KNOW WHEN TO HOLD 'EM: CORRECT-TOKEN RETENTION IN UNIFORM-STATE DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Uniform-state diffusion models (USDMs) can revise any token at any denoising step, which lets them correct their own mistakes, a key advantage over masked diffusion. Self-correction, however, requires both revising incorrect tokens and retaining correct ones, and we show that current USDMs lack the latter. Even under greedy-tail decoding, state-of-the-art USDMs (DUO, UDLM, and uniform-noise SEDD) keep revising 173–270 of 512 positions at every step, and these large, uncoordinated edits collapse sample diversity. A random-token corruption experiment traces this deficit to the models themselves: they reconstruct clean and corrupted tokens with nearly identical accuracy, even though clean tokens are easier targets. A decomposition of the validation NELBO shows that training barely rewards retention: incorrect predictions are heavily penalized at corrupted positions but almost free at clean ones. We propose Correct-Token Retention Regularization (CTR-Reg), a simple but effective auxiliary loss that trains the model to retain tokens left unperturbed by the forward process and requires no change to the sampler. CTR-Reg improves clean-token accuracy by 26.5 percentage points on average across six benchmarks, while leaving corrupted-token accuracy virtually unchanged, and its per-step revisions converge to only 3–11 positions. With just five greedy-tail steps, generative perplexity more than halves under CTR-Reg for all three models while diversity is preserved, and these gains hold across sampling budgets. Our results identify correct-token retention as a key missing ingredient for self-correcting diffusion language models, and demonstrate an effective fix.

1 INTRODUCTION

Diffusion models have achieved remarkable success in image, video, and audio generation (Sohl-Dickstein et al., 2015; Ho et al., 2020; Polyak et al., 2025; Huang et al., 2023; Rombach et al., 2021; Blattmann et al., 2022), driven largely by iterative refinement: progressively correcting errors over many denoising steps yields realistic and diverse samples (Zhao et al., 2025; Dhariwal & Nichol, 2021; Song et al., 2023; Ma et al., 2025). Recent work has extended diffusion to discrete data, with promising results in language modeling (Sahoo et al., 2025; Shi et al., 2024; Lou et al., 2024; Austin et al., 2021): unlike left-to-right autoregressive models, diffusion language models decode many tokens in parallel using bidirectional context, enabling substantially faster generation while remaining competitive on downstream tasks (Labs et al., 2025; Song et al., 2025).

Discrete diffusion models interpolate between clean data x and a fixed prior π via the forward process $q_t(z_t \mid x; \alpha_t) = \mathrm{Cat}(z_t; \alpha_t x + (1 - \alpha_t)\pi)$, with a monotonically decreasing noise schedule $\alpha_t \in [0, 1]$ (Austin et al., 2021; Sahoo et al., 2024). The choice of π yields two families. Masked Diffusion Models (MDMs), or absorbing-state diffusion (Austin et al., 2021; Sahoo et al., 2024), set π to a `[MASK]` token: a masked position is revealed once during denoising and never revisited, permanently locking in any early mistake. Uniform-state Diffusion Models (USDMs) (Schiff et al., 2024; Sahoo et al., 2025) instead set π to the uniform distribution over the vocabulary; since a corrupted token is then indistinguishable from a clean one, the model must reconsider every position at every step, enabling self-correction (Deschenaux et al., 2026).

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

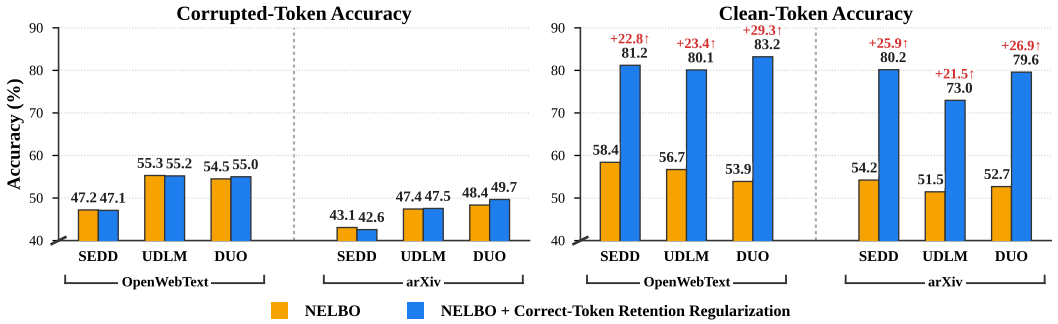

Figure 1: **Effect of CTR-Reg on corrupted- vs. clean-token reconstruction accuracy under random-token perturbation.** For SEDD, UDLM, and DUO, we compare base NELBO training (orange) to NELBO + Correct-Token Retention Regularization (CTR-Reg; blue), measuring reconstruction accuracy on corrupted tokens (left) and clean tokens (right), averaged over corruption rates $p \in \{10, 20, 30, 40\}\%$. Each panel shows OpenWebText and arXiv results, separated by a dotted line. CTR-Reg leaves corrupted-token accuracy essentially unchanged while improving clean-token accuracy by 21.5–29.3 percentage points, across all models and both corpora. All models are trained on 20B OpenWebText tokens (169M parameters, sequence length 512); evaluation uses the OpenWebText validation set ($\sim$110M tokens) and Scientific Papers (arXiv) validation set ($\sim$55M tokens).

Self-correction, however, requires two distinct capabilities: revising incorrect tokens and retaining correct ones (*correct-token retention*); we find that current USDMs lack the latter. The deficit surfaces under the *Greedy-Tail Sampler* of DUO (Sahoo et al., 2025), which appends one per-position argmax step after ancestral sampling; we also consider k such steps, each a single forward pass followed by a per-position argmax, with no stochastic sampling. Under this sampler, three state-of-the-art USDMs, DUO, UDLM (Schiff et al., 2024), and SEDD with uniform noise (Lou et al., 2024) (hereafter SEDD), exhibit a substantial collapse in sample diversity, measured by entropy (Figure 4).

To identify its source, Figure 2 tracks how many positions change prediction at each greedy-tail step: 173–270 of the 512 positions are revised at every step, with no sign of convergence. Such churn indicates a lack of correct-token retention: because positions are predicted independently rather than jointly, these large-scale simultaneous edits drive the collapse, since even locally plausible edits, applied by the hundreds at once, can compound into globally inconsistent sequences (Gu et al., 2018). Samplers such as ancestral sampling mitigate this by limiting the average amount of revision per step, but do not determine *which* positions are revised; absent a reliable model preference for correct tokens, this selection is governed by sampling stochasticity.

To disentangle revision from retention, we design a random-token corruption experiment (Figure 1): we split a corpus into 512-token chunks, replace a random $p\%$ of tokens in each with tokens drawn uniformly from the vocabulary, reconstruct each sequence with a single forward pass, and report accuracy separately on the corrupted $p\%$ and the clean $(100 - p)\%$ of tokens, averaged over $p \in \{10, 20, 30, 40\}$. For current state-of-the-art USDMs the two accuracies are nearly identical, even though clean tokens are the easier target at these corruption rates: the models modify tokens indiscriminately, failing to recognize and preserve already-correct ones.

We trace this deficit to the training objective: standard NELBO training provides little incentive for retention. Decomposing the validation NELBO of current USDMs by position type (Figure 5), we find that at low noise levels an incorrect prediction costs 13–100 nats more than a correct one at a corrupted position, but only 0.09–1.32 nats more at a clean one. Overall, mispredicted clean positions make up 23–31% of all positions yet only 3.5–5.2% of the NELBO, versus 90–92% for mispredicted corrupted ones. We therefore propose **C**orrect-**T**oken **R**etention **Reg**ularization (**CTR-Reg**), a simple auxiliary loss term that supplies this missing signal by explicitly training the model to retain tokens left unperturbed by the noising process, and hence already correct.

Our contributions are: (i) we identify a *correct-token retention* deficit in USDMs: they revise correct and incorrect tokens indiscriminately, reconstructing both with nearly identical accuracy, and thus collapse in sample diversity under greedy-tail sampling; (ii) we trace this deficit to the NELBO,

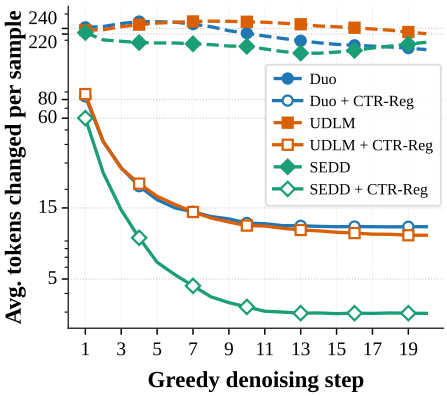

Figure 2: **Token changes per greedy-tail step.** Mean positions (of 512) changed at each of $g = 20$ greedy-tail steps after 500 ancestral steps (512 samples; broken y-axis) for DUO, UDLM, and SEDD, NELBO-only (dashed) vs. CTR-Reg (solid). NELBO-only models never converge, still revising 173–270 positions at step 20 and driving the diversity collapse in Figure 4; CTR-Reg plateaus at 3–11 positions within $\sim$10 steps.

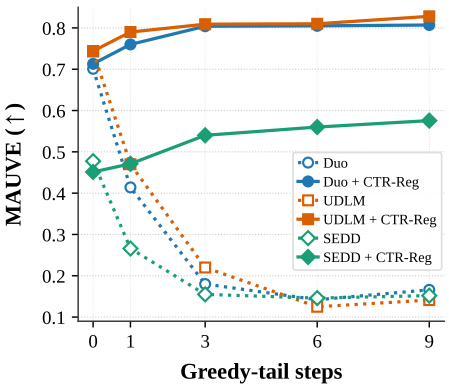

Figure 3: **MAUVE (↑) vs. greedy-tail steps.** DUO, UDLM, and SEDD with (solid) and without (dotted) CTR-Reg; $g \in \{0, 1, 3, 6, 9\}$ greedy-tail steps follow 500 ancestral steps; 5,000 samples vs. 5,000 OWT sequences. NELBO-only models degrade sharply as g grows, consistent with the diversity collapse in Figure 4; CTR-Reg models instead improve with g.

which barely penalizes incorrect predictions at clean positions, and propose CTR-Reg, which supplies this missing signal without modifying the sampler; (iii) CTR-Reg improves clean-token accuracy by 21.5–29.3 points per model and corpus (Figure 1; 26.5 on average over six benchmarks) with corrupted-token accuracy virtually unchanged, and cuts the positions changed at the first greedy-tail step to about a third of the *NELBO-only* baseline's, plateauing at 3–11 (Figure 2); and (iv) across DUO, UDLM, and SEDD, this yields substantially higher-quality generations with diversity above the training-data entropy (Figure 4): with 5 greedy-tail steps, perplexity drops from 105.86 to 49.57 (DUO), 93.28 to 44.31 (UDLM), and 194.59 to 87.19 (SEDD), with gains holding across all step budgets, including ultra-low ones (Figure 7), and under MAUVE (Pillutla et al., 2021) (Figure 3).

2 Background and Related Work

Notation. Let $V := \{v \in \{0,1\}^K : \sum_{i=1}^K v_i = 1\}$ denote the set of one-hot encodings of discrete random variables over K categories (the vocabulary size), and let Δ^K denote the corresponding K-simplex. For $v \in \Delta^K$, $\mathrm{Cat}(\cdot\,; v)$ denotes the categorical distribution with $\mathbb{P}(u_i = 1) = v_i$ for $u \sim \mathrm{Cat}(\cdot\,; v)$, $u \in V$. For vectors a, b, let $\langle a, b\rangle$ and $a \odot b$ denote their dot and Hadamard products, and let $\mathbf{1}$ denote the all-ones vector. We write $x \in V$ for a single token and $x^{(1:L)} \in V^L$ for a sequence of L tokens, with $x^{(\ell)}$ its ℓ-th entry; the same convention applies to the noisy latents z_t. Let $\pi \in \Delta^K$ denote a designated prior distribution.

2.1 Discrete Diffusion Models

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) learn a parameterized reverse process p_θ that inverts a fixed forward corruption process q, which gradually maps clean data x to increasingly noisy latent variables z_t over time $t \in [0, 1]$. Discrete diffusion models extend this framework to categorical data such as text (Hoogeboom et al., 2021; Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024). Following the interpolating formulation of Sahoo et al. (2024), the forward marginal at time t is

$$q_t(z_t \mid x) = \mathrm{Cat}\big(z_t;\, \alpha_t x + (1 - \alpha_t)\pi\big), \tag{1}$$

where the noise schedule $\alpha_t \in [0, 1]$ decreases monotonically in t, with $\alpha_0 \approx 1$ and $\alpha_1 \approx 0$, and noise is applied independently to each position of a length-L sequence. With T denoising steps, we

follow Schiff et al. (2024) and set $t(i) = (i+1)/T$ and $s(i) = i/T$ for $i = 0, \dots, T-1$, and write $q_{s|t}(z_s \mid z_t, x)$ for the forward posterior between consecutive times $s < t$. A denoising network $x_\theta(z_t^{(1:L)}, t) \in (\Delta^K)^L$ predicts the clean sequence from the noisy one, and the reverse transition substitutes this prediction into the forward posterior (Sahoo et al., 2024; Schiff et al., 2024):

$$p_\theta\big(z_s^{(1:L)} \mid z_t^{(1:L)}\big) = \prod_{\ell=1}^{L} q_{s|t}\Big(z_s^{(\ell)} \,\Big|\, z_t^{(\ell)},\, x = x_\theta^{(\ell)}\big(z_t^{(1:L)}, t\big)\Big). \tag{2}$$

The reverse process therefore factorizes across positions given $z_t^{(1:L)}$: each position is updated independently of the others at every step. Following Austin et al. (2021) and Schiff et al. (2024), the network is trained by minimizing the negative evidence lower bound (NELBO), writing $q_{s|t}^{(i)} := q_{s|t}(z_{s(i)} \mid z_{t(i)}, x)$ and $p_\theta^{(i)} := p_\theta(z_{s(i)} \mid z_{t(i)})$, and $q_{t(T)}, p_{t(T)}$ for $q(z_{t(T)} \mid x), p(z_{t(T)})$:

$$\mathcal{L}_{\mathrm{vb}} = \mathbb{E}_q\Big[\underbrace{-\log p_\theta(x \mid z_{t(0)})}_{\mathcal{L}_{\mathrm{recons}}} + \underbrace{\textstyle\sum_{i=1}^{T} D_{\mathrm{KL}}\big(q_{s|t}^{(i)} \,\|\, p_\theta^{(i)}\big)}_{\mathcal{L}_{\mathrm{diffusion}}} + \underbrace{D_{\mathrm{KL}}\big(q_{t(T)} \,\|\, p_{t(T)}\big)}_{\mathcal{L}_{\mathrm{prior}}}\Big] \tag{3}$$

whose continuous-time limit $T \to \infty$ gives a tighter bound (Sahoo et al., 2024; Schiff et al., 2024).

