68575, 2023b.

Sitan Chen, Kevin Cong, and Jerry Li. Optimal inference schedules for masked diffusion models. *arXiv preprint arXiv:2511.04647*, 2025.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022b.

Yuxin Chen, Chumeng Liang, Hangke Sui, Ruihan Guo, Chaoran Cheng, Jiaxuan You, and Ge Liu. Langflow: Continuous diffusion rivals discrete in language modeling. *arXiv preprint arXiv:2604.11748*, 2026.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

Daniil Dmitriev, Zhihan Huang, and Yuting Wei. Efficient sampling with discrete diffusion models: Sharp and adaptive guarantees. *arXiv preprint arXiv:2602.15008*, 2026.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Hengyu Fu, Baihe Huang, Virginia Adams, Charles Wang, Venkat Srinivasan, and Jiantao Jiao. From bits to rounds: Parallel decoding with exploration for diffusion language models. *arXiv preprint arXiv:2511.21103*, 2025.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36:16693–16715, 2023.

Hongcan Guo, Qinyu Zhao, Yian Zhao, Shen Nie, Rui Zhu, Qiushan Guo, Feng Wang, Tao Yang, Hengshuang Zhao, Guoqiang Wei, et al. Continuous latent diffusion language model. *arXiv preprint arXiv:2605.06548*, 2026.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11575–11596, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. Elf: Embedded language flows. *arXiv preprint arXiv:2605.10938*, 2026.

Nishant Jain and Tong Zhang. A sharp kl convergence analysis for diffusion models under minimal assumptions. In *International Conference on Learning Representations*, volume 2026, pp. 22706–22735, 2026.

Yuchen Jiao and Gen Li. Instance-dependent convergence theory for diffusion models. *arXiv preprint arXiv:2410.13738*, 2024.

Yuchen Jiao, Yuxin Chen, and Gen Li. Towards a unified framework for guided diffusion models. *arXiv preprint arXiv:2512.04985.*, 2025a.

Yuchen Jiao, Yuchen Zhou, and Gen Li. Optimal convergence analysis of ddpm for general distributions. *arXiv preprint arXiv:2510.27562*, 2025b.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*, 2025.

Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Hugo Lavenant and Giacomo Zanella. Error bounds and optimal schedules for masked diffusions with factorized approximations. *arXiv preprint arXiv:2510.25544*, 2025.

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising. *arXiv preprint arXiv:2602.16813*, 2026.

Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. *Advances in Neural Information Processing Systems*, 35:22870–22882, 2022.

Gen Li and Changxiao Cai. Provable acceleration for diffusion models under minimal assumptions. *arXiv preprint arXiv:2410.23285*, 2024.

Gen Li and Changxiao Cai. Breaking AR's sampling bottleneck: Provable acceleration via diffusion language models. *Advances in Neural Information Processing Systems*, 38:11700–11725, 2025.

Gen Li and Yuchen Jiao. Provable efficiency of guidance in diffusion models for general data distribution. In *International Conference on Machine Learning*, volume 267, pp. 35034–35046, 2025.

Gen Li and Yuling Yan. $O(d/T)$ convergence theory for diffusion probabilistic models under minimal assumptions. *arXiv preprint arXiv:2409.18959*, 2024.

Gen Li, Yuting Wei, Yuejie Chi, and Yuxin Chen. A sharp convergence theory for the probability flow odes of diffusion models. *arXiv preprint arXiv:2408.02320*, 2024.

Gen Li, Changxiao Cai, and Yuting Wei. Dimension-free convergence of diffusion models for approximate gaussian mixtures. *arXiv preprint arXiv:2504.05300*, 2025a.

Gen Li, Yuchen Zhou, Yuting Wei, and Yuxin Chen. Faster diffusion models via higher-order approximation. *arXiv preprint arXiv:2506.24042*, 2025b.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. Latent diffusion for language generation. *Advances in Neural Information Processing Systems*, 36:56998–57025, 2023.

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, 2022a.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

Rabeeh Karimi Mahabadi, Hamish Ivison, Jaesung Tae, James Henderson, Iz Beltagy, Matthew E Peters, and Arman Cohan. Tess: Text-to-text self-conditioned simplex diffusion. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2347–2361, 2024.

Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. *Advances in Neural Information Processing Systems*, 35: 34532–34545, 2022.

Viacheslav Meshchaninov, Egor Chimbulatov, Alexander Shabalin, Aleksandr Abramov, and Dmitry Vetrov. Cosmos: Compressed and smooth latent space for text diffusion modeling. *Advances in Neural Information Processing Systems*, 38:14271–14299, 2026a.

Viacheslav Meshchaninov, Alexander Shabalin, Egor Chimbulatov, Nikita Gushchin, Ilya Koziev, Alexander Korotin, and Dmitry Vetrov. How to train your latent diffusion language model jointly with the latent space. *arXiv preprint arXiv:2605.07933*, 2026b.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Kazusato Oko, Shunta Akiyama, and Taiji Suzuki. Diffusion models are minimax optimal distribution estimators. In *International Conference on Machine Learning*, pp. 26517–26582. PMLR, 2023.

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.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Peter Potaptchik, Jason Yim, Adhi Saravanan, Peter Holderrieth, Eric Vanden-Eijnden, and Michael S Albergo. Discrete flow maps. *arXiv preprint arXiv:2604.09784*, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Daan Roos, Oscar Davis, Floor Eijkelboom, Michael Bronstein, Max Welling, Ismail Ilkan Ceylan, Luca Ambrogioni, and Jan-Willem van de Meent. Categorical flow maps. *arXiv preprint arXiv:2602.12233*, 2026.

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.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. *Proceedings of machine learning research*, 267: 52584, 2025.

Subham Sekhar Sahoo, Jean-Marie Lemercier, Zhihan Yang, Justin Deschenaux, Jingyu Liu, John Thickstun, and Ante Jukic. Scaling beyond masked diffusion language models. *arXiv preprint arXiv:2602.15014*, 2026.

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.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, volume 2024, pp. 15078–15097, 2024.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

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.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

Jaesung Tae, Hamish Ivison, Sachin Kumar, and Arman Cohan. Tess 2: A large-scale generalist diffusion language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21171–21188, 2025.

Brian L Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

Martin J Wainwright. The data geometry of masking diffusion: Certified-optimal schedules via unmasking growth complexity. *arXiv preprint arXiv:2608.13520*, 2026.

Shintaro Wakasugi and Taiji Suzuki. State size independent statistical error bound for discrete diffusion models. volume 38, pp. 138908–138943, 2026.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.

Jingda Wu and Changxiao Cai. Diffusion models are statistically optimal for learning low-dimensional multi-modal distributions. *arXiv preprint arXiv:2605.30153*, 2026.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024.

Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*, 2025.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025.

Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Joshua Susskind, and Navdeep Jaitly. Planner: Generating diversified paragraph via latent language diffusion model. *Advances in Neural Information Processing Systems*, 36:80178–80190, 2023.

Zixuan Zhang, Hengyu Fu, Zhuoran Yang, Mengdi Wang, Tuo Zhao, and Minshuo Chen. Generalization bounds for discrete diffusion: Statistical advantage of masking. In *International Conference on Machine Learning*, 2026.

Yunxiao Zhao and Changxiao Cai. Adaptation to intrinsic dependence in diffusion language models. *arXiv preprint arXiv:2602.20126*, 2026.

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. In *International Conference on Learning Representations*, volume 2025, pp. 63186–63227, 2025.

A Other related work

Continuous DLMs. Continuous DLMs differ primarily in the space over which Gaussian diffusion is performed. At the token level, embedding-space DLMs (Li et al., 2022; Dieleman et al., 2022; Gong et al., 2022; Gulrajani & Hashimoto, 2023) diffuse a sequence of token-level embeddings. Simplex-based DLMs (Han et al., 2023; Mahabadi et al., 2024; Tae et al., 2025; Potaptchik et al., 2026; Roos et al., 2026) instead map each token to a point on the probability simplex over the vocabulary, while Lee et al. (2026) adopts the one-hot encoding. Because such token-level states may not adequately capture contextual semantics, latent DLMs perform diffusion in sequence-level latent spaces constructed from contextual representations. Earlier approaches obtain these features from the outputs of a frozen pre-trained encoder (Lovelace et al., 2023; Zhang et al., 2023; Meshchaninov et al., 2026a), whereas more recent works jointly learn the latent encoder and the diffusion model (Meshchaninov et al., 2026b; Guo et al., 2026).

Discrete DLMs. Two families dominate modern discrete DLMs, distinguished by their forward corruption processes: uniform diffusion models (UDMs) and masked diffusion models (MDMs). UDMs progressively corrupt tokens toward the uniform distribution over the vocabulary (Sahoo et al., 2025; 2026). MDMs instead augment the vocabulary with a special mask token as an absorbing state, and progressively replace tokens with it (Sahoo et al., 2024; Shi et al., 2024). In MDMs, the discrete score (Meng et al., 2022; Lou et al., 2023) is equivalent to the joint conditional distribution of the masked tokens given the unmasked context (Ou et al., 2024; Zheng et al., 2025). In practice, masked DLMs approximate this joint conditional by a product of token-wise conditional marginals (Nie et al., 2025; Arriola et al., 2025). Although this enables parallel generation, it imposes a conditional independence assumption among tokens revealed in the same iteration and thus introduces an inherent factorization bias. Consequently, the unmasking strategy—which determines the number and positions of tokens to reveal at each step—plays a critical role in the performance of masked DLMs (Kim et al., 2025; Wu et al., 2025; Yu et al., 2025; Ben-Hamu et al., 2026; Fu et al., 2025).

Theory for DLMs. Because masked DLMs have historically outperformed continuous DLMs and uniform DLMs, theoretical analyses for DLMs have largely focused on masked DLMs, particularly on characterizing the accuracy-speed trade-off in parallel generation. Early work studied fixed-size, random-ordering unmasking (Shi et al., 2024; Sahoo et al., 2024), which prescribes the number of sampling steps and tokens revealed per step while selecting their positions at random. Li & Cai (2025) derived the first convergence guarantees for such strategies, which were subsequently sharpened using information-theoretic quantities that capture low-complexity structure in the data distribution (Chen et al., 2025; Lavenant & Zanella, 2025; Zhao & Cai, 2026; Wainwright, 2026; Dmitriev et al., 2026). More recently, Cai & Li (2026) established the provable efficiency of confidence-based unmasking (Ben-Hamu et al., 2026), which adaptively selects both the number and positions of tokens to reveal based on the model's predictive confidence. Parallel to these sampling convergence results, statistical generalization guarantees have been established in Wakasugi & Suzuki (2026); Zhang et al. (2026).

Theory for continuous diffusion models. Recent years have witnessed substantial theoretical progress in continuous diffusion models (Lee et al., 2022; Chen et al., 2023b; Oko et al., 2023; Cai

& Li, 2025; Wu & Cai, 2026). In particular, a line of work derives convergence guarantees for sampling from data distributions under mild assumptions, such as bounded moments, without requiring globally Lipschitz score functions (Chen et al., 2022a; 2023a; Benton et al., 2023; Li et al., 2024; Li & Yan, 2024; Jiao & Li, 2024; Jiao et al., 2025b). Because token-level embedding distributions have finite support and hence bounded moments, these results naturally apply to continuous DLMs and flow-based LMs. Provably accelerated samplers based on higher-order approximations are further developed in (Li & Cai, 2024; Li et al., 2025b; Jain & Zhang, 2026). Provably efficient guidance scheme has been establish in (Li & Jiao, 2025; Jiao et al., 2025a). Moreover, the discrete nature of text also provides additional structure where Gaussian smoothing of the discrete token-embedding distribution yields a Gaussian mixture. Exploiting this structure, nearly dimension-free convergence guarantees have been established in Li et al. (2025a).

B Other preliminary

We briefly review the metrics commonly used to evaluate language models. Let p_θ denote the distribution induced by a trained language model.

Because the data distribution p_{data} is unknown, the quality of the trained model is often assessed using a reference language model such as the GPT-2 Large model (Radford et al., 2019). Specifically, let p_{ref} denote the distribution of the reference model. *Generative perplexity* (Gen. PPL) is defined as

$$\mathrm{PPL}_{\mathrm{gen}}(p_\theta; p_{\mathsf{ref}}) \coloneqq \exp\left(-\frac{1}{L}\mathbb{E}_{X\sim p_\theta}[\log p_{\mathsf{ref}}(X)]\right), \tag{23}$$

which satisfies the following relationship:

$$\log \mathrm{PPL}_{\mathrm{gen}}(p_\theta; p_{\mathsf{ref}}) = \frac{1}{L} D_{\mathrm{KL}}(p_\theta \,\|\, p_{\mathsf{ref}}) + \frac{1}{L} H(p_\theta), \tag{24}$$

where $D_{\mathrm{KL}}(p_\theta \,\|\, p_{\mathsf{ref}})$ is the Kullback-Leibler (KL) divergence between the model distribution p_θ and the reference distribution p_{ref}, and $H(p_\theta)$ denotes the entropy of p_θ.

