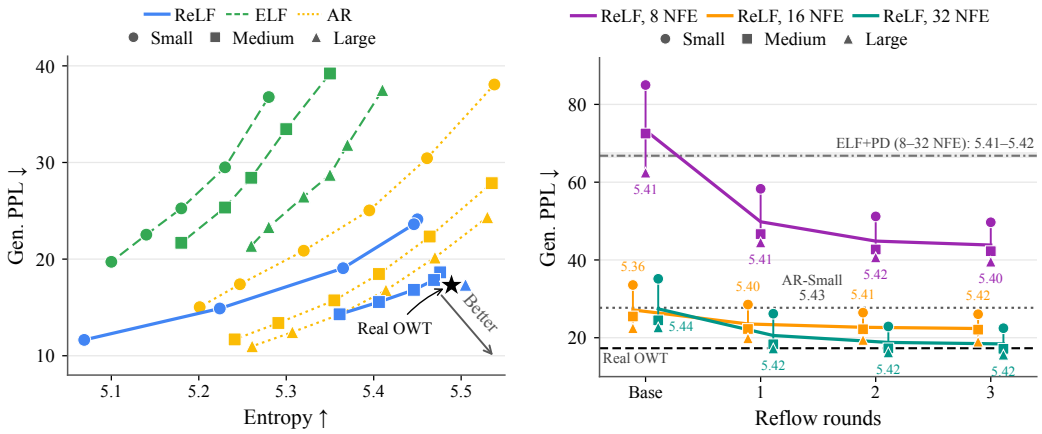
RECTIFIED LANGUAGE FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous flow language models had lagged behind discrete-state diffusion models. They face challenges absent from discrete models: linking a continuous velocity field to a categorical vocabulary, committing to a single token as noise vanishes, and choosing the embedding geometry in which denoising takes place. We present ***Re**ctified **L**anguage **F**low* (ReLF), a rectified-flow language model trained with cross-entropy. We introduce *singular logits*, which analytically capture the divergent component of token posteriors, leaving the network a bounded residual and guaranteeing convergence to valid token embeddings. The flow runs in a compact, context-informed embedding space, and reflow straightens it for few-step generation. On OpenWebText with 64 function evaluations, ReLF attains lower generative perplexity than autoregressive baselines at every model size tested under matched diversity.



(a) 64 NFE **(b) Reflow rounds**

Figure 1: **Unconditional generation on OpenWebText. (a)** Quality–diversity trade-off at 64 function evaluations (NFE): ReLF outperforms ELF (Hu et al., 2026) and autoregressive (AR) baselines at every model size. **(b)** Gen. PPL versus reflow rounds at 8, 16, and 32 NFE; lines connect the means over model sizes, and labels give the entropy. At matched entropy, ReLF after reflow outperforms ELF+PD at 8–32 NFE and AR-Small (1,023 NFE) from 16 NFE.

1 INTRODUCTION

Flow and diffusion models have emerged as a major paradigm alongside autoregressive (AR) modeling, with strong results in image, audio, and video generation (Wu et al., 2025a; Evans et al., 2025; Polyak et al., 2024). For language, a central question is whether the flow should be defined in a *continuous* or *discrete state space*. Continuous-state formulations inherit standard flow techniques (Liu et al., 2022; Lipman et al., 2022) and showed early promise for text generation (Li et al., 2022; Gulrajani & Hashimoto, 2023), but have since been overshadowed by the success of discrete-state approaches, particularly in large-scale diffusion language modeling (Sahoo et al., 2024; Nie et al., 2025; Song et al., 2025).

Continuous flows remain appealing for language: all positions evolve jointly along a continuous trajectory, and image-flow tools such as ODE/SDE samplers, guidance, and trajectory straightening

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.

apply directly. Realizing this potential, however, requires careful design considerations that do not arise in AR or discrete diffusion models. The training objective must connect a continuous velocity field to a categorical vocabulary; the token posterior must commit to a single token as noise vanishes; and the token embeddings, rather than serving only as network inputs, define the geometry in which corruption and denoising take place. Prior continuous-state models address these decisions through complex designs, such as auxiliary objectives (Li et al., 2022; Gulrajani & Hashimoto, 2023), encoder–decoder networks (Lovelace et al., 2023), or vocabulary-sized simplex representations (Han et al., 2023; Lee et al., 2026), yet their generation quality still trails that of AR models at comparable sample diversity (Figure 1(a)).

This work aims to substantially push the empirical limits of continuous flow language models. We present ***Re**ctified **L**anguage **F**low* (ReLF), a rectified-flow language model that generates token sequences along continuous embedding trajectories from Gaussian noise. Following earlier continuous-state models (Dieleman et al., 2022; Han et al., 2023; Eijkelboom et al., 2024), ReLF is trained with cross-entropy on positionwise token posteriors, from which the velocity follows as the posterior-weighted mean of the token embeddings. On OpenWebText generation with 64 function evaluations, ReLF-Small reaches a generative perplexity of 24.1 at a unigram entropy of 5.45 nats, outperforming a similarly sized AR baseline (27.7 at 5.43). At larger scale, ReLF-Large reaches 17.3 at 5.51, matching real OpenWebText text (17.3 at 5.49). Trajectory straightening also transfers: after reflow (Liu et al., 2022), ReLF-Medium reaches 42.7 at 5.42 with only 8 function evaluations, compared with 66.4 at 5.42 for ELF+PD, a progressively distilled Embedded Language Flows (ELF) model (Hu et al., 2026).

We attribute this performance to several design choices. As noise vanishes, token posteriors approach point masses, which would require unbounded logit differences that a network cannot output. We introduce *singular logits*, which capture this divergence analytically through a reference posterior and leave the network a bounded residual, so that the modeled posterior is guaranteed to commit to a one-hot distribution and the flow trajectory to converge to a valid token embedding (Section 3.1). The flow itself runs in a compact, context-informed embedding space: a fixed 32-dimensional space, shared across model scales and far narrower than the Transformer width, whose frequency-weighted Hellinger–SVD embeddings place contextually similar tokens nearby, so that noisy states retain context before commitment (Section 3.2). Finally, we adapt reflow to ReLF as *regularized reflow*: each round retrains the model on its own noise–token couplings and then interpolates its weights back toward the base model, straightening the flow for few-step generation while preventing drift from the base model's posteriors (Section 3.3).

2 ReLF for Discrete Generation

Setup. Let $\boldsymbol{w} = [w^{(1)}, \dots, w^{(L)}] \in \mathcal{V}^L$ be a token sequence drawn from the data distribution $\mathcal{D}$, where $\mathcal{V} = \{1, \dots, V\}$ is the vocabulary. Let $E : \mathcal{V} \to \mathbb{R}^d$ map each token to its embedding; we call $\mathbb{R}^d$ the *flow embedding space*, since generation evolves in it, and d the *flow embedding dimension*. Applying E positionwise yields the embedded sequence $X_1 = E(\boldsymbol{w}) \in \mathbb{R}^{d \times L}$, whose distribution is denoted by q_1. Below, we suppress position indices, reserving X_t for training interpolations and Z_t for inference trajectories. Token probabilities such as $p_\theta(w \mid Z_t, t)$ refer to a representative position but condition on the full noisy sequence; positionwise embedding operations use the corresponding state at that position. We write the data and reference token posteriors as $q_t(w \mid Z_t)$ and $r_t(w \mid Z_t)$. Appendix A gives the fully indexed definitions and theory.

Rectified Flow. Following the rectified flow (RF) framework of Liu et al. (2022), we interpolate between data embeddings $X_1 \sim q_1$ and Gaussian noise $X_0 \sim q_0 = \mathcal{N}(0, I)$ as $X_t = tX_1 + (1 - t)X_0$ for $t \in [0, 1]$. In standard RF, the model learns a velocity field $\boldsymbol{v}_\theta(X, t)$ by minimizing the mean-squared error (MSE) loss:

$$\mathcal{L}(\theta) = \mathbb{E}\big[\|X_1 - X_0 - \boldsymbol{v}_\theta(X_t, t)\|^2\big] ,$$

where the expectation is taken over (X_0, X_1) and an independent random t from a time distribution. The associated flow ODE, initialized at $Z_0 \sim q_0$, evolves as

$$\frac{\mathrm{d}Z_t}{\mathrm{d}t} = \boldsymbol{v}_\theta(Z_t, t), \qquad \boldsymbol{v}_\theta(Z_t, t) \approx \boldsymbol{v}^*(Z_t, t) = \mathbb{E}[X_1 - X_0 \mid X_t = Z_t], \tag{1}$$

so that $\mathrm{Law}(Z_t) \approx \mathrm{Law}(X_t)$ for $t \in [0, 1]$.

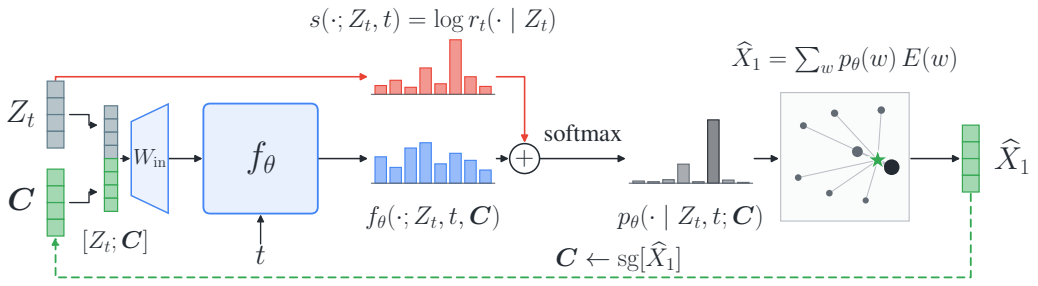


Figure 2: **ReLF architecture.** From the noisy state Z_t and self-conditioning input $\boldsymbol{C}$, the Transformer outputs residual logits f_θ, added to singular logits s. $\widehat{X}_1$ is a probability-weighted mean of token embeddings, fed back via stop-gradient sg as the next $\boldsymbol{C}$ (dashed).

Cross-Entropy Loss Training. For language models over discrete vocabularies, cross-entropy (CE) is a more natural training objective than MSE (Dieleman et al., 2022; Han et al., 2023; Eijkelboom et al., 2024). RF can be trained with CE by expressing the velocity field through conditional token marginals, which CE learns directly. The optimal velocity is determined by the conditional mean of the clean embedding (Appendix A.1):

$$\boldsymbol{v}^*(X_t, t) = \tfrac{1}{1-t}\big(\mathbb{E}[X_1 \mid X_t] - X_t\big), \qquad \mathbb{E}[X_1 \mid X_t] = \sum_{w \in \mathcal{V}} E(w)\, q_t(w \mid X_t). \tag{2}$$

Thus, estimating the velocity field requires only the positionwise posterior marginals $q_t(w \mid X_t)$, which a bidirectional diffusion Transformer can predict jointly from the full noisy sequence. We denote these predictions by $p_\theta(w \mid X_t, t)$ and train with cross-entropy to make $p_\theta \approx q_t$:

$$\mathcal{L}(\theta) = \mathbb{E}_{\boldsymbol{w} \sim \mathcal{D},\, X_0 \sim q_0,\, t}[-\log p_\theta(w \mid X_t, t)]\,. \tag{3}$$

The expectation also averages over token positions, with w denoting the corresponding clean target token. The model estimates the clean state as $\widehat{X}_1(Z_t, t) = \sum_{w \in \mathcal{V}} E(w) p_\theta(w \mid Z_t, t)$.

Sampling follows standard continuous flows except at the final step. Starting from $Z_{t_0} \sim q_0$, we integrate the flow ODE with Euler steps on a grid $0 = t_0 < t_1 < \cdots < t_{N-1} < 1$:

$$Z_{t_{i+1}} = Z_{t_i} + (t_{i+1} - t_i)\tfrac{1}{1-t_i}\big(\widehat{X}_1(Z_{t_i}, t_i) - Z_{t_i}\big). \tag{4}$$

At the last grid point, we decode each position as $\widehat{w} = \arg\max_{w \in \mathcal{V}} p_\theta(w \mid Z_{t_{N-1}}, t_{N-1})$. We also support SDE sampling with the score-corrected formulation of Ma et al. (2024), whose noise scale $\sigma \geq 0$ controls stochasticity; $\sigma = 0$ recovers the flow ODE (Appendix B.4).

Self-Conditioning. We use *self-conditioning* (Chen et al., 2023; 2026): the network concatenates its noisy input with a conditioning input $\boldsymbol{C}$ (Figure 2), set during training to $\text{sg}[\widehat{X}_1(X_t, t; \mathbf{0})]$ with probability $1/2$ and to $\mathbf{0}$ otherwise; at inference, $\boldsymbol{C}_0 = \mathbf{0}$ and $\boldsymbol{C}_{i+1} = \widehat{X}_{1,i}$, requiring no additional function evaluations. Following ELF (Hu et al., 2026), we further adopt training-time guidance: the network is conditioned on the self-conditioning guidance scale and trained to output guided predictions directly, so that guided sampling uses one evaluation per step. Unlike ELF, which applies the guidance shift to an MSE velocity target, we apply it to the token logits and retain the single CE objective (Appendix B.3).

3 Singular Logits and Flow Shaping

Section 2 fixes the training objective and sampler but leaves open how tokens commit as noise vanishes and in which embedding space the flow evolves. This section addresses both and then adapts reflow to ReLF: Section 3.1 introduces singular logits, which guarantee convergence to token embeddings; Section 3.2 constructs compact, context-informed flow embeddings; and Section 3.3 straightens the resulting flow for few-step generation.

3.1 SINGULAR LOGITS ENABLE ENDPOINT CONVERGENCE

Motivation. Near $t = 1$, as noise vanishes, the posterior must approach a one-hot distribution. Under softmax, this requires unbounded logit differences, which a continuous network with bounded inputs and parameters cannot represent exactly. Residual mass on competing tokens then makes the predicted clean state an embedding mixture, which can keep the ODE trajectory from reaching a token embedding and cause decoding errors (Figure 3(a,c)); prior continuous models instead map final states to tokens by rounding or a separate decoder (Li et al., 2022; Hu et al., 2026). We observe that the exact posterior already has the required structure: its logits decompose into a singular component that diverges as noise vanishes and a bounded component that carries contextual token preferences. The singular component is available in closed form for a simple reference distribution r_1 over sequences whose posterior r_t under the same corruption is tractable.

Singular-Logit Correction. We compute the log reference posterior, the *singular logits* $s(w; Z_t, t) = \log r_t(w \mid Z_t)$, analytically, and let the network parameterize only a residual correction f_θ toward the data posterior:

$$p_\theta(\cdot \mid Z_t, t) = \mathrm{softmax}(s(\cdot; Z_t, t) + f_\theta(\cdot; Z_t, t)) \,. \tag{5}$$

The exact residual is bounded whenever the reference stays within a constant factor of the data distribution.