2.2 Uniform-State Diffusion Language Models

Uniform-state diffusion models (USDMs) set the prior to the uniform distribution over the vocabulary, $\pi = \mathbf{1}/K$, so a corrupted token is replaced by a token drawn uniformly at random (Hoogeboom et al., 2021; Austin et al., 2021). In the discrete-time D3PM framework of Austin et al. (2021), the n-th forward step applies $Q_n = (1-\beta_n)I + \beta_n \mathbf{1}\mathbf{1}^\top/K$ with $\beta_n \in [0,1]$ to every token; products of such matrices keep the same form, so the marginal after n steps is equation 1 with $\pi = \mathbf{1}/K$ and $\alpha = \prod_{n' \le n}(1-\beta_{n'})$, and the forward posterior has the closed form (Austin et al., 2021; Schiff et al., 2024)

$$q_{s|t}(z_s \mid z_t, x) = \mathrm{Cat}\left(z_s;\, \frac{K\alpha_t\, z_t \odot x + (\alpha_{t|s} - \alpha_t)\, z_t + (\alpha_s - \alpha_t)\, x + (1-\alpha_{t|s})(1-\alpha_s)\, \mathbf{1}/K}{K\alpha_t \langle z_t, x\rangle + 1 - \alpha_t}\right), \tag{4}$$

with $\alpha_{t|s} = \alpha_t/\alpha_s$. Unlike absorbing-state diffusion, where a decoded token is kept deterministically (Sahoo et al., 2024), equation 4 assigns non-zero probability both to keeping z_t and to moving to any other token, so every position can be revised at every step; this is the source of the self-correction ability of USDMs (Sahoo et al., 2025; Deschenaux et al., 2026).

Continuous-time objective (NELBO). Current state-of-the-art USDMs adopt different parameterizations, but all optimize a bound on the negative log-likelihood (NELBO), and up to parameterization these bounds coincide with a single continuous-time objective, which we state here for $\pi = \mathbf{1}/K$. UDLM (Schiff et al., 2024) reaches it from the variational perspective, by bounding a discrete-time chain and taking that bound to the continuous-time limit, as MDLM (Sahoo et al., 2024) does for the absorbing state. DUO (Sahoo et al., 2025) trains with the same bound but rewrites its per-token term to avoid materializing an intermediate K-dimensional quantity, and computes certain expectations analytically via Rao-Blackwellization, lowering memory, training time and gradient variance. SEDD (Lou et al., 2024) instead formulates the process directly as a continuous-time Markov chain (Campbell et al., 2022) and predicts ratios of marginals (the concrete score (Meng et al., 2022)) rather than the clean data, training with the diffusion-weighted denoising score entropy; Schiff et al. (2024) show that under a mean parameterization of the concrete score this objective is equivalent to the same bound, and report that CTMC-based models underperform MDLM and UDLM.

Concretely, the bound is the $T \to \infty$ limit of equation 3: $\mathcal{L}_{\mathrm{prior}}$ and $\mathcal{L}_{\mathrm{recons}}$ vanish, and the diffusion term converges to

$$\mathcal{L}^\infty = \mathbb{E}_{t \sim \mathcal{U}[0,1]}\, \mathbb{E}_{z_t \sim q_t(\cdot|x)} \textstyle\sum_{\ell=1}^{L} f\big(z_t^{(\ell)}, x_\theta^{(\ell)}(z_t^{(1:L)}, t), \alpha_t; x^{(\ell)}\big), \tag{5}$$

$$f\big(z_t, x_\theta(z_t,t), \alpha_t; x\big) = \frac{\alpha_t'}{K\alpha_t}\left[\frac{K}{\bar{x}_i} - \frac{K}{(\bar{x}_\theta)_i} - \sum_{j \neq i} \frac{\bar{x}_j}{\bar{x}_i} \log \frac{(\bar{x}_\theta)_i\, \bar{x}_j}{(\bar{x}_\theta)_j\, \bar{x}_i}\right], \tag{6}$$

where i indexes the non-zero entry of z_t, $\bar{x} = K\alpha_t x + (1-\alpha_t)\mathbf{1}$, $\bar{x}_\theta = K\alpha_t x_\theta(z_t,t) + (1-\alpha_t)\mathbf{1}$, and $\alpha_t' = d\alpha_t/dt$. The explicit per-token objectives of DUO and SEDD, and their equivalence to this bound, are given in Section C. equation 5 is the objective to which CTR-Reg is added in Section 3.2.

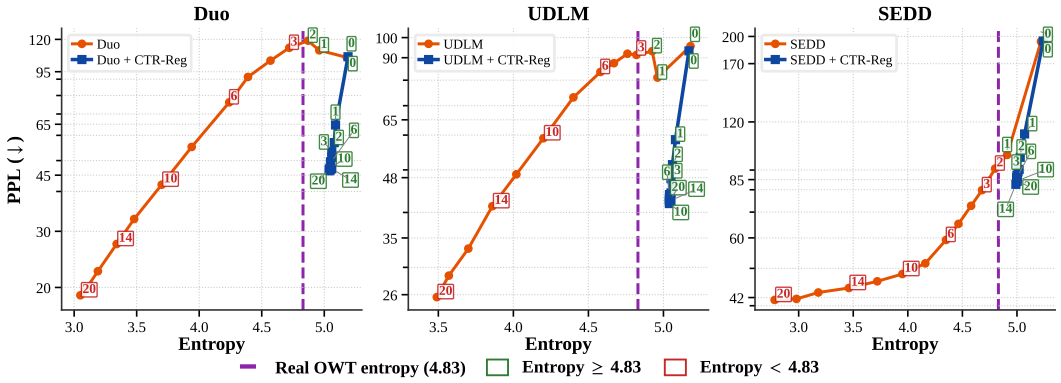

Figure 4: **PPL–entropy frontier vs. greedy-tail steps.** Unconditional generation with DUO, UDLM, and SEDD (one per panel) at a fixed budget of 512 steps, sweeping the number of terminal greedy-tail steps g from 0 (fully ancestral) to 20 (annotated at each point); PPL is measured with GPT2-Large over 1024 samples. The dashed line marks real OpenWebText entropy (4.83), below which lower PPL reflects degenerate repetition rather than genuine quality. Greedy-tail steps collapse diversity in NELBO-only models, which fail to preserve correct tokens, whereas CTR-Reg shifts the frontier toward lower PPL at higher entropy and reaches high-quality generations within a few steps.

3 Methodology

This section develops both the diagnosis and the remedy. We first show that current USDMs revise positions without regard to whether the token already there is correct, that this deficit surfaces as a collapse in sample diversity under greedy-tail decoding, and that the NELBO training objective barely penalizes it (Section 3.1). We then propose Correct-Token Retention Regularization (CTR-Reg), a single auxiliary term that targets correct-token retention directly (Section 3.2).

3.1 Current USDMs Lack Correct-Token Retention

Multi-step greedy-tail sampling. We study three state-of-the-art USDMs, DUO (Sahoo et al., 2025), UDLM (Schiff et al., 2024), and SEDD (Lou et al., 2024), under a multi-step extension of the *Greedy-Tail Sampler* of Sahoo et al. (2025). Given a total step budget $T = n + g$, the sampler takes n ancestral steps (equation 2 with $(s, t) = (s(i), t(i))$) followed by g greedy-tail steps, each a single forward pass at the terminal diffusion time δ followed by a per-position $\arg\max$, with no stochastic sampling:

$$z^{(\ell)} \leftarrow \text{onehot}\Big(\arg\max_j \big[x_\theta^{(\ell)}\big(z^{(1:L)}, \delta\big)\big]_j \Big) \qquad \forall \ell \in \{1, \dots, L\}. \tag{7}$$

Setting $g = 0$ recovers pure ancestral sampling and $g = 1$ the sampler of Sahoo et al. (2025), whose final greedy step lowers sample entropy much as nucleus sampling does (Holtzman et al., 2020); fixing T while sweeping g keeps comparisons compute-matched. Under this sampler, all three USDMs suffer a substantial collapse in sample diversity, measured by entropy (Figure 4), even the official DUO checkpoint trained on 26× more data, whose entropy falls from 5.54 to 2.14 over 20 greedy-tail steps and stays below the real-text level at its sequence length of 1024 (5.40) for every $g \geq 1$ (Section F).

Where the collapse comes from. To identify the mechanism, we generate 512 sequences of length 512 per USDM with $n = 500$ ancestral and $g = 20$ greedy-tail steps, and track the number of positions whose decoded token changes at each greedy-tail step, averaged over samples (Figure 2). For all three NELBO-only models, this number starts near half the sequence and shows no sign of convergence: even at the twentieth step, 173–270 positions still change. Because all positions are updated independently and simultaneously (equation 2), each step rewrites a large fraction of the sequence, and even locally plausible edits, made by the hundreds at once, compound into globally inconsistent sequences (Gu et al., 2018), producing the low-diversity outputs in Figures 4 and 7. This persistent churn stems from **the models' inability to recognize and retain tokens that are already correct**: without such a signal, a greedy step has no basis for leaving a position unchanged.

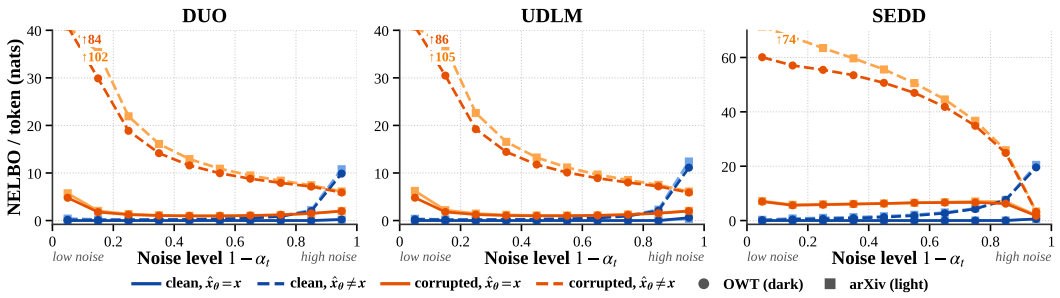

Figure 5: **The NELBO barely penalizes failures of retention.** Mean validation NELBO per token of NELBO-only models on OpenWebText (dark) and arXiv (light) versus the noise level $1 - \alpha_t$ (fraction of corrupted tokens), for clean (blue) and corrupted (orange) positions where the argmax prediction $\hat{x}_\theta$ equals the clean token x (solid) or not (dashed). The solid–dashed gap is the likelihood penalty for an incorrect prediction: large at corrupted positions, near zero at clean positions except at the highest noise. Magnitudes are comparable within, not across, panels.

Ancestral sampling masks the correct-token retention deficit. The retention seen under ancestral sampling comes from the noise schedule, not from the model. Under equation 2, a position that disagrees with the model's prediction adopts it with probability only $\frac{\alpha_s - \alpha_t}{1-\alpha_t}$ (Section D), which is $O(1/T)$ for any smooth schedule (Section D.2): decoded tokens persist largely because the sampler rarely acts on the model's prediction, not because the model would keep them. This masking is weakest near the terminal region, where the greedy tail operates, yet even there at most roughly one in five disagreeing positions adopts the prediction (under the log-linear schedule of DUO and UDLM, for which $\alpha_t = 1 - t$; Table 2). Absent genuine correct-token retention, which positions get revised is thus decided by sampling stochasticity rather than by the model; removing this bound, as the greedy tail does, exposes the deficit directly.

The NELBO barely rewards retention. To quantify how much the training objective penalizes this deficit, we split validation positions into clean ($z_t = x$) and corrupted ($z_t \neq x$), and by whether the model's prediction $\hat{x}_\theta = \mathrm{onehot}(\arg\max_j [x_\theta]_j)$ equals the clean token, and average the per-token NELBO of NELBO-only models within each group and noise level (Figure 5). At corrupted positions, an incorrect prediction is heavily penalized: for $1 - \alpha_t < 0.4$ it costs 13–100 nats more than a correct one. At clean positions, the same error costs only 0.09–1.32 nats more; overall, mispredicted clean positions make up 23–31% of all positions yet only 3.5–5.2% of the NELBO, versus 90–92% for mispredicted corrupted ones (Section E.2). This holds for all three models on all six corpora (Figures 5 and 8). The NELBO thus provides almost no signal for retaining correct tokens, consistent with their near-identical clean- and corrupted-token accuracies (Figure 1).