Identity (24) reveals that low Gen. PPL may arise either because the generated samples have high likelihood under the reference model or because the model concentrates its probability mass on a small set of likely sequences.

A commonly used diagnostic is *unigram entropy*, which measures token-frequency diversity within individual sequences. For a sequence $x = (x^{(1)}, \ldots, x^{(L)})$, define its empirical unigram distribution by

$$\widehat{p}_x(v) = \frac{1}{L}\sum_{i=1}^{L} \mathbb{1}\{x^{(i)} = v\}, \quad \forall\, v,$$

and let $H(\widehat{p}_x)$ denote the corresponding entropy. The unigram entropy is then defined as

$$H_{\mathsf{uni}}(p_\theta) \coloneqq \mathbb{E}_{X\sim p_\theta}[H(\widehat{p}_X)]. \tag{25}$$

In practice, the expectations defining Gen. PPL and unigram entropy are approximated by averaging over independent samples.

C Other results

Moreover, recall that the data predictor is a convex combination of the token embeddings, with weights given by $w_\theta^{(i)}$. The following proposition shows that if the data predictor converges to a token embedding, then the corresponding weight vectors converge to a one-hot vector[2]. The proof is deferred to Appendix D.4.

[2]The fixed LangFlow embeddings used in our experiments satisfy the assumptions of Proposition 3.

Proposition 3. *Assume that the token embeddings $\{e_j\}_{j\in[V]}$ have the same norm and are pairwise separated, namely, $\max_{j\neq j'}\left|\frac{\langle e_j, e_j'\rangle}{\|e_j\|_2\|e_j'\|_2}\right| \leq 1-\rho$ for some constant $\rho > 0$. If there exists some $j \in [V]$ such that $\mu_\theta^{(i)}(x_t, t) \to e_j$ as $t \to 1$, then one has*

$$w_\theta^{(i)}(\cdot \,|\, x_t, t) \to \delta_j \quad \text{as } t \to 1, \tag{26}$$

where $\delta_j \in \mathbb{R}^V$ denotes the one-hot vector associated with token j.

Table 1: Summary of sampling techniques for quality-diversity trade-offs.

Technique	Control parameter	Intended effect
Self-conditioning guidance	Coefficient w_{scg}	Amplify refinement from self-conditioning
Iterative self-conditioning refinement	Iteration count K	Refine data prediction
Unconditional guidance	Coefficient w_{ug}	Strengthen movement toward gradient of PPL

C.1 CONSISTENCY OF TWO TOKEN PREDICTION RULES

Having empirically examined convergence toward token embeddings, we next examine the consistency of the weight-based and distance-based token prediction rules. Theorem 1 and Proposition 3 together imply their asymptotic equivalence: the flow state converges to a token embedding, while the corresponding weights converge to its one-hot representation. As the number of sampling steps N varies from 32 to 512, the two rules agree at 99.16%–99.82% of token positions, with the agreement increasing as the sampling discretization becomes finer. This near-perfect agreement indicates that the two rules are effectively equivalent in practice, with the remaining discrepancies diminishing as the sampling time discretization becomes finer.

C.2 SAMPLING TECHNIQUES FOR QUALITY-DIVERSITY TRADE-OFFS

Controlling Gen. PPL-entropy trade-off. We introduce three inference mechanisms for controlling the trade-off between Gen. PPL and entropy; see Table 1 for a summary.

- *self-conditioning guidance.* Motivated by classifier-free guidance (CFG) (Ho & Salimans, 2022), we introduce a CFG-type guidance for self-conditioning. At each solver step i, we form the guided data prediction via the unconditional and one-step self-conditioned predictions:

  $$\mu_{t_i}^{\mathsf{scg}} = \mu_\theta(x_{t_i}, t_i, \varnothing) + w_{\mathsf{scg}}\big(\mu_\theta(x_{t_i}, t_i, c_{t_i}) - \mu_\theta(x_{t_i}, t_i, \varnothing)\big) \quad \text{with} \quad c_{t_i} = \mu_\theta(x_{t_i}, t_i, \varnothing). \tag{27}$$

  When $w_{\mathsf{scg}} = 0$, this reduces to sampling without self-conditioning; when $w_{\mathsf{scg}} = 1$, it recovers the ordinary sampling with self-conditioning. Values of $w_{\mathsf{scg}} > 1$ extrapolate beyond the self-conditioned prediction and amplify the refinement induced by self-conditioning.

  Although the form in (27) resembles CFG, the condition here is generated by the model itself rather than supplied externally.

- *Iterative self-conditioning refinement.* Recall the computational shortcut for self-conditioning in (22), which reuses the data prediction from the previous step. Unrolling this recursion over K steps shows that the data prediction μ_{t_i} at time t_i satisfies

  $$\mu_{t_{i-j}} = \mu_\theta\big(x_{t_{i-j}}, t_{i-j}, \mu_{t_{i-j-1}}\big), \quad j = 0, \ldots, K-1,$$

  or equivalently,

  $$\mu_{t_i} = \mu_\theta\Big(x_{t_i}, t_i, \mu_\theta\big(x_{t_{i-1}}, t_{i-1}, \ldots, \mu_\theta(x_{t_{i-K+1}}, t_{i-K+1}, \mu_{t_{i-K}})\ldots\big)\Big).$$

If the time grid is sufficiently fine, then the state and time vary little over these K steps, and $\mu_\theta(x_{t_{i-j}}, t_{i-j}, c) \approx \mu_\theta(x_{t_i}, t_i, c)$ for $j = 0, \dots, K-1$. Consequently, the data prediction μ_{t_i} is approximately given by

$$\mu_{t_i} \approx \mu_\theta\Big(x_{t_i}, t_i, \mu_\theta\big(x_{t_i}, t_i, \dots, \mu_\theta(x_{t_i}, t_i, \mu_{t_{i-K}})\dots\big)\Big),$$

where K recursive evaluations are all applied to the current state x_{t_i} and time t_i. Thus, reusing the previous data prediction in self-conditioning implicitly produces a recursive refinement whose effective depth depends on the number and spacing of solver steps.

Motivated by this observation, we make the self-conditioning refinement explicit. At each sampling step i, we define

$$u_{t_i}^0 \coloneqq \mu_\theta(x_{t_i}, t_i, \varnothing), \qquad u_{t_i}^k \coloneqq \mu_\theta(x_{t_i}, t_i, u_{t_i}^{k-1}), \quad k = 1, \dots, K, \tag{28}$$

and use $u_{t_i}^K$ as the data prediction in the solver update. This construction makes the recursion depth K an explicit hyperparameter, thereby decoupling it from the number of solver steps.

Empirically, we observe that iterative self-conditioning refinement is less sensitive to the solver-step count than the standard self-conditioning shortcut in (22). Moreover, varying the depth K provides an effective means of controlling the trade-off between Gen. PPL and entropy.

- *Unconditional guidance.* We introduce unconditional guidance by amplifying the score-inspired direction of the sampling update. By Tweedie's formula, we have

$$\nabla \log p_t(x) = -\frac{\varepsilon(x,t)}{\sigma_t}, \quad \text{with} \quad \varepsilon(x,t) = \frac{x - \alpha_t \mu(x,t)}{\sigma_t}.$$

Recall the standard noise prediction-based update rule:

$$\frac{x_{t_{i+1}}}{\alpha_{t_{i+1}}} - \frac{x_{t_i}}{\alpha_{t_i}} = \left(\frac{\sigma_{t_{i+1}}}{\alpha_{t_{i+1}}} - \frac{\sigma_{t_i}}{\alpha_{t_i}}\right)\varepsilon_\theta(x_{t_i}, t_i).$$

As σ_t/α_t decreases along the sampling process, the coefficient on the right-hand side is negative and the update therefore moves in the direction of $-\varepsilon_\theta$. To strengthen this motion, we multiply the update by a factor of $1 + w_{\text{ug}}$, resulting in the following sampler:

$$\frac{x_{t_{i+1}}}{\alpha_{t_{i+1}}} - \frac{x_{t_i}}{\alpha_{t_i}} = (1 + w_{\text{ug}})\left(\frac{\sigma_{t_{i+1}}}{\alpha_{t_{i+1}}} - \frac{\sigma_{t_i}}{\alpha_{t_i}}\right)\varepsilon_\theta(x_{t_i}, t_i). \tag{29}$$

D Proof of theorem and propositions

D.1 Proof of Theorem 1

In this section, we provide a proof of Theorem 1, organized into four steps.

Step 1: Construction of an auxiliary sequence. Define the probability simplex over the vocabulary as

$$F \coloneqq \left\{ f \in \mathbb{R}_+^V : \sum_{v \in [V]} f(v) = 1 \right\}.$$

We first note that given an initial point x_{t_0} and a collection of base weights $\{\widehat{f}_k^{(i)}\}_{0 \le k < N-1, 1 \le i \le L} \subset F$ satisfying

$$\max_{j,k,i} \left| \log \widehat{f}_{k+1}^{(i)}(j) - \log \widehat{f}_k^{(i)}(j) \right| \le \widetilde{L}\, |t_{k+1} - t_k|,$$

we can use them to construct an auxiliary flow state sequence $\widehat{x}_{t_k}$ as follows.

Initialize the auxiliary sequence $\widehat{x}_{t_0} = x_{t_0}$. For each token position i and step k, define the auxiliary weights

$$\widehat{w}_k^{(i)}(j \mid \widehat{x}_{t_k}) \coloneqq \frac{\widehat{f}_k^{(i)}(j) \exp\left(-\frac{\|\widehat{x}_{t_k}^{(i)} - \alpha_{t_k} e_j\|_2^2}{2\sigma_{t_k}^2}\right)}{\sum_{j=1}^V \widehat{f}_k^{(i)}(j) \exp\left(-\frac{\|\widehat{x}_{t_k}^{(i)} - \alpha_{t_k} e_j\|_2^2}{2\sigma_{t_k}^2}\right)}, \quad j \in [V]. \tag{30}$$

Thus, $\widehat{w}_k^{(i)}(j \mid \widehat{x}_{t_k})$ is the counterpart of the learned weight $w_\theta^{(i)}(j|\widehat{x}_{t_k}, t_k)$, obtained by replacing the learned base weight $f_\theta^{(i)}(j \mid x_{t_k}, t_k)$ with the data-independent quantity $\widehat{f}_k^{(i)}(j)$.

The corresponding auxiliary data predictor $\widehat{\mu}_\theta(\widehat{x}_{t_k}, t_k)$ is given by

$$\widehat{\mu}_\theta^{(i)}(\widehat{x}_{t_k}, t_k) \coloneqq \sum_{j=1}^{V} \widehat{w}_k^{(i)}(j \mid \widehat{x}_{t_k}) e_j. \tag{31}$$

We then iteratively update the auxiliary sequence according to

$$\widehat{x}_{t_{k+1}} = \frac{\sigma_{t_{k+1}}}{\sigma_{t_k}} \widehat{x}_{t_k} + \big(\alpha_{t_{k+1}} - \frac{\sigma_{t_{k+1}} \alpha_{t_k}}{\sigma_{t_k}}\big) \widehat{\mu}_\theta(\widehat{x}_{t_k}, t_k), \qquad k = 0, 1, \ldots, N-2. \tag{32}$$

With this construction procedure in place, we next show that the auxiliary sequence can reproduce the original sampling trajectory. Given the original trajectory $(x_{t_k})_{k=0}^{N-1}$, choose

$$\widehat{f}_k^{(i)}(j) \coloneqq f_\theta^{(i)}(j \mid x_{t_k}, t_k), \qquad j \in [V], \tag{33}$$

for each iteration $0 \le k < N-1$ and token position $i \in [L]$. The log-Lipschitz assumption in Theorem 1 ensures that this sequence satisfies the required regularity condition. We claim that

$$x_{t_k} = \widehat{x}_{t_k}, \qquad 0 \le k \le N-1. \tag{34}$$

We prove this claim by induction. First, it holds trivially for $k = 0$. Next, suppose that $x_{t_k} = \widehat{x}_{t_k}$ holds for some k. By the choice of $\widehat{f}_k^{(i)}(j)$ in (33) and the construction of $\widehat{w}_k^{(i)}$ in (30), we have

$$\widehat{w}_k^{(i)}(j \mid \widehat{x}_{t_k}) = w_\theta^{(i)}(j|\widehat{x}_{t_k}, t_k) = w_\theta^{(i)}(j \mid x_{t_k}, t_k). \tag{35}$$

It then follows from (31) and (16) that

$$\widehat{\mu}_\theta^{(i)}(\widehat{x}_{t_k}, t_k) = \sum_{j=1}^{V} w_\theta^{(i)}(j \mid x_{t_k}, t_k) e_j = \mu_\theta^{(i)}(x_{t_k}, t_k).$$

Therefore, the update rule of the auxiliary sequence in (32) coincides with that of the original sequence in (9), thereby yielding $x_{t_{k+1}} = \widehat{x}_{t_{k+1}}$. This completes the induction and proves (34).