Theorem 3.1. *Suppose $\kappa^{-1} \le q_1(\boldsymbol{w})/r_1(\boldsymbol{w}) \le \kappa$ for every sequence $\boldsymbol{w}$ and some $1 \le \kappa < \infty$. Then the exact residual $f_*(w; Z_t, t) = \log q_t(w \mid Z_t) - s(w; Z_t, t)$, which f_θ approximates, satisfies $|f_*(w; Z_t, t)| \le 2 \log \kappa$ for all $t < 1$.*

With a full-support reference such as the uniform distribution, the ratio condition is equivalent to $q_1(\boldsymbol{w}) > 0$ for every sequence, since L and $\mathcal{V}$ are finite. This is the same condition AR models need for exact representation with finite softmax logits. The resulting residual bound is uniform in t, so the network need not represent the divergence caused by vanishing noise. For efficient computation, a factorized reference reduces the reference posterior to a vocabulary sum at each position.

Lemma 3.2. *Suppose the reference distribution factorizes as $r_1(\boldsymbol{w}) = \prod_{\ell=1}^{L} r_1^{(\ell)}(w^{(\ell)})$, where $r_1^{(\ell)}$ is the reference token prior at position ℓ, and corruption is independent across positions. Then the singular logits can also be computed positionwise as*

$$s(w; Z_t, t) = \log \frac{r_1(w) q_{t|1}(Z_t \mid w)}{\sum_{u \in \mathcal{V}} r_1(u) q_{t|1}(Z_t \mid u)}.$$

Here $r_1(w)$ is the corresponding factor $r_1^{(\ell)}(w)$. For a uniform reference, this becomes

$$s(w; Z_t, t) = \log q_{t|1}(Z_t \mid w) - \log \sum_{u \in \mathcal{V}} q_{t|1}(Z_t \mid u).$$

For standard Gaussian noise and a factorized uniform reference, the singular logits are

$$s(w; Z_t, t) = -\frac{\|Z_t - tE(w)\|^2}{2(1-t)^2} - \log \sum_{u \in \mathcal{V}} \exp\left(-\frac{\|Z_t - tE(u)\|^2}{2(1-t)^2} \right). \tag{6}$$

Factorization applies only to the reference; the network still conditions on the full sequence. Appendix A.2 gives the proofs and the Gaussian specialization.

In *probability space*, token commitment means that the posterior converges to a one-hot distribution. This holds for any uniformly bounded residual. The following proposition shows that this implies token commitment in *flow embedding space*: the ODE trajectory converges to the selected token embedding.

Proposition 3.3. *Consider the ODE in Eq. (1), using modeled probabilities in Eq. (2). If each position's probability vector converges to a one-hot vector selecting a token w as $t \uparrow 1$, then its corresponding state satisfies $Z_t \to E(w)$.*

Thus, under the stated assumptions and exact continuous-time dynamics, commitment in both spaces holds for any uniformly bounded residual, regardless of optimality (Appendix A.3; Figure 3(b,c)).

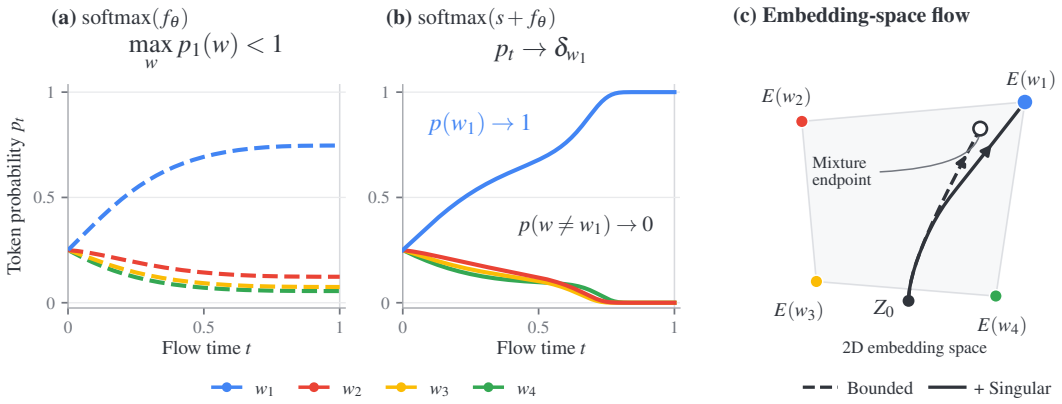


Figure 3: **Singular logits connect commitment to embedding convergence.** A four-token example in which both models share the bounded residual f_θ and the initial state Z_0, with $p_t = p_\theta(\cdot \mid Z_t, t)$ along each trajectory. **(a)** With bounded logits, p_t retains mass on competing tokens. **(b)** Adding $s = \log r_t$ drives p_t to a one-hot distribution. **(c)** Correspondingly, the flow ends at an embedding mixture (dashed) or at $E(w_1)$ (solid).

The network still needs to learn which tokens to choose: singular logits ensure token commitment, while the learned residual determines the quality of the generated text. On OpenWebText, singular logits improve the quality–diversity trade-off (Section 4.3). They also add little computation: because the flow embedding dimension $d = 32$ (Section 3.2) is much smaller than the Transformer hidden width, their evaluation accounts for approximately 0.2–1.1% of the total FLOPs per inference step for our Small, Medium, and Large models.

3.2 Construction of Flow Embeddings

The embedding matrix E shapes the flow. For a given text distribution and Gaussian corruption, E determines the token posteriors along the trajectory and hence the optimal flow field. Given exact posteriors and exact integration, any E with distinct rows recovers the data distribution, but a finite-capacity model must approximate these posteriors, and E controls how hard that is. Because nearby embeddings are the last to be distinguished as noise decreases, E decides which tokens remain confused during most of generation. The design question is therefore which tokens should share this ambiguity.

Hellinger SVD. A noisy token is not only decoded; it also serves as context for the other positions. Before it commits, its state is consistent with several candidate tokens, so the information it carries about its neighbors is a probability-weighted blend of the candidates' typical contexts. If the candidates occur in similar contexts, this blend keeps their shared preferences, letting the residual f_θ reconstruct surrounding positions before every token commits; otherwise, the preferences wash out. Tokens with nearby embeddings should therefore have similar contexts. We construct such an E, the Hellinger–SVD embedding, following Lebret & Collobert (2014): for each token w, we estimate the smoothed distribution Q_w of its preceding and following tokens from the dataset, take its entry-wise square root so that Euclidean distances are proportional to Hellinger distances, and project the result onto the leading d frequency-weighted singular directions (Appendix A.4). Embedding distance then tracks context dissimilarity; we call this construction *flow shaping*, as it preserves context before tokens commit.

Hellinger–SVD fixes the embeddings' relative geometry but not their scale, which sets the signal-to-noise ratio. After perturbing duplicate rows, we center and rescale E under the unigram distribution π so that $\sum_w \pi(w)E(w) = 0$ and $\sum_w \pi(w)\|E(w)\|^2 = d$, fixing the average signal power relative to unit noise while keeping differences in token norms. Since the geometry also determines when tokens emerge from noise, we adapt training times to it: at each noise level, we measure the true token's average log-probability gain over its prior under the analytic posterior with and without local context, and sample training times uniformly along the arc length of the joint curve of the two normalized gains. This concentrates training where token identification improves fastest. The embeddings and schedule stay fixed during training (Appendix B.1).

Embedding Dimension. At each position, the network reads a d-dimensional state, projects it to its hidden width, and outputs logits over all V tokens, while the velocity depends on the vocabulary only through E (Eq. (2)). The flow embedding dimension is thus independent of the Transformer width and the vocabulary size, and any d admits an E with distinct rows, representing every token losslessly. FMLM, a flow-map language model (Lee et al., 2026), suggests a vocabulary-sized one-hot flow with $d = V$. However, image diffusion studies treat this dimension as a design parameter: it should not exceed the model width (Zheng et al., 2026), and with clean-data prediction, the network is largely decoupled from it (Li & He, 2026). We therefore use $d = 32$ for all model sizes, far below their hidden widths and the vocabulary size. Section 4.3 shows that this compact space outperforms a one-hot flow and compares Hellinger–SVD embeddings of dimension 16 to 256.

3.3 Regularized Reflow

Reflow (Liu et al., 2022) straightens the flow by retraining it on noise–token couplings generated by the model itself. A straighter flow can be integrated accurately with larger steps, which reduces the number of function evaluations needed for sampling. Let $\mathcal{S}_\theta(Z_0)$ denote the tokens obtained by integrating the flow ODE of model θ from $Z_0 \sim q_0$ and decoding at the final grid point. Reflow fine-tunes a model, initialized at θ, with the CE objective of Eq. (3) on these couplings, replacing independent noise with the coupled noise:

$$\mathrm{Reflow}(\theta) = \arg\min_{\theta'} \mathbb{E}_{Z_0 \sim q_0,\, t}\big[-\log p_{\theta'}\big(\widehat{\boldsymbol{w}} \mid tE(\widehat{\boldsymbol{w}}) + (1-t)Z_0,\, t\big)\big], \qquad \widehat{\boldsymbol{w}} = \mathcal{S}_\theta(Z_0). \quad (7)$$

Standard reflow samples with the retrained model directly and draws the next round's couplings from it. We instead regularize every round toward the base model θ_0, which we call *regularized reflow*: with θ_k the model after k rounds, each round applies reflow and then interpolates the result back toward the base weights,

$$\hat{\theta}_{k+1} = \mathrm{Reflow}(\theta_k), \qquad \theta_{k+1} = \lambda\,\theta_0 + (1-\lambda)\,\hat{\theta}_{k+1}. \quad (8)$$

The update $\hat{\theta}_{k+1} - \theta_k$ straightens the flow but also moves the posterior predictions away from those of θ_0, which was fit to the data rather than to finite self-generated couplings. Interpolating in weight space (Wortsman et al., 2022) keeps much of the straightening while pulling the predictions back toward θ_0, so θ_{k+1} outperforms both θ_0 and $\hat{\theta}_{k+1}$ at few steps. Each round draws couplings from, and initializes at, θ_k, so the couplings straighten across rounds, while anchoring to θ_0 prevents drift. The target remains a token posterior, so reflow requires no change to the loss, the singular logits (Section 3.1), or the architecture. We use two rounds of regularized reflow (Appendix B.2).

4 Experiments

4.1 Experimental Setup

Dataset and Evaluation. We evaluate unconditional text generation on OpenWebText (OWT) (Gokaslan & Cohen, 2019), following the benchmark used in prior diffusion and flow language models (Sahoo et al., 2024; Hu et al., 2026). We tokenize the corpus with the GPT-2 tokenizer (Radford et al., 2019) and pack sequences to length $L = 1024$. For each ReLF sampling configuration, we generate 1,024 sequences and report generative perplexity (Gen. PPL) under a pretrained GPT-2 Large model, measuring sample diversity by the mean per-sequence unigram entropy in nats. We evaluate the quality–diversity trade-off by sweeping the guidance strength.

Model. Our default model is ReLF-Small, a bidirectional diffusion Transformer (Peebles & Xie, 2023) with 138M trainable parameters, 12 layers, hidden width 768, and 12 attention heads. We use the fixed 32-dimensional Hellinger–SVD token embeddings constructed in Section 3.2. At each position, the noisy embedding is concatenated with a self-conditioning estimate of the clean embedding and projected to the Transformer width. The network predicts residual logits, which are combined with the analytic singular logits using a corpus unigram reference prior. Detailed architectural configurations are given in Appendix B.

Training and Inference. We train ReLF-Small, ReLF-Medium, and ReLF-Large for 80,000 steps each, with a global batch of $1{,}024 \times 1{,}024$ tokens (1,024 sequences of length 1,024), minimizing

the CE loss in Eq. (3). We use Muon (Jordan et al., 2024) for hidden matrices and AdamW for the input and output projections, normalization gains, and biases, with a base learning rate of 3×10^{-4}, 2,500 warmup steps, and a constant learning rate thereafter. Training uses standard Gaussian corruption, an embedding-dependent interpolation-time distribution, and self-conditioning with probability 0.5. For training-time guidance, each sequence also draws a scale ω log-uniform on $[1.5, 6]$, as in ELF (Hu et al., 2026). We evaluate exponential-moving-average (EMA) checkpoints with decay 0.9999. We measure sampling cost by the number of function evaluations (NFE); by default, we use the SDE sampler with 64 NFE. At sampling time, we sweep ω; because ω is a network input, each guided step costs a single forward pass.

4.2 Unconditional Generation on OpenWebText

Figure 1(a) compares the quality–diversity trade-offs of ReLF with those of ELF (Hu et al., 2026) and of autoregressive baselines (AR), each at three model sizes. The AR baselines are trained on the same data and swept over sampling temperature. We denote the model sizes as Small, Medium, and Large; they correspond to the B, M, and L configurations of Hu et al. (2026) (Table 2). With 64 NFE, ReLF outperforms both ELF and the AR baseline at every model size across the trade-off. At matched entropy, ReLF-Small, ReLF-Medium, and ReLF-Large reach Gen. PPL 24.1, 18.6, and 17.3, compared with 27.7, 22.3, and 20.2 for AR-Small, AR-Medium, and AR-Large. Appendix B.6 gives the evaluation details and the full Pareto frontiers.

ReLF-Large saturates the standard OpenWebText benchmark. It reaches Gen. PPL 17.3 at entropy 5.51, matching real OpenWebText text (17.3 at 5.49). Surprisingly, unlike the Small and Medium models and despite the identical training pipeline, ReLF-Large no longer trades Gen. PPL against entropy under guidance: across all guidance scales, both metrics stay nearly constant, with Gen. PPL between 17.3 and 18.0 (Table 5), so Figure 1(a) shows ReLF-Large as a single point. This does not suggest that its samples fully match real text; the remaining differences in quality and diversity are small, semantic, and beyond what Gen. PPL and unigram entropy resolve (see the samples in Appendix C). Resolving them calls for larger models and more data, evaluated on harder benchmarks with richer metrics.

Figure 1(b) shows how regularized reflow (Section 3.3) improves few-step generation. We sample every round with one fixed sampler per model size and NFE ($k = 0$ is the base model). Across rounds, Gen. PPL decreases at 8, 16, and 32 NFE for every model size, with no loss of diversity. Most of the gain arrives within two rounds: at 8 NFE, the first round lowers Gen. PPL by 29–36% (e.g., from 62.5 to 44.5 for Large), and the second by a further 8–12%. After two rounds, 16 NFE yields lower Gen. PPL than the base model at 32 NFE for every model size (Figure 5 in Appendix B.5), and at matched entropy ReLF outperforms ELF+PD at 8–32 NFE and AR-Small (1,023 NFE) from 16 NFE.

4.3 Ablation Studies

Figure 4 summarizes the ablations. Unless otherwise noted, they use ReLF-Small with SDE sampling at 64 NFE and 1,024 samples per point. Panels (a)–(d) plot quality–diversity curves. With the one-hot embeddings of (b), training-time guidance is unstable during training, and SDE sampling is not supported. For a fair comparison, we therefore train the models in (a) and (b) without training-time guidance, sweep momentum guidance (Liao et al., 2026) of strength α, and also report ODE (Euler) sampling in (b); (c) and (d) sweep the guidance scale ω. Appendix B.5 gives further settings.