3.2 Correct-Token Retention Regularization (CTR-Reg)

Since the NELBO provides almost no signal at clean positions (Figure 5), we add one explicitly. CTR-Reg applies only at positions left unperturbed by the forward process ($z_t = x$), where the displayed token is already correct, and rewards the model for predicting that token with high probability:

$$f_{\mathrm{CTR}}\big(z_t, x_\theta(z_t, t); x\big) \;=\; \mathbb{I}\big[z_t = x\big]\,\Big(-\log\big\langle x_\theta(z_t, t),\, x\big\rangle\Big), \tag{8}$$

where $\langle x_\theta, x\rangle = (x_\theta)_m$ is the probability the denoiser assigns to the clean token and m indexes its non-zero entry; the indicator restricts the term to correct tokens and directly reinforces their retention. The full objective adds this term to the model's NELBO $\mathcal{L}^\infty$ (equation 5, whose per-token form differs across USDMs) through a single scalar $\lambda \geq 0$:

$$\mathcal{L} \;=\; \mathcal{L}^\infty \;+\; \lambda\, \mathbb{E}_{t\sim\mathcal{U}[0,1]}\, \mathbb{E}_{z_t\sim q_t(\cdot|x)} \sum_{\ell=1}^{L} f_{\mathrm{CTR}}\big(z_t^{(\ell)}, x_\theta^{(\ell)}(z_t^{(1:L)}, t); x^{(\ell)}\big). \tag{9}$$

CTR-Reg preserves the likelihood bound. Since $\langle x_\theta, x\rangle \in (0, 1]$, we have $-\log\langle x_\theta, x\rangle \geq 0$; with $\lambda \geq 0$ and the indicator non-negative, the CTR-Reg term is non-negative pointwise. Hence

$$\mathcal{L} \;=\; \mathcal{L}^\infty + \lambda\, \mathbb{E}\big[f_{\mathrm{CTR}}\big] \;\geq\; \mathcal{L}^\infty \;\geq\; -\log p_\theta(x). \tag{10}$$

Adding a non-negative quantity to an upper bound leaves an upper bound: training with CTR-Reg still minimizes a valid bound on the negative log-likelihood, merely a looser one. Empirically, the likelihood cost is small: validation perplexity changes by $+1.5\%$, $+1.2\%$, and -2.4% on average for DUO, UDLM, and SEDD (Section E.1), even though clean-token accuracy rises by 26.5 points (Section 4.1), further indicating that the likelihood is largely insensitive to retention.

Cost and hyperparameter. CTR-Reg acts only at training time and reuses the same forward pass as the NELBO, so it adds no computational overhead and leaves the sampler unchanged. We use $\lambda = 0.02$ for every USDM, and ablate it in Section 4.4.

3.3 Applying CTR-Reg to Score-Parameterized Models

UDLM and DUO predict the clean data directly, so equation 8 applies verbatim. SEDD, in contrast, outputs the log concrete score $\log s_\theta(z_t, t)$, whose j-th entry estimates $\log\big(q_t(j)/q_t(i)\big)$ relative to the displayed token i. These ratios determine the reverse transition rates from i to j (Lou et al., 2024); because the uniform forward rate is identical across all $j \neq i$, $s_\theta(z_t, t)_j$ directly measures how strongly the model favors moving from i to j. The displayed token has a fixed self-score $s_\theta(z_t, t)_i = 1$, so the preference for retaining i is encoded jointly by the off-diagonal scores and increases as they decrease. Consequently, equation 8 does not apply directly to SEDD. We instead formulate the penalty over the off-diagonal scores, penalizing scores above SEDD's training target with a smooth hinge:

$$f_{\mathrm{CTR}}^{\mathrm{score}} \;=\; \mathbb{I}\big[z_t = x\big]\,\frac{1}{K-1}\sum_{j\neq i}\mathrm{softplus}\!\left(\log\frac{s_\theta(z_t,t)_j}{\rho_t}\right), \qquad \rho_t = \frac{1-\alpha_t}{K\alpha_t + 1 - \alpha_t}. \tag{11}$$

Here, $\rho_t = q_t(j \mid x)/q_t(m \mid x)$ is the target value of every off-diagonal score at an unperturbed position ($i = m$) (Section G.2). We use a softplus hinge to obtain a non-negative penalty that vanishes smoothly below the target; adding $\lambda\,\mathbb{E}\big[f_{\mathrm{CTR}}^{\mathrm{score}}\big]$ to SEDD's objective therefore preserves the bound argument of Section 3.2. Dividing by ρ_t places the hinge at the training target: without this normalization, $\mathrm{softplus}(\log s)$ becomes negligible as soon as $s < 1$, which can be substantially above ρ_t. With it, the penalty pushes scores above ρ_t downward and fades below it, where SEDD's own objective already holds them (Section G.3).

4 Experiments

We evaluate the effectiveness of CTR-Reg on three USDMs, namely DUO (Sahoo et al., 2025), UDLM (Schiff et al., 2024), and SEDD with uniform noise (Lou et al., 2024), each trained with the standard NELBO objective (*NELBO-only*) and with NELBO + CTR-Reg. We address three questions: (i) does CTR-Reg improve correct-token retention without impairing the revision of incorrect tokens (Section 4.1); (ii) does better retention translate into better unconditional generation (Section 4.2); and (iii) how does CTR-Reg prevent the diversity collapse under greedy-tail decoding (Section 4.3)?

Implementation Details. We train the three models with the two objectives above. All models use the same 169M-parameter transformer and are trained from scratch on OpenWebText (Gokaslan & Cohen, 2019) ($L = 512$, 77K steps, batch size 512, $\approx$20B tokens), with the optimizer and remaining settings following each model's official implementation: log-linear noise for DUO and UDLM, geometric noise for SEDD, and DUO's Gaussian curriculum (Sahoo et al., 2025) over the first half of training. CTR-Reg uses a constant $\lambda = 0.02$ for all models; NELBO-only baselines use identical settings with $\lambda = 0$. Unless stated otherwise, we sample with 512 steps and report Gen PPL under GPT-2 Large (Radford et al., 2019) and mean unigram entropy over 1,024 samples, and MAUVE (Pillutla et al., 2021) between 5,000 generated and 5,000 OpenWebText sequences.

4.1 CTR-Reg Improves Retention Without Hurting Revision

Protocol. We evaluate retention using a controlled random-token corruption experiment. We split each evaluation corpus into sequences of $L = 512$ tokens and randomly replace $p\%$ of the positions with tokens drawn uniformly from the vocabulary. Each model then reconstructs the corrupted sequence in a single forward pass, taking the argmax of $x_\theta^{(\ell)}$ at each position, with the time input

set to the corruption level corresponding to p. We report accuracy, i.e., the fraction of positions whose prediction matches the original token, separately on the corrupted positions (*Corrupted-Token Accuracy*, measuring revision) and the remaining $(100 - p)\%$ of positions (*Clean-Token Accuracy*, measuring retention), averaged over $p \in \{10, 20, 30, 40\}\%$. For a model that recognizes and preserves correct tokens, clean positions are the easier target: the correct prediction is simply the observed token.

Results. Figure 1 reports both accuracies on the OpenWebText validation split (last 100K documents) and arXiv (Cohan et al., 2018); Table 9 (Section H) extends the evaluation to six corpora spanning web, news, scientific, biomedical, and encyclopedic text by adding LM1B (Chelba et al., 2014), AG News (Zhang et al., 2016), PubMed (Cohan et al., 2018), and WikiText-103 (Merity et al., 2017). NELBO-only models achieve nearly identical accuracy on clean and corrupted positions despite clean positions being the easier target, indicating that they revise tokens indiscriminately. This is not an artifact of undertraining: the official DUO checkpoint, trained on 26× more data, raises both accuracies by 5–6 points but leaves the clean-minus-corrupted gap at 1.1 points (vs. 2.0 at 20B), and CTR-Reg at 20B exceeds the checkpoint's clean-token accuracy by 24.7 points (77.4 vs. 52.7; Section F). CTR-Reg improves Clean-Token Accuracy for every model and corpus, by 26.5 points on average, while changing Corrupted-Token Accuracy by only 0.7 points per cell on average (Section H). CTR-Reg thus substantially improves retention without compromising revision.

4.2 Better Retention Yields Better Generations

We next test whether the single-step retention gains from CTR-Reg translate into better full generations. We sample with the multi-step greedy-tail sampler (Section 3.1), which appends a few deterministic denoising steps (a forward pass followed by a per-position argmax, with no stochastic sampling) after standard ancestral sampling. Because no sampling randomness is involved, whether a position is kept or rewritten depends only on the model's prediction. A model that cannot recognize already-correct tokens therefore has no basis for leaving them untouched, so greedy-tail decoding directly exposes the retention deficit identified in Section 3.1. We evaluate the quality and diversity of the resulting unconditionally generated samples to assess the effectiveness of CTR-Reg.

Generative Perplexity vs. Entropy. Figure 4 plots Gen PPL against entropy as the number of greedy-tail steps grows from $g = 0$ to $g = 20$ at $T = 512$ (exact values in Section I.1). For NELBO-only models, greedy-tail steps lower Gen PPL but push entropy below that of real text, so the lower Gen PPL reflects a collapse toward repetitive output rather than better text, consistent with the retention deficit measured in Section 4.1. CTR-Reg instead moves all three models toward lower Gen PPL at higher entropy, reaching substantially lower Gen PPL while keeping entropy at or above that of real OpenWebText. A few steps suffice: from $g = 0$ to $g = 5$, CTR-Reg lowers Gen PPL from 105.86 to 49.57 for DUO, from 93.28 to 44.31 for UDLM, and from 194.59 to 87.19 for SEDD. The benefit already appears under DUO's default sampler ($g = 1$), with no change to sampling: CTR-Reg reduces DUO's Gen PPL from 110.88 (NELBO-only) to 64.67 while raising entropy from 4.96 to 5.09. These gains hold across step budgets from 16 to 1024, including the ultra-low-step regime (Section B).

MAUVE. To confirm the Gen PPL and entropy trends, we additionally evaluate MAUVE (Pillutla et al., 2021). Whereas Gen PPL and entropy measure quality and diversity separately, MAUVE captures both jointly by comparing the distributions of generated and human-written text. Figure 3 reports MAUVE as a function of g (exact values in Section I.3). At $g = 0$, NELBO-only and CTR-Reg models score comparably, so CTR-Reg does not degrade standard ancestral sampling. As g grows, NELBO-only models collapse: within 6–9 steps, MAUVE falls from ∼0.70–0.75 to as low as 0.13–0.17 for DUO and UDLM, and from 0.47 to 0.15 for SEDD. CTR-Reg models instead improve with additional steps, reaching 0.81 (DUO), 0.83 (UDLM), and 0.58 (SEDD).

4.3 How CTR-Reg Resolves the Diversity Collapse

We repeat the token-change analysis of Section 3.1 with the same setup (Figure 2). There, NELBO-only models still revise 173–270 of the 512 positions at the twentieth greedy-tail step, and these independent, simultaneous edits compound into the entropy and MAUVE collapse of Section 4.2.

Table 1: **Effect of CTR-Reg coefficient λ on the corruption probe.** Corrupted- and clean-token accuracy and copy rate (corrupted positions that keep the random replacement token instead of recovering the original), in %, avg. over $p \in \{10, 20, 30, 40\}\%$, for DUO and UDLM at $\lambda = 0$ (NELBO-only), 0.02 (**our default**), and 0.1 (**blue**). **Avg**: mean over six corpora; **bold**: best accuracy across λ per model and corpus. Clean-token accuracy rises monotonically with λ; corrupted-token accuracy does not; copy rate stays below 0.25%.

		Corrupted-Token Acc. ↑			Clean-Token Acc. ↑			Copy Rate ↓		
Model	**Dataset**	0	**0.02**	**0.1**	0	**0.02**	**0.1**	0	**0.02**	**0.1**
DUO	OWT	54.5	**55.0**	54.6	53.9	83.2	**91.6**	0.003	0.04	0.16
	AG News	41.5	**41.8**	41.7	40.7	72.5	**84.3**	0.004	0.05	0.22
	arXiv	48.4	**49.7**	48.7	52.7	79.6	**88.2**	0.005	0.06	0.24
	LM1B	35.2	**36.5**	35.5	36.3	69.9	**82.3**	0.003	0.05	0.24
	PubMed	46.3	**47.7**	46.8	50.2	78.4	**87.1**	0.005	0.06	0.23
	WT-103	45.5	**47.2**	46.2	49.1	80.8	**89.6**	0.006	0.05	0.24
	Avg	45.2	**46.3**	45.6	47.2	77.4	**87.2**	0.004	0.05	0.22
UDLM	OWT	55.3	55.2	**55.5**	56.7	80.1	**89.6**	0.003	0.02	0.09
	AG News	42.8	42.6	**43.7**	46.0	68.2	**81.0**	0.007	0.03	0.09
	arXiv	47.4	**47.6**	47.4	51.5	73.0	**83.0**	0.006	0.03	0.11
	LM1B	**38.8**	38.0	37.7	40.9	66.2	**78.7**	0.005	0.03	0.12
	PubMed	45.4	**45.9**	45.6	49.6	72.8	**82.3**	0.006	0.04	0.11
	WT-103	**47.9**	47.4	47.7	51.8	76.4	**86.8**	0.007	0.03	0.11
	Avg	**46.3**	46.1	**46.3**	49.4	72.8	**83.6**	0.006	0.03	0.11

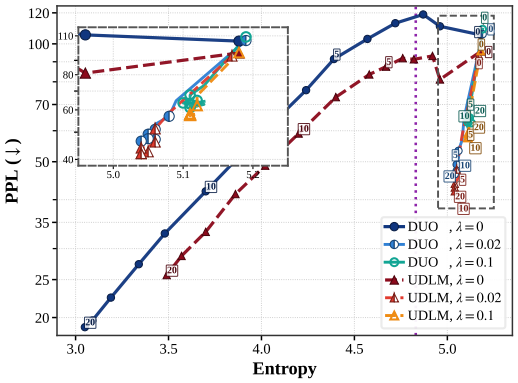

Figure 6: **Effect of CTR-Reg coefficient λ on greedy-tail generation.** PPL (GPT2-Large, 1024 samples) vs. entropy over greedy-tail steps g (budget 512) for DUO (blue, solid, circles) and UDLM (red/orange, dashed, triangles); marker fill encodes λ (filled: 0, half: 0.02, hollow: 0.1). Dotted line: real-OWT entropy (4.83); inset: zoom on the converging high-entropy cluster.