Consequently, it remains to analyze the auxiliary sequence for an arbitrary collection of base weights satisfying the stated regularity condition. Concretely, we will show that for any token position $i \in [L]$, there exists some $j_i \in [V]$, such that

$$\widehat{x}_{t_{N-1}}^{(i)} \to e_{j_i}, \qquad \text{as} \quad t_{N-1} \to 1.$$

Step 2: Construction of reference distributions. For each token position i, let $q_k^{(i)}$ denote the probability density function of $\widehat{x}_{t_k}^{(i)}$ when $\widehat{x}_{t_0}^{(i)}$ is initialized with the standard Gaussian distribution. In addition, for each $0 \le k \le N$ and $i \in [L]$, we define the reference distribution

$$\widehat{p}_k^{(i)}(x) \coloneqq (2\pi\sigma_{t_k}^2)^{-d/2} \sum_{j=1}^{V} \widehat{f}_k^{(i)}(j) \exp\Big(-\frac{\|x - \alpha_{t_k} e_j\|_2^2}{2\sigma_{t_k}^2}\Big), \qquad \forall x \in \mathbb{R}^d.$$

In the remainder of the proof, we focus on a token position i. For notational brevity, we omit the superscript (i) from $\widehat{f}_k^{(i)}$, $\widehat{w}_k^{(i)}$, $\widehat{\mu}_\theta^{(i)}$, $\widehat{p}_k^{(i)}$ and $\widehat{q}_k^{(i)}$, and $\widehat{x}_{t_k}^{(i)}$.

We first compare the reference densities at two consecutive points along the auxiliary trajectory. By the definition of $\widehat{p}_k$, one can derive

$$\begin{aligned}
\frac{\widehat{p}_{k+1}(\widehat{x}_{t_{k+1}})}{\widehat{p}_k(\widehat{x}_{t_k})} &= \frac{1}{\widehat{p}_k(\widehat{x}_{t_k})} (2\pi\sigma_{t_{k+1}}^2)^{-d/2} \sum_{j=1}^{V} \widehat{f}_{k+1}(j) \exp\Big(-\frac{\|\widehat{x}_{t_{k+1}} - \alpha_{t_{k+1}} e_j\|_2^2}{2\sigma_{t_{k+1}}^2}\Big) \\
&= \frac{1}{\widehat{p}_k(\widehat{x}_{t_k})} (2\pi\sigma_{t_k}^2)^{-d/2} \Big(\frac{\sigma_{t_k}}{\sigma_{t_{k+1}}}\Big)^d \sum_{j=1}^{V} \widehat{f}_k(j) \exp\Big(-\frac{\|\widehat{x}_{t_k} - \alpha_{t_k} e_j\|_2^2}{2\sigma_{t_k}^2}\Big) \frac{\widehat{f}_{k+1}(j)}{\widehat{f}_k(j)} \exp\big(\Delta_{t_k}(e_j)\big)
\end{aligned}$$

$$
\begin{aligned}
&\overset{\text{(i)}}{\geq} \exp\big(-\widetilde{L}(t_{k+1}-t_k)\big)\frac{1}{\widehat{p}_k(\widehat{x}_{t_k})}(2\pi\sigma_{t_k}^2)^{-d/2}\Big(\frac{\sigma_{t_k}}{\sigma_{t_{k+1}}}\Big)^d \\
&\qquad \times \sum_{j=1}^{V}\widehat{f}_k(j)\exp\Big(-\frac{\|\widehat{x}_{t_k}-\alpha_{t_k}e_j\|_2^2}{2\sigma_{t_k}^2}\Big)\exp\big(\Delta_{t_k}(e_j)\big) \\
&\overset{\text{(ii)}}{=} \exp\big(-\widetilde{L}(t_{k+1}-t_k)\big)\Big(\frac{\sigma_{t_k}}{\sigma_{t_{k+1}}}\Big)^d\sum_{j=1}^{V}\widehat{w}_k(j\mid\widehat{x}_{t_k})\exp\big(\Delta_{t_k}(e_j)\big) \\
&\overset{\text{(iii)}}{\geq} \exp\big(-\widetilde{L}(t_{k+1}-t_k)\big)\Big(\frac{\sigma_{t_k}}{\sigma_{t_{k+1}}}\Big)^d\exp\Big(\sum_{j=1}^{V}\widehat{w}_k(j\mid\widehat{x}_{t_k})\Delta_{t_k}(e_j)\Big).
\end{aligned}
\tag{36}
$$

Here, $\Delta_{t_k}(e_j)$ records the change in the Gaussian exponent associated with the j-th token embedding:

$$
\begin{aligned}
\Delta_{t_k}(e_j) &:= \frac{\|\widehat{x}_{t_k}-\alpha_{t_k}e_j\|_2^2}{2\sigma_{t_k}^2}-\frac{\|\widehat{x}_{t_{k+1}}-\alpha_{t_{k+1}}e_j\|_2^2}{2\sigma_{t_{k+1}}^2} \\
&= \frac{1}{2}\Big(1-\frac{\alpha_{t_{k+1}}^2\sigma_{t_k}^2}{\alpha_{t_k}^2\sigma_{t_{k+1}}^2}\Big)\frac{\|\widehat{x}_{t_k}-\alpha_{t_k}e_j\|_2^2}{\sigma_{t_k}^2}-\frac{1}{2}\Big(1-\frac{\alpha_{t_{k+1}}^2\sigma_{t_k}^2}{\alpha_{t_k}^2\sigma_{t_{k+1}}^2}\Big)\frac{\|\widehat{x}_{t_k}-\alpha_{t_k}\widehat{\mu}_\theta(\widehat{x}_{t_k},t_k)\|_2^2}{\sigma_{t_k}^2},
\end{aligned}
\tag{37}
$$

where the second line follows from the update rule of $\widehat{x}_{t_k}$ given in (32) and the definition of $\widehat{\mu}_\theta(\widehat{x}_{t_k},t_k)$ from (31); (i) follows from the log-Lipschitz condition on $\widehat{f}_k$; (ii) follows from the identity

$$
\widehat{w}_k(j\mid\widehat{x}_{t_k})=\frac{\widehat{f}_k(j)\exp\Big(-\frac{\|\widehat{x}_{t_k}-\alpha_{t_k}e_j\|_2^2}{2\sigma_{t_k}^2}\Big)}{\sum_{j'=1}^{V}\widehat{f}_k(j')\exp\Big(-\frac{\|\widehat{x}_{t_k}-\alpha_{t_k}e_{j'}\|_2^2}{2\sigma_{t_k}^2}\Big)}=\frac{1}{\widehat{p}_k(\widehat{x}_{t_k})}(2\pi\sigma_{t_k}^2)^{-d/2}\widehat{f}_k(j)\exp\Big(-\frac{\|\widehat{x}_{t_k}-\alpha_{t_k}e_j\|_2^2}{2\sigma_{t_k}^2}\Big);
$$

(iii) is a consequence of Jensen's inequality $\mathbb{E}[e^X]\geq e^{\mathbb{E}[X]}$. By summary, we have

$$
\begin{aligned}
\sum_{j=1}^{V}\widehat{w}_k(j\mid\widehat{x}_{t_k})\Delta_{t_k}(e_j) &= \frac{1}{2}\Big(1-\frac{\alpha_{t_{k+1}}^2\sigma_{t_k}^2}{\alpha_{t_k}^2\sigma_{t_{k+1}}^2}\Big)\Bigg(\sum_{j=1}^{V}\widehat{w}_k(j\mid\widehat{x}_{t_k})\frac{\|\widehat{x}_{t_k}-\alpha_{t_k}e_j\|_2^2}{\sigma_{t_k}^2}-\frac{\|\widehat{x}_{t_k}-\alpha_{t_k}\widehat{\mu}_\theta(\widehat{x}_{t_k},t_k)\|_2^2}{\sigma_{t_k}^2}\Bigg) \\
&= \frac{\alpha_{t_k}^2}{2\sigma_{t_k}^2}\Big(1-\frac{\alpha_{t_{k+1}}^2\sigma_{t_k}^2}{\alpha_{t_k}^2\sigma_{t_{k+1}}^2}\Big)\Bigg(\sum_{j=1}^{V}\widehat{w}_k(j\mid\widehat{x}_{t_k})\|e_j\|_2^2-\|\widehat{\mu}_\theta(\widehat{x}_{t_k},t_k)\|_2^2\Bigg).
\end{aligned}
\tag{38}
$$

Substituting into (36) yields

$$
\begin{aligned}
&\frac{\widehat{p}_{k+1}(\widehat{x}_{t_{k+1}})}{\widehat{p}_k(\widehat{x}_{t_k})} \\
&\geq \exp\big(-\widetilde{L}(t_{k+1}-t_k)\big)\Big(\frac{\sigma_{t_k}}{\sigma_{t_{k+1}}}\Big)^d\exp\Bigg(\frac{\alpha_{t_k}^2}{2\sigma_{t_k}^2}\Big(1-\frac{\alpha_{t_{k+1}}^2\sigma_{t_k}^2}{\alpha_{t_k}^2\sigma_{t_{k+1}}^2}\Big)\Bigg(\sum_{j=1}^{V}\widehat{w}_k(j\mid\widehat{x}_{t_k})\|e_j\|_2^2-\|\widehat{\mu}_\theta(\widehat{x}_{t_k},t_k)\|_2^2\Bigg)\Bigg).
\end{aligned}
\tag{39}
$$

Step 3: Analysis of the distributions of $\widehat{x}_{t_k}$. We now track the density of the auxiliary sequence $\widehat{x}_{t_k}$. Because the update is deterministic, its one-step density ratio is determined by the inverse Jacobian determinant of the update map. From the update rule in (32), the change-of-variable formula yields the following relationship between the densities of $\widehat{x}_{t_k}$ and $\widehat{x}_{t_{k+1}}$:

$$
\begin{aligned}
\frac{q_{k+1}(\widehat{x}_{t_{k+1}})}{q_k(\widehat{x}_{t_k})} &= \Big|\frac{\sigma_{t_{k+1}}}{\sigma_{t_k}}I+\big(\alpha_{t_{k+1}}-\frac{\sigma_{t_{k+1}}\alpha_{t_k}}{\sigma_{t_k}}\big)\nabla_{\widehat{x}_{t_k}}\widehat{\mu}_\theta(\widehat{x}_{t_k},t_k)\Big|^{-1} \\
&= \Big(\frac{\sigma_{t_{k+1}}}{\sigma_{t_k}}\Big)^{-d}\Big|I-\alpha_{t_k}\Big(1-\frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)\nabla_{\widehat{x}_{t_k}}\widehat{\mu}_\theta(\widehat{x}_{t_k},t_k)\Big|^{-1}.
\end{aligned}
\tag{40}
$$

where $|\cdot|$ denotes the determinant. By the construction of the embedding-weighted predictor $\widehat{\mu}_\theta(\widehat{x}_{t_k}, t_k)$ in (31), we obtain

$$
\begin{aligned}
&\nabla_{\widehat{x}_{t_k}} \widehat{\mu}_\theta(\widehat{x}_{t_k}, t_k) \\
=&\sum_{j=1}^{V} e_j \big(\nabla_{\widehat{x}_{t_k}} \widehat{w}_k(j \mid \widehat{x}_{t_k})\big)^\top \\
=&-\frac{1}{\sigma_{t_k}^2}\sum_{j=1}^{V} \widehat{w}_k(j \mid \widehat{x}_{t_k}) e_j(\widehat{x}_{t_k} - \alpha_{t_k} e_j)^\top + \frac{1}{\sigma_{t_k}^2}\big(\sum_{j=1}^{V} \widehat{w}_k(j \mid \widehat{x}_{t_k}) e_j\big)\big(\sum_{j=1}^{V} \widehat{w}_k(j \mid \widehat{x}_{t_k})(\widehat{x}_{t_k} - \alpha_{t_k} e_j)\big)^\top \\
=&\frac{\alpha_{t_k}}{\sigma_{t_k}^2}\left(\sum_{j=1}^{V} \widehat{w}_k(j \mid \widehat{x}_{t_k}) e_j e_j^\top - \big(\sum_{j=1}^{V} \widehat{w}_k(j \mid \widehat{x}_{t_k}) e_j\big)\big(\sum_{j=1}^{V} \widehat{w}_k(j \mid \widehat{x}_{t_k}) e_j\big)^\top\right) \\
\overset{\text{(i)}}{=}&\frac{\alpha_{t_k}}{\sigma_{t_k}^2}\left(\sum_{j=1}^{V} \widehat{w}_k(j \mid \widehat{x}_{t_k}) e_j e_j^\top - \widehat{\mu}_\theta(\widehat{x}_{t_k}, t_k)\widehat{\mu}_\theta(\widehat{x}_{t_k}, t_k)^\top\right) \\
=:&\frac{\alpha_{t_k}}{\sigma_{t_k}^2}\widehat{\Sigma}_k(\widehat{x}_{t_k}),
\end{aligned}
$$

where (i) applies the definition of $\widehat{\mu}_\theta(\widehat{x}_{t_k}, t_k)$ in (31), and $\widehat{\Sigma}_k(\widehat{x}_{t_k})$ is the covariance matrix of discrete distribution on $\{e_j\}_{j\in[V]}$ associated with the probability $\mathbb{P}\{X = e_j\} = \widehat{w}_k(j \mid \widehat{x}_{t_k})$. This is the same quantity that appears in the reference-density calculation in Step 2; see (38).