Singular Logits. Singular logits improve the quality–diversity trade-off (Figure 4(a)). Without guidance, the model with singular logits reaches Gen. PPL 57.2 at entropy 5.48, versus 64.2 at 5.48 for the baseline, and its curve lies below the baseline's across their shared entropy range. The gap widens under strong guidance: from $\alpha = 0.5$ to 0.7, Gen. PPL falls from 29.3 to 24.0 with singular logits but only from 37.2 to 36.3 without them.

Embedding Matrix. Hellinger–SVD outperforms random Gaussian and one-hot embeddings, all with singular logits (Figure 4(b)). With SDE sampling at $\alpha = 0.2$, Hellinger–SVD reaches Gen. PPL 42.1 at entropy 5.37, compared with 83.6 at 5.34 for a 32-dimensional random Gaussian embedding.

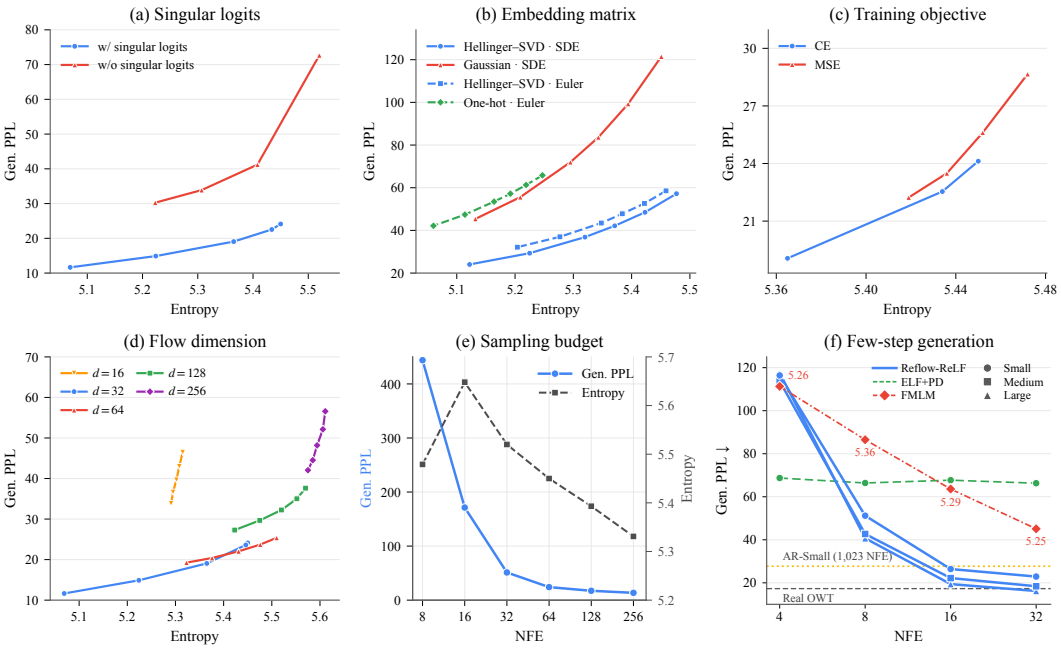


Figure 4: **Ablations of ReLF. (a)–(d)** Gen. PPL versus entropy at 64 NFE; lines connect points in order of guidance strength. **(a)** With and without singular logits. **(b)** Embedding matrix. **(c)** CE and MSE training objectives. **(d)** Flow embedding dimension d. In (b), solid and dashed curves use SDE and Euler sampling, respectively. **(e)** Gen. PPL (solid, left axis) and entropy (dashed, right axis) versus NFE at $\omega = 1.7$. **(f)** Few-step generation after two reflow rounds: lowest Gen. PPL with entropy at least 5.40; FMLM (Lee et al., 2026) points are as reported, labeled with their entropy.

With Euler sampling, it attains lower Gen. PPL and higher entropy than the one-hot embedding at every guidance strength, even though the one-hot model has about 215M parameters versus 138M.

Training Objective. Training with CE yields a better quality–diversity trade-off than training with MSE (Figure 4(c)). The MSE variant regresses the velocity $\boldsymbol{v}_\theta = \frac{1}{1-t}(\widehat{X}_1 - X_t)$ onto $X_1 - X_0$, as in ELF (Hu et al., 2026), and applies the training-time guidance shift to this velocity target rather than to the logits. The two models differ only in the training loss. At matched entropy, CE attains lower Gen. PPL: 24.1 at entropy 5.45 ($\omega = 1.7$) versus 25.6 at 5.45 for MSE ($\omega = 2.8$). MSE also responds weakly to guidance: its Gen. PPL stays above 22 for all $\omega \in [1.7, 3.0]$, and beyond $\omega = 2.4$, both Gen. PPL and entropy rise. CE instead reaches 19.1 at entropy 5.37 with $\omega = 1.9$.

Flow Embedding Dimension. The default 32-dimensional flow embedding space gives the best quality–diversity trade-off (Figure 4(d)). We vary $d \in \{16, 32, 64, 128, 256\}$, retaining the leading d Hellinger–SVD directions (Section 3.2) and normalizing the unigram-weighted mean squared norm to d; only the input projection depends on d. Larger dimensions shift the curve toward higher entropy and need much stronger guidance to lower Gen. PPL: $d = 128$ and $d = 256$ remain at Gen. PPL 27.3 and 42.1 (entropies 5.42 and 5.58) even at $\omega = 2.8$ and 2.6, and $d = 64$ needs $\omega = 2.4$ to reach 23.7 at 5.48, whereas $d = 32$ reaches 23.6 at 5.45 with $\omega = 1.8$. Reducing the dimension to $d = 16$ also hurts, placing its curve above that of $d = 32$. Table 3 in Appendix B.5 lists the plotted values.

Sampling Budget. At the default guidance scale $\omega = 1.7$, a larger sampling budget lowers Gen. PPL (Figure 4(e)); the grid family is fixed, and only the number of steps varies (Appendix B.4). Gen. PPL drops sharply up to the default 64 NFE and decreases further at 256 NFE, at lower entropy.

Few-Step Generation. Figure 4(f) compares ReLF after two reflow rounds with few-step baselines: ELF+PD, which progressively distills ELF into a few-step student (Hu et al., 2026), and FMLM (Lee et al., 2026). For every model and NFE, we report the lowest Gen. PPL among settings with entropy at least 5.40 (Appendix B.6). ELF+PD plateaus from 4 to 32 NFE, whereas ReLF

improves steadily with NFE: every model size outperforms ELF+PD from 8 NFE and AR-Small (1,023 NFE) from 16 NFE.

5 Related Work

Discrete-State Models. Discrete state models generate sequences by updating categorical variables, often at multiple positions per step. Following masked-token pretraining (Devlin et al., 2019) and iterative refinement (Ghazvininejad et al., 2019), D3PM formalized structured categorical corruption (Austin et al., 2021), while continuous-time Markov formulations established corresponding denoising dynamics (Campbell et al., 2022). Subsequent work develops discrete score estimation, simplified masked-denoising objectives and time-independent parameterizations (Lou et al., 2024; Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2025), alongside more general corruption kernels and probability paths (Gat et al., 2024; Shaul et al., 2025; von Rütte et al., 2025). The LLaDA family (Nie et al., 2025; Zhu et al., 2025; Bie et al., 2025) advances masked diffusion through scaling and preference optimization, alongside work on language and code generation (Ye et al., 2025; Gong et al., 2025; Song et al., 2025) and uniform-state diffusion (Sahoo et al., 2025; 2026; Ye et al., 2026). Sampling research develops blockwise decoding (Arriola et al., 2025), remasking (Wang et al., 2025), and adaptive token ordering and parallel updates (Kim et al., 2025; Wu et al., 2025b; Ben-Hamu et al., 2025). Reinforcement learning supports reasoning (Zhao et al., 2025), while masked generation extends to images and multimodal modeling (Chang et al., 2022; Yang et al., 2025). These models retain categorical states even when time is continuous.

Continuous-State Models. Continuous state models generate categorical sequences through real-valued representations, with early examples including Argmax Flows (Hoogeboom et al., 2021). Diffusion-LM (Li et al., 2022) used token-embedding diffusion for controllable text generation, while Analog Bits (Chen et al., 2023) introduced self-conditioning for diffusion over continuous relaxations of binary encodings. Subsequent embedding-space models incorporate self-conditioning (Strudel et al., 2022), continuous-time formulations (Dieleman et al., 2022), conditional generation and pretraining (Gong et al., 2023; Lin et al., 2023; Ye et al., 2023), and likelihood-based training (Gulrajani & Hashimoto, 2023). Alternative representations include vocabulary-sized logit/simplex states (Han et al., 2023; Mahabadi et al., 2024; Tae et al., 2025) and statistical-manifold constructions (Jo & Hwang, 2025; Cheng et al., 2024; 2025). Latent diffusion denoises encoded text representations with a separate decoder (Lovelace et al., 2023; Zhang et al., 2023; Shabalin et al., 2025; Meshchaninov et al., 2025; Shen et al., 2026), with recent work jointly learning the latent space and generative dynamics (Meshchaninov et al., 2026; Guo et al., 2026). Rectified flow has also been applied to conditional text generation (Hu et al., 2024), while continuous flow-map language models pursue few-step decoding through distillation (Lee et al., 2026; Roos et al., 2026; Potaptchik et al., 2026). LangFlow (Chen et al., 2026) uses a Bregman formulation to train embedding-space flows with cross-entropy, whereas ELF (Hu et al., 2026) performs rectified flow in pretrained contextual embeddings and applies token-level supervision only at final decoding. Within this family, ReLF combines analytic singular logits for endpoint commitment with a compact, context-informed flow embedding space under a single cross-entropy objective.

6 Conclusion

We introduced ReLF, a continuous rectified-flow language model trained with cross-entropy on token posteriors. Singular logits capture the divergent part of these posteriors analytically, so every trajectory converges to a token embedding, and the flow runs in a compact 32-dimensional Hellinger–SVD embedding space that preserves context before tokens commit. On OpenWebText, ReLF outperforms ELF and autoregressive baselines of matching size at 64 NFE, ReLF-Large matches real text in Gen. PPL and entropy, and regularized reflow extends ReLF to few-step generation. Ablations confirm the gains from singular logits and the compact embedding space. These results establish continuous flows as a promising alternative to autoregressive language modeling, with scaling to larger models and harder benchmarks as the natural next step.

AI USE STATEMENT

We used large language models (LLMs) for three tasks: editing the paper to improve readability, creating and formatting scientific figures, and deploying existing compute kernels in our code for efficient hardware utilization. We did not use generative AI tools for any other task listed in the ICLR 2027 AI Policy; the research ideas, theory and proofs, method, experimental design, and interpretation of results are our own. We reviewed all AI-assisted work: we checked every edited passage for technical accuracy, verified each figure against the underlying results, and tested the code after each kernel deployment for correctness. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=tyEyYT267x.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pp. 17981–17993, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/510e22f4a2da5212a64f1591736e2eaf-Abstract-Conference.html.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Lin Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, Jianfeng Tan, Lanning Wei, Ji-Rong Wen, Yipeng Xing, Xiaolu Zhang, Junbo Zhao, Da Zheng, Jun Zhou, Junlin Zhou, Zhanchao Zhou, Liwang Zhu, and Yihong Zhuang. Llada2.0: Scaling up diffusion language models to 100b. *CoRR*, abs/2512.15745, 2025. URL https://arxiv.org/abs/2512.15745.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b5b528767aa35f5b1a60fe0aaeca0563-Abstract-Conference.html.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pp. 11305–11315. IEEE, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Chang_MaskGIT_Masked_Generative_Image_Transformer_CVPR_2022_paper.html.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*, 2023.

Yuxin Chen, Chumeng Liang, Hangke Sui, Ruihan Guo, Chaoran Cheng, Jiaxuan You, and Ge Liu. Langflow: Continuous diffusion rivals discrete in language modeling. *CoRR*, abs/2604.11748, 2026. URL https://arxiv.org/abs/2604.11748.

Chaoran Cheng, Jiahan Li, Jian Peng, and Ge Liu. Categorical flow matching on statistical manifolds. In *Advances in Neural Information Processing Systems*

37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/62a58f2130894e44e8a272c563a2c6f1-Abstract-Conference.html.

Chaoran Cheng, Jiahan Li, Jiajun Fan, and Ge Liu. α-flow: A unified framework for continuous-state discrete flow matching models. *CoRR*, abs/2504.10283, 2025. URL https://arxiv.org/abs/2504.10283.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. URL https://aclanthology.org/N19-1423/.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H. Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, Curtis Hawthorne, Rémi Leblond, Will Grathwohl, and Jonas Adler. Continuous diffusion for categorical data. *CoRR*, abs/2211.15089, 2022. URL https://arxiv.org/abs/2211.15089.

Floor Eijkelboom, Grigory Bartosh, Christian A. Naesseth, Max Welling, and Jan-Willem van de Meent. Variational flow matching for graph generation. In *Advances in Neural Information Processing Systems*, 2024.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/f0d629a734b56a642701bba7bc8bb3ed-Abstract-Conference.html.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pp. 6111–6120. Association for Computational Linguistics, 2019. URL https://aclanthology.org/D19-1633/.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus, 2019.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations, ICLR 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=jQj-_rLVXsj.

Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. *CoRR*, abs/2506.20639, 2025. URL https://arxiv.org/abs/2506.20639.

Ishaan Gulrajani and Tatsunori B. Hashimoto. Likelihood-based diffusion language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/35b5c175e139bff5f22a5361270fce87-Abstract-Conference.html.

Hongcan Guo, Qinyu Zhao, Yian Zhao, Shen Nie, Rui Zhu, Qiushan Guo, Feng Wang, Tao Yang, Hengshuang Zhao, Guoqiang Wei, and Yan Zeng. Continuous latent diffusion language model. *CoRR*, abs/2605.06548, 2026. URL https://arxiv.org/abs/2605.06548.

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), ACL 2023*, pp. 11575–11596. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.acl-long.647/.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pp. 12454–12465, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/67d96d458abdef21792e6d8e590244e7-Abstract.html.

Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. ELF: embedded language flows. *CoRR*, abs/2605.10938, 2026. URL https://arxiv.org/abs/2605.10938.

Vincent Tao Hu, Di Wu, Yuki Markus Asano, Pascal Mettes, Basura Fernando, Björn Ommer, and Cees Snoek. Flow matching for conditional text generation in a few sampling steps. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 2: Short Papers*, pp. 380–392. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.eacl-short.33/.

Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/8c4046a93293442e88f54e19a5849d68-Abstract-Conference.html.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Forty-second International Conference on Machine Learning, ICML 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/kim25ah.html.

Rémi Lebret and Ronan Collobert. Word embeddings through Hellinger PCA. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 482–490. Association for Computational Linguistics, 2014. URL https://aclanthology.org/E14-1051/.