CTR-Reg removes this churn: for all three models, the number of changed positions at the first greedy-tail step is already about one third of the NELBO-only count and keeps decreasing to a stable plateau of 3–11 positions, so greedy decoding converges to a near-fixed point instead of repeatedly rewriting the sequence. The corruption experiment (Section 4.1), where ground truth is available, confirms that this selectivity is well placed: CTR-Reg models retain clean tokens while still correcting corrupted ones as often as NELBO-only models. Bounded, targeted revision thus preserves sequence structure, whereas indiscriminate revision destroys it, which accounts for the Gen PPL, entropy, and MAUVE gaps in Section 4.2.

4.4 Regularization Strength Analysis

We vary $\lambda \in \{0, 0.02, 0.1\}$ on DUO and UDLM ($\lambda = 0$: NELBO-only; main experiments use $\lambda = 0.02$). Raising λ from 0.02 to 0.1 further improves Clean-Token Accuracy (DUO: 77.4%→87.2%; UDLM: 72.8%→83.6%), while Corrupted-Token Accuracy stays largely unchanged (Table 1; per-rate results in Section I.6). To verify that CTR-Reg does not simply learn to copy its input, we also report the copy rate: the percentage of corrupted positions where the model keeps the random replacement token instead of recovering the original one. It is near zero for NELBO-only models (0.003–0.007%) and grows with λ (DUO: 0.05%/0.22%; UDLM: 0.03%/0.11% at $\lambda = 0.02/0.1$), yet stays below 0.25% throughout: CTR-Reg retains tokens selectively rather than copying them. Stronger retention, however, does not yield better generation (Figure 6): $\lambda = 0.1$ matches smaller λ under ancestral sampling ($g = 0$) and avoids the diversity collapse under greedy-tail decoding, but gains less from additional greedy-tail steps than $\lambda = 0.02$. Consistent with its higher copy rate, excessive retention can thus hinder the revision of earlier sampling mistakes, whereas $\lambda = 0.02$ better balances retaining correct tokens and revising uncertain ones, yielding stronger generation for both models.

5 Conclusion

Self-correction in USDMs requires both revising incorrect tokens and retaining correct ones; current USDMs lack retention, and their NELBO barely rewards it. CTR-Reg, a simple auxiliary loss on unperturbed tokens that leaves the sampler unchanged, raises clean-token accuracy by 26.5 points on average without hurting revision and more than halves generative perplexity within five greedy-tail steps while preserving diversity, across DUO, UDLM, and SEDD. Correct-token retention is thus a key ingredient for self-correcting diffusion language models.

AI USE STATEMENT.

Generative AI tools were used as assistants during manuscript preparation, primarily for language refinement, organization, LaTeX and table formatting, reference checking, numerical consistency checks between the main text and appendix tables, and feedback on clarity and presentation, as well as for syntax and debugging assistance during coding. The research problem, methodology, experimental design, implementation, experiments, analyses, and scientific conclusions were developed and carried out by the authors. All AI-assisted edits, code suggestions, and checks were reviewed and verified by the authors, and the authors take full responsibility for the content of the paper.

ETHICS STATEMENT

This work involves no human subjects or personal data collection. All models are trained and evaluated on publicly available research corpora, which as web-derived text may contain biases that trained models can reproduce. Our models are small (169M parameters) and intended solely for research, and CTR-Reg, as a training objective, does not introduce new risks or misuse potential beyond those of existing diffusion language models.

REPRODUCIBILITY STATEMENT

Section 4 specifies the datasets, backbone, optimization, and evaluation protocols, and Section 4.4 reports the CTR-Reg coefficient used for training. The CTR-Reg objective and its score-space variant for SEDD are fully specified in Sections 3.2–3.3, with derivations in Appendices D and G, and exact values for all reported results are given in Appendix I. The supplementary material contains, for each USDM (DUO, UDLM, and SEDD), the full training code and the code for the random-token corruption experiment and the MAUVE and Gen PPL/entropy evaluations.

REFERENCES

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

Andreas Blattmann, Robin Rombach, Kaan Oktay, and Björn Ommer. Retrieval-augmented diffusion models, 2022. URL https://arxiv.org/abs/2204.11824.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=DmT862YAieY.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling, 2014. URL https://arxiv.org/abs/1312.3005.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2097. URL https://aclanthology.org/N18-2097/.

Justin Deschenaux, Caglar Gulcehre, and Subham Sekhar Sahoo. The diffusion duality, chapter II: ψ-samplers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RSIoYWIzaP.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1l8BtlCb.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12454–12465. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/67d96d458abdef21792e6d8e590244e7-Paper.pdf.

Qingqing Huang, Daniel S. Park, Tao Wang, Timo I. Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, Jesse Engel, Quoc V. Le, William Chan, Zhifeng Chen, and Wei Han. Noise2music: Text-conditioned music generation with diffusion models, 2023. URL https://arxiv.org/abs/2302.03917.

Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion, 2025. URL https://arxiv.org/abs/2506.17298.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2024.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-time scaling for diffusion models beyond scaling denoising steps, 2025. URL https://arxiv.org/abs/2501.09732.

Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_RL7wtHkPJK.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=Tqx7nJp7PR.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu,

Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models, 2025. URL https://arxiv.org/abs/2410.13720.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

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*, 2025. URL https://openreview.net/forum?id=9P9Y8FOSOk.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.

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*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

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, 2025. URL https://arxiv.org/abs/2508.02193.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2016. URL https://arxiv.org/abs/1509.01626.

Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=HDeIb67lJe.

CONTENTS

1 Introduction **1**

2 Background and Related Work **3**

2.1 Discrete Diffusion Models 3

2.2 Uniform-State Diffusion Language Models 4

3 Methodology **5**

3.1 Current USDMs Lack Correct-Token Retention 5

3.2 Correct-Token Retention Regularization (CTR-Reg) 6

3.3 Applying CTR-Reg to Score-Parameterized Models 7

4 Experiments **7**

4.1 CTR-Reg Improves Retention Without Hurting Revision 7

4.2 Better Retention Yields Better Generations 8

4.3 How CTR-Reg Resolves the Diversity Collapse 8

4.4 Regularization Strength Analysis 9

5 Conclusion **9**

A Appendix **14**

B Step-Budget Analysis **14**

C Per-Model USDM Objectives **14**

D Ancestral Sampling Suppresses Indiscriminate Token Changes **15**

D.1 Single-position transition probabilities 15

D.2 The schedule bounds the number of changes per step 16

D.3 Why the greedy tail reveals what ancestral sampling hides 16

E Likelihood Analysis **16**

E.1 Validation Perplexity 17

E.2 Decomposing the NELBO by Position Type and Prediction 17

F Training on a Larger Corpus Does Not Fix the Problem **17**

G CTR-Reg for Score-Parameterized Models: Details **22**

G.1 Reading predictions from scores 22

G.2 The training target ρ_t 23

G.3 Placing the hinge at the target 23

H Random-Token Corruption on Additional Corpora **24**

I Additional Experimental Results **25**

I.1 Exact Values for the Greedy-Tail Step Sweep 25

I.2 Exact Values Across Sampling-Step Budgets 25

I.3 Exact Values for the MAUVE Greedy-Tail Sweep 25

I.4 Per-Corruption-Rate Breakdown for the 524B-Token Checkpoint 25

I.5 Exact Values for the CTR-Reg Coefficient Ablation 25

I.6 Per-Corruption-Rate Breakdown for the CTR-Reg Coefficient Ablation 25

A Appendix

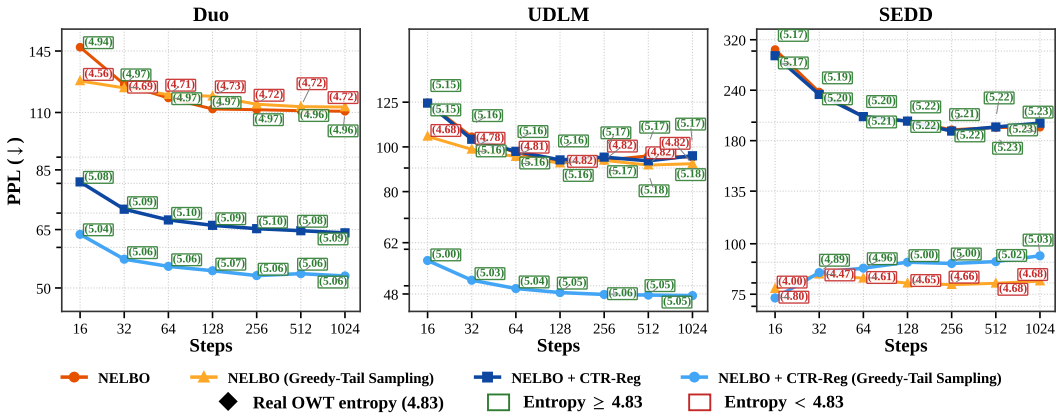

Figure 7: **Generation quality across sampling budgets (16–1024 steps).** PPL (↓) vs. total sampling steps for DUO, UDLM, and SEDD trained with NELBO or NELBO + CTR-Reg, under default (DUO: +1 greedy step) and greedy-tail sampling (final three steps greedy); each point is annotated with its generation entropy. Under the default sampler, CTR-Reg matches or outperforms NELBO-only training; under greedy-tail sampling, CTR-Reg substantially lowers PPL while preserving diversity, whereas NELBO-only models lower PPL only at the cost of diversity. These gains hold across the full step-budget range, including the ultra-low-step regime.

B Step-Budget Analysis

Figure 7 extends the greedy-tail analysis of Section 4.2 across total sampling budgets from 16 to 1024 steps, comparing each model's default sampler with greedy-tail sampling (final three steps greedy); exact values are given in Section I.2. Under each model's default sampler (for DUO, with one trailing greedy step), CTR-Reg matches or outperforms the NELBO-only baseline at every budget, including the ultra-low-step regime. Adding greedy-tail steps widens the gap: CTR-Reg substantially lowers Gen PPL while preserving diversity, whereas the apparent gains of NELBO-only models come at the cost of diversity, consistent with the collapse in Figure 4. The benefit of improved retention therefore holds across sampling budgets.

C Per-Model USDM Objectives

This appendix gives the explicit training objectives of DUO and SEDD, both of which are equivalent to the continuous-time bound of equation 5 that UDLM optimizes directly. We reuse the notation of Section 2.2: i indexes the non-zero entry of z_t, $\bar{x} = K\alpha_t x + (1-\alpha_t)\mathbf{1}$, $\bar{x}_\theta = K\alpha_t x_\theta(z_t, t) + (1-\alpha_t)\mathbf{1}$, and $\alpha'_t = d\alpha_t/dt$.

DUO. Sahoo et al. (2025) rewrite equation 6 so that $\bar{x}$ never has to be materialized, and compute certain expectations analytically via Rao-Blackwellization; the authors describe the result as an efficient, low-variance formulation with minimal memory overhead. With m the index of the non-zero entry of x and $\rho_t = (1-\alpha_t)/(K\alpha_t + 1 - \alpha_t)$, the per-token loss becomes

$$f_{\mathrm{DUO}} = \frac{\alpha_t'}{K\alpha_t}\Bigg[\frac{K}{\bar{x}_i} - \frac{K}{(\bar{x}_\theta)_i} - \big(\rho_t\,\mathbb{I}[z_t = x] + \mathbb{I}[z_t \neq x]\big)\sum_{j=1}^{K}\log\frac{(\bar{x}_\theta)_i}{(\bar{x}_\theta)_j} \\ - \frac{K\alpha_t}{1-\alpha_t}\,\mathbb{I}[z_t \neq x]\log\frac{(\bar{x}_\theta)_i}{(\bar{x}_\theta)_m} - \Big((K-1)\rho_t\,\mathbb{I}[z_t = x] - \frac{\mathbb{I}[z_t \neq x]}{\rho_t}\Big)\log\rho_t\Bigg]. \tag{12}$$

DUO further accelerates training with a curriculum that relaxes the discrete latents through an underlying Gaussian diffusion (Sahoo et al., 2025), which Deschenaux et al. (2026) later make more memory-efficient.