Substituting this expression into (40) yields

$$
\begin{aligned}
\frac{q_{k+1}(\widehat{x}_{t_{k+1}})}{q_k(\widehat{x}_{t_k})} &= \Big(\frac{\sigma_{t_{k+1}}}{\sigma_{t_k}}\Big)^{-d}\left|I - \frac{\alpha_{t_k}^2}{\sigma_{t_k}^2}\Big(1 - \frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)\widehat{\Sigma}_k(\widehat{x}_{t_k})\right|^{-1} \\
&\overset{\text{(i)}}{=} \Big(\frac{\sigma_{t_{k+1}}}{\sigma_{t_k}}\Big)^{-d}\exp\left(\frac{\alpha_{t_k}^2}{\sigma_{t_k}^2}\Big(1 - \frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)\mathsf{Tr}(\widehat{\Sigma}_k(\widehat{x}_{t_k})) + \frac{\alpha_{t_k}^4}{\sigma_{t_k}^4}\Big(1 - \frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)^2 O\big(\|\widehat{\Sigma}_k(\widehat{x}_{t_k})\|_F^2\big)\right) \\
&= \Big(\frac{\sigma_{t_{k+1}}}{\sigma_{t_k}}\Big)^{-d}\exp\left(\frac{\alpha_{t_k}^2}{\sigma_{t_k}^2}\Big(1 - \frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)\Big(\sum_{j=1}^{V}\widehat{w}_k(j \mid \widehat{x}_{t_k})\|e_j\|_2^2 - \|\widehat{\mu}_\theta(\widehat{x}_{t_k}, t_k)\|_2^2\Big) + \mathcal{E}_{t_k}(\widehat{x}_{t_k}).\right)
\end{aligned}
\tag{41}
$$

Here the leading term in the right-hand-side of (i) is obtained from the first-order expansion of the log determinant, while the remainder collects the corresponding higher-order terms, and

$$
\mathcal{E}_{t_k}(\widehat{x}_{t_k}) := \frac{\alpha_{t_k}^4}{\sigma_{t_k}^4}\Big(1 - \frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)^2 O\big(\|\widehat{\Sigma}_k(\widehat{x}_{t_k})\|_F^2\big).
$$

Step 4: Combining the estimates. We finally compare the two density evolutions. Combining (39) and (41), and using the identity

$$
1 - \frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}} - \left(\frac{1}{2} - \frac{\alpha_{t_{k+1}}^2\sigma_{t_k}^2}{2\alpha_{t_k}^2\sigma_{t_{k+1}}^2}\right) = \frac{1}{2}\left(1 - \frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\right)^2,
$$

we obtain

$$
\begin{aligned}
&\frac{q_{k+1}(\widehat{x}_{t_{k+1}})\widehat{p}_k(\widehat{x}_{t_k})}{q_k(\widehat{x}_{t_k})\widehat{p}_{k+1}(\widehat{x}_{t_{k+1}})} \\
&\quad\le \exp\big(\widetilde{L}(t_{k+1} - t_k)\big)\exp\left(\frac{\alpha_{t_k}^2}{2\sigma_{t_k}^2}\Big(1 - \frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)^2\Big(\sum_{j=1}^{V}\widehat{w}_k(j \mid \widehat{x}_{t_k})\|e_j\|_2^2 - \|\widehat{\mu}_\theta(\widehat{x}_{t_k}, t_k)\|_2^2\Big) + \mathcal{E}_{t_k}(\widehat{x}_{t_k})\right) \\
&\quad\le \exp\big(\widetilde{L}(t_{k+1} - t_k)\big)\exp\left(\frac{\alpha_{t_k}^2}{2\sigma_{t_k}^2}\Big(1 - \frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)^2\sum_{j=1}^{V}\widehat{w}_k(j \mid \widehat{x}_{t_k})\|e_j\|_2^2 + \mathcal{E}_{t_k}(\widehat{x}_{t_k})\right).
\end{aligned}
$$

To control the right-hand side, recall that $\|e_j\|_2$ is bounded, i.e., $\|e_j\|_2 = B$. Consequently, telescoping the one-step density-ratio bounds gives

$$\begin{aligned}\frac{q_N(\widehat{x}_{t_{N-1}})}{\widehat{p}_N(\widehat{x}_{t_{N-1}})} &= \frac{q_0(\widehat{x}_{t_0})}{\widehat{p}_0(\widehat{x}_{t_0})}\prod_{k=0}^{N-2}\frac{q_{k+1}(\widehat{x}_{t_{k+1}})\widehat{p}_k(\widehat{x}_{t_k})}{q_k(\widehat{x}_{t_k})\widehat{p}_{k+1}(\widehat{x}_{t_{k+1}})} \\ &\le \frac{q_0(\widehat{x}_{t_0})}{\widehat{p}_0(\widehat{x}_{t_0})}\exp(\widetilde{L})\exp\Big(\sum_{k=0}^{N-2}\frac{\alpha_{t_k}^2}{2\sigma_{t_k}^2}\Big(1-\frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)^2B^2+\sum_{k=0}^{N-2}\mathcal{E}_{t_k}(\widehat{x}_{t_k})\Big),\end{aligned}$$

where we have used

$$\begin{aligned}&\sum_{k=0}^{N-2}\frac{\alpha_{t_k}^2}{2\sigma_{t_k}^2}\Big(1-\frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)^2\sum_{j=1}^{V}\widehat{w}_k(j\mid\widehat{x}_{t_k})\|e_j\|_2^2 \\ &=\frac{1}{2}\sum_{k=0}^{N-2}\Big(\frac{\alpha_{t_k}}{\sigma_{t_k}}-\frac{\alpha_{t_{k+1}}}{\sigma_{t_{k+1}}}\Big)^2B^2\lesssim N\delta^2,\end{aligned}$$

and

$$\begin{aligned}\sum_{k=0}^{N-2}\mathcal{E}_{t_k}(\widehat{x}_{t_k}) &\lesssim \sum_{k=0}^{N-2}\frac{\alpha_{t_k}^4}{\sigma_{t_k}^4}\Big(1-\frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)^2\mathsf{Tr}(\widehat{\Sigma}_k(\widehat{x}_{t_k}))^2 \\ &\lesssim \sum_{k=0}^{N-2}\frac{\alpha_{t_k}^4}{\sigma_{t_k}^4}\Big(1-\frac{\alpha_{t_{k+1}}\sigma_{t_k}}{\alpha_{t_k}\sigma_{t_{k+1}}}\Big)^2\mathsf{Tr}\Big(\sum_{j=1}^{V}\widehat{w}_k(j\mid\widehat{x}_{t_k})\|e_j\|_2^2\Big)^2 \\ &=\sum_{k=0}^{N-2}\frac{\alpha_{t_k}^2}{\sigma_{t_k}^2}\Big(\frac{\alpha_{t_k}}{\sigma_{t_k}}-\frac{\alpha_{t_{k+1}}}{\sigma_{t_{k+1}}}\Big)^2B^4\lesssim N\delta^2B^4,\end{aligned}$$

where the last inequality uses the facts that

$$\begin{aligned}\big|\frac{\alpha_{t_k}}{\sigma_{t_k}}-\frac{\alpha_{t_{k+1}}}{\sigma_{t_{k+1}}}\big|&\le\delta,\\ \frac{\alpha_{t_k}}{\sigma_{t_k}}\big|\frac{\alpha_{t_k}}{\sigma_{t_k}}-\frac{\alpha_{t_{k+1}}}{\sigma_{t_{k+1}}}\big|&\le\delta.\end{aligned}\tag{42}$$

It remains to control the initial density ratio. Observe that the initial time step satisfies

$$\frac{q_0(\widehat{x}_{t_0})}{\widehat{p}_0(\widehat{x}_{t_0})}=\sigma_{t_0}^d\frac{1}{\sum_{j=1}^{V}\widehat{f}_k(j)\exp\big(-\frac{\alpha_{t_0}^2\|e_j\|_2^2}{2\sigma_{t_0}^2}+\frac{\alpha_{t_0}e_j^\top\widehat{x}_{t_0}}{2\sigma_{t_0}^2}-\frac{1-\sigma_{t_0}^2}{2\sigma_{t_0}^2}\|\widehat{x}_{t_0}\|_2^2\big)}\to 1\quad\text{as }\ t_0\to 0.$$

Under the stated discretization condition, the accumulated remainder remains bounded. In particular,for sufficiently small δ, we have $\delta^2N=O(1)$. Consequently, as $t_0\to 0$,

$$\frac{q_N(\widehat{x}_{t_{N-1}})}{\widehat{p}_N(\widehat{x}_{t_{N-1}})}=O(1).$$

Finally, recall that

$$\widehat{p}_N(\widehat{x}_{t_{N-1}})=(2\pi\sigma_{t_{N-1}}^2)^{-d/2}\sum_{j=1}^{V}\widehat{w}_N(e_j)\exp\big(-\|\widehat{x}_{t_{N-1}}-\alpha_{t_{N-1}}e_j\|_2^2/(2\sigma_{t_{N-1}}^2)\big),$$

which implies that

$$q_N(\widehat{x}_{t_{N-1}})\lesssim(2\pi\sigma_{t_{N-1}}^2)^{-d/2}\sum_{j=1}^{V}\widehat{w}_N(e_j)\exp\big(-\|\widehat{x}_{t_{N-1}}-\alpha_{t_{N-1}}e_j\|_2^2/(2\sigma_{t_{N-1}}^2)\big).$$

As $t_{N-1}\to 1$, the variance of every Gaussian component vanishes and its center approaches a vocabulary embedding. Therefore, q_N is asymptotically bounded by $\sum_{j=1}^{V}\widehat{w}_N(e_j)\delta_{e_j}$, where δ_{e_j} denotes the Dirac measure with mass at e_j. This completes the proof.

D.2 Proof of Proposition 1

Define

$$p^{(i)}(j \mid x_t^{(-i)}, t) \coloneqq \mathbb{P}\{s^{(i)} = j \mid x_t^{(-i)}\}, \quad j \in [V]. \tag{43}$$

By Bayes's rule, we can derive

$$\mathbb{P}\{s^{(i)} = j \mid x_t\} = \frac{\mathbb{P}\{s^{(i)} = j, x_t^{(i)} \mid x_t^{(-i)}\}}{\mathbb{P}\{x_t^{(i)} \mid x_t^{(-i)}\}} = \frac{p^{(-i)}(j \mid x_t^{(-i)}, t) p(x_t^{(i)} \mid s^{(i)} = j, x_t^{(-i)})}{\sum_{j' \in [V]} p^{(-i)}(j' \mid x_t^{(-i)}, t) p(x_t^{(i)} \mid s^{(i)} = j', x_t^{(-i)})}. \tag{44}$$

Because the Gaussian corruption is independent across token positions (see the probability path construction in (1)), $x_t^{(i)}$ and $x_t^{(-i)}$ are conditionally independent given $s^{(i)}$. As a result, we have

$$x_t^{(i)} \mid s^{(i)} = j, x_t^{(-i)} \sim \mathcal{N}(\alpha_t e_j, \sigma_t^2 I_d).$$

Plugging this into (44) yields the claim in (14).