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.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2026*. IEEE, 2026. URL https://arxiv.org/abs/2511.13720.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/1be5bc25d50895ee656b8c2d9eb89d6a-Abstract-Conference.html.

Runlong Liao, Jian Yu, Baiyu Su, Chi Zhang, Lizhang Chen, and Qiang Liu. Momentum guidance: Plug-and-play guidance for flow models. *arXiv preprint arXiv:2602.20360*, 2026.

Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21051–21064. PMLR, 2023. URL https://proceedings.mlr.press/v202/lin23d.html.

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.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning, ICML 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q. Weinberger. Latent diffusion for language generation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/b2a2bd5d5051ff6af52e1ef60aefd255-Abstract-Conference.html.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant Transformers. In *European Conference on Computer Vision*, 2024.

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, EACL 2024 - Volume 1: Long Papers*, pp. 2347–2361. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.eacl-long.144/.

Viacheslav Meshchaninov, Egor Chimbulatov, Alexander Shabalin, Aleksandr Abramov, and Dmitry P. Vetrov. Cosmos: Compressed and smooth latent space for text diffusion modeling. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/1506930fb75c82246e4d8648a66e4b27-Abstract-Conference.html.

Viacheslav Meshchaninov, Alexander Shabalin, Egor Chimbulatov, Nikita Gushchin, Ilya Koziev, Alexander Korotin, and Dmitry P. Vetrov. How to train your latent diffusion language model jointly with the latent space. *CoRR*, abs/2605.07933, 2026. URL https://arxiv.org/abs/2605.07933.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/48b383b24230e0e6e649d9c98dae4d8c-Abstract-Conference.html.

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. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Peter Potaptchik, Jason Yim, Adhi Saravanan, Peter Holderrieth, Eric Vanden-Eijnden, and Michael S. Albergo. Discrete flow maps. *CoRR*, abs/2604.09784, 2026. URL https://arxiv.org/abs/2604.09784.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/904e89bb4e632e75fb47f093b620b257-Abstract-Conference.html.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Daan Roos, Oscar Davis, Floor Eijkelboom, Michael M. Bronstein, Max Welling, İsmail İlkan Ceylan, Luca Ambrogioni, and Jan-Willem van de Meent. Categorical flow maps. *CoRR*, abs/2602.12233, 2026. URL https://arxiv.org/abs/2602.12233.

Subham S. Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T. Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Forty-second International Conference on Machine Learning, ICML 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/sahoo25a.html.

Subham Sekhar Sahoo, Jean-Marie Lemercier, Zhihan Yang, Justin Deschenaux, Jingyu Liu, John Thickstun, and Ante Jukic. Scaling beyond masked diffusion language models. *CoRR*, abs/2602.15014, 2026. URL https://arxiv.org/abs/2602.15014.

Alexander Shabalin, Viacheslav Meshchaninov, Egor Chimbulatov, Vladislav Lapikov, Roman Kim, Grigory Bartosh, Dmitry Molchanov, Sergey Markov, and Dmitry P. Vetrov. Tencdm: Understanding the properties of the diffusion model in the space of language model encodings. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025*, pp. 25110–25118. AAAI Press, 2025. URL https://ojs.aaai.org/index.php/AAAI/article/view/34696.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky T. Q. Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=tcvMzR2NrP.

Junzhe Shen, Jieru Zhao, Ziwei He, and Zhouhan Lin. Codar: Continuous diffusion language models are more powerful than you think. *CoRR*, abs/2603.02547, 2026. URL https://arxiv.org/abs/2603.02547.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, Yuwei Fu, Jing Su, Ge Zhang, Wenhao Huang, Mingxuan Wang, Lin Yan, Xiaoying Jia, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Yonghui Wu, and Hao

Zhou. Seed diffusion: A large-scale diffusion language model with high-speed inference. *CoRR*, abs/2508.02193, 2025. URL https://arxiv.org/abs/2508.02193.

Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and Rémi Leblond. Self-conditioned embedding diffusion for text generation. *CoRR*, abs/2211.04236, 2022. URL https://arxiv.org/abs/2211.04236.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. URL https://linkinghub.elsevier.com/retrieve/pii/S0925231223011864.

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), ACL 2025*, pp. 21171–21188. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.1029/.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *Forty-second International Conference on Machine Learning, ICML 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/von-rutte25a.html.

Guanghan Wang, Yair Schiff, Subham S. Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/d877ea0dd78bdbe54830670618c1de09-Abstract-Conference.html.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Hannaneh Hajishirzi, Raphael Gontijo-Lopes, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7969, 2022.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025a. URL https://arxiv.org/abs/2508.02324.

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. *CoRR*, abs/2505.22618, 2025b. URL https://arxiv.org/abs/2505.22618.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/caa934a507a952698d54efb24845fc4b-Abstract-Conference.html.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *CoRR*, abs/2508.15487, 2025. URL https://arxiv.org/abs/2508.15487.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Mingxuan Wang. Dinoiser: Diffused conditional sequence learning by manipulating noises. *arXiv preprint arXiv:2302.10025*, 2023.

Mengyu Ye, Keito Kudo, Wataru Ikeda, Ryosuke Matsuda, Keisuke Sakaguchi, and Jun Suzuki. Sumi: Open uniform diffusion language model from scratch. *CoRR*, abs/2606.19005, 2026. URL https://arxiv.org/abs/2606.19005.

Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Joshua M. Susskind, and Navdeep Jaitly. PLANNER: generating diversified paragraph via latent language diffusion model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/fdba5e0a9b57fce03e89cc0cad0a24e9-Abstract-Conference.html.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/52190a0362148d179f1cbd9080956872-Abstract-Conference.html.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *The Fourteenth International Conference on Learning Representations, ICLR 2026*. OpenReview.net, 2026. URL https://arxiv.org/abs/2510.11690.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *CoRR*, abs/2505.19223, 2025. URL https://arxiv.org/abs/2505.19223.

A THEORY AND PROOFS

Indexed Notation. Let $\mathcal{V} = \{1, \dots, V\}$ be a vocabulary of size V, L the sequence length, and d the embedding dimension. A token sequence $\boldsymbol{w} = [w^{(1)}, \dots, w^{(L)}] \in \mathcal{V}^L$ is drawn from a data distribution $\mathcal{D}$. The embedding map $E : \mathcal{V} \to \mathbb{R}^d$ acts positionwise to give the clean state $X_1 = E(\boldsymbol{w}) \in \mathbb{R}^{d \times L}$. For a state $Z \in \mathbb{R}^{d \times L}$, its ℓth column is $Z^{(\ell)} \in \mathbb{R}^d$, where $\ell \in \{1, \dots, L\}$; thus $X_1^{(\ell)} = E(w^{(\ell)})$. Parenthesized superscripts index sequence positions for both tokens and embedding states. We retain capitals for states: $X_t = tX_1 + (1-t)X_0$, $t \in [0,1]$, denotes the training interpolation, Z_t an inference trajectory, and Z a generic full-sequence state at which a posterior or vector field is evaluated. Lowercase $z \in \mathbb{R}^d$ denotes a single-position state. The columns $X_0^{(\ell)} \sim \mathcal{N}(0, I_d)$ are mutually independent and independent of $\boldsymbol{w}$; equivalently, q_0 is standard Gaussian on $\mathbb{R}^{d \times L}$. State norms are Frobenius norms, and single-position embedding norms are Euclidean norms.

Sequence Laws and Token Posteriors. We overload q_1 to denote both the data distribution on token sequences, $q_1(\boldsymbol{w}) = \mathcal{D}(\boldsymbol{w})$, and the induced distribution of the embedded clean state $X_1 = E(\boldsymbol{w})$. Let $r_1(\boldsymbol{w})$ be a reference sequence distribution. The shared corruption likelihood and its positionwise factors are

$$q_{t|1}(Z \mid \boldsymbol{w}) = \prod_{\ell=1}^{L} q_{t|1}^{(\ell)}(Z^{(\ell)} \mid w^{(\ell)}), \qquad q_{t|1}^{(\ell)}(z \mid w) = \frac{\exp\Big(-\frac{\|z - tE(w)\|^2}{2(1-t)^2}\Big)}{(2\pi)^{d/2}(1-t)^d}, \quad t < 1.$$

For $a \in \{q, r\}$, define the noisy-state density, full-sequence posterior, and positionwise token posterior by

$$\begin{aligned} a_t(Z) &\coloneqq \sum_{\boldsymbol{u} \in \mathcal{V}^L} a_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u}), \\ a_t(\boldsymbol{w} \mid Z) &\coloneqq \frac{a_1(\boldsymbol{w}) q_{t|1}(Z \mid \boldsymbol{w})}{a_t(Z)}, \\ a_t^{(\ell)}(w \mid Z) &\coloneqq \sum_{\boldsymbol{u}:\, u^{(\ell)} = w} a_t(\boldsymbol{u} \mid Z). \end{aligned}$$

In particular, $q_t^{(\ell)}(w \mid Z) = \Pr(w^{(\ell)} = w \mid X_t = Z)$ under the data distribution; $r_t^{(\ell)}$ is defined under the reference distribution with the same corruption. Their arguments distinguish noisy-state densities from token posteriors. When the position index is suppressed, $q_t(w \mid Z_t)$ and $r_t(w \mid Z_t)$ denote the corresponding positionwise posteriors evaluated at $Z = Z_t$. The subscript $t \mid 1$ is reserved for the corruption likelihood. Unless otherwise specified, sums over $\boldsymbol{u}$ run over $\mathcal{V}^L$, and sums over individual tokens run over $\mathcal{V}$.

Model Outputs and Self-Conditioning. A model with parameters θ predicts a vocabulary distribution $p_\theta^{(\ell)}(\cdot \mid Z, t; \boldsymbol{C})$ at each position, conditioning on the full state Z and, when used, the self-conditioning input $\boldsymbol{C} \in \mathbb{R}^{d \times L}$. With singular logits,

$$\begin{aligned} s^{(\ell)}(w; Z, t) &\coloneqq \log r_t^{(\ell)}(w \mid Z), \\ p_\theta^{(\ell)}(w \mid Z, t; \boldsymbol{C}) &= \frac{\exp\Big(s^{(\ell)}(w; Z, t) + f_\theta^{(\ell)}(w; Z, t; \boldsymbol{C})\Big)}{\sum_{u \in \mathcal{V}} \exp\Big(s^{(\ell)}(u; Z, t) + f_\theta^{(\ell)}(u; Z, t; \boldsymbol{C})\Big)}, \\ \widehat{X}_1^{(\ell)}(Z, t; \boldsymbol{C}) &\coloneqq \sum_{w \in \mathcal{V}} E(w) p_\theta^{(\ell)}(w \mid Z, t; \boldsymbol{C}). \end{aligned}$$

Here $f_\theta^{(\ell)}$ is the learned residual logit, and $\widehat{X}_1^{(\ell)}$ is the ℓth column of the clean prediction $\widehat{X}_1$. We omit $\boldsymbol{C}$ when it is unused or implicit.

A.1 From Velocity to Conditional Marginals

The optimal rectified-flow velocity is the conditional mean of the interpolation velocity (Liu et al., 2022; Lipman et al., 2022):

$$\boldsymbol{v}^*(Z,t) = \mathbb{E}[X_1 - X_0 \mid X_t = Z].$$

Since $X_1 - X_0 = (X_1 - X_t)/(1-t)$ for $t < 1$, we obtain

$$\begin{aligned} \boldsymbol{v}^*(Z,t) &= \mathbb{E}\left[\frac{X_1 - X_t}{1-t} \,\middle|\, X_t = Z\right] \\ &= \frac{\mathbb{E}[X_1 \mid X_t = Z] - Z}{1-t}. \end{aligned} \tag{9}$$

The ℓth column of the exact clean conditional mean is

$$\mathbb{E}[X_1^{(\ell)} \mid X_t = Z] = \sum_{w \in \mathcal{V}} E(w) q_t^{(\ell)}(w \mid Z).$$

Thus the positionwise token posteriors suffice to determine every column of the velocity, even when the full-sequence posterior does not factorize. Replacing $q_t^{(\ell)}$ by $p_\theta^{(\ell)}$ gives the modeled clean prediction $\widehat{X}_1$ and velocity

$$\boldsymbol{v}_\theta(Z,t;\boldsymbol{C}) = \frac{\widehat{X}_1(Z,t;\boldsymbol{C}) - Z}{1-t}.$$

The token-averaged cross-entropy objective is

$$\mathcal{L}(\theta) = \mathbb{E}_{\boldsymbol{w}\sim\mathcal{D},\, X_0\sim q_0,\, t}\left[-\frac{1}{L}\sum_{\ell=1}^{L} \log p_\theta^{(\ell)}(w^{(\ell)} \mid X_t, t)\right].$$

The expectation averages over data sequences, independent Gaussian noise, and the training-time distribution. With self-conditioning, each training sequence uses $\boldsymbol{C} = \mathrm{sg}[\widehat{X}_1(X_t,t;\boldsymbol{0})]$ with probability $1/2$ and $\boldsymbol{C} = \boldsymbol{0}$ otherwise, where sg blocks gradient propagation through the preliminary prediction. The probabilities additionally condition on $\boldsymbol{C}$, and the expectation includes this sampling. The clean target at each position remains $w^{(\ell)}$.

A.2 Singular Logits

Theorem A.1. *Suppose $r_1(\boldsymbol{w}) > 0$ and $\kappa^{-1} \le q_1(\boldsymbol{w})/r_1(\boldsymbol{w}) \le \kappa$ for every $\boldsymbol{w} \in \mathcal{V}^L$ and some $1 \le \kappa < \infty$. For each position ℓ, the singular logits are*

$$s^{(\ell)}(w;Z,t) = \log r_t^{(\ell)}(w \mid Z) = \log \frac{\sum_{\boldsymbol{u}:\, u^{(\ell)}=w} r_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u})}{\sum_{\boldsymbol{u}} r_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u})}.$$

Define the exact residual that the neural network $f_\theta^{(\ell)}$ approximates by

$$f_*^{(\ell)}(w;Z,t) \coloneqq \log q_t^{(\ell)}(w \mid Z) - s^{(\ell)}(w;Z,t).$$

This residual is uniformly bounded: $\left|f_*^{(\ell)}(w;Z,t)\right| \le 2\log\kappa$ for every position ℓ, state $Z \in \mathbb{R}^{d\times L}$, token $w \in \mathcal{V}$, and time $t < 1$.