SEDD and CTMC-based formulations. Rather than bounding a discrete-time chain and taking that bound to the continuous-time limit, Campbell et al. (2022) and SEDD (Lou et al., 2024) formulate discrete diffusion directly with continuous-time Markov chains (CTMCs); both still optimize a bound on the likelihood. The forward process is defined by a rate matrix $R_t \in \mathbb{R}^{K\times K}$ through $q(z_t = z' \mid z_s = z) = \delta_{z',z} + R_t(z,z')/T + o(1/T)$ with $s = t - 1/T$. For the uniform interpolating process, $R_t = -\frac{\alpha_t'}{K\alpha_t}(\mathbf{1}\mathbf{1}^\top - KI)$ (Schiff et al., 2024), which coincides with SEDD-Uniform's rate $\sigma(t)(\mathbf{1}\mathbf{1}^\top/K - I)$ for $\alpha_t = e^{-\bar{\sigma}(t)}$ and $\bar{\sigma}(t) = \int_0^t \sigma(u)\,du$. Reversing this chain requires ratios of marginals: for $z' \neq z$, the reverse rate is $\hat{R}_t(z,z') = \frac{q_t(z')}{q_t(z)}R_t(z',z)$. These ratios, the concrete score (Meng et al., 2022), replace the score $\nabla \log q_t$ used by continuous score-based models (Song et al., 2021), since discrete data admit no gradient. Because tokens are corrupted independently, only sequences that differ from z_t in a single position have non-zero rate. Whereas UDLM, DUO and Campbell et al. (2022) predict the clean data, SEDD's network outputs these ratios directly, one per position and candidate token: $s_\theta(z_t,t) \in \mathbb{R}^{L\times K}$ with $s_\theta(z_t,t)_{z'} \approx q_t(z')/q_t(z_t)$. It is trained with the diffusion-weighted denoising score entropy, which forms a NELBO (Lou et al., 2024):

$$\mathcal{L}_{\mathrm{SEDD}} = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\,\mathbb{E}_{z_t\sim q_t(\cdot|x)}\sum_{z'\neq z_t} R_t(z',z_t)\Big[s_\theta(z_t,t)_{z'} - r_t(z')\log s_\theta(z_t,t)_{z'} + g\big(r_t(z')\big)\Big], \tag{13}$$

where $r_t(z') = q_t(z' \mid x)/q_t(z_t \mid x)$ and $g(a) = a(\log a - 1)$. Under the mean parameterization $s_\theta(z_t,t)_{z'} = (\bar{x}_\theta)_j/(\bar{x}_\theta)_i$, with j indexing the non-zero entry of z', Schiff et al. (2024) show that equation 13 is equivalent to equation 5.

D Ancestral Sampling Suppresses Indiscriminate Token Changes

We show that the ancestral sampler's noise schedule, rather than the denoiser's predictions, is what prevents the sampler from adopting the model's predictions en masse. The schedule assigns most of its probability mass to retaining the currently decoded token, which bounds the number of positions that can be revised toward the model's prediction at any single step.

D.1 Single-position transition probabilities

Consider a position ℓ with $z_t^\ell \neq x_\theta^\ell$ (both one-hot), so that we are asking how readily the sampler incorporates a change the model has already decided on. The cross term in the reverse posterior (equation 2) then vanishes, since $z_t^\ell \odot x_\theta^\ell = \mathbf{0}$ and $\langle z_t^\ell, x_\theta^\ell\rangle = 0$, leaving

$$q_{s|t}(z_s^\ell \mid z_t^\ell, x_\theta^\ell) = \mathrm{Cat}\left(\cdot\,;\,\frac{(\alpha_{t|s}-\alpha_t)z_t^\ell + (\alpha_s - \alpha_t)x_\theta^\ell + (1-\alpha_{t|s})(1-\alpha_s)\mathbf{1}/K}{1-\alpha_t}\right).$$

Collecting the weight assigned to each of the K categories and dropping $O(1/K)$ terms (negligible for $K \approx 50\text{k}$) yields three outcomes:

$$P(\text{stay}) \approx \frac{\alpha_{t|s} - \alpha_t}{1-\alpha_t}, \tag{14}$$

$$P(\text{jump-to-prediction}) \approx \frac{\alpha_s - \alpha_t}{1 - \alpha_t}, \tag{15}$$

$$P(\text{jump-elsewhere}) \approx \frac{(1 - \alpha_{t|s})(1 - \alpha_s)}{1 - \alpha_t}. \tag{16}$$

Summing the exact, un-truncated weights and applying $\alpha_{t|s}\alpha_s = \alpha_t$ confirms that these normalize to one. Note that equation 14–equation 16 assume no functional form for α_t and therefore hold for any interpolating noise schedule.

D.2 The schedule bounds the number of changes per step

Under the log-linear noise schedule used in our experiments, $\bar{\sigma}(t) = -\log\big(1 - (1 - \epsilon)t\big)$, so $\alpha_t = e^{-\bar{\sigma}(t)} = 1 - (1 - \epsilon)t$ is linear in t; with $\epsilon = 10^{-3}$ we write $\alpha_t \approx 1 - t$. With step size $h = 1/T$, we have $\alpha_s - \alpha_t = (1 - \epsilon)h \approx h$ at every step, so equation 15 simplifies to

$$P(\text{jump-to-prediction}) \approx \frac{h}{1 - \alpha_t}. \tag{17}$$

This is $O(1/T)$ for any α_t bounded away from one, and approaches $O(1)$ only as the remaining noise budget $1 - \alpha_t$ shrinks toward h. For a general smooth schedule, $\alpha_s - \alpha_t \approx |\alpha_t'|h$, so equation 17 becomes $|\alpha_t'|h/(1 - \alpha_t)$: the $O(1/T)$ per-step suppression holds for any such schedule, which enters only through $|\alpha_t'|$. We specialize to the linear case for concreteness, matching our experimental setup.

Consequently, if m positions disagree with the denoiser's prediction at step t, the expected number that adopt that prediction on the step is

$$\mathbb{E}[\#\text{changed}] \approx \frac{mh}{1 - \alpha_t}, \tag{18}$$

which remains small whenever $h \ll 1 - \alpha_t$. Through most of the trajectory the sampler therefore revises only a small fraction of the disagreeing positions per step, never all m simultaneously; the bound loosens only near the terminal region.

Table 2: Single-step transition probabilities at a disagreeing position under the log-linear schedule ($\alpha_t \approx 1 - t$) with $T = 512$ ($h \approx 1.95 \times 10^{-3}$).

α_t	stay	jump-to-prediction	jump-elsewhere
0.01	0.835	0.002	0.163
0.50	0.992	0.004	0.004
0.99	0.803	0.195	0.002

Table 2 evaluates equation 14–equation 16 at three points along the trajectory. Even at $\alpha_t = 0.99$, where the bound is loosest, roughly four out of five disagreeing positions still fail to update in a single step. The bound constrains *model-driven* revisions specifically: early in generation ($\alpha_t = 0.01$) many positions do change, but almost all of these are draws from the uniform prior (equation 16) rather than adoptions of the denoiser's prediction, which remains suppressed at 0.002 throughout that regime.

D.3 Why the greedy tail reveals what ancestral sampling hides

The greedy tail sets $z_s^\ell = x_\theta^\ell$ deterministically, so $P_g(\text{jump-to-prediction}) = 1$ for every α_t and h. This removes the suppression term in equation 17 and lifts the bound in equation 18 entirely: all m disagreeing positions update at once, rather than the $O(mh)$ fraction permitted by ancestral sampling. A denoiser whose argmax is unstable across steps therefore exhibits little visible instability under ancestral sampling, while the same instability is fully exposed under the greedy tail. The near-complete, non-converging rewrites in Figure 2 indicate that this instability was present throughout ancestral sampling and was simply not surfaced.

E Likelihood Analysis

This appendix supports the likelihood analysis of Section 3.1 (Figure 5). All quantities are computed from the validation NELBO of each model *without* the CTR-Reg term, using the EMA weights and

the six evaluation corpora of Section H at sequence length 512. For each sequence, we draw one diffusion time t (stratified within each batch) and one corruption $z_t \sim q_t(\cdot \mid x)$, and evaluate the per-token NELBO integrand of each model: equation 5 for UDLM, and the equivalent per-token objectives of DUO and SEDD given in Section C (for SEDD, the diffusion-weighted score entropy of equation 13). NELBO-only and CTR-Reg models of the same USDM are evaluated with the same seed and batch size, so both see identical t and corruptions. DUO and SEDD are evaluated on identical token sequences, whereas UDLM uses its own data pipeline over the same corpora, which yields slightly different chunks.

E.1 Validation Perplexity

Table 3 reports the resulting perplexity upper bound, $\exp(\text{NELBO per token})$. CTR-Reg increases perplexity by 1.5% on average for DUO and 1.2% for UDLM, with a maximum of 4.4% (DUO on PubMed), and decreases it by 2.4% on average for SEDD, on five of six corpora. The in-distribution (OWT) change is below 1% for DUO and UDLM.

E.2 Decomposing the NELBO by Position Type and Prediction

To measure how much the likelihood rewards retention, we split every validation position into four groups: *clean* ($z_t = x$) or *corrupted* ($z_t \neq x$), and whether the model's argmax prediction $\hat{x}_\theta$ equals the clean token ($\hat{x}_\theta = x$) or not ($\hat{x}_\theta \neq x$). For SEDD, $\hat{x}_\theta$ is taken directly over the scores, which yields the same ranking as x_θ (Section G.1). Positions where a uniform draw happens to coincide with the original token count as clean, consistent with the NELBO. We average the per-token NELBO within each group and within ten bins of the noise level $1 - \alpha_t$, the expected fraction of corrupted tokens. For DUO and UDLM, whose log-linear schedule gives $1 - \alpha_t \approx t$, these bins are bins of t. SEDD uses a geometric schedule, so its bins are bins of $1 - \alpha_t = 1 - e^{-\sigma(t)}$, and its per-token magnitudes are not directly comparable to those of the other two models.

Figure 8 extends Figure 5 to the remaining four corpora, and Tables 4 to 6 give the exact values underlying both figures, together with the number of tokens behind each mean. The pattern is consistent across all models and corpora. Clean positions where the prediction is incorrect ($\hat{x}_\theta \neq x$) make up 23–31% of all validation positions yet contribute only 3.5–5.2% of the NELBO, whereas corrupted positions with incorrect predictions make up 25–41% of positions and contribute 90–92%; correctly predicted clean positions contribute almost nothing. At noise levels $1 - \alpha_t < 0.4$, an incorrect prediction at a corrupted position costs 13–100 nats more than a correct one, whereas at a clean position it costs only 0.09–1.32 nats more. The penalty at clean positions grows only near $1 - \alpha_t = 1$, where almost every displayed token is noise and replacing it is reasonable. Means at the highest noise levels rest on few clean tokens and are correspondingly noisier; these cells are grayed out in the tables.

F Training on a Larger Corpus Does Not Fix the Problem

Owing to computational cost, our main experiments train DUO, UDLM, and SEDD from scratch on 20B OpenWebText tokens (169M parameters, sequence length 512). A natural question is whether the correct-token retention deficit we identify is simply an artifact of undertraining, and would disappear if the model were trained on substantially more data. To test this, we repeat both the random-token corruption analysis and the greedy-tail sampling analysis on the official, publicly released DUO checkpoint (Sahoo et al., 2025), which is trained on approximately 524B tokens ($\sim$26$\times$ more than our 20B-token models) with a native sequence length of 1024 (rather than 512). For the corruption experiment, sequences are chunked to this checkpoint's native length of 1024 tokens rather than 512, so that the model is evaluated in-distribution; because of this sequence-length mismatch, we do not directly compare PPL or entropy magnitudes between the 524B-token checkpoint and our 512-length models (see below).

Random-token corruption at scale. Table 7 reports corrupted-token, clean-token, and overall reconstruction accuracy for the 524B-token DUO checkpoint alongside our 20B-token NELBO-only and NELBO + CTR-Reg models, following the same protocol as Table 9. Scaling to 26$\times$ more training tokens raises corrupted-token accuracy substantially (avg. 45.2% $\rightarrow$ 51.6%), as expected:

Table 3: **Validation perplexity (↓) of NELBO-only and CTR-Reg models.** Perplexity upper bound computed from the validation NELBO alone; the CTR-Reg term is excluded at evaluation. All models have 169M parameters and are trained on 20B OpenWebText (OWT) tokens with sequence length 512. Δ: relative change of NELBO + CTR-Reg with respect to NELBO-only. CTR-Reg changes perplexity by at most a few percent, and lowers it on average for SEDD.