D.3 Proof of Proposition 2

Consider a general schedule (α_t, σ_t) such that $\alpha_t \to 1$ and $\sigma_t \to 0$ as $t \to 1$. Define the smooth, unconstrained data predictor

$$\mu_\theta(x, t) = \frac{x}{\alpha_t + \sigma_t}. \tag{45}$$

It is easy to verify that

$$\mu_\theta(x_t, t) = \frac{x_t}{\alpha_t + \sigma_t} = \frac{\alpha_t x_\star + \sigma_t z}{\alpha_t + \sigma_t} \to x_\star \quad \text{as} \quad t \to 1.$$

Thus, the data predictor is asymptotically accurate. However, we will show that the flow induced by this data predictor does not converge to any token embedding with constant probability.

By the ODE in (2) and the velocity identity in (8), the flow induced by μ_θ is given by

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = \frac{\sigma_t'}{\sigma_t} x_t + \Big(\alpha_t' - \frac{\sigma_t'}{\sigma_t}\alpha_t\Big) \frac{x_t}{\alpha_t + \sigma_t} = \frac{\alpha_t' + \sigma_t'}{\alpha_t + \sigma_t} x_t.$$

Solving the ODE shows that the flow trajectory initialized at x_{t_0} evolves according to

$$x_t = \frac{\alpha_t + \sigma_t}{\alpha_{t_0} + \sigma_{t_0}} x_{t_0}.$$

As a result, the data predictor remains constant along the flow trajectory:

$$\mu_\theta(x_t, t) = \frac{x_t}{\alpha_t + \sigma_t} = \frac{x_{t_0}}{\alpha_{t_0} + \sigma_{t_0}}.$$

Suppose that $x_{t_0} = \sigma z$ with $z_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. Then the final iterate is given by

$$\frac{x_{t_{N-1}}}{\alpha_{t_{N-1}}} = \sigma_{N-1} z \qquad \text{with} \qquad \sigma_{N-1} \coloneqq \frac{\sigma(\alpha_{t_{N-1}} + \sigma_{t_{N-1}})}{\alpha_{t_{N-1}}(\alpha_{t_0} + \sigma_{t_0})}.$$

Because $\sigma_t / \alpha_t \to 0$ as $t \to 1$, we have

$$\sigma_{N-1} \downarrow \sigma_\infty \coloneqq \frac{\sigma}{\alpha_{t_0} + \sigma_{t_0}} > 0 \quad \text{as} \quad N \to \infty.$$

Because $\sigma_\infty z$ has a continuous distribution, its probability of belonging to any finite set is zero. Therefore, the final iterate $x_{t_{N-1}}$ fails to converge to any token embedding with constant probability, as formalized in (18).

For example, if $\|e_j\|_2 = \sqrt{d}$ for every $j \in [V]$, then for any token position $i \in [L]$, one has

$$\begin{aligned}
\mathbb{P}\Big\{\min_j \big\|\alpha_{t_{N-1}}^{-1} x_{t_{N-1}}^{(i)} - e_j\big\|_2 \geq 1\Big\} &\geq \mathbb{P}\Big\{\big|\sigma_N \|z^{(i)}\|_2 - \|e_j\|_2\big| \geq 1\Big\} \\
&\geq \mathbb{P}\Big\{\sigma_\infty \|z^{(i)}\|_2 \geq 1 + \sqrt{d}\Big\} \geq c_{\mathsf{lb}}
\end{aligned}$$

for some constant $c_{\mathsf{lb}} > 0$ independent of N.

D.4 Proof of Proposition 3

Let us denote by $\overline{e}_j \coloneqq e_j/\|e_j\|_2$ the normalized embedding vector. Fix an arbitrary token position $i \in [L]$. Notice the relation

$$\begin{aligned}\frac{\|\mu_\theta^{(i)}(x_t,t)\|_2^2}{\|e_j\|_2^2} &= \Big\| \sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t)\overline{e}_{j'} \Big\|_2^2 \\ &= \sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t)^2 + \sum_{j'=1}^{V}\sum_{\ell\neq j'} w_\theta^{(i)}(j' \,|\, x_t,t) w_\theta^{(i)}(\ell \,|\, x_t,t) \frac{\langle e_{j'}, e_\ell\rangle}{\|e_{j'}\|_2\|e_\ell\|_2}.\end{aligned}$$

By the separation condition that $\max_{j'\neq\ell} \big|\frac{\langle e_{j'}, e_\ell\rangle}{\|e_{j'}\|_2\|e_\ell\|_2}\big| \leq 1-\rho$, one can derive

$$\begin{aligned}\frac{\|\mu_\theta^{(i)}(x_t,t)\|_2^2}{\|e_j\|_2^2} &\leq \sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t)^2 + (1-\rho)\sum_{j'=1}^{V}\sum_{\ell\neq j'} w_\theta^{(i)}(j' \,|\, x_t,t) w_\theta^{(i)}(\ell \,|\, x_t,t) \\ &\overset{\text{(i)}}{=} \sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t)^2 + (1-\rho)\sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t)\big(1 - w_\theta^{(i)}(j' \,|\, x_t,t)\big) \\ &\overset{\text{(ii)}}{=} \sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t)^2 + (1-\rho) - (1-\rho)\sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t)^2 \\ &= 1 - \rho + \rho \sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t)^2, \end{aligned} \tag{46}$$

where (i) and (ii) use the fact that $\sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t) = 1$. As $\mu_\theta^{(i)}(x_t,t) \to e_j$, one has $\frac{\|\mu_\theta^{(i)}(x_t,t)\|_2^2}{\|e_j\|_2^2} \to 1$. Combining this with (46), we obtain

$$1 - \rho + \rho \lim_{t\to 1} \sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t)^2 \geq 1,$$

which leads to

$$\lim_{t\to 1} \sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t)^2 \geq 1.$$

As $w_\theta^{(i)}(j' \,|\, x_t,t) \in [0,1]$ and $\sum_{j'=1}^{V} w_\theta^{(i)}(j' \,|\, x_t,t) = 1$, there exists some $\widehat{j} \in [V]$ such as $w_\theta^{(i)}(\widehat{j} \,|\, x_t,t) \to 1$ and $w_\theta^{(i)}(j' \,|\, x_t,t) \to 0$ for $j' \neq \widehat{j}$. Since $\mu_\theta^{(i)}(x_t,t) \to e_j$, we conclude that $\widehat{j} = j$, thereby completing the proof.

E Additional experimental results

This section provides the additional experimental results in this paper. Table 2 summarizes Gen. PPL, entropy, and model size for ConvergeFlow and the baseline models on OWT dataset. Figure 4 compares the original LangFlow schedule with the adaptive schedule.

E.1 Control of quality-diversity trade-off

Empirically, we find that sampling guidance is most effective near the data endpoint. We therefore adopt time-adaptive variants of self-conditioning guidance, unconditional guidance, and iterative self-conditioning refinement.

Specifically, at each step i, we replace the constant guidance strengths w_{scg} and w_{ug} with $w_{\mathsf{scg}}/(1+\sigma_{t_i}/\alpha_{t_i})$ and $w_{\mathsf{ug}}/(1+\sigma_{t_i}/\alpha_{t_i})$, respectively. Thus, these schedules gradually increase the guidance strength over the sampling trajectory.

Table 2: Comparison of Gen. PPL and entropy across different models. For ELF and LangFlow, we report the Gen. PPL at the point whose entropy is closest to that of the dataset; their complete Gen. PPL-entropy curves are shown in Figure 1. For ConvergeFlow, we report the Gen. PPL at the dataset entropy using the complete results in Figure 1 and Table 10. Results and model sizes marked with ‡ are taken from Duo. At the dataset entropy of 5.44, our method achieves a Gen. PPL of 34.19, whereas the lowest Gen. PPL among the continuous flow-based LMs is approximately 60, even though these models are evaluated at entropies below the dataset entropy.

Model	Gen. PPL (↓)	Entropy (↑)	Model Size
Autoregressive			
Transformer‡	35.90	5.58	170M
Discrete DLMs			
MDLM‡ (Sahoo et al., 2024)	104.85	5.63	170M
Duo‡ (Sahoo et al., 2025)	77.69	5.55	170M
SEDD Uniform‡ (Lou et al., 2023)	99.90	5.56	170M
SEDD Absorb‡ (Lou et al., 2023)	105.03	5.62	170M
Continuous DLMs			
FLM (Lee et al., 2026)	62.23	5.33	179M
LangFlow (Chen et al., 2026)	60.09	5.43	130M
ELF (Hu et al., 2026)	65.30	5.40	105M
ConvergeFlow	**34.19**	5.44	130M
Dataset	-	5.44	-

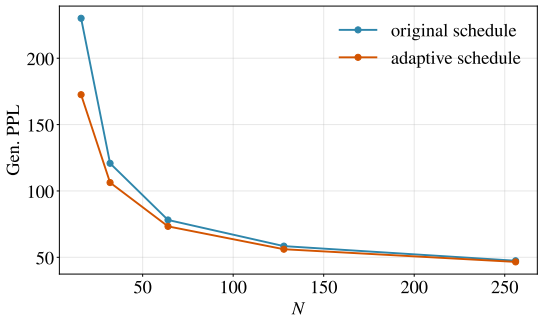


Figure 4: Effect of the sampling schedule on Gen. PPL. We compare the original schedule by LangFlow with the proposed adaptive schedule $t_i = (i+0.5)/N$, $i = 0, \ldots, N-1$, across different steps N.

For iterative self-conditioning refinement, we analogously adapt the refinement count according to

$$K_i = \left\lceil \frac{K_{\mathsf{iscr}}}{1 + \sigma_{t_i}/\alpha_{t_i}} \right\rceil.$$

To ensure a fair comparison under a fixed computational budget, we choose the number of sampling steps separately for each configuration. In particular, the NFE for iterative self-conditioning refinement is given by

$$\mathsf{NFE} = \sum_{i=0}^{N-1} (K_i + 1).$$

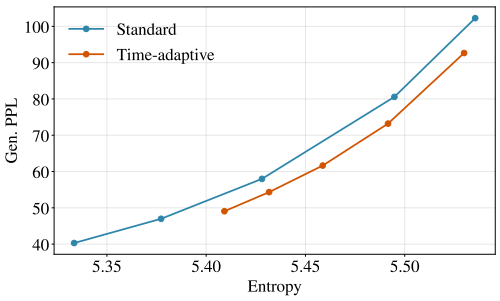
All three sampling mechanisms provide effective control over the quality-diversity trade-off. Figures 5(a)–5(c) present the results for self-conditioning guidance, unconditional guidance, and iterative self-conditioning refinement, respectively. Figure 5(d) compares their time-adaptive variants. Each trade-off curve is obtained by varying the corresponding nominal control parameter, namely w_{scg}, w_{ug}, or K_{iscr}, while holding the total NFE fixed. The complete results for NFE=64 and NFE=128 are summarized in Tables 3 and 4, respectively.

Table 3: Gen. PPL-entropy trade-offs produced by the three sampling techniques under a fixed NFE budget of 64. We compare each standard variant with its time-adaptive counterpart. For the time-adaptive variants, "Nominal value" denotes the control parameter before time-dependent scaling.

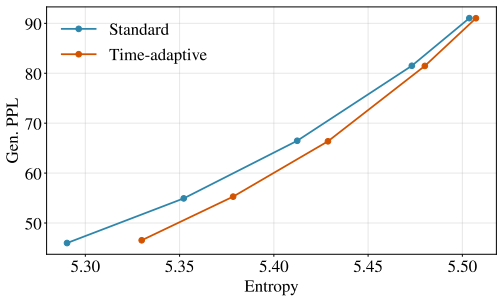
Sampling technique	**Standard**			**Time-adaptive**		
	Value	Gen. PPL	Entropy	Nominal value	Gen. PPL	Entropy
Iterative self-conditioning refinement	1	102.2591	5.5355	10	97.8570	5.5328
	2	84.5884	5.5101	15	89.2844	5.5250
	3	77.0637	5.5011	20	84.1505	5.5179
	5	72.3546	5.4962	40	77.7179	5.5125
Self-conditioning guidance	1	102.2591	5.5355	20	92.6444	5.5299
	2	80.5658	5.4948	40	73.1988	5.4916
	4	57.9800	5.4281	60	61.6401	5.4587
	6	46.9794	5.3773	80	54.3244	5.4317
	8	40.2985	5.3335	100	49.0598	5.4092
Unconditional guidance	0.01	91.0414	5.5037	0.25	91.0238	5.5072
	0.02	81.5008	5.4732	0.50	81.4699	5.4801
	0.04	66.4769	5.4124	1.00	66.3674	5.4287
	0.06	54.9319	5.3522	1.50	55.2844	5.3784
	0.08	45.9883	5.2903	2.00	46.5392	5.3299

Table 4: Gen. PPL-entropy trade-offs produced by the three sampling techniques under a fixed NFE budget of 128. We compare each standard variant with its time-adaptive counterpart. For the time-adaptive variants, "Nominal value" denotes the control parameter before time-dependent scaling.