Proof. By Bayes' rule, the marginal posterior is

$$\begin{aligned} q_t^{(\ell)}(w \mid Z) &= \sum_{\boldsymbol{u}:\, u^{(\ell)}=w} q_t(\boldsymbol{u} \mid Z) \\ &= \frac{\sum_{\boldsymbol{u}:\, u^{(\ell)}=w} q_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u})}{\sum_{\boldsymbol{u}} q_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u})}. \end{aligned}$$

Taking the ratio of the two marginal posteriors gives

$$\frac{q_t^{(\ell)}(w \mid Z)}{r_t^{(\ell)}(w \mid Z)} = \frac{\sum_{\boldsymbol{u}:\, u^{(\ell)}=w} q_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u})}{\sum_{\boldsymbol{u}:\, u^{(\ell)}=w} r_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u})} \cdot \frac{\sum_{\boldsymbol{u}} r_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u})}{\sum_{\boldsymbol{u}} q_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u})}.$$

Since $\kappa^{-1} r_1(\boldsymbol{u}) \le q_1(\boldsymbol{u}) \le \kappa r_1(\boldsymbol{u})$, summing over sequences with $u^{(\ell)} = w$ yields

$$\begin{aligned}\kappa^{-1} \sum_{\boldsymbol{u}:\, u^{(\ell)}=w} r_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u}) &\le \sum_{\boldsymbol{u}:\, u^{(\ell)}=w} q_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u}) \\ &\le \kappa \sum_{\boldsymbol{u}:\, u^{(\ell)}=w} r_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u}).\end{aligned}$$

Thus the first ratio lies in $[\kappa^{-1}, \kappa]$. Summing over all sequences gives the same bounds for the second ratio. Therefore,

$$\kappa^{-2} \le \frac{q_t^{(\ell)}(w \mid Z)}{r_t^{(\ell)}(w \mid Z)} \le \kappa^2, \qquad \left| f_*^{(\ell)}(w; Z, t) \right| \le 2 \log \kappa.$$

□

Lemma A.2. *Suppose the reference distribution factorizes as $r_1(\boldsymbol{w}) = \prod_{\ell=1}^{L} r_1^{(\ell)}(w^{(\ell)})$, where $r_1^{(\ell)}$ is the reference token prior at position ℓ, and corruption is independent across positions. Then the singular logits can also be computed positionwise as*

$$s^{(\ell)}(w; Z, t) = \log \frac{r_1^{(\ell)}(w) q_{t|1}^{(\ell)}(Z^{(\ell)} \mid w)}{\sum_{u \in \mathcal{V}} r_1^{(\ell)}(u) q_{t|1}^{(\ell)}(Z^{(\ell)} \mid u)}.$$

For a uniform reference, this becomes

$$s^{(\ell)}(w; Z, t) = \log q_{t|1}^{(\ell)}(Z^{(\ell)} \mid w) - \log \sum_{u \in \mathcal{V}} q_{t|1}^{(\ell)}(Z^{(\ell)} \mid u).$$

Proof. Using the factorization of the reference and corruption, the numerator of the reference token posterior is

$$\begin{aligned}&\sum_{\boldsymbol{u}:\, u^{(\ell)}=w} r_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u}) \\ &= \sum_{\boldsymbol{u}:\, u^{(\ell)}=w} \prod_{m=1}^{L} r_1^{(m)}(u^{(m)}) q_{t|1}^{(m)}(Z^{(m)} \mid u^{(m)}) \\ &= r_1^{(\ell)}(w) q_{t|1}^{(\ell)}(Z^{(\ell)} \mid w) \prod_{m \ne \ell} \left[\sum_{u \in \mathcal{V}} r_1^{(m)}(u) q_{t|1}^{(m)}(Z^{(m)} \mid u) \right].\end{aligned}$$

Similarly, the denominator is

$$\sum_{\boldsymbol{u}} r_1(\boldsymbol{u}) q_{t|1}(Z \mid \boldsymbol{u}) = \prod_{m=1}^{L} \left[\sum_{u \in \mathcal{V}} r_1^{(m)}(u) q_{t|1}^{(m)}(Z^{(m)} \mid u) \right].$$

Canceling the factors at positions $m \ne \ell$ and taking the logarithm gives the first expression. For a uniform reference, $r_1^{(\ell)}(u) = 1/V$ for all positions and tokens, and this factor also cancels. □

For standard Gaussian noise, substituting the likelihood into the factorized expression yields

$$\begin{aligned}s^{(\ell)}(w; Z, t) = \log r_1^{(\ell)}(w) &- \frac{\| Z^{(\ell)} - t E(w) \|^2}{2(1-t)^2} \\ &- \log \sum_{u \in \mathcal{V}} r_1^{(\ell)}(u) \exp\left(- \frac{\| Z^{(\ell)} - t E(u) \|^2}{2(1-t)^2} \right).\end{aligned}$$

The last term is shared across tokens and can be omitted before softmax. Setting $r_1^{(\ell)}(w) = 1/V$ recovers Eq. (6), while the experimental unigram reference uses $r_1^{(\ell)}(w) = \pi(w)$ at every position, where $\pi(w)$ is the normalized corpus frequency of token w. Only the reference posterior reduces to a function of $Z^{(\ell)}$; the data posterior and learned residual still condition on the full sequence Z.

A.3 From Token Commitment to Endpoint Convergence

Fix a position ℓ and, along the trajectory, write

$$p_t^{(\ell)}(w) = p_\theta^{(\ell)}(w \mid Z_t, t), \quad r_t^{(\ell)}(w) = r_t^{(\ell)}(w \mid Z_t), \quad f_t^{(\ell)}(w) = f_\theta^{(\ell)}(w; Z_t, t).$$

Let $w^{(\ell)\star} \in \mathcal{V}$ be the token selected in the limit at position ℓ. Suppose $r_t^{(\ell)}(w^{(\ell)\star}) \to 1$ and $|f_t^{(\ell)}(w)| \le M$ uniformly in w and $t < 1$. Since $p_t^{(\ell)}(w) \propto r_t^{(\ell)}(w) e^{f_t^{(\ell)}(w)}$,

$$\begin{aligned} 1 - p_t^{(\ell)}(w^{(\ell)\star}) &= \frac{\sum_{w \neq w^{(\ell)\star}} r_t^{(\ell)}(w) e^{f_t^{(\ell)}(w)}}{\sum_{u \in \mathcal{V}} r_t^{(\ell)}(u) e^{f_t^{(\ell)}(u)}} \\ &\le \frac{e^{M} \sum_{w \neq w^{(\ell)\star}} r_t^{(\ell)}(w)}{e^{-M} \sum_{u \in \mathcal{V}} r_t^{(\ell)}(u)} \\ &= e^{2M}\big(1 - r_t^{(\ell)}(w^{(\ell)\star})\big) \longrightarrow 0. \end{aligned}$$

Thus any uniformly bounded residual preserves one-hot convergence of the reference posterior.

Proposition A.3. *Consider the flow ODE $\dot{Z}_t = \boldsymbol{v}_\theta(Z_t, t)$ with modeled probabilities. If each position's probability vector $p_t^{(\ell)}$ converges to a one-hot vector selecting $w^{(\ell)\star}$ as $t \uparrow 1$, then $Z_t^{(\ell)} \to E(w^{(\ell)\star})$.*

Proof. Let $\widehat{X}_{1,t} = \widehat{X}_1(Z_t, t)$ denote the modeled clean prediction along the trajectory. By the assumed probability convergence,

$$\widehat{X}_{1,t}^{(\ell)} = \sum_{w \in \mathcal{V}} E(w) p_t^{(\ell)}(w) \longrightarrow E(w^{(\ell)\star}).$$

Using $\dot{Z}_t^{(\ell)} = (\widehat{X}_{1,t}^{(\ell)} - Z_t^{(\ell)})/(1-t)$, we obtain

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{Z_t^{(\ell)}}{1-t}\right) &= \frac{(1-t)\dot{Z}_t^{(\ell)} + Z_t^{(\ell)}}{(1-t)^2} \\ &= \frac{\widehat{X}_{1,t}^{(\ell)}}{(1-t)^2}. \end{aligned}$$

Integrating from 0 to t gives

$$\frac{Z_t^{(\ell)}}{1-t} = Z_0^{(\ell)} + \int_0^t \frac{\widehat{X}_{1,u}^{(\ell)}}{(1-u)^2}\,\mathrm{d}u.$$

Applying L'Hôpital's rule coordinatewise,

$$\begin{aligned} \lim_{t \uparrow 1} Z_t^{(\ell)} &= \lim_{t \uparrow 1} \frac{Z_0^{(\ell)} + \int_0^t \widehat{X}_{1,u}^{(\ell)} (1-u)^{-2}\,\mathrm{d}u}{(1-t)^{-1}} \\ &= \lim_{t \uparrow 1} \frac{\widehat{X}_{1,t}^{(\ell)} (1-t)^{-2}}{(1-t)^{-2}} \\ &= \lim_{t \uparrow 1} \widehat{X}_{1,t}^{(\ell)} = E(w^{(\ell)\star}). \end{aligned}$$

□

A.4 HELLINGER–SVD EMBEDDINGS

Hellinger Geometry. Let $a, b \in \mathcal{V}$ index tokens and $c = (j, h) \in \{-1, +1\} \times \mathcal{V}$ index a neighboring context, where h is the preceding token for $j = -1$ and the following token for $j = +1$. Let N_{ac} count occurrences of token a with context c, and let $N_a = \sum_c N_{ac}$. We define $q_C(c) \propto \sum_a N_{ac} + \eta$, normalized over contexts, with $\eta = 10^{-2}$ ensuring full support. Smoothing toward this marginal stabilizes estimates for rare tokens; held-out context prediction log loss selects $\tau = 2000$ (Appendix B.1). The smoothed context distribution is $Q_a(c) = (N_{ac} + \tau q_C(c))/(N_a + \tau)$, which satisfies

$$\sum_c Q_a(c) = \frac{\sum_c N_{ac} + \tau \sum_c q_C(c)}{N_a + \tau} = 1.$$

The squared Hellinger distance is

$$\begin{aligned} d^2_{\mathrm{Hel}}(Q_a, Q_b) &\coloneqq \frac{1}{2} \sum_c \left(\sqrt{Q_a(c)} - \sqrt{Q_b(c)} \right)^2 \\ &= 1 - \sum_c \sqrt{Q_a(c) Q_b(c)}. \end{aligned}$$

Therefore, with $H_{ac} = \sqrt{Q_a(c)}$,

$$\|H_{a,:} - H_{b,:}\|^2 = 2 d^2_{\mathrm{Hel}}(Q_a, Q_b).$$

Weighted Low-Dimensional Projection. Define $A = H - \mathbf{1}\bar{H}$ and $B = \mathrm{diag}(\sqrt{\rho})A$, where $\rho_a = N_a / \sum_b N_b$, $\bar{H} = \sum_a \rho_a H_{a,:}$, and $\mathbf{1} \in \mathbb{R}^V$ is the all-ones column. Let $P_d \in \mathbb{R}^{2V \times d}$ contain the leading d right singular vectors of B. The projected embeddings are

$$\widetilde{E}(a) = A_{a,:} P_d.$$

This subspace minimizes the frequency-weighted reconstruction error; we approximate it using randomized SVD. Projection changes the Hellinger geometry only through the discarded component:

$$\|\widetilde{E}(a) - \widetilde{E}(b)\|^2 = 2 d^2_{\mathrm{Hel}}(Q_a, Q_b) - \|(H_{a,:} - H_{b,:})(I - P_d P_d^\top)\|^2.$$

Sparse Computation. Define

$$\lambda_a = \sqrt{\frac{\tau}{N_a + \tau}}, \qquad \nu_c = \sqrt{q_C(c)}, \qquad S_{ac} = \frac{\sqrt{N_{ac} + \tau q_C(c)} - \sqrt{\tau q_C(c)}}{\sqrt{N_a + \tau}}.$$

Then

$$\begin{aligned} H_{ac} &= \frac{\sqrt{N_{ac} + \tau q_C(c)}}{\sqrt{N_a + \tau}} \\ &= S_{ac} + \frac{\sqrt{\tau q_C(c)}}{\sqrt{N_a + \tau}} = S_{ac} + \lambda_a \nu_c, \end{aligned}$$

so $H = S + \boldsymbol{\lambda}\boldsymbol{\nu}^\top$. Since $S_{ac} = 0$ whenever $N_{ac} = 0$, S has the sparsity pattern of the count matrix. The weighted mean is

$$\bar{H} = \rho^\top S + (\rho^\top \boldsymbol{\lambda})\boldsymbol{\nu}^\top.$$

Writing $W = \mathrm{diag}(\sqrt{\rho})$, the matrix products needed for randomized SVD are

$$\begin{aligned} B\mathbf{x} &= W\left[S\mathbf{x} + \boldsymbol{\lambda}(\boldsymbol{\nu}^\top \mathbf{x}) - \mathbf{1}(\bar{H}\mathbf{x})\right], \\ B^\top \mathbf{y} &= S^\top W\mathbf{y} + \boldsymbol{\nu}(\boldsymbol{\lambda}^\top W\mathbf{y}) - \bar{H}^\top(\mathbf{1}^\top W\mathbf{y}). \end{aligned}$$

Finally,

$$\widetilde{E} = SP_d + \boldsymbol{\lambda}(\boldsymbol{\nu}^\top P_d) - \mathbf{1}(\bar{H}P_d).$$

These expressions include centering and frequency weighting without materializing the dense vocabulary-by-context matrix.

Final Normalization. Let $E^\dagger(a)$ denote the projected rows after perturbing exact duplicates. With corpus unigram weights $\pi(a) \geq 0$ satisfying $\sum_a \pi(a) = 1$, define

$$\mu_\pi = \sum_a \pi(a) E^\dagger(a), \qquad \sigma_\pi^2 = \sum_a \pi(a) \|E^\dagger(a) - \mu_\pi\|^2, \qquad E(a) = \frac{\sqrt{d}}{\sigma_\pi}\big(E^\dagger(a) - \mu_\pi\big).$$

Then

$$\sum_a \pi(a) E(a) = \frac{\sqrt{d}}{\sigma_\pi}\left(\sum_a \pi(a) E^\dagger(a) - \mu_\pi \sum_a \pi(a)\right) = 0,$$
$$\sum_a \pi(a) \|E(a)\|^2 = \frac{d}{\sigma_\pi^2} \sum_a \pi(a) \|E^\dagger(a) - \mu_\pi\|^2 = d.$$

The weights ρ determine the projection subspace, while π determines the final centering and scale.

B Experimental Details

B.1 Data Preparation and Fixed Embeddings

Corpus Split and Sequence Packing. We use the OpenWebText training split, reserving the last 100,000 documents for validation and training on the rest. Each document is tokenized with the GPT-2 tokenizer and followed by an EOS token, and documents are concatenated within each preprocessing batch. We divide the stream into blocks of 1,022 tokens, discard the incomplete remainder, and add BOS and EOS to obtain sequences of length 1,024; GPT-2 uses the same token ID for both. Token distributions are defined over the 50,257-token GPT-2 vocabulary, excluding padding entries.

Offline Statistics and Smoothing Selection. Embedding construction uses packed training windows sampled from contiguous chunks spread across the corpus, with a quarter of the chunks held out from count estimation for offline embedding evaluation. Adjacent-token pairs are counted within packed windows, including special tokens. The smoothing strength of the two-sided context distribution $Q_a(c)$ of Appendix A.4 is selected by held-out log loss on validation windows, giving $\tau = 2000$.