		In-dist.	Zero-shot					
Model	Loss	OWT	LM1B	AG News	arXiv	PubMed	WT-103	Avg.
DUO	NELBO	36.59	93.21	114.21	62.53	74.81	51.31	72.11
	+ CTR-Reg	36.74	93.66	114.47	64.21	78.09	51.78	73.16
	Δ	+0.4%	+0.5%	+0.2%	+2.7%	+4.4%	+0.9%	+1.5%
UDLM	NELBO	40.31	128.92	139.37	77.24	91.03	62.80	89.95
	+ CTR-Reg	40.65	129.84	143.68	77.37	91.07	63.69	91.05
	Δ	+0.8%	+0.7%	+3.1%	+0.2%	+0.0%	+1.4%	+1.2%
SEDD	NELBO	68.07	155.47	229.39	115.25	154.18	98.33	136.78
	+ CTR-Reg	66.87	152.93	228.08	109.38	144.97	98.79	133.50
	Δ	−1.8%	−1.6%	−0.6%	−5.1%	−6.0%	+0.5%	−2.4%

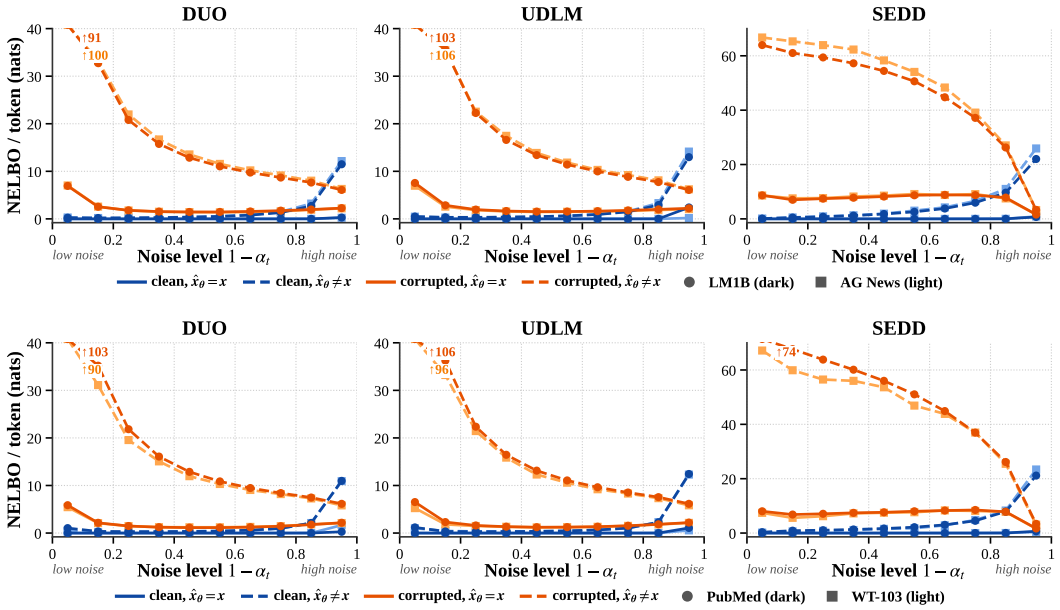

Figure 8: **The NELBO barely penalizes failures of retention on the remaining corpora.** Mean validation NELBO per token of NELBO-only DUO, UDLM, and SEDD on LM1B and AG News (top) and PubMed and WikiText-103 (bottom), with the first corpus of each pair dark and the second light, versus the noise level $1 - \alpha_t$ (fraction of corrupted tokens). Lines show clean (blue) and corrupted (orange) positions where the argmax prediction $\hat{x}_\theta$ equals the clean token x (solid) or not (dashed), so the solid–dashed gap is the likelihood penalty for an incorrect prediction. As on OpenWebText and arXiv (Figure 5), this penalty is large at corrupted positions and near zero at clean ones, except at the highest noise. Magnitudes are comparable within, not across, panels.

more data simply makes the model a better denoiser. However, clean-token accuracy improves by only a comparable amount (avg. 47.2% → 52.7%), so the model still cannot tell already-correct tokens apart from corrupted ones: the average clean-minus-corrupted gap is 2.0 points at 20B tokens and only 1.1 points at 524B tokens, i.e., scaling by 26× leaves this gap unchanged within noise. In contrast, CTR-Reg opens a 31.1-point average gap using only the 20B-token budget, and its clean-token accuracy exceeds the 524B-token checkpoint's by nearly 25 percentage points on average (77.4% vs. 52.7%) despite seeing 26× fewer training tokens. This indicates that correct-token retention is not a

Table 4: **Exact values underlying Figures 5 and 8 for DUO.** Mean validation NELBO per token (nats) of the NELBO-only model and the number of tokens behind each mean, by corpus and noise-level bin $1-\alpha_t$ (fraction of corrupted tokens), for **clean** and **corrupted** positions where the argmax prediction $\hat{x}_\theta$ equals the clean token x or not. †In units of 10^{-3} nats. Gray: fewer than 1,000 tokens (noisier estimate).

		Mean NELBO per token (nats)				Number of tokens			
		Clean		Corrupted		Clean		Corrupted	
Corpus	$1-\alpha_t$	$\hat{x}_\theta = x^\dagger$	$\hat{x}_\theta \neq x$	$\hat{x}_\theta = x$	$\hat{x}_\theta \neq x$	$\hat{x}_\theta = x$	$\hat{x}_\theta \neq x$	$\hat{x}_\theta = x$	$\hat{x}_\theta \neq x$
OpenWebText	[0.0, 0.1)	0.019	0.1587	4.7935	84.0081	7,062,434	3,572,467	378,832	186,779
	[0.1, 0.2)	0.027	0.1111	1.7759	29.8928	6,139,616	3,545,775	1,074,039	632,058
	[0.2, 0.3)	0.040	0.1255	1.2435	18.8710	5,059,097	3,420,197	1,627,315	1,195,791
	[0.3, 0.4)	0.063	0.1586	1.0497	14.1688	4,087,763	3,298,748	2,052,332	1,915,781
	[0.4, 0.5)	0.110	0.2154	0.9840	11.5886	3,121,303	3,108,428	2,283,455	2,808,670
	[0.5, 0.6)	0.213	0.3119	0.9903	9.9526	2,253,591	2,872,115	2,341,637	3,914,417
	[0.6, 0.7)	0.480	0.4894	1.0615	8.7982	1,454,797	2,506,955	2,157,867	5,178,685
	[0.7, 0.8)	1.358	0.8690	1.2128	7.9157	805,436	2,039,583	1,804,974	6,700,535
	[0.8, 0.9)	6.010	1.9837	1.4987	7.1404	327,824	1,384,040	1,294,731	8,358,269
	[0.9, 1.0]	222.624	9.8801	1.9753	5.9091	66,583	507,961	732,809	9,994,535
LM1B	[0.0, 0.1)	0.025	0.2502	6.9276	91.0346	452,913	344,306	23,380	19,081
	[0.1, 0.2)	0.036	0.1799	2.5497	32.7434	393,046	350,682	66,084	65,196
	[0.2, 0.3)	0.055	0.2066	1.7987	20.7934	304,654	329,088	94,251	115,783
	[0.3, 0.4)	0.089	0.2638	1.5314	15.7570	242,774	320,485	115,453	185,544
	[0.4, 0.5)	0.158	0.3617	1.4475	12.8561	176,240	298,517	123,104	266,907
	[0.5, 0.6)	0.307	0.5154	1.4576	11.0393	120,936	267,303	118,772	355,197
	[0.6, 0.7)	0.690	0.7823	1.5611	9.7381	73,915	227,277	103,354	452,542
	[0.7, 0.8)	1.908	1.3204	1.7287	8.6864	38,960	176,301	83,444	559,919
	[0.8, 0.9)	8.310	2.7153	1.9655	7.6393	16,485	115,337	60,845	681,829
	[0.9, 1.0]	275.655	11.4829	2.2610	6.0831	3,642	38,471	40,216	761,959
AG News	[0.0, 0.1)	0.027	0.2943	7.0253	100.4483	20,614	14,506	985	759
	[0.1, 0.2)	0.039	0.1826	2.5778	33.2478	20,138	17,230	3,513	3,151
	[0.2, 0.3)	0.058	0.2181	1.7661	21.9576	12,619	12,733	3,943	4,497
	[0.3, 0.4)	0.092	0.2715	1.4993	16.6985	12,103	14,451	5,671	8,223
	[0.4, 0.5)	0.175	0.4011	1.4325	13.5510	8,656	13,967	5,759	12,578
	[0.5, 0.6)	0.331	0.5495	1.3784	11.5594	5,886	13,036	5,624	17,438
	[0.6, 0.7)	0.768	0.8787	1.4398	10.1959	2,967	9,720	4,146	20,031
	[0.7, 0.8)	1.905	1.4776	1.5987	9.1441	1,688	8,262	3,473	26,513
	[0.8, 0.9)	7.810	3.2152	1.9279	8.0340	550	5,176	2,142	30,532
	[0.9, 1.0]	234.666	12.1551	2.2062	6.2942	103	1,920	1,411	35,478
arXiv	[0.0, 0.1)	0.025	0.4159	5.7355	102.1255	3,152,387	1,900,869	160,269	109,227
	[0.1, 0.2)	0.035	0.2448	2.0284	35.4463	2,728,095	1,831,867	449,252	353,986
	[0.2, 0.3)	0.051	0.2377	1.3667	21.9201	2,269,589	1,745,305	685,259	650,247
	[0.3, 0.4)	0.079	0.2625	1.1066	16.1067	1,845,448	1,639,599	866,826	1,007,743
	[0.4, 0.5)	0.131	0.3176	0.9943	12.9259	1,433,287	1,511,996	978,438	1,429,239
	[0.5, 0.6)	0.243	0.4165	0.9703	10.9009	1,055,985	1,367,986	1,013,330	1,944,331
	[0.6, 0.7)	0.523	0.6040	1.0185	9.4821	704,371	1,173,911	945,656	2,531,070
	[0.7, 0.8)	1.399	1.0083	1.1704	8.3863	403,960	935,107	780,469	3,219,600
	[0.8, 0.9)	5.846	2.1731	1.4860	7.4202	177,842	636,076	558,903	4,022,635
	[0.9, 1.0]	182.380	10.7864	2.0129	6.1090	32,987	237,425	315,807	4,752,917
PubMed	[0.0, 0.1)	0.024	1.0386	5.8592	102.7561	1,545,685	983,268	80,294	54,689
	[0.1, 0.2)	0.034	0.3510	2.1517	35.1877	1,323,358	967,203	221,774	182,321
	[0.2, 0.3)	0.050	0.3024	1.4764	21.8310	1,089,726	913,998	332,266	333,066
	[0.3, 0.4)	0.079	0.3072	1.2327	16.0828	878,997	872,799	414,117	525,159
	[0.4, 0.5)	0.137	0.3471	1.1381	12.8692	671,605	811,982	456,947	756,170
	[0.5, 0.6)	0.264	0.4392	1.1460	10.8607	478,143	734,833	455,013	1,024,619
	[0.6, 0.7)	0.593	0.6074	1.2299	9.4668	308,135	637,871	410,483	1,340,727
	[0.7, 0.8)	1.695	0.9848	1.4401	8.4028	162,859	506,431	317,414	1,682,864
	[0.8, 0.9)	7.300	2.1245	1.7702	7.4685	65,295	344,824	222,004	2,086,597
	[0.9, 1.0]	264.126	10.9695	2.1681	6.1455	12,086	122,576	129,028	2,403,318
WikiText-103	[0.0, 0.1)	0.022	0.5311	5.4428	89.7271	12,942	8,491	677	418
	[0.1, 0.2)	0.031	0.1856	2.0839	31.1121	12,921	8,849	2,213	1,617
	[0.2, 0.3)	0.047	0.2213	1.4747	19.5526	8,929	7,163	2,845	2,567
	[0.3, 0.4)	0.073	0.2350	1.2366	15.0690	7,891	7,451	3,889	4,321
	[0.4, 0.5)	0.144	0.2694	1.2133	11.9555	6,032	7,534	4,427	6,583
	[0.5, 0.6)	0.256	0.3623	1.2105	10.3137	4,477	6,851	4,595	9,165
	[0.6, 0.7)	0.589	0.5344	1.2951	9.0539	2,404	5,255	3,556	10,801
	[0.7, 0.8)	1.543	0.9669	1.4541	8.1912	1,527	4,458	3,286	14,793
	[0.8, 0.9)	7.320	2.1099	1.8352	7.2770	565	2,903	2,257	17,827
	[0.9, 1.0]	1658.877	10.9192	2.1069	5.8171	113	985	1,403	20,539

Table 5: **Exact values underlying Figures 5 and 8 for UDLM.** Mean validation NELBO per token (nats) of the NELBO-only model and the number of tokens behind each mean, by corpus and noise-level bin $1 - \alpha_t$ (fraction of corrupted tokens), for **clean** and **corrupted** positions where the argmax prediction $\hat{x}_\theta$ equals the clean token x or not. †In units of 10^{-3} nats. Gray: fewer than 1,000 tokens (noisier estimate).