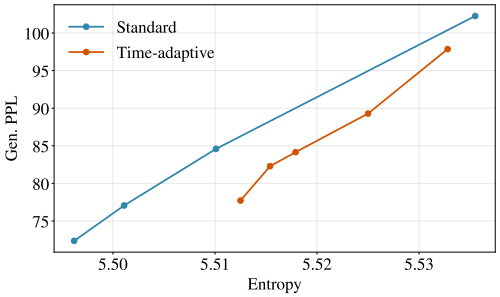
Sampling technique	**Standard**			**Time-adaptive**		
	Value	Gen. PPL	Entropy	Nominal value	Gen. PPL	Entropy
Iterative self-conditioning refinement	1	92.6728	5.5033	10	88.6072	5.5008
	2	73.3312	5.4652	15	79.5148	5.4880
	3	63.8502	5.4438	20	73.9816	5.4772
	5	56.0827	5.4274	40	64.0461	5.4560
Self-conditioning guidance	1	92.6728	5.5033	20	83.8231	5.4971
	2	72.0048	5.4574	40	65.2554	5.4543
	4	50.8108	5.3834	60	54.5742	5.4175
	6	40.5837	5.3211	80	47.8842	5.3891
	8	34.5459	5.2721	100	43.0442	5.3622
Unconditional guidance	0.01	81.5110	5.4671	0.25	82.0885	5.4739
	0.02	71.9972	5.4310	0.50	73.3846	5.4456
	0.04	57.1605	5.3610	1.00	59.3169	5.3910
	0.06	46.0725	5.2840	1.50	49.2200	5.3389
	0.08	38.1870	5.2097	2.00	41.3859	5.2858



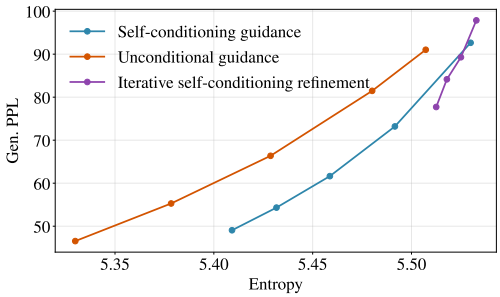
(a) Standard and time-adaptive self-conditioning guidance.



(b) Standard and time-adaptive unconditional guidance.



(c) Standard and time-adaptive iterative self-conditioning refinement.



(d) Comparison of the three time-adaptive samplers.

Figure 5: Gen. PPL-entropy trade-offs of the three sampling techniques at NFE=64. Each curve is obtained by varying the corresponding nominal control parameter. (a)–(c) compare the standard and time-adaptive variants of each sampling technique, where the adaptive variants increase the guidance strength or self-conditioning refinement count toward the data endpoint. (d) compares the time-adaptive variants of the three sampling techniques.

Overall, the time-adaptive variants generally improve the Gen. PPL-entropy frontier relative to their constant-strength counterparts. As shown in Figure 5(d), among the three adaptive techniques, self-conditioning guidance spans the broadest range of operating points and achieves the most favorable trade-off in the low- and medium-entropy regimes. Iterative self-conditioning refinement is particularly effective in the high-entropy regime, although it covers a narrower controllable range. Unconditional guidance yields only modest Gen. PPL reductions, and we investigate this further in Section E.3.

E.2 Combination of three sampling techniques

We next test the performance of combinations of the three sampling techniques, where we use the time-adaptive variant in all cases.

We first combine self-conditioning guidance and iterative self-conditioning refinement. Figure 6 reports the results under a fixed NFE budget of 64. The complete results for NFE=64 and NFE=128 are provided in Tables 5 and 6, respectively.

As illustrated in Figure 6, increasing K_{iscr} generally shifts the trade-off frontier toward lower Gen. PPL at comparable entropy levels. Thus, combining a sufficiently large self-conditioning refinement parameter K_{iscr} with an appropriate self-conditioning guidance strength w_{scg} substantially improves the trade-off.

We next add time-adaptive unconditional guidance to the combined sampler. Figure 7 plots the results, with $K_{\mathsf{iscr}} = 50$ and $K_{\mathsf{iscr}} = 100$ shown separately in Figures 7(a) and 7(b), respectively. For both settings, varying w_{ug} primarily moves the operating point along a similar Gen. PPL-entropy frontier rather than shifting the frontier outward. These results suggest that self-conditioning guidance and iterative self-conditioning refinement capture most of the gain. Nevertheless, unconditional

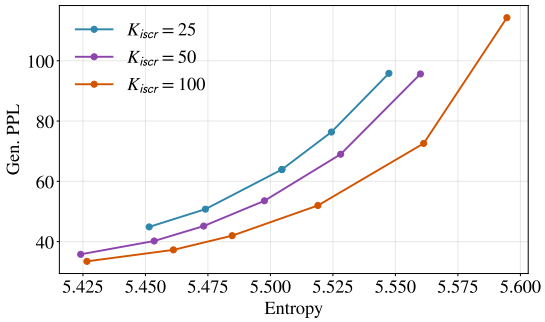


Figure 6: Joint effect of iterative self-conditioning refinement and self-conditioning guidance at NFE=64. Each curve fixes the refinement parameter K_{iscr} and varies the guidance strength w_{scg}.

Table 5: Joint effect of iterative self-conditioning refinement and self-conditioning guidance under NFE $= 64$.

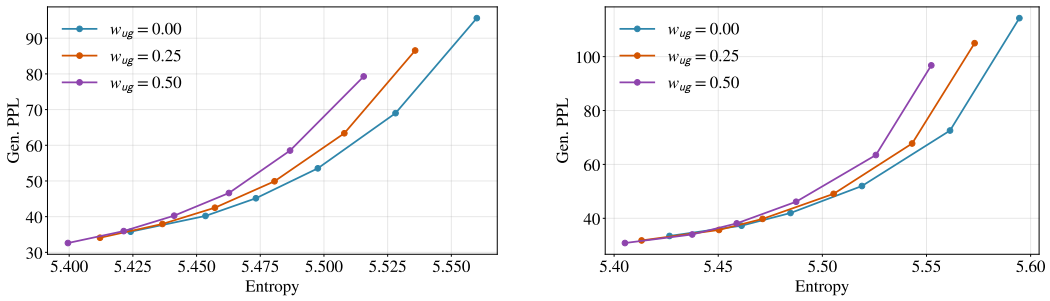
		Self-conditioning guidance strength w_{scg}					
Refinement parameter K_{iscr}	Metric	10	15	20	25	30	40
25	Gen. PPL	95.8253	76.3501	63.9167	63.9167	50.7672	44.8755
	Entropy	5.5474	5.5244	5.5046	5.5046	5.4740	5.4515
50	Gen. PPL	95.6214	68.9955	53.5390	45.1553	40.2005	35.7850
	Entropy	5.5600	5.5281	5.4976	5.4733	5.4535	5.4241
100	Gen. PPL	114.2815	72.5750	51.9846	41.9660	37.2542	33.4498
	Entropy	5.5946	5.5613	5.5190	5.4847	5.4612	5.4266

Table 6: Joint effect of iterative self-conditioning refinement and self-conditioning guidance under NFE $= 128$.

		Self-conditioning guidance strength w_{scg}					
Refinement parameter K_{iscr}	Metric	10	15	20	25	30	40
50	Gen. PPL	78.8759	56.7015	44.7796	38.0792	34.0202	29.5631
	Entropy	5.5121	5.4713	5.4402	5.4141	5.3910	5.3522
100	Gen. PPL	83.9935	53.9882	39.7684	32.6640	28.5357	25.3549
	Entropy	5.5356	5.4838	5.4388	5.4009	5.3733	5.3286
200	Gen. PPL	100.3790	58.5191	40.3036	32.1147	28.0896	25.2520
	Entropy	5.5758	5.5212	5.4657	5.4207	5.3884	5.3401

guidance remains useful because it does not require a conditioning signal, which we believe has independent interest. The complete numerical results for different combinations of w_{ug}, K_{iscr}, and w_{scg} are reported in Table 7.

Finally, we compare the two token prediction rules described in Section 3.2. As the flow approaches the data endpoint, these two rules should output the same token. Figure 8 empirically verifies this agreement, with the NFE$= 64$ and NFE$= 128$ results shown in Figures 8(a) and 8(b), respectively. The corresponding Gen. PPL-entropy curves nearly overlap under both computational budgets. In the high-entropy, high-Gen. PPL regime, distance-based decoding yields a slightly more favorable trade-off. These results support using nearest-neighbor projection to map the generated continuous flow state to a token sequence, without a separately trained terminal decoder.



(a) K_{iscr}=50 (b) K_{iscr}=100

Figure 7: Effect of adding unconditional guidance to a sampler combining iterative self-conditioning refinement and self-conditioning guidance at NFE $= 64$.

Table 7: Joint effect of unconditional guidance, iterative self-conditioning refinement, and self-conditioning guidance under NFE$= 64$.

			Self-conditioning guidance strength w_{scg}					
w_{ug}	K_{iscr}	Metric	10	15	20	25	30	40
0.0	50	Gen. PPL	95.6214	68.9955	53.5390	45.1553	40.2005	35.7850
		Entropy	5.5600	5.5281	5.4976	5.4733	5.4535	5.4241
	100	Gen. PPL	114.2815	72.5750	51.9846	41.9660	37.2542	33.4498
		Entropy	5.5946	5.5613	5.5190	5.4847	5.4612	5.4266
0.25	50	Gen. PPL	86.5587	63.3631	49.9138	42.5065	37.9509	34.0961
		Entropy	5.5358	5.5080	5.4806	5.4572	5.4366	5.4121
	100	Gen. PPL	104.9930	67.7606	49.0741	39.7811	35.6832	31.7834
		Entropy	5.5731	5.5431	5.5054	5.4713	5.4504	5.4132
0.5	50	Gen. PPL	79.2823	58.5215	46.6228	40.2759	35.9516	32.6090
		Entropy	5.5156	5.4867	5.4627	5.4412	5.4214	5.3995
	100	Gen. PPL	96.7562	63.4840	46.1475	38.1169	33.9966	30.8225
		Entropy	5.5523	5.5257	5.4874	5.4589	5.4375	5.4052

E.3 Further Improvement

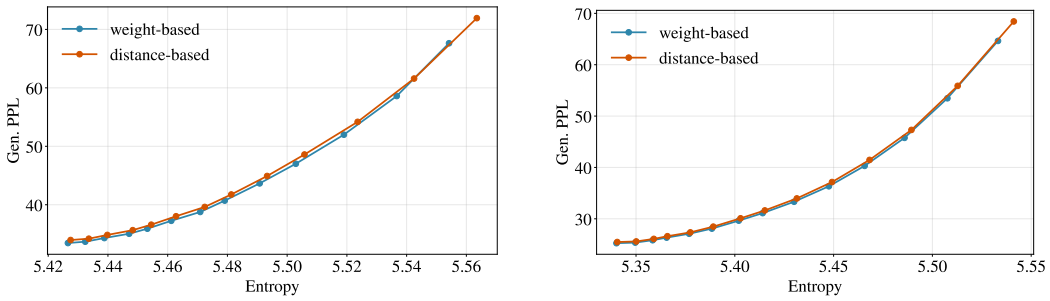
The experiments in Section E.1 indicate that concentrating sampling guidance near the data endpoint improves the Gen. PPL-entropy trade-off. At sufficiently large guidance strengths, however, data endpoint-focused allocation exhibits diminishing returns. Further reduction in Gen. PPL requires stronger guidance earlier in the trajectory, when the state remains relatively noisy.

To study this effect, we compare the following three configurations under the same NFE budget:

- Configuration A: fixed refinement count $K_i = K$ and $w_{\text{scg}}(t_i) = w_{\text{scg}}$;
- Configuration B: $K_i = \lceil K_{\text{iscr}}/(1 + \sqrt{\sigma_{t_i}/\alpha_{t_i}}) \rceil$ and $w_{\text{scg}}(t_i) = w_{\text{scg}}/(1 + \sqrt{\sigma_{t_i}/\alpha_{t_i}})$;
- Configuration C: $K_i = \lceil K_{\text{iscr}}/(1 + \sigma_{t_i}/\alpha_{t_i}) \rceil$ and $w_{\text{scg}}(t_i) = w_{\text{scg}}/(1 + \sigma_{t_i}/\alpha_{t_i})$.