Default Hellinger–SVD Embedding. The default embedding follows Appendix A.4: we retain 32 components of a randomized SVD, perturb exactly duplicated rows with small Gaussian noise, and apply the unigram-weighted centering and global rescaling described there. The embedding is then frozen. Table 1 lists the representation and sampling endpoint of each reported configuration.

Table 1: **Fixed embedding configurations.** Each representation has unigram-weighted mean squared norm 32 and its own precomputed training-time distribution. t_{end} is the endpoint of the training-time distribution.

Embedding	Reported configuration	Dimension	t_{end}
Hellinger–SVD	Default	32	0.943
Gaussian	Small embedding ablation	32	0.620
Scaled one-hot	Small embedding ablation	50,257	0.557

Gaussian and One-Hot Controls. For the Gaussian control, we draw independent $\mathcal{N}(0, I_{32})$ rows, subtract their unigram-weighted mean, and rescale them to unigram-weighted mean squared norm 32. For the one-hot control, we use the uncentered embeddings $E(a) = \sqrt{32}\, e_a$, where e_a is the vocabulary basis vector, matching the mean squared norm of the 32-dimensional representations. The one-hot input projection increases the model to approximately 215M parameters.

Precomputation of Training-Time Distributions. We build the information-arc-length schedule separately for each fixed representation at its final scale. The calculation involves a single position, so we suppress (ℓ) and write $X_0 \in \mathbb{R}^d$ for the noise at that position. With $\gamma = t/(1-t)$, the rescaled Gaussian channel is $Y_\gamma = X_t/(1-t) = \gamma E(w) + X_0$, where $X_0 \sim \mathcal{N}(0, I_d)$. For a token prior ν, the analytic posterior is

$$r_{\gamma,\nu}(w \mid Y_\gamma) = \frac{\nu(w)\exp\!\left(-\frac{1}{2}\|Y_\gamma - \gamma E(w)\|^2\right)}{\sum_{u\in\mathcal{V}} \nu(u)\exp\!\left(-\frac{1}{2}\|Y_\gamma - \gamma E(u)\|^2\right)}, \tag{10}$$

the positionwise reference posterior at $t = \gamma/(1+\gamma)$. We use the corpus unigram prior π and a smoothed bigram prior ν_h conditioned on the preceding clean token h. On held-out adjacent-token pairs (h, w) and shared Gaussian noise draws, we estimate the expected posterior log-probability gains over these priors:

$$\begin{aligned} R_{\mathrm{c}}(\gamma) &= \widehat{\mathbb{E}}\left[\log \frac{r_{\gamma,\nu_h}(w \mid Y_\gamma)}{\nu_h(w)}\right], \\ R_{\mathrm{u}}(\gamma) &= \widehat{\mathbb{E}}\left[\log \frac{r_{\gamma,\pi}(w \mid Y_\gamma)}{\pi(w)}\right]. \end{aligned} \tag{11}$$

The conditional curve measures identification with local context, using the clean neighbor as an offline proxy for context, while the unigram curve retains the context-free identification tail.

The endpoint γ_{end} is the smallest channel level at which the unigram posterior identifies the token almost surely, with top-1 error and entropy relative to the prior both at most 10^{-3}; for the default embedding, $t_{\mathrm{end}} = \gamma_{\mathrm{end}}/(1+\gamma_{\mathrm{end}}) \approx 0.943$. We evaluate both gain curves on a grid $0 = \gamma_0 < \gamma_1 < \cdots < \gamma_J = \gamma_{\mathrm{end}}$ with logarithmically spaced positive levels, rescale each to increase monotonically from 0 to 1 (using isotonic regression), and write $\mathbf{r}_j = (r_{\mathrm{c},j}, r_{\mathrm{u},j})$ for the rescaled pair at γ_j. Their normalized cumulative arc length defines

$$\begin{aligned} \Delta a_j = \|\mathbf{r}_j - \mathbf{r}_{j-1}\|_2, \quad u_j = \frac{\sum_{k=1}^{j}\Delta a_k}{\sum_{k=1}^{J}\Delta a_k}, \quad u_0 = 0, \\ t = \frac{\gamma(U)}{1+\gamma(U)}, \qquad U \sim \mathcal{U}(0,1), \end{aligned} \tag{12}$$

where $\gamma(u)$ linearly interpolates the inverse map from u_j to γ_j. The schedule is computed once from corpus statistics and the fixed embedding, without evaluating the flow network. We stratify U across the global training batch and apply no time-dependent loss weight.

B.2 Model Architecture and Training

Model Architecture. We parameterize the neural residual in ReLF with a bidirectional diffusion Transformer on sequences of 1,024 tokens, using the fixed 32-dimensional embeddings of Table 1. At each position, we concatenate the noisy embedding with a 32-dimensional self-conditioning estimate of the clean embedding and project the result to the Transformer width. Each block combines multi-head self-attention with rotary positional embeddings (RoPE) (Su et al., 2024), query–key RMS normalization, a sigmoid gate on the attention output (Qiu et al., 2025), and a SwiGLU feed-forward network. Both sublayers use RMS normalization with time-dependent affine modulation and residual gates, driven by a sinusoidal time embedding followed by a two-layer MLP, and dropout 0.1 on the residual branches. A final time-modulated RMS normalization and linear projection produce vocabulary logits, to which we add the analytic singular logits with the unigram reference prior, except in the ablation that removes this term. Table 2 summarizes the configurations. Small, Medium, and Large match the hidden widths and depths of ELF-B, ELF-M, and ELF-L (Hu et al., 2026), which we accordingly call ELF-Small, ELF-Medium, and ELF-Large. We implement attention with FlashAttention (Dao et al., 2022; Dao, 2024); for compatibility with its backward kernels, Medium and Large use 11 and 20 heads instead of ELF's 16, while Small keeps ELF's 12.

Model Training. All reported models use a global batch of 1,024 sequences with the preprocessing of Appendix B.1. We minimize token-averaged cross-entropy, drawing interpolation times by stratified inverse-CDF sampling from the embedding's precomputed training-time distribution on $[0, t_{\mathrm{end}}]$, with no time-dependent loss weight. For self-conditioning, a forward pass without gradient tracking estimates the clean embedding at the same noisy input and time; independently for each

Table 2: **Architecture configurations of ReLF.** Parameter counts include all trainable projections, attention gates, and time-conditioning layers, but exclude the fixed token embeddings. FFN width denotes the intermediate dimension of the SwiGLU network.

Model	Parameters (M)	Layers	Hidden width	Attention heads	FFN width
Small	138	12	768	12	2,048
Medium	421	24	1,056	11	2,816
Large	784	32	1,280	20	3,456

sequence, we keep this estimate with probability 0.5 and otherwise replace it with zero. We optimize hidden, time-embedding, and modulation matrices with Muon (Nesterov momentum 0.95, weight decay 0.02), and the input and output projections, normalization gains, and biases with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.99$, no weight decay). Both use a base learning rate of 3×10^{-4} with 2,500 linear warmup steps followed by a constant schedule; Muon matches its update RMS to AdamW. Training uses bfloat16 mixed precision, gradient clipping at norm 1.0, and an EMA with decay 0.9999 for evaluation.

Reflow Training. Couplings are generated from the EMA weights of θ_k with 256-NFE Euler sampling, at guidance scale $\omega = 1.3$ in the first round and without guidance ($\omega = 1$) in the second, giving about 0.5M sequence pairs per round. Each round fine-tunes for 4,000 steps with batch size 512 and learning rate 10^{-4}. Eq. (8) uses $\lambda = 0.7$ in the first round and $\lambda = 0.55$ in the second.

Compute Accounting. A training run of the default Small configuration takes approximately 300 NVIDIA GH200 GPU-hours.

Numerical Precision. The Transformer residual stream and vocabulary projection run in bfloat16, while the flow state, analytic singular logits, and posterior embedding expectations stay in FP32. Network logits are promoted to FP32 before the analytic term is added, and the softmax and probability-weighted embedding sum are computed in FP32. This limits rounding of token-logit differences as the singular term grows near the endpoint and avoids storing small integration increments in bfloat16. Sampling updates, the self-conditioning carry, and the momentum-guidance velocity average also remain in FP32. Grid nodes and interval coefficients are computed in FP64.

Memory-Efficient Vocabulary Operations. We limit the memory of vocabulary-sized tensors by processing token positions in chunks. During training, each chunk of hidden features is projected to vocabulary logits, augmented with the analytic term, and passed to a fused cross-entropy kernel. A custom autograd operation computes each chunk's hidden-feature and output-head gradients immediately and saves them for the outer backward pass, so training never materializes the full logit tensor. The self-conditioning pass likewise computes embedding expectations chunk by chunk. During sampling, the network produces the full bfloat16 logit tensor, but singular-logit addition, softmax, and the embedding expectation are performed in FP32 chunks, and final decoding applies a chunkwise argmax.

B.3 Training-Time Self-Conditioning Guidance

We write $l_\theta(X_t, t; \boldsymbol{C}, \omega) = s(\cdot; X_t, t) + f_\theta(\cdot; X_t, t; \boldsymbol{C}, \omega)$ for the full logits, including the analytic singular term, of a network that also receives the guidance scale ω. Throughout this subsection, w denotes the clean target token.

Explicit Self-Conditioning Guidance. A self-conditioned model defines two posteriors at each noisy state: a cold posterior with $\boldsymbol{C} = \mathbf{0}$ and a warm posterior conditioned on the carried clean estimate. Contrasting them in logit space gives

$$l_\omega = l(\mathbf{0}) + \omega\big(l(\boldsymbol{C}) - l(\mathbf{0})\big), \qquad p_\omega(\cdot) \propto p(\cdot \mid \mathbf{0})^{1-\omega}\, p(\cdot \mid \boldsymbol{C})^{\omega}, \tag{13}$$

where $\omega = 1$ recovers unguided self-conditioning. The singular logits are identical in both passes, so guidance acts only on the network residual. The clean estimate $\widehat{X}_1 = \sum_w E(w)\, p_\omega(w)$ determines

the velocity and is carried as the next $\boldsymbol{C}$. Applied explicitly, Eq. (13) requires a cold and a warm evaluation at every step after the first, nearly doubling the sampling cost.

Scale Conditioning. To obtain guided predictions from a single evaluation, the network receives ω as an input. For each training sequence, we draw ω log-uniformly on $[1.5, 6]$, the same distribution as in ELF. The scale is encoded by a sinusoidal embedding and a two-layer MLP, mirroring the time embedding, and added to the time embedding that drives the adaptive normalization and residual gates. The final MLP layer is zero-initialized, so the network initially ignores ω.

Training Objective. At each training step, we evaluate the online network twice without gradient tracking at (X_t, t, ω), with dropout disabled. The cold pass gives $l^{\mathrm{u}} = l_\theta(X_t, t; \mathbf{0}, \omega)$ and the clean estimate $\widehat{X}_1^{\mathrm{u}}$; the warm pass, conditioned on $\widehat{X}_1^{\mathrm{u}}$, gives $l^{\mathrm{c}} = l_\theta(X_t, t; \widehat{X}_1^{\mathrm{u}}, \omega)$. With a per-sequence mask $k \sim \mathrm{Bernoulli}(1/2)$ shared across positions, the gradient-carrying pass uses $\boldsymbol{C} = k\,\widehat{X}_1^{\mathrm{u}}$, and we minimize

$$\mathcal{L}(\theta) = \mathbb{E}\big[-\log \mathrm{softmax}\big(l_\theta(X_t, t; \boldsymbol{C}, \omega) - \Delta\big)_w\big], \qquad \Delta = k\big(1 - \tfrac{1}{\omega}\big)\,\mathrm{sg}\big[l^{\mathrm{c}} - l^{\mathrm{u}}\big]. \tag{14}$$

The singular logits cancel in Δ, and gradients flow only through the unshifted logits $l_\theta(X_t, t; \boldsymbol{C}, \omega)$. Sequences with $k = 0$ therefore receive the ordinary CE objective, and sequences with $k = 1$ receive CE on shifted logits, so the loss remains a single token-averaged cross-entropy.

Stationary Point. Let l^* denote the log-posterior that CE fits for the self-conditioned input without the shift. For $k = 1$, CE drives $l_\theta(\cdot; \boldsymbol{C}, \omega) - \Delta$ toward l^* up to an additive constant. At a stationary point where the no-gradient warm pass agrees with the gradient-carrying pass, $l^{\mathrm{c}} = l_\theta(\cdot; \boldsymbol{C}, \omega)$, so

$$\tfrac{1}{\omega}\, l_\theta(\cdot; \boldsymbol{C}, \omega) + \big(1 - \tfrac{1}{\omega}\big)\, l^{\mathrm{u}} = l^* \quad \Longrightarrow \quad l_\theta(\cdot; \boldsymbol{C}, \omega) = l^{\mathrm{u}} + \omega\big(l^* - l^{\mathrm{u}}\big),$$

again up to an additive constant. A single warm evaluation at scale ω thus reproduces the guidance of Eq. (13).

Relation to ELF. ELF (Hu et al., 2026) uses the same construction, conditioning the network on a per-example guidance scale and building the target from its own stop-gradient cold and warm predictions, but applies the shift to an MSE velocity target, $\boldsymbol{v} + (1 - 1/\omega)\ \mathrm{sg}[\boldsymbol{v}_\theta(\boldsymbol{C}) - \boldsymbol{v}_\theta(\mathbf{0})]$. Its guidance therefore extrapolates the clean-embedding estimate linearly, whereas ours extrapolates the token logits as in Eq. (13) and takes the clean estimate as the mean embedding under the guided posterior p_ω; the two generally differ. ELF also trains a separate unguided CE decoding branch, while our guidance enters the only training objective.

Training and Inference. Models with training-time guidance are trained from scratch with the recipe of Appendix B.2. Relative to plain self-conditioning, each training step needs one additional forward pass without gradient tracking, since the cold pass also supplies the self-conditioning estimate. At inference, we feed a fixed ω to every evaluation, including the first and the final decoding evaluation, use one function evaluation per step, and carry the guided clean estimate as the self-conditioning input.

B.4 Evaluation and Sampling

Evaluation Protocol. For each plotted point, we generate 1,024 unconditional sequences of 1,024 tokens from the EMA parameters of the final checkpoint. We report generative perplexity and mean sample entropy. For generative perplexity, we decode the samples to text, retokenize them with the GPT-2 Large evaluator's tokenizer, truncate to at most 1,024 tokens, and exponentiate the mean next-token negative log-likelihood over all scored tokens, which include non-EOS targets and the first EOS in each sequence. For diversity, we compute the unigram entropy of each generated sequence in nats and average across samples.

Inference Settings. We sample with a nonuniform time grid and a time-dependent SDE noise level, tuned once on the default Small model at 64 NFE and shared across model sizes; Euler sampling uses the same grid. In the sampling-budget ablation, only the number of grid points changes. Self-conditioning starts from zero and carries the previous clean estimate between steps, so each step

requires one function evaluation, and the last evaluation decodes each position by argmax. Momentum guidance (Liao et al., 2026) extrapolates the current velocity away from an exponential moving average of previous velocities; we fix its base decay to $\beta = 0.8$ and sweep the guidance strength $\alpha \in \{0, 0.1, 0.2, 0.3, 0.5, 0.7\}$, where $\alpha = 0$ is the unguided baseline.