		Mean NELBO per token (nats)				Number of tokens			
		Clean		Corrupted		Clean		Corrupted	
Corpus	$1-\alpha_t$	$\hat{x}_\theta = x$†	$\hat{x}_\theta \neq x$	$\hat{x}_\theta = x$	$\hat{x}_\theta \neq x$	$\hat{x}_\theta = x$	$\hat{x}_\theta \neq x$	$\hat{x}_\theta = x$	$\hat{x}_\theta \neq x$
OpenWebText	[0.0, 0.1)	0.018	0.2514	4.8082	86.1054	7,101,661	3,543,893	374,566	191,656
	[0.1, 0.2)	0.025	0.1482	1.8055	30.4774	6,149,687	3,543,007	1,056,380	651,118
	[0.2, 0.3)	0.038	0.1566	1.2567	19.2184	5,063,872	3,423,035	1,601,837	1,223,896
	[0.3, 0.4)	0.060	0.1894	1.0628	14.4402	4,078,135	3,314,353	2,011,194	1,960,158
	[0.4, 0.5)	0.107	0.2455	0.9950	11.7728	3,103,788	3,131,260	2,238,431	2,858,105
	[0.5, 0.6)	0.210	0.3488	1.0089	10.0972	2,221,855	2,908,232	2,271,244	3,990,157
	[0.6, 0.7)	0.473	0.5339	1.0817	8.9078	1,424,470	2,541,104	2,084,878	5,258,604
	[0.7, 0.8)	1.353	0.9326	1.2354	8.0044	783,849	2,063,379	1,736,773	6,775,231
	[0.8, 0.9)	6.160	2.0812	1.5205	7.1903	319,177	1,394,077	1,243,837	8,416,989
	[0.9, 1.0]	604.616	11.1240	1.9855	5.8919	68,777	506,349	714,534	10,022,980
LM1B	[0.0, 0.1)	0.029	0.4898	7.5498	102.6641	440,923	383,999	22,789	21,153
	[0.1, 0.2)	0.042	0.3293	2.8552	35.6002	381,491	387,979	63,564	72,182
	[0.2, 0.3)	0.062	0.3297	1.9539	22.2561	297,758	359,300	90,095	127,855
	[0.3, 0.4)	0.101	0.3760	1.6471	16.5993	235,784	346,085	109,475	201,072
	[0.4, 0.5)	0.179	0.4685	1.5275	13.4078	171,836	321,355	117,685	287,172
	[0.5, 0.6)	0.342	0.6299	1.5377	11.4153	116,833	285,343	112,320	378,432
	[0.6, 0.7)	0.747	0.9177	1.6264	9.9845	72,231	239,601	97,880	477,584
	[0.7, 0.8)	2.023	1.4827	1.7817	8.8357	38,251	184,311	79,517	585,729
	[0.8, 0.9)	8.333	2.9082	2.0005	7.7406	15,953	120,730	59,727	710,342
	[0.9, 1.0]	2399.443	12.9781	2.2236	6.0835	3,680	39,901	38,159	792,244
AG News	[0.0, 0.1)	0.028	0.4651	6.9650	106.3791	20,936	15,130	1,025	797
	[0.1, 0.2)	0.042	0.2893	2.5737	35.4581	20,493	17,257	3,353	3,441
	[0.2, 0.3)	0.062	0.2840	1.8510	22.5358	12,831	13,311	3,861	4,813
	[0.3, 0.4)	0.104	0.3552	1.5563	17.4600	12,153	14,700	5,420	8,687
	[0.4, 0.5)	0.190	0.4509	1.4576	13.8635	8,643	14,521	5,686	13,134
	[0.5, 0.6)	0.376	0.6376	1.4925	11.8373	5,894	13,237	5,423	17,942
	[0.6, 0.7)	0.847	0.9601	1.5335	10.2963	2,955	10,135	3,859	20,939
	[0.7, 0.8)	2.207	1.5690	1.6712	9.2171	1,613	8,456	3,447	26,932
	[0.8, 0.9)	7.840	3.3168	1.8626	8.1033	567	5,359	2,154	31,856
	[0.9, 1.0]	173.881	14.1470	2.0505	6.2160	116	1,852	1,258	35,686
arXiv	[0.0, 0.1)	0.031	0.3746	6.2288	105.4303	3,072,423	1,981,318	153,449	116,074
	[0.1, 0.2)	0.044	0.2311	2.1870	36.6228	2,627,959	1,932,454	425,086	378,213
	[0.2, 0.3)	0.064	0.2361	1.4674	22.6117	2,182,498	1,833,190	649,660	686,076
	[0.3, 0.4)	0.100	0.2674	1.1919	16.5478	1,758,235	1,727,158	819,365	1,055,370
	[0.4, 0.5)	0.170	0.3318	1.0840	13.2530	1,353,026	1,592,558	917,289	1,490,599
	[0.5, 0.6)	0.317	0.4437	1.0612	11.1607	984,752	1,439,683	939,615	2,018,606
	[0.6, 0.7)	0.667	0.6482	1.1078	9.6734	653,254	1,225,230	876,791	2,600,245
	[0.7, 0.8)	1.782	1.0788	1.2697	8.5224	374,800	964,388	718,299	3,282,161
	[0.8, 0.9)	7.503	2.2944	1.6030	7.5019	162,872	651,203	505,555	4,076,850
	[0.9, 1.0]	359.492	12.4100	2.0410	6.1071	32,247	238,181	300,010	4,769,210
PubMed	[0.0, 0.1)	0.028	1.1884	6.5149	106.0867	1,511,100	1,018,355	76,518	58,475
	[0.1, 0.2)	0.040	0.3343	2.3100	36.2489	1,292,426	998,600	212,358	191,784
	[0.2, 0.3)	0.060	0.2941	1.5958	22.3527	1,053,738	950,382	315,722	349,726
	[0.3, 0.4)	0.097	0.3104	1.3390	16.4529	843,782	909,010	392,601	547,215
	[0.4, 0.5)	0.171	0.3555	1.2293	13.1226	638,614	845,229	429,737	783,636
	[0.5, 0.6)	0.331	0.4562	1.2390	11.0649	451,682	761,286	423,499	1,056,141
	[0.6, 0.7)	0.736	0.6458	1.3387	9.6304	285,946	660,783	373,888	1,378,647
	[0.7, 0.8)	2.045	1.0518	1.5358	8.5216	151,044	518,149	291,492	1,708,371
	[0.8, 0.9)	8.441	2.2659	1.8391	7.5513	60,669	349,768	205,718	2,104,613
	[0.9, 1.0]	1073.970	12.4168	2.2015	6.1451	11,573	123,055	123,553	2,408,827
WikiText-103	[0.0, 0.1)	0.024	1.0118	5.2412	96.3114	13,569	8,338	672	461
	[0.1, 0.2)	0.031	0.4319	1.8248	33.2287	13,189	8,989	2,293	1,641
	[0.2, 0.3)	0.052	0.3447	1.4828	21.4422	9,027	7,433	2,874	2,682
	[0.3, 0.4)	0.077	0.3383	1.3226	15.8585	7,867	7,516	3,728	4,441
	[0.4, 0.5)	0.138	0.3656	1.1833	12.3010	6,283	7,541	4,464	6,800
	[0.5, 0.6)	0.261	0.4640	1.2436	10.5837	4,399	6,955	4,379	9,355
	[0.6, 0.7)	0.605	0.6751	1.3156	9.2074	2,421	5,446	3,445	11,216
	[0.7, 0.8)	1.705	1.0182	1.5106	8.3065	1,403	4,737	3,031	15,405
	[0.8, 0.9)	7.846	2.2968	1.8117	7.3334	590	2,775	2,065	17,610
	[0.9, 1.0]	567.680	12.2931	2.1372	5.8313	117	1,074	1,470	21,403

Table 6: **Exact values underlying Figures 5 and 8 for SEDD.** Mean validation NELBO per token (nats) of the NELBO-only model and the number of tokens behind each mean, by corpus and noise-level bin $1-\alpha_t$ (fraction of corrupted tokens), for **clean** and **corrupted** positions where the argmax prediction $\hat{x}_\theta$ equals the clean token x or not. †In units of 10^{-3} nats. Gray: fewer than 1,000 tokens (noisier estimate).

		Mean NELBO per token (nats)				Number of tokens			
		Clean		Corrupted		Clean		Corrupted	
Corpus	$1-\alpha_t$	$\hat{x}_\theta = x$†	$\hat{x}_\theta \neq x$	$\hat{x}_\theta = x$	$\hat{x}_\theta \neq x$	$\hat{x}_\theta = x$	$\hat{x}_\theta \neq x$	$\hat{x}_\theta = x$	$\hat{x}_\theta \neq x$
OpenWebText	[0.0, 0.1)	0.006	0.1122	6.9664	60.0927	44,975,769	18,586,155	570,010	382,626
	[0.1, 0.2)	0.094	0.4139	5.6180	57.0556	3,958,369	2,032,886	559,789	454,652
	[0.2, 0.3)	0.224	0.6183	5.8430	55.4429	2,015,786	1,270,128	539,883	536,443
	[0.3, 0.4)	0.441	0.8860	6.0644	53.4810	1,204,759	953,960	513,454	640,979
	[0.4, 0.5)	0.823	1.2522	6.2645	50.6722	767,219	773,081	480,113	773,571
	[0.5, 0.6)	1.516	1.8145	6.5382	46.9844	510,867	679,067	460,118	990,332
	[0.6, 0.7)	2.976	2.6879	6.6226	41.8275	312,820	567,255	405,832	1,228,525
	[0.7, 0.8)	6.859	4.2608	6.6664	34.8671	190,432	494,755	372,513	1,691,740
	[0.8, 0.9)	21.992	7.5930	6.2161	24.8689	98,726	386,897	324,732	2,500,749
	[0.9, 1.0]	589.139	19.5619	1.7468	3.1931	43,632	255,949	866,816	18,896,835
LM1B	[0.0, 0.1)	0.008	0.0920	8.6592	63.9557	2,903,919	1,910,925	34,754	37,954
	[0.1, 0.2)	0.113	0.4848	7.0151	61.0286	247,147	205,419	32,877	43,453
	[0.2, 0.3)	0.263	0.7810	7.4407	59.3964	126,499	130,755	31,949	52,301
	[0.3, 0.4)	0.519	1.1916	7.7749	57.2406	69,556	90,792	27,981	57,943
	[0.4, 0.5)	0.949	1.7878	8.2122	54.4366	44,642	75,249	25,875	71,322
	[0.5, 0.6)	1.751	2.5869	8.6842	50.6192	27,966	62,018	23,535	86,161
	[0.6, 0.7)	3.530	3.8091	8.8262	44.7385	15,524	47,176	18,788	98,736
	[0.7, 0.8)	7.961	5.9508	8.7397	37.1627	9,609	41,732	17,699	137,808
	[0.8, 0.9)	25.439	9.6504	7.8313	26.2638	5,305	33,093	16,183	205,003
	[0.9, 1.0]	697.746	22.0292	1.5896	3.3037	2,665	19,929	61,284	1,432,666
AG News	[0.0, 0.1)	0.010	0.1004	8.6059	66.7374	134,034	87,786	1,657	1,803
	[0.1, 0.2)	0.133	0.5284	7.5544	65.3083	9,845	8,651	1,286	1,722
	[0.2, 0.3)	0.344	0.9272	7.6416	63.9260	5,775	6,188	1,452	2,457
	[0.3, 0.4)	0.689	1.3195	8.2262	62.3019	2,843	4,133	1,190	2,586
	[0.4, 0.5)	1.136	1.9244	8.5453	58.3105	2,456	4,718	1,376	4,250
	[0.5, 0.6)	2.172	3.0250	9.1466	54.0694	889	2,160	662	2,945
	[0.6, 0.7)	4.283	4.2171	8.7848	48.3291	833	2,955	938	6,026
	[0.7, 0.8)	9.560	6.5947	9.1107	39.1040	302	1,809	570	6,023
	[0.8, 0.9)	27.113	11.0171	7.5178	27.0388	176	1,470	523	9,095
	[0.9, 1.0]	865.570	25.9089	1.6462	3.2496	72	898	1,820	65,818
arXiv	[0.0, 0.1)	0.011	0.2776	7.1886	73.7853	19,128,699	10,928,086	246,368	203,855
	[0.1, 0.2)	0.174	0.7912	5.8121	67.4313	1,659,101	1,170,999	243,408	236,060
	[0.2, 0.3)	0.408	0.9846	5.9938	63.4647	859,485	713,369	238,386	277,208
	[0.3, 0.4)	0.784	1.2472	6.1180	59.6908	516,468	498,321	229,582	314,157
	[0.4, 0.5)	1.386	1.5850	6.2932	55.5943	339,169	400,115	219,514	381,106
	[0.5, 0.6)	2.424	2.1210	6.5232	50.5840	224,746	326,927	206,187	468,380
	[0.6, 0.7)	4.390	2.9880	6.7732	44.5527	142,722	267,698	183,365	579,207
	[0.7, 0.8)	9.079	4.4954	7.0221	36.7034	91,322	232,920	168,081	807,645
	[0.8, 0.9)	27.609	7.6246	6.9697	25.8699	45,721	186,298	138,457	1,213,652
	[0.9, 1.0]	585.046	20.4808	1.9786	3.3328	16,723	125,233	345,527	8,985,029
PubMed	[0.0, 0.1)	0.010	0.3319	8.0136	74.1592	9,411,118	5,659,156	119,174	107,064
	[0.1, 0.2)	0.154	0.8721	6.8068	67.6566	815,313	600,642	115,061	124,792
	[0.2, 0.3)	0.368	1.0510	7.0889	63.8145	421,854	373,939	111,687	149,288
	[0.3, 0.4)	0.718	1.3161	7.4342	60.0986	245,672	265,166	102,650	171,408
	[0.4, 0.5)	1.304	1.6583	7.6368	56.0094	156,019	212,201	95,114	205,338
	[0.5, 0.6)	2.378	2.1629	8.0167	51.0325	100,770	175,732	85,598	252,812
	[0.6, 0.7)	4.654	3.0120	8.2824	44.8822	62,556	143,790	73,123	311,379
	[0.7, 0.8)	9.997	4.5093	8.4930	36.9475	38,043	122,512	63,110	422,479
	[0.8, 0.9)	29.820	7.9068	7.8295	26.1446	19,206	97,746	53,520	626,200
	[0.9, 1.0]	645.013	21.1488	1.6938	3.3757	7,597	63,835	173,982	4,503,898
WikiText-103	[0.0, 0.1)	0.008	0.1784	7.4300	67.1533	88,611	44,509	1,153	895
	[0.1, 0.2)	0.100	0.6166	5.5952	59.9009	7,273	4,184	913	942
	[0.2, 0.3)	0.252	0.8249	6.2021	56.4863	4,077	2,852	1,033	1,254
	[0.3, 0.4)	0.584	1.1212	7.2267	56.0038	2,007	2,019	796	1,322
	[0.4, 0.5)	0.909	1.5419	7.4784	53.6313	1,873	2,145	1,027	2,123
	[0.5, 0.6)	1.816	1.7696	7.6558	46.8919	741	1,108	672	1,575
	[0.6, 0.7)	3.421	2.8474	8.3333	43.8387	921	2,123	1,098	4,562
	[0.7, 0.8)	6.716	4.8043	8.1447	36.9778	139	598	358	1,977
	[0.8, 0.9)	25.539	8.2892	7.7705	25.4684	186	835	539	5,096
	[0.9, 1.0]	414.697	23.3882	1.6172	3.1123	82	487	1,903	39,512

Table 7: **Scaling training data alone does not close the retention gap.** Corrupted-token, clean-token, and overall reconstruction accuracy (%, averaged over corruption rates $p \in \{10, 20, 30, 40\}\%$) on six evaluation corpora, comparing our from-scratch DUO trained on 20B tokens (NELBO-only, black), the same 20B-token model with **NELBO + CTR-Reg** (purple), and the officially released DUO checkpoint trained on **524B tokens** (blue, $\sim$26$\times$ more data, native sequence length 1024). The **bolded** value in each column is the highest of the three settings. Scaling to 524B tokens substantially improves corrupted-token accuracy but leaves clean-token accuracy nearly as low as the 20B baseline, so the clean-minus-corrupted gap does not close with scale; CTR-Reg opens a large gap using 26$\times$ less data. See Table 18 for the per-corruption-rate breakdown underlying the 524B-token row.