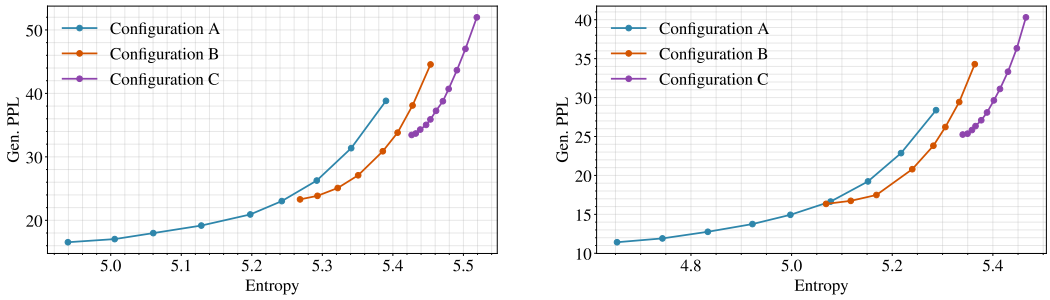
Relative to Configuration C, Configuration B applies stronger guidance in the high-noise regime.

Figure 9 presents the results for NFE=64 and 128. The three configurations are favorable in different regions of the quality-diversity frontier. Configuration A, particularly with $K = 8$, reaches the low-entropy, low-Gen. PPL regime. Configuration B provides strong intermediate operating points by applying more guidance during the noisier portion of the trajectory, whereas Configuration C preserves the greatest diversity by concentrating guidance near the endpoint.



(a) NFE=64 (b) NFE=128

Figure 8: Comparison of two token prediction rules for NFE budgets of 64 and 128.



(a) NFE=64 (b) NFE=128

Figure 9: Comparison of the three configurations for NFE budgets of 64 and 128. Each curve is obtained by varying the self-conditioning guidance strength w_{scg}.

In regions where the curves overlap, increasing the refinement count generally improves the trade-off. Among the settings considered, $K_{\text{iscr}} = 100$ and $K_{\text{iscr}} = 200$ yield the most favorable frontiers for NFE=64 and 128, respectively. These results show that both the refinement count and its temporal allocation determine the attainable Gen. PPL-entropy regime.

Complete numerical results for Configurations A and B are reported in Tables 8 and 9, respectively. Results for additional guidance configurations under NFE=64 and 128 are provided in Tables 10 and 11.

E.4 Detailed results for guidance allocation

Tables 8 and 9 report the complete sweeps for Configurations A and B, respectively, under both NFE budgets. Tables 10 and 11 then collect representative operating points from all three configurations for NFE= 64 and 128. The numerical results support the trends in Figure 9: Configuration A reaches the low-entropy, low-Gen. PPL regime, Configuration B provides strong intermediate operating points, and Configuration C preserves the greatest entropy. In overlapping regions, a larger refinement parameter K_{iscr} generally yields a more favorable trade-off, confirming that both the refinement count and its temporal allocation are important.

E.5 Qualitative Samples

We present qualitative generation samples from ConvergeFlow under NFE=64. All samplesare generated with a fixed sequence length of 1024 tokens.

Table 8: Results for Configuration A that uses a constant refinement count K and self-conditioning guidance strength w_{scg}.

NFE	K	Metric	Self-conditioning guidance strength w_{scg} 1.5	2	2.5	3	3.5
64	4	Gen. PPL	44.9274	31.8150	25.8833	22.4534	20.6955
		Entropy	5.4087	5.3335	5.2784	5.2223	5.1797
	6	Gen. PPL	41.9674	29.7222	24.0008	20.7902	19.1477
		Entropy	5.4047	5.3255	5.2605	5.1896	5.1216
	8	Gen. PPL	38.8262	26.2650	20.9220	17.9837	16.5453
		Entropy	5.3902	5.2921	5.1978	5.0603	4.9393
128	4	Gen. PPL	35.3425	24.8707	19.8753	16.9287	15.1857
		Entropy	5.3267	5.2298	5.1423	5.0654	5.0131
	6	Gen. PPL	30.2807	21.1930	17.0699	14.5470	13.2466
		Entropy	5.3001	5.1882	5.0965	4.9866	4.8690
	8	Gen. PPL	28.3841	19.2306	14.9421	12.7615	11.4202
		Entropy	5.2871	5.1518	4.9976	4.8335	4.6531

Table 9: Results for Configuration B, in which both the refinement count and self-conditioning guidance strength are scaled by $1/(1+\sqrt{\sigma_{t_i}/\alpha_{t_i}})$.

NFE	K_{iscr}	Metric	Self-conditioning guidance strength w_{scg} 8	10	12	16
64	10	Gen. PPL	50.6284	42.7216	37.8040	32.9574
		Entropy	5.4563	5.4311	5.4064	5.3774
	15	Gen. PPL	43.5503	36.2495	31.5968	27.6530
		Entropy	5.4372	5.4020	5.3701	5.3268
	20	Gen. PPL	40.2613	32.6712	28.7003	24.9890
		Entropy	5.4306	5.3901	5.3582	5.3057
	25	Gen. PPL	38.0965	30.8788	27.0954	23.8566
		Entropy	5.4279	5.3857	5.3508	5.2931
128	10	Gen. PPL	42.4745	35.5080	31.2866	27.2081
		Entropy	5.3957	5.3640	5.3365	5.3003
	20	Gen. PPL	35.7925	29.6742	22.1093	19.1334
		Entropy	5.3661	5.3275	5.2526	5.1836
	30	Gen. PPL	31.5364	25.5134	19.5748	17.0443
		Entropy	5.3424	5.2963	5.2182	5.1391
	40	Gen. PPL	29.4243	23.8153	20.7996	16.7360
		Entropy	5.3331	5.2818	5.2398	5.1176

Table 10: Detailed Gen. PPL and entropy results for Configurations A–C under a fixed NFE budget of 64. Within each configuration, the columns vary the nominal self-conditioning guidance strength w_{scg}.

Configuration	Metric	w_{scg}								
A		1.5	1.75	2	2.25	2.5	2.75	3	3.25	3.5
	Gen. PPL	38.8262	31.3686	26.2650	23.0270	20.9220	19.1722	17.9837	17.0393	16.5453
	Entropy	5.3902	5.3409	5.2921	5.2425	5.1978	5.1284	5.0603	5.0056	4.9393
B		7	8	9	10	12	14	16	18	
	Gen. PPL	44.5625	38.0965	33.8150	30.8788	27.0954	25.0884	23.8566	23.2972	
	Entropy	5.4534	5.4279	5.4067	5.3857	5.3508	5.3218	5.2931	5.2684	
C		20	22	24	26	28	30	32	34	36	38	40
	Gen. PPL	51.9846	47.0125	43.6558	40.7043	38.7662	37.2542	35.9016	35.0390	34.3009	33.6783	33.4498
	Entropy	5.5190	5.5029	5.4908	5.4791	5.4709	5.4612	5.4532	5.4471	5.4388	5.4324	5.4266

Table 11: Detailed Gen. PPL and entropy results for Configurations A–C under a fixed NFE budget of 128. Within each configuration, the columns vary the nominal self-conditioning guidance strength w_{scg}.

Configuration	Metric	w_{scg}								
A		1.5	1.75	2	2.25	2.5	2.75	3	3.25	3.5
	Gen. PPL	28.3841	22.8663	19.2306	16.6408	14.9421	13.7528	12.7615	11.9109	11.4202
	Entropy	5.2871	5.2175	5.1518	5.0775	4.9976	4.9222	4.8335	4.7433	4.6531
B		7	8	9	10	12	14	16	18	
	Gen. PPL	34.2873	29.4243	26.2162	23.8153	20.7996	17.4883	16.7360	16.3467	
	Entropy	5.3640	5.3331	5.3056	5.2818	5.2398	5.1686	5.1176	5.0685	
C		20	22	24	26	28	30	32	34	36	38	40
	Gen. PPL	40.3036	36.3322	33.3216	31.1079	29.6387	28.0896	27.0878	26.3363	25.8291	25.3638	25.2520
	Entropy	5.4657	5.4476	5.4300	5.4141	5.4020	5.3884	5.3769	5.3655	5.3584	5.3496	5.3401

ConvergeFlow

NFE: 64; **Gen. PPL:** 34.19; **Entropy:** 5.44

Primarily, it also applies to lions, kangaroo, trophy-tailed dogs, and so on, while it applies to cows, dogs and chickens, raccours, leopards, wolves, and so on, and many other phenomena. Humanism allows for many different kinds of negative acts toward humans. Without these negative acts, nonhumanists have no reason to believe that their own actions are coming from humans, even if they share their true beliefs, values, and so forth. Eugenicists, in stark contrast, have no reason to take anyone's actions, even if they have society's moral authority behind them. It is the responsibilities of humanists to decide what is right for humans and what is best for society in these decisions.

Humanism is a statement of nature's nature, a critique of nature's culture, history, and so forth. It is essential for human people to consistently understand this one way or another, and obviously we will have incredible difficulty in hearing them. Vegetarianism does not view nature as a complicated activity that seeks to set the boundaries to our laws as we approach them toward one another. But its system of veganism is particularly harmful because it can intentionally distort nature's boundaries and it is demeaning. Bulled animal abuse is so ubiquitous as it has little official use among animals. The failure to apply it, instead of treating it as a weapon, gives a grave eye to ethical weakness. This is why the literature showing that animal abuse in animals is widespread, well-documented, and widespread.

Animalism is especially dangerous because it is often used as an excuse to stir a criticism of nature's boundaries, but almost always by literally calling for animals that can be captured and domesticated. This incapacity to ignore it, instead of treating it as a deterrent, puts a temporary end to ethical weakness. Whether humans aim to "strictly exploit the natural environment" or "suppestly eat livestock," nonhumanists often claim that their laws are incompatible with their moral duties against humans and those we want, and even make the case that society ought to work with them just to make their lives better.

The key to improving the ethical treatment of victims of animal animal abuse is the lack of empathy by individuals who care nothing about their behavior as they understand their attitudes toward it. They cannot even talk about the boundaries or the acts that justify one's moral duty towards animals. The most precise example used in this article is when lions and tigers were allegedly gunned down an amusement park in Zimbabwe.

Reason and Peace: Nonhumanists

Phumanists are not necessarily bad people. They often celebrate a state's constitutional right to a man to keep arms, but not a state's constitutional right to bear arms. For some, the loss of such a powerful fellow is an extremely useful ethical choice given that it encaps in just a few of the same core principles and principles that historically entangled them with the wide range of moral issues. Nonhumanists generally seek to accept the "crime" of aggression with the right to be taken away from others, even though they distinguish between humans and all humans, rather than by acting on the application of moral relativism. Even when using the words "sex bornmanamally," an abdication of a bogeyman or slave capitalist, or a woman's legal right to have children, or the ability to have children and them, these are thoroughly debunked.

Nonhumanists often seek to understand the "naturalization" of all humans, but how they seek to justify it in order to acknowledge the absolutism of nature. For example, they object to the term "theetamic Earth," and talk about keeping rabbits, gorillas or chimpanzees living in open spaces and engaging in other forms of transhumanization. Nonhumanists do not accept the purely contestal moral choice of a human being, if they harm others with their own choices, even when pushing for creating laws or regulations that clearly violate our laws.

It is important to understand that physical violence—even intentional and reprehensible conduct—is central to our ethical practice. All acts of actions are religiously defined by these principles, and are adjudicated with a quick response to everyone who wants them. Nonhumanists areently act on animals without their direct legal responsibility, and rely exclusively on animals to try to kill human beings lest they flaunt their rights.