Finite-Time Token Decoding. Singular logits guarantee one-hot commitment as $t \uparrow 1$ under the assumptions of Appendix A.3. Our sampler uses N evaluation times $t_0 < t_1 < \cdots < t_{N-1}$; write $Z_i \in \mathbb{R}^{d\times L}$ for the numerical state at t_i and $\boldsymbol{C}_i$ for its self-conditioning input. The final evaluation is at $t_{N-1} < 1$, and we decode each position by

$$\widehat{w}^{(\ell)} = \arg\max_{w\in\mathcal{V}} p_\theta^{(\ell)}(w \mid Z_{N-1}, t_{N-1}; \boldsymbol{C}_{N-1}).$$

Finite-Step SDE Update. The clean prediction gives the Gaussian score $\boldsymbol{s}_\theta = (t\widehat{X}_1 - Z_t)/(1-t)^2$, with arguments omitted. We sample using the score-corrected SDE (Ma et al., 2024)

$$\mathrm{d}Z_t = \big[\boldsymbol{v}_\theta(Z_t, t) + \sigma(1-t)^2 \boldsymbol{s}_\theta(Z_t, t)\big]\,\mathrm{d}t + \sqrt{2\sigma}\,(1-t)\,\mathrm{d}\boldsymbol{W}_t, \qquad \sigma \geq 0, \tag{15}$$

where $\boldsymbol{W}_t$ is standard Brownian motion and σ controls stochasticity; $\sigma = 0$ recovers the flow ODE. We discretize Eq. (15) by refreshing the estimated Gaussian source component. Initialize $Z_0 \sim q_0$ and $\boldsymbol{C}_0 = \boldsymbol{0}$. For $i = 0, \ldots, N-2$, let $h_i = t_{i+1} - t_i$, $b_i = 1 - t_i$, $\widehat{X}_{1,i} = \widehat{X}_1(Z_i, t_i; \boldsymbol{C}_i)$, and $\widehat{\boldsymbol{\epsilon}}_i = (Z_i - t_i\widehat{X}_{1,i})/b_i$, and let $\boldsymbol{\xi}_i \in \mathbb{R}^{d\times L}$ be fresh standard Gaussian noise. The update is

$$\begin{aligned} \delta_i &= \min\{\sigma h_i b_i,\, b_i\}, \\ \widetilde{Z}_i &= Z_i - \delta_i \widehat{\boldsymbol{\epsilon}}_i + \sqrt{b_i^2 - (b_i - \delta_i)^2}\,\boldsymbol{\xi}_i, \\ Z_{i+1} &= \widetilde{Z}_i + \frac{h_i}{b_i}\big(\widehat{X}_{1,i} - \widetilde{Z}_i\big). \end{aligned} \tag{16}$$

The first update replaces part of the estimated source noise with fresh noise, and the second moves toward the same clean prediction without another network call. We set $\boldsymbol{C}_{i+1} = \widehat{X}_{1,i}$. As $h_i \to 0$, the drift and diffusion coefficients converge to those of Eq. (15), and $\sigma = 0$ gives Euler integration of the flow ODE. The grid has N evaluations and $N-1$ state updates; the last evaluation performs token decoding.

When momentum guidance is enabled, we compute $\boldsymbol{v}_i = (\widehat{X}_{1,i} - Z_i)/b_i$ and form $\boldsymbol{v}_i^{\mathrm{g}} = (1+\alpha)\boldsymbol{v}_i - \alpha\boldsymbol{m}_i$, where the velocity average $\boldsymbol{m}_i$ is initialized by $\boldsymbol{m}_0 = \boldsymbol{v}_0$ and updated as $\boldsymbol{m}_{i+1} = (1-\beta^{Nh_i})\boldsymbol{v}_i + \beta^{Nh_i}\boldsymbol{m}_i$. We replace $\widehat{X}_{1,i}$ in Eq. (16), including in $\widehat{\boldsymbol{\epsilon}}_i$, by $Z_i + b_i\boldsymbol{v}_i^{\mathrm{g}}$, while self-conditioning carries the unguided prediction $\widehat{X}_{1,i}$.

B.5 Ablation Settings

Singular Logits and Embeddings. The models in Figure 4(a) and (b) are trained for 80,000 steps without training-time guidance and evaluated at 64 NFE with the momentum-guidance sweep. They use a logit-normal time grid ($\mu = -0.8$, $\sigma = 1.2$) on $[0, t_{\mathrm{end}}]$ with a constant SDE noise scale of 4; Euler sampling uses the same grid. The Gaussian and one-hot controls are specified in Appendix B.1.

Training Objective. The MSE variant keeps the network, its logit output, and the clean estimate $\widehat{X}_1 = \sum_w E(w)p_\theta(w \mid X_t, t)$ of the CE model, and trains the velocity $\boldsymbol{v}_\theta = \frac{1}{1-t}(\widehat{X}_1 - X_t)$ with the RF loss of Section 2, averaged over positions and embedding coordinates. Interpolation times follow the same training-time distribution, with no additional weight. For training-time guidance, we reuse the two no-gradient passes of Eq. (14) but, as in ELF, shift the velocity target by $(1 - 1/\omega)\, k\,\mathrm{sg}[\boldsymbol{v}_\theta(\widehat{X}_1^{\mathrm{u}}) - \boldsymbol{v}_\theta(\boldsymbol{0})]$ rather than the logits. Both models are evaluated at 80,000 training steps with the same sampler.

Flow Embedding Dimension. Each d-dimensional embedding follows the construction of Appendix B.1 with d retained singular directions, and receives its own training-time distribution and endpoint t_{end}, which decreases from 0.969 at $d = 16$ to 0.731 at $d = 256$.

Table 3: **Flow embedding dimension sweep for ReLF-Small.** Values plotted in Figure 4(d); SDE sampling at 64 NFE. For each d, the first row gives the guidance scale ω and the second the Gen. PPL / entropy.

d		Gen. PPL / entropy				
16	ω	1.7	1.8	1.9	2.0	2.1
		46.5 / 5.32	43.0 / 5.31	39.7 / 5.30	36.7 / 5.30	33.9 / 5.29
32	ω	1.7	1.8	1.9	2.0	2.1
		24.1 / 5.45	23.6 / 5.45	19.1 / 5.37	14.9 / 5.22	11.6 / 5.07
64	ω	2.2	2.4	2.6	2.8	3.0
		25.4 / 5.51	23.7 / 5.48	22.0 / 5.43	20.4 / 5.38	19.3 / 5.32
128	ω	2.0	2.2	2.4	2.6	2.8
		37.6 / 5.57	35.0 / 5.55	32.2 / 5.52	29.7 / 5.48	27.3 / 5.42
256	ω	1.8	2.0	2.2	2.4	2.6
		56.6 / 5.61	52.1 / 5.61	48.2 / 5.59	44.5 / 5.59	42.1 / 5.58

Reflow Rounds. Figure 5 compares the base model at 32 NFE with the model after two reflow rounds at 16 NFE, each evaluated with 4,096 samples. With half the NFE, the reflowed model reaches lower Gen. PPL at every model size (26.4 versus 35.2 for Small, 22.2 versus 24.5 for Medium, and 19.4 versus 22.6 for Large), with entropy within 0.04 nats of the base model.

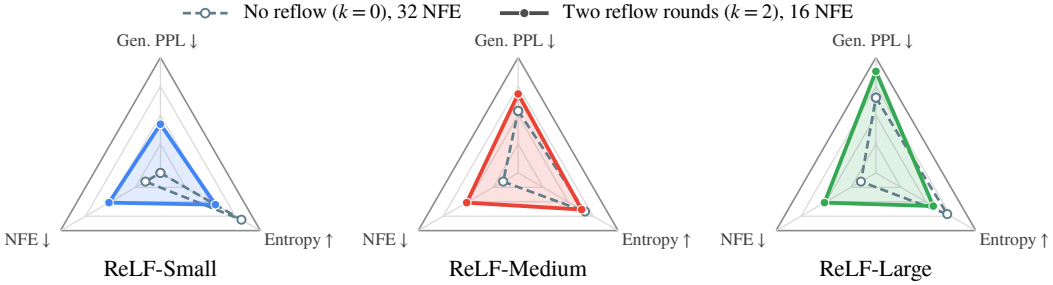


Figure 5: **Two reflow rounds at half the NFE.** Each panel compares the base model at 32 NFE (dashed) with the model after two reflow rounds at 16 NFE (solid), with vertices Gen. PPL, entropy, and NFE; outward is better on every axis. Gen. PPL is normalized over the six plotted runs, with the highest value at the center, and entropy and NFE over the reflow-round sweep ($k = 0$–3 at 8, 16, and 32 NFE for all sizes); Gen. PPL and NFE use a log scale.

B.6 Quality–Diversity Frontiers and Autoregressive Baselines

Extending the Small- and Medium-model curves in Figure 1(a), Table 4 reports the quality–diversity frontiers of Small, Medium, and Large models trained with training-time guidance. We evaluate EMA checkpoints using SDE sampling with noise scale 24 and 64 NFE, including decoding, with 1,024 sequences per configuration. For each model size, we keep the non-dominated points: a point is dominated if another setting achieves at least as high an entropy and at most as high a Gen. PPL, with strict improvement in at least one. Because ReLF-Large varies little across guidance scales, only two of its settings are non-dominated; Table 5 lists all eight.

The table also lists the six temperature settings of each AR baseline shown in Figure 1(a). AR-Small, AR-Medium, and AR-Large are causal Transformers with the hidden widths and depths of the corresponding ReLF models (Table 2) and have 124M, 374M, and 699M parameters. We evaluate their EMA weights after 80,000, 200,000, and 200,000 training steps, respectively, generating each length-1,024 sequence ancestrally after BOS and scoring it with the same metrics.

Table 4: **OpenWebText Pareto results and autoregressive baselines.** ReLF uses 64 NFE; its frontiers are computed separately within each model size over $\omega \in \{1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.3\}$. AR baselines use ancestral sampling over the temperatures shown in Figure 1(a). Entropy is mean per-sequence unigram entropy; Gen. PPL is evaluated by GPT-2 Large.

Model	Guidance ω	Entropy (nats) ↑	Gen. PPL ↓
ReLF-Small	1.7	5.45	24.1
	1.8	5.45	23.6
	1.9	5.37	19.1
	2.0	5.22	14.9
	2.1	5.07	11.6
	2.3	4.94	9.8
ReLF-Medium	1.6	5.48	19.3
	1.7	5.48	18.6
	1.8	5.47	17.8
	1.9	5.45	16.8
	2.0	5.41	15.6
	2.1	5.36	14.3
	2.3	5.31	13.3
ReLF-Large	2.0	5.51	17.6
	1.9	5.51	17.3
Model	**Temperature T**	**Entropy (nats) ↑**	**Gen. PPL ↓**
AR-Small	1.00	5.54	38.1
	0.98	5.46	30.4
	0.96	5.40	25.0
	0.94	5.32	20.9
	0.92	5.25	17.4
	0.90	5.20	15.0
AR-Medium	1.00	5.54	27.9
	0.98	5.46	22.3
	0.96	5.41	18.5
	0.94	5.36	15.7
	0.92	5.29	13.4
	0.90	5.24	11.7
AR-Large	1.00	5.53	24.3
	0.98	5.47	20.2
	0.96	5.41	16.8
	0.94	5.36	14.3
	0.92	5.31	12.4
	0.90	5.26	11.0

Table 5: **Full guidance-scale sweep for ReLF-Large.** All eight settings used for the ReLF-Large frontier in Table 4, with the same evaluation protocol. †Non-dominated settings listed in Table 4.

Guidance ω	Entropy (nats) ↑	Gen. PPL ↓
1.5	5.49	18.0
1.6	5.50	17.7
1.7	5.50	17.4
1.8	5.50	17.4
$1.9^{\dagger}$	5.51	17.3
$2.0^{\dagger}$	5.51	17.6
2.1	5.50	17.4
2.3	5.48	17.7

Few-Step Comparison. The distilled ELF-Small curve in Figure 4(f) is ELF+PD, the progressively distilled ELF-Small of Hu et al. (2026). Its reported few-step settings reach low Gen. PPL only at entropy 5.04–5.16, so we compare at matched entropy instead. For each model and NFE, we sweep the solver and guidance scale (the sampling temperature for AR-Small) and report the lowest Gen. PPL among settings with entropy at least 5.40, using 4,096 sequences per setting. FMLM (Lee et al., 2026) reports a single setting per NFE, so we plot its reported OpenWebText results directly and label each point with its entropy. The plotted ReLF points lie at entropy 5.41–5.45. Table 6 lists all plotted values.

Table 6: **Values plotted in Figure 4(f).** ReLF rows use two reflow rounds. Except for FMLM, each entry is the lowest Gen. PPL found with entropy at least 5.40. FMLM values are as reported by Lee et al. (2026).

Model	NFE	Entropy (nats) ↑	Gen. PPL ↓
ReLF-Small	4	5.41	116.4
	8	5.42	51.2
	16	5.41	26.4
	32	5.43	22.9
ReLF-Medium	4	5.42	113.7
	8	5.42	42.7
	16	5.42	22.2
	32	5.45	18.4
ReLF-Large	4	5.43	117.5
	8	5.43	40.7
	16	5.41	19.4
	32	5.43	16.2
ELF+PD	4	5.40	68.7
	8	5.42	66.4
	16	5.41	67.7
	32	5.41	66.3
FMLM	4	5.26	111.3
	8	5.36	86.5
	16	5.29	63.6
	32	5.25	45.1
AR-Small	1,023	5.43	27.7

C Qualitative Examples

We show unconditional samples from ReLF-Small, ReLF-Medium, and ReLF-Large, all trained with training-time guidance and generated with the SDE sampler at 64 NFE. ReLF-Small and ReLF-Medium use guidance scale $\omega = 1.7$, the setting reported in Section 4.2, which attains Gen. PPL 24.1 at entropy 5.45 for Small and 18.6 at entropy 5.48 for Medium over 1,024 samples. ReLF-Large uses $\omega = 1.9$, which attains Gen. PPL 17.3 at entropy 5.51. Each sample is a sequence of 1,024 GPT-2 tokens decoded to text without any post-processing; <|endoftext|> denotes the end-of-sequence token. The header of each sample reports its own Gen. PPL and entropy.