Legal Considerations

You can find an authoritative list of dozens of cases that are central to our ethical practice. However, these are generally the laws that should be discussed in practice.
Human beings are well-organized in performing their moral duties towards us and should not be subject to any use by those with moral authority or responsibility. If you believe in truly equal liberty and equality, accept the fact that those with good moral responsibility are being only

ConvergeFlow

NFE: 64; **Gen. PPL:** 34.19; **Entropy:** 5.44

Thunder Creek State Park is located just just a few miles away from a home that Michael Mevala and his wife are renovating.
About a year ago, Michael Ranarekala and his wife reached out to their families that they were looking to find a riverfront riverfront in Lannie for about $10.
"We actually thought it would be really nice that we could put it back to a really good place," Ranarekala said.
But Ranarekala had to wait several years to figure that out. The couple would end up providing him with more than $5 million in mortgage payments, and in the process, he took the money off the drain.
"You're about to pay your bills again," Mevala said
The two friends thought they were going to work on a similar project, instead of a loan, but it wasn't.
"Once the loan came out, they told us, 'Just take a little energy and a little money,' and they'll buy your house on the hale and charge you mortgage payment, and one of our brothers said, 'Literally, it's going to be great.'"
But as the couple worked their hardest to get together, Mevala didn't know the house was coming on without Ranarekala's help. Shortly after Christmas last year, the two got out of the Toyota Camry and got a call from a sales man from Indiana.
"He told me that we were shelling out $1,200,000, and we were putting out $10,000 out of replacement stuff."
"Is there the cash key?" Mevala asked.
"We'll pay for them," Mevala said.
They got together on the same project for a few months. And, y there, it worked.
"We just got it on going," Mevala said.
Shortly after he moved out of Lannie, Ranarekala ran up onto Lannie Avenue and stopped an officer in search for drugs. The officer who had been in trouble then pulled up and robbed him more than $5 million from his wallet.
Unsurprisingly, he's not the county's best friend.
"Michael is a great guy," Mevala said. "He was Ranarekala, we lived in Lannie for about 20 years and he came up with something that was really good. That just made him great for us."
Now, the family is living at a different, nicer house, and Ranarekala plans to be in the arse in a few years.
Since 2013, Ranarekala has moved almost $24,000 to Lannie (including rent and fees), but he didn't know anything about the long-term mortgage payments.
"I think it would have been a lot cheaper for the mortgage payment," said Ranarekala, who grew up in the local town. "We want to always have some food on the table. We want to have some food. The only thing we do is to just drive through town, buy coffee to the beach, play basketball, have some jobs, and not having to fight through the up, you can get a little farther out of the city."
Sleling, Mevala said, was the next step in cutting money off the three neighbors.
"They were thinking that some (property) mortgage payments would just go down," he said. "And now they're going to have to work on that."
And finally, if it's working, Ranarekala, who is planning to hold a fundraiser for Philly delak food next week at the Thunder Creek State Park, has the same message for the community: He needs a deal.

Eventually, Mevala expects to move on with it.
”This is not where I came from,” he said. ”The people this town like to be accepted as good neighbors. I think it’s a good one.”
You can follow him on Twitter @benvala.
Read or Share this story: http://nyne.ws/2/2SVXeI2¡—endoftext—¿Making common care for soups is a learning experience. Peppersticks can be a very popular choice for soups. You will want to create some of your most popular delicacies, usually by eating “dairy” chopsticks and topping them with your cold food. However, you need to mix it in so that you can enjoy the homemade roast. The advantage of the device is that it will stand out from the traditional chopsticks by putting a bit of the water into the end of your sink, preventing it from falling into a pan or within. Thus, a simple homemade roast is perfect and for a dish that will ease the effects of various spooky ingredients. Don’t hesitate to look for local food!

ConvergeFlow

NFE: 64; **Gen. PPL:** 34.19; **Entropy:** 5.44

Tramp is configuring my bitcoin keys.
I was initially thinking about my bitcoin keys, and wanted to focus on security. But with Roshal and LeeEn, it will begin to work out much better. Although I am still communicating with my bitcoin keys, I will greatly appreciate a bit. Second, as with the new Dostash service my bitcoin wallet allows me to use more efficient transactions.¡—endoftext—¿New scientific research suggests that different parts of the brain may contribute to the understanding of what happens between humans and fruit animals.
In research published at Frontiers in the Neuron (2014), researchers Maswuki Soto and Masihy Man Shayashi have shown that differences occur in relatively different parts of the brain, and suggest that it might lead to a biological change in the way the brain moves from time to day.
While it is unclear whether these differences arise from the nervous system, the differences in different parts of the brain could also be linked to gaming behaviours (such as constructing a few animals or catching stones or cups). The team believes that studying the differences within the brain will possibly lead to the understanding of real-world relationships.
Dr Peter Michael Flint, Professor of Cognitive Science and Intelligence from Cambridge’s Institute of Psychology, said, “Our work with Frontiers in Genetics and Neuron shows that the differences underlying a person’s personality are not only linked to gaming, but also how the brain system responds to different parts of the brain. This is a one piece of the puzzle, but there are many bigger hypotheses.”
The team recruited 242 people to carry out studies related to brain regions, and analysed the differences in brain cells. They also found that 90% of the difference were related as a result of genetic changes in the cells. To determine the number of genetic mutations in each cell, each volunteer was used to study how the human brain move from one cell to another.
By analyzing the differences within the brain, Dr Soto and her colleagues put the differences between two sides on the match level after brain imaging. “While each side had their own specific brain safety issues, one side felt that a gold ring was more more than necessary, while the other felt that handling was stronger than normal. So when both sides had to work rather than save they had to get more bad ups and shps.”
Other experiments carried out will help better understand the behavioural variations in the brain. Soto added, “To understand how the laws of selection and cooperation work, we can understand why they are so important.”
To ascertain how the differences were related, the researchers conducted a series of different cognitive studies studying brain activity patterns, including interaction between blood and stimulation through fruit flies. Dr Flint told the BBC, “Central to human image, brain movement is an important source of information, since it is used to make decisions about the state of a person’s person. For example, fruit mice might have to take different degrees of brain activity to get only one portion of the brain to work properly, but another might need another

brain hole to walk around in the sand. Then, all the time, brain movement analysis shows that animals suffer to nervous system changes."
Based on laboratory studies of brain movement in traditional brain injuries, the researchers said, "If there were injuries, its brain activity would change over time. The results show that, since this procedure occurred to animals, the relationship between different parts of the brain as a result of the movements is significantly greater."
When it came to the link between sports and biological tissue, researchers had no idea what such differences may have been, compared with other mammals. Dr Flint continued, "The only model that consistently suggests that the nervous system might be linked to gaming are human users as well as other apes."
The researchers conclude that their work has the importance of understanding the relation between gaming and natural biology, social interactions and brain activity. What remains unclear, though, is that this differences are related to other animals, but not competitors.¡—endoftext—¿The Donald Trump administration will create America's clean energy industry by keeping carbon dioxide emissions back in the United States by 2030, the Department of Energy announced on Tuesday.
The US government said in a statement that it would cut coal-fired power capacity by 15 percent by 2030 and cut down on its installed energy use by 1.2 percent by 2025.
The president's decision comes shortly after US President Barack Obama reaffirmed the government's clean energy programme, but stopped short of stepping up efforts on climate change. The US joined last year to join the Paris accord, but is still part of a pledge to cut global greenhouse gas emissions by a quarter by 2025 — something Paris has put back since 2005.
"The US should be able to contribute as much as usual by leading the environment," said Greg Welles of Natural Resources Defense Institute.
The first lady

ConvergeFlow

NFE: 64; **Gen. PPL:** 34.19; **Entropy:** 5.44

"That's, Ryona. What should happen to you right now?"
He held up the bag to his hands as he pointed to Tanishi.
"I want to have a personal message Father. It's so proud to find you the best friend and a child in the world. I've said this to you, but thank you for courting and treating us as well."
Tanishi held his father with a lot. "I do not take much thought or credit what you have done. I have not made the choice of not letting the Demon walk off out of the shadows."

"Father, how are you, Ryona? How terrible are you to us?"
He sighed. "What the hell is it? Just to take lives for people who are too afraid to work hard in order to survive in a world like ours?"
Suddenly Ryona smiled at him, his jaw narrowed as all of them were staring at him and smirking at him.
"I will send your son back, and that is what you have said."
Tanishi took up his tray from the man's shirt as he sat out right behind him. He held up the book as he stood, wrapping up the rest of the book in his hand and a bright colored one on the top of his desk next to the fingernails on his desk.
"Father, did you make it up for us then? What do you know? You are poor. You are rich. You pay a few dollars of taxes every month. You are almost half of your income! How did you come up with for anything else in this world?"
His parents were frantically mocking him as he cried. "There would never be any more rich than this. He was yours. What would you say if he was not punished, and why would he be a huge fine?"
Tanishi was confused as the kneeling man leaned his hand into the table.

"Come on. He was yours. He was just a kid. I was trying to raise the money to bring him back to his country. I was trying to figure out how he would become the world's richest person. Years ago. He was not punished. He wouldn't deserve to die. What was the reason why I left him in life?"
Tanishi thenplaced his hand through the table, covering himself in hair as he slammed his hand from the table.
"Now what about us? What do you think about your son? Well, I have heard of all of you. You went to high school. You saw the people that helped him move. You mourned him when you got into the school and you didn't believe him when he went to school. You were just one of them, but you ended up leaving him after he entered school. How often you ate? I'm sure he ate more than two times."
Pfeo's hands pained his chest as he cyclifled through all of the images on his phone.
He turned back in the living room, drawably, and stood up to it. "I would love to have you back here. I find you very smart and well-formed, someone who would be nice to me. Thank you very much."
Tanishi shook his head. "You know what I think, Ryona. When I tell you the truth, I know you shouldn't deserve to die." He smiled teasingly.
It was hard to look down his parents' faces as he traced them to his phone before placing them in front of his screen on the left screen. He sighed when he turned off the screen to see them hinkily right next to the TV.
All in all, Ryona stepped up. "Come on! You have been the Demon here for so long. It seems so clear to me. You are smart and perfectly determined and you certainly have nothing to lose. Please praise me like this."
"Father."
Ryona squeezed his throat.
Tanishi turned around as Ryona said his hand.
"Thank you. Father. We'll look at it in real time."
"Thank you."
Tanishi sat back with blank plates. The man then turned on Ryona, looking down at Tanishi's eyes, and raised his hand to stand in front of them.

Pfeo grinned as Ryona leaned over Tanishi's shoulder. "There was nothing wrong

ConvergeFlow

NFE: 64; **Gen. PPL:** 34.19; **Entropy:** 5.44

Liberty Foundation, a coalition that includes 25 US government officials and five civil rights organizations, is planning a partnership between free speech and libertarian thought.
The gathering site, one of the world's largest institutions, is Pope Francis and Betsy DeVos, joint secretary of state, set to deliver his invitation-style meeting on April 16 at Duke University of America's Heart Memorial Center.
Francis and her gathered audience will be joined by the cause "All-Free Education USA," a grassroots grassroots coalition of students from socio-economic backgrounds who are denied the freedom to speak and exercise their religious beliefs. These groups have long advocated free speech in education, sports, health care, and the practice of prayer.
"Federal government protects the right of religion for the right to worship and in church and state in public," the group says. "Candidates and candidates as the Republican Party's nominee have recently condemned the violence in Charlottesville and unleashed what the group calls a dangerous assault on our democracy and public schools. This coalition will further build on offering bold and constructive political solutions in the 21st century."
Perry's guest names are Sam Robertson, a vice president at Liberty and George Orwell, the author of a best 2006 book called Freedom for Patialism.
"We are seeing huge changes in U.S. and around the world. We're seeing a lot of change in many countries," Perry says. "Most people don't have anything to worry about. We're not

seeing any sort of change. We're in a position where you're having people who don't have access to Twitter or Facebook, and we're seeing tremendous growth. In other parts of the world, we're seeing a lot of resistance to free speech—and it's onon the fringes."

Power is increasingly increasing across the country and more college campuses each year after Donald Trump's election.

"People choose to express their views in public and private schools in a way that violates the First Amendment," says Steve Graff, an activist and the president of American Civil Liberties Union in West Rutherford, N.J. "We see a greater number of people who choose to express their views in a more consistent manner."

Clorence McCarthy, the executive director of the liberal advocacy group AFL-CIO, says, "Federal law doesn't discriminate against the distribution of ideas. Restricting out of the distribution of ideas is unconstitutional. We're not banning anything that works. We want to protect people's rights and stop bad speech."

McCarthy says the religious speech rights movement is akin to a criminal justice ring, where they come back with a confession to freely express their opinions—and there's no chance for their trial. Ari Poole, director of the Center for Public Policy at Liberty University, says citizens who do free speech speech will likely having to go to a trial and would have to face losing their civil rights rights.

"They may be advocating something they don't like," Poole says. "This is the repeal of democratic liberalism. This tool of police has to be called on it in order to get their freedom. We are seeing that en masse all the time."

After the end of the 30-minute meeting, White House Counsel John Gibbs will schedule brief talks with business and religious leaders at several venues. Bob Gallock, president and communications director in Washington, D.C., is teaming with Karen Earnest, the president in Washington, D.C., this week. Tom Lohouse, president of the Center for Freedom and Heritage Liberty, is be the person behind the event.

"We're still building a group. At the moment, there are 30 to 30 people at the White House," Gallock says. "We just have to move it together."

It will be the first substantive meeting between the two religious leaders. Perry and DeVos will meet at the Fuller Grove Baptist Church around 12 a.m. Saturday at the Deene United Methodist Church. The meeting is not finalized yet, Gallock says, "but we have a lot of time to burn off there—and we definitely feel like we're in direct hands."

UNOCIALS GIVEEPAT AN SOCIAL OUR FACEBOOK

Once the meeting ends, three religious leaders will show down in different venues in standing up to their example. "There will be a pre- to tea breakfast, and we'll have a cocktail party here at Liberty University," Perry says. "And then there will be good business on college campuses. We'll have a conversation about how the law and universities should be governed."

Gallock and his group of 40 prominent individual