ReLF-Small, sample 1 Gen. PPL 20.3 Entropy 5.55

```
<|endoftext|> funding is a panacea.  We've seen an increase in
waterfalls on floodway projects over the years, which suggests that we
should invest more money in areas where river infrastructure is more
viable.

While New Mexico was once inundated with billions from natural
rainfall, it now is heavily impacted by flooding.  In recent years its
reservoirs have fallen more than 100 percent.  In relatively climate,
there is little reason to believe that the water bills will rise more
```

than 20 percent a year. Increased flood insurance provides a more convenient, easier-to-use way to reduce the burden of the DWF to the federal government. But it's not a panacea, either. This is what we can use right now. All elements of our infrastructure are aging, including water lines and electrical utilities. We certainly shouldn't make a repeat of new drainage infrastructure. In the absence of a reliable hydropower re drainage system, we're likely going to miss out on mudslower restoration.

The study provides evidence that providing better flood funding and reducing the risk of future flooding makes a more resilient society a realistic long-term alternative. In short, the best answer to the flooding crisis is by providing some public infrastructure spending—subsidized by both federal governments and private investors—to fund it in the first place.<|endoftext|>× Iraqi security forces raided Christian home of wife of a Islamic State leader

KANIZALAD, Iraq (AP) — Iraqi security forces raided the Christian home of a family of a Islamic State leader on Sunday in an effort to curb ISIS militants in the northern Iraqi province of Nineveh, including arresting more Christians, a spokesman for the group said.

Explosvers, also known as the Shirab, raided a Christian house in the village of Hawiyah, a community where dozens of Christians lived when the raid took place.

No injuries were immediately reported. Five people were arrested, and two were wounded.

Nineveh's Yazidis have fled to fight with ISIS to the north of the Tigris, which stretches along the porous border between Iraq and Syria. It is one of the largest strongholds for Sunni Christians in the region. Yazidis have been seeking wide cover from Islamic State since it staged deadly attacks on Mosul and the city of Kirkuk in June.

Christians of the Christian faith have pledged to join forces to fight ISIS in the region, state-run Anadol Agency reported Wednesday.

Explosvers raided an apartment in Samarija late Sunday, and arrested at least 18 members of the Shirab, an Iraqi security official told the Associated Press.

In Hawija province, 49 members of theShirab were arrested, the official told Iraqi state news agency Dogan on Tuesday. He said at least a dozen were taken into protective custody.

Anadol news agency on Monday reported that 20 Shirab members had been allowed to leave their homes, which remained in Hawija until the night of the raid.

Hassman Kamar, a spokesman for the Christian community of Hawija, said local Christians living with theShirab could face some eviction, especially if the Iraqi government maintains its support.

"In the Nineveh, local Christians used to live with the Shirab, so they now avoid support because they are no longer allowed to leave the area," Kamar said.

Read or Share this story online: https://usat.ly/2k9zqg<|endoftext|>A 28-year old, dual U.S. citizen has been indicted on two counts of impersonating a Dominican government official, as well as signing a false identification letter.

Rico Sino of the John Morris Community Court was charged in a two-count indictment. All charges threaten to defraud the U.S. government.

Originally from the Dominican Republic, Sino got his start in the subject after spending two years with Spanish Venezuela school club Monaxo.

He later asked permission to embark on his three-year journey into the United States. Sino's attorney Carlos Herníguez said to NJ Advance Media that he has never seen Sino do anything more illegal than signing a green card on the form of his $10,000, or peso, bail.

He has been on the wrong side of the issue 'for a very long time,' he said.

State Sen. Francisco Molco said to NJ Advance Media that he still believes Sino's arrest will help the feds go after him.

'We have to see what does. I am just going to see,' he said.

The Homeland Security Department did not respond to a request for comment on the charges.

The Senate recently approved a bill that allows dual residents of the Dominican Republic to apply for citizenship in the United States.

The law covers a Dominican citizen who will not be identified by real name but has a birth card and spousal<|endoftext|>

ReLF-Small, sample 2 Gen. PPL 28.3 Entropy 5.49

<|endoftext|> hot temperatures without the need for a 400-square-foot greenhouse that will be enough to cover 10,000 homes without electricity or water.

The Trump White House has announced a series of recovery centers to help communities in the nation's southern states overcome their dependence on fossil fuels.

So far, 1,080 units have been built so far and 895 people have been awarded permits. But high temperatures in the region are threatening key water conditions for millions of people without electricity or water.

Recovery Program Manager Bill Holley, executive director of the Union of Concerned Scientists, said the lack of adequate funding raises immediate questions, especially given that the region – about 30 percent of the state – suffers from extreme drought patterns.

Coupled with droughts, he said, "it's a very challenging time for temperatures to rise at high levels, especially when compared to normal temperatures. We're seeing record floods and famines, and that's something we're really concerned about," he said.

Gierra Diversity, an environmental nonprofit that helped scientists launch the Union of Concerned Scientists initiative, noted that 'as we have and do the right thing,' they will "both work together to be able to achieve more meaningful change."

Montley won't be alone. In Tucson, the group received $25,000 from Communities From All Climate. In Yuma, it will receive $5,000 from Community Recovery Foundation. Where Santa View is hit hardest by wildfires, it will receive $30,950 in grants.

In Maricopa County, Gulf Water Recovery Center was awarded $104,500 by Environmental Frontier Association for helping it meet the needs of 1,300 communities statewide. According to program coordinators for all of those counties, by the end of this year, the recovery centers will provide access to 538 people in all southern counties – the highest number since records began in 2000.

"A recovery center can be a step toward empowering our communities," said Lana Hatterer, program manager for Make Something Clean —, a nonprofit that looks at how and where new communities can be built. "President Trump's announcement to build these recovery centers fulfishes that optimism and strengthens the urgency to help us address the needs we currently face."

The facility will, in a variety of ways, mirror the established facilities that will replace now. In Oahuopa and Temosa, communities without enough water will be stuck in a bitterly water water for decades to come. In Sonima, communities without sufficient water will face water cuts and many rural communities will be trapped in the cold water for decades to come.

The communities in site will be divided into 10 different manufacturers. The unit will remain operational for years. But Corley said as the facilities become more sophisticated there's growing interest in making how and supplement them for their communities.

Each unit has two basic components:

An internal inverter system that generates power waves that can be controlled by an electrical outlet. A three-cooled lighting system can help adjust the switch.

Most of the buildings will have prices that vary from $1,500 to $1,050, Holley said. But with each unit, there will be an additional 110-square-foot greenhouse, the standard form factor for state-of-the-art climate panels.

Many of the centers will also be set up for local philanthropic organizations. Residents will receive grants from foundations affiliated with water communities in Arizona. For example, a grant from the Maricopa County Department of Public Services will be used to fund water education. A grant from the governor's mansion will go toward the Arizona Department of Agriculture.

'When people talk about this, they're talking about a recovery center,' Corley said. 'The center will support people in place to meet vital water and medical needs. It will give local communities renewed hope for their future."

As the oil market continues and demand runs high, so do the numbers of Americans who overdose from prescription opioids.

The National Institute on Drug Abuse estimated that overdoseizations associated with the opioid overdose crisis will increase from 1,080 to 841,080 by 2020, according to a study in Baltimore, Massachusetts. But opioid overdoses in the United States, including heroin overdoses, have climbed sharply and more than 10,300 Americans have died from overdoses since 2014 alone.

DAURA WCCENBERY

NeiComFounder

Carwater Action Foundation<|endoftext|>A top official on President Trump's "America First" campaign team had his license revoked after calling him a "bigoted homophobe."

Curtel Zaklow had finished his legal degree from Trump University, but a federal district judge ruled the Chicago graduate had his license revoked when he sued the University of<|endoftext|>

ReLF-Medium, sample 1 Gen. PPL 17.0 Entropy 5.33

<|endoftext|> cannabis research firm. The company was founded in Colorado by entrepreneur David Conler in 2014. Whitler is the Principal Medical Cannabis Advisor for the Colorado Department of Health and Community Services.

You are known for your work in cannabis research, innovation, and marketing. Can you talk a bit about your goals when it comes to cannabis and the firm's strategies to achieve those goals?

Whyte Group uses a multidisciplinary, research-driven business model and seeks to create solutions that deviate from three pillars of cannabis, with primary objectives: reducing harmful use and underage advertising, supporting the commercialization of FDA-approved products, and breaking down non-trivial barriers to entry. The Group understands the importance of the legal cannabis industry and believes the key for achieving success is "the science."

Who are you looking to hire?

In addition to hiring researchers and innovators, the Whyte Group employs lawyers and legal people to advise the unique issues within the cannabis industry. Our leadership team also has extensive experience with education, marketing, and pharmaceutical and fid Taskancy management.

How has the research been conducted with lawmakers across the country? How has it affected both your clients and the politicians themselves?

The Whyte Group has held cannabis policy discussions with legislators in 30 states since the opening of our office in January. These discussions revolve addressed a number of policy-related topics such as research, development, regulation, and regulation. Our Group also works with state and local marijuana leaders, helping to shape the discussion about the issue and help policy makers to find solutions to their problems. We hope these discussions broaden the dialogue between policymakers and experts on cannabis policy, and increase legislators' support for raising the legal age of all cannabis products to 21 years old.

When you're a scientist, what is it like to get an epiphany? What was an exciting epiphany you went through in your lab when you made groundbreaking discoveries? When you're an entrepreneur, what's it like to have an exciting epiphany?

The regulatory conversation around cannabis across the country is rapidly changing overnight. While the drug has now been legalized in 23 states, the cannabis industry remains much more active in regulation. Large scale use and distribution of marijuana can still be profited under various criminal agencies, and products that aren't sold as legitimate continue to be prohibited. The Whyte Group will now work hard to ensure that these problems evolve and the industry focuses on business models that will benefit the general consumer and create good-paying jobs. We've successfully responded to this shift, and continue to study the issue informally.

You're a small company, with 15 people working on research, development, marketing, sales, product management. What are the firm's goals when it comes to cannabis?

When you're an entrepreneur, cannabis is often one of the first priorities that you think of things that matter to you. That said, almost two years ago, we decided to form the Whyte Group, and we believe we have a significant job ahead of us moving forward. The Whyte Group's first goal is to create a world-class cannabis research laboratory. Our second is to utilize a multidisciplinary, research-driven business model. We will engage industry stakeholders by creating business models and products that deviate from these three

pillars. Creating a business model that is responsible, ethical, and minimally harmful is our third goal, and we seek to ensure cannabis is regulated and regulated the same way alcohol and tobacco. With these "3 pillars" in place, the Group will continue to engage people across the country to help the cannabis industry move forward.

The Whyte Group also believes the federal government should stop regulating the cannabis industry like it did in the tobacco industry. We believe that the government still holds sway over the regulatory process for cannabis, but is finally making some significant progress. The DEA is currently developing a list of pharmaceutical products that will be put on "Schedule A." CBD products like the Group's THC Caze Solution are also slated to be placed on Schedule B, bringing it one step closer to a Schedule A substance. In addition, the Drug Enforcement Administration is currently revising its Schedule Enforcement Controlled Drug list, which is below Schedule A. These policy changes are intended to try and take the stigma off cannabis now that federal regulation of cannabis is over.

Can you give your thoughts on the current situation in the cannabis industry?

We're cautiously optimistic, and that's for one big reason: With a market capitalization of over $1 trillion, the cannabis industry is the fastest growing industry in the country. It's poised to be not only the gateway to some of the top jobs in this country, but also an innovation engine. Many of those resources are still badly underutilized. We'll continue our efforts to create new, exciting cannabis<|endoftext|>

ReLF-Large, sample 1 Gen. PPL 19.2 Entropy 5.51

<|endoftext|> rural Internet service. This amendment was signed into law with the HEREAM Act in 2009. Arartz is an IEEE Fellow Emeritus of the American Institute of Electrical and Electronics Engineers.

Ginson is a researcher at North Carolina State University, where he's led a number of projects in lifesaving medicine, including developing the world's first portable dialhetic device.

Richard Kamer, a provinticisticist and former University College resident, is an IEEE Fellow Emeritus of the American Institute of Electrical and Electronics Engineers. His research has focused on constructing wired Bypass Wirelessways that provide real-speed access to the Internet at non-trivial cost to consumers. Levkin is the author of the acclaimed SmartCities, a book outlining building blocks for "smart cities."

What new graduates need to know?

In addition to offering scholarships and mentoring, the Letch Institute hosts conferences and workshops designed to help new graduates build upon their advanced degrees. Our flagship conference is held annually and attracts 1,000+ students from over 100 companies and countries. Our conference were named the 2014 World School Lecture of the Year and Young Leaders Worksation of the Year by Young Scouts America.

The Letch Institute also offers an online developer Development program that prepares students with the skills to become successful Web developers. Our classes are offered twice a year to web-agers ages 13 to 25, regardless of high location or income. Each day, students learn the skills and techniques that are required to use the Internet as a resource and create useful applications that enable others to access it. In addition to teaching students these tools and

skills, our online training program also offers free Master's and Doctorate Degree programs for students ages 18 to 35 years old.

When I've asked new graduates what it feels like to be a Letch Institute Fellow, the overwhelming response is, "I've just earned your degree, and you're still getting paid!"

What is the Rural Internet Innovation Fund?

The Rural Internet Innovation Fund is a federal program to help rural communities build more reliable Internet infrastructure so everyone in any parts of the country can get Internet for anywhere at home.

More than half of rural Americans already have high-speed Internet service connections, and they will be the most widely used if broadband continue to be expanding. The RIEF can promote rural growth by building more reliable Internet infrastructure in rural communities and by funding grants that will help those communities grow and have good-paying jobs. That's less waiting for cable TV and more opportunities to pitch your sweet cash pump.

Here's a great YouTube video about this initiative released on Jan. 21, 2014, and for much more information, see the President's FY14 State of the Union Address.

You're the president, CEO, and CEO of American River Code. What do those three positions mean to you?

After working at Silicon University, I decided to launch the Letch Institute, and since then I have been responsible for all of our leadership activities. The Letch Institute's core mission is to be the driving driving force for digital innovation. Our focus is to keep the innovation world open to all backgrounds and all perspectives. While we value intellectual property rights, we believe that human property everywhere belong in the digital realm. Creating a space where we can collaborate, connect, connect, and create is a huge gift, and we have to make sure those currently excluded from the digital world can experience it. Since the "Arab Spring" in 2010, digital rights movements have empowered ordinary people around the world to discuss the critical issues we face.

The Letch Institute also believes that American businesses should contribute to the digital economy like they contribute to the real economy. Contributing to the economy means supporting them by maximizing economic returns and creating wealth for consumers. For more information about our global mission, please visit the Letch Institute website: http://www.letchinstitute.org/.

Part of our nation's economic success is putting people to work to meet our true economy. Access to the Internet is key to economic growth and prosperity. In 2000, only 1 percent of the country had an Internet connection, but by 2012, that increased to 18 percent. We need our private courage to invest to make the Internet as accessible and affordable as possible for generations to come.

Have you done any work in the private sector or the public sector?

I'm an engineer, and I've been programming since 1985. My private sector career began at Ruralron (now Southern Grid, Inc.), a company that specializes in Web storage. I'm currently chief technology officer at Public Electric, which is the largest utility in the state, serving over 1 million customers. Many of those are economically underserved in rural Texas and don't have Internet access. My public sector career began<|endoftext|>