		Accuracy (%) by Evaluation Dataset						
Evaluated On	**Setting**	**OWT**	**AG News**	**arXiv**	**LM1B**	**PubMed**	**WT-103**	**Avg**
Corrupted-Tokens	DUO (20B)	54.5	41.5	48.4	35.2	46.3	45.5	45.2
	+ CTR-Reg (20B)	55.0	41.8	49.7	36.5	47.7	47.2	46.3
	DUO (524B)	**60.7**	**48.8**	**54.4**	**39.9**	**53.6**	**52.3**	**51.6**
Clean-Tokens	DUO (20B)	53.9	40.7	52.7	36.3	50.2	49.1	47.2
	+ CTR-Reg (20B)	**83.2**	**72.5**	**79.6**	**69.9**	**78.4**	**80.8**	**77.4**
	DUO (524B)	61.5	49.3	55.8	41.5	53.7	54.7	52.7
Overall	DUO (20B)	53.9	40.7	51.5	35.8	49.0	48.0	46.5
	+ CTR-Reg (20B)	**75.9**	**64.6**	**71.8**	**61.4**	**70.4**	**72.0**	**69.3**
	DUO (524B)	61.1	48.9	55.2	40.9	53.4	53.9	52.2

data-scale bottleneck: training on 26$\times$ more data does not close the gap, whereas CTR-Reg closes it within the 20B-token budget.

Greedy-tail sampling at scale. Table 8 reports PPL and generation entropy for the 524B-token DUO checkpoint under greedy-tail sampling, following the protocol of Table 10. The real-OpenWebText reference entropy at sequence length 1024 is 5.40 (vs. 4.83 at length 512, Table 10), and we use it to color-code entropy in Table 8 in the same way. As with our 20B-token NELBO-only baselines, PPL falls monotonically as the number of greedy-tail steps increases (77.77 at $g = 0$ down to 9.90 at $g = 20$), but this is accompanied by a steep, monotonic collapse in entropy (5.54 down to 2.14) with no sign of stabilizing. Only the $g = 0$ setting (no greedy-tail steps at all) reaches the real-data entropy of 5.40; every greedy-tail step count $g \geq 1$ falls below it, so the PPL improvements from $g = 1$ onward are, by the same criterion as Table 10, achieved through degenerate, low-diversity generation rather than genuine quality gains. Because this checkpoint uses a native sequence length of 1024 rather than 512, we do not compare PPL or entropy magnitudes directly against Table 10; each model is instead judged against its own sequence-length-matched reference entropy. Together with Table 7, this shows that training on substantially more data improves corrupted-token denoising but does not improve the model's ability to leave correct tokens unchanged, and the diversity collapse under greedy-tail sampling persists at scale.

These results indicate that the correct-token retention deficit in USDMs is not a symptom of insufficient training data. Scaling training data by 26$\times$ leaves the clean-vs-corrupted discrimination gap essentially unchanged and does not prevent diversity collapse under greedy-tail sampling, whereas CTR-Reg induces this discrimination directly, at a fraction of the training cost.

G CTR-Reg for Score-Parameterized Models: Details

This appendix supports Section 3.3. Throughout, i indexes the displayed token, m the clean token, and we write s_v for $s_\theta(z_t, t)_v$.

G.1 Reading predictions from scores

The network emits $\log s_\theta(z_t, t)$, so the scores s_v are positive. Under the mean parameterization $s_v = (\bar{x}_\theta)_v / (\bar{x}_\theta)_i$ they determine $\bar{x}_\theta$ only up to its common factor $(\bar{x}_\theta)_i$, which is positive since

Table 8: **Greedy-tail sampling collapses diversity at scale, too.** PPL (GPT2-Large, 1024 generated samples) and generation entropy for the officially released DUO checkpoint (524B training tokens, sequence length 1024, total step budget 512) across greedy-tail step counts g. Entropy is colored green when it meets or exceeds the real-OpenWebText reference entropy at sequence length 1024 (5.40) and red when it falls short, following the same convention as Table 10 (whose reference entropy, 4.83, is computed at sequence length 512 and is therefore not the value used here). A red entropy means the corresponding PPL was achieved via degenerate, low-diversity generation rather than genuine quality. The **bolded** PPL is the lowest across all g. Because this checkpoint's native sequence length (1024) differs from our 20B-token models (512), PPL and entropy values here are *not* directly comparable in magnitude to Table 10; only the qualitative trend, and each model's standing relative to its own reference entropy, should be compared.

Greedy-Tail Steps (g)	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
PPL (↓)	77.77	71.14	71.91	68.67	57.55	47.61	36.01	22.76	16.19	12.56	11.22	10.44	9.97	**9.90**
Entropy	5.54	5.21	4.89	4.58	4.26	3.93	3.63	3.07	2.83	2.61	2.49	2.34	2.21	2.14

$\bar{x}_\theta \geq 1 - \alpha_t > 0$ entrywise. Inverting $\bar{x}_\theta = K\alpha_t x_\theta + (1-\alpha_t)\mathbf{1}$ then gives

$$(x_\theta)_v = \frac{(\bar{x}_\theta)_i \, s_v - (1-\alpha_t)}{K\alpha_t}, \tag{19}$$

which is strictly increasing in s_v because $(\bar{x}_\theta)_i > 0$ is shared by all v. Ranking tokens by score therefore ranks them by x_θ, whatever the value of the common factor, and argmax predictions, in greedy-tail decoding (equation 7) and in the random-token corruption experiment (Section 4.1), are taken directly over the scores. With $s_i = 1$, the rule reads directly: $s_j < 1 \iff (x_\theta)_j < (x_\theta)_i$, so the displayed token is kept exactly when the model ranks it above every alternative.

G.2 The training target ρ_t

SEDD's objective is a per-coordinate regression onto marginal ratios. In equation 13, each off-diagonal score is trained through a term $s_j - r_t(j)\log s_j$, minimized at $s_j = r_t(j)$, where $r_t(j) = q_t(j \mid x)/q_t(i \mid x)$; the remaining factor $g(r_t(j))$ does not involve θ, and for uniform noise the weight $R_t(z_t, z')$ is identical across all $j \neq i$. The sum runs over $j \neq i$ only, so the displayed token's own score is never trained: it equals $q_t(i \mid x)/q_t(i \mid x) = 1$ by definition, which the network enforces by fixing the corresponding log-score to zero. Writing the forward marginal equation 1 with $\pi = \mathbf{1}/K$ as $q_t(v \mid x) = \alpha_t[v = m] + (1-\alpha_t)/K$, and recalling $\bar{x} = K\, q_t(\cdot \mid x)$, the targets are $r_t(j) = \bar{x}_j/\bar{x}_i$, that is

$$r_t(j) = \begin{cases} \rho_t = \dfrac{1-\alpha_t}{K\alpha_t + 1 - \alpha_t}, & \text{clean position } (i = m),\ \forall j \neq i, \\ 1/\rho_t, & \text{corrupted position } (i \neq m),\ j = m, \\ 1, & \text{corrupted position},\ j \notin \{i, m\}. \end{cases} \tag{20}$$

Whenever $K\alpha_t \gg 1 - \alpha_t$, that is for all but the noisiest t, $\rho_t \ll 1$: at a clean position the objective pulls every off-diagonal score well below one, so that the displayed token is retained, while at a corrupted one it pulls s_m far above one, so that the position is revised to m, leaving the remaining wrong tokens at one. Substituting the parameterization $s_j = (\bar{x}_\theta)_j/(\bar{x}_\theta)_i$ into equation 13 reproduces equation 5 term by term, which is the sense in which the two objectives are equivalent (Schiff et al., 2024).

equation 20 is what equation 11 builds on: at an unperturbed position every off-diagonal score has the same target ρ_t, and the hinge is placed there (Section G.3).

G.3 Placing the hinge at the target

The derivative of $\mathrm{softplus}\big(\log(s_j/c)\big)$ with respect to $\log s_j$ is $s_j/(s_j + c)$: close to one for $s_j \gg c$, one half at $s_j = c$, and vanishing for $s_j \ll c$. The threshold c therefore sets how far the penalty pushes. With $c = 1$ the push fades as soon as a score falls below one, which by equation 20 is

Table 9: **Generalization of CTR-Reg across evaluation corpora.** For SEDD, UDLM, and DUO, we compare NELBO-only training (black) to **NELBO + CTR-Reg** (purple) on six corpora: OWT, AG News, arXiv, LM1B, PubMed, and WT-103, reporting corrupted-token, clean-token, and overall reconstruction accuracy averaged over $p \in \{10, 20, 30, 40\}\%$; **Avg** is the mean across all six corpora, and the **bolded** value in each pair is higher. CTR-Reg leaves corrupted-token accuracy essentially unchanged (0.7 points per cell on average) while substantially improving clean-token and overall accuracy across every dataset and model. All models are trained on 20B OpenWebText tokens (169M parameters, sequence length 512); evaluation uses each corpus's validation split.

			Accuracy (%) by Evaluation Dataset						
Evaluated On	**Model**	**Loss**	**OWT**	**AG News**	**arXiv**	**LM1B**	**PubMed**	**WT-103**	**Avg**
Corrupted-Tokens	**DUO**	NELBO	54.5	41.5	48.4	35.2	46.3	45.5	45.2
		NELBO + CTR-Reg	**55.0**	**41.8**	**49.7**	**36.5**	**47.7**	**47.2**	**46.3**
	SEDD	NELBO	**47.2**	**33.6**	**43.1**	28.9	**39.7**	**39.0**	**38.6**
		NELBO + CTR-Reg	47.1	32.6	42.6	28.9	38.5	38.1	38.0
	UDLM	NELBO	**55.3**	**42.8**	47.4	**38.8**	45.4	**47.9**	**46.3**
		NELBO + CTR-Reg	55.2	42.6	**47.6**	38.0	**45.9**	47.4	46.1
Clean-Tokens	**DUO**	NELBO	53.9	40.7	52.7	36.3	50.2	49.1	47.2
		NELBO + CTR-Reg	**83.2**	**72.5**	**79.6**	**69.9**	**78.4**	**80.8**	**77.4**
	SEDD	NELBO	58.4	45.1	54.2	42.5	52.1	55.6	51.3
		NELBO + CTR-Reg	**81.2**	**71.9**	**80.2**	**72.5**	**78.0**	**79.5**	**77.2**
	UDLM	NELBO	56.7	46.0	51.5	40.9	49.6	51.8	49.4
		NELBO + CTR-Reg	**80.1**	**68.2**	**73.0**	**66.2**	**72.8**	**76.4**	**72.8**
Overall	**DUO**	NELBO	53.9	40.7	51.5	35.8	49.0	48.0	46.5
		NELBO + CTR-Reg	**75.9**	**64.6**	**71.8**	**61.4**	**70.4**	**72.0**	**69.3**
	SEDD	NELBO	55.6	42.3	51.3	39.3	48.9	51.4	48.1
		NELBO + CTR-Reg	**72.3**	**62.1**	**70.2**	**61.7**	**67.9**	**69.0**	**67.2**
	UDLM	NELBO	56.1	45.0	50.3	40.3	48.4	50.7	48.5
		NELBO + CTR-Reg	**73.6**	**61.9**	**66.7**	**59.3**	**66.0**	**69.1**	**66.1**

the target for tokens at a corrupted position, far above ρ_t: at $K = 50257$ and $\alpha_t = 0.9$, where $\rho_t \approx 2.2 \times 10^{-6}$, the gradient at $s_j = 10^{-3}$ is about 10^{-3} with $c = 1$ but close to one with $c = \rho_t$. With $c = \rho_t$ the push persists until the score reaches the clean-position target and fades below it, since ρ_t is already the value SEDD's own objective trains toward. Note that $\log \rho_t$ is constant in θ; it matters only because it lies inside the nonlinearity.

H Random-Token Corruption on Additional Corpora

To test whether our findings generalize across domains, we repeat the random-token corruption experiment of Section 4.1 on six corpora: OpenWebText (web text), LM1B and AG News (news), arXiv (scientific papers), PubMed (biomedical abstracts), and WikiText-103 (encyclopedic text). The protocol is unchanged: each corpus is split into sequences of $L = 512$ tokens, $p\%$ of positions are replaced with tokens drawn uniformly from the vocabulary, and each model reconstructs the sequence in a single forward pass.

Table 9 reports Clean- and Corrupted-Token Accuracy, averaged over corruption rates $p \in \{10, 20, 30, 40\}\%$. Two patterns hold on every corpus. First, NELBO-only models reach nearly identical accuracy on clean and corrupted positions, even though clean positions are the easier target, confirming that they revise tokens indiscriminately regardless of domain. Second, CTR-Reg raises Clean-Token Accuracy for all three models on all six corpora, by 26.5 percentage points on average, while changing Corrupted-Token Accuracy by less than 1.7 percentage points in every cell. The retention gain therefore generalizes across domains and does not come at the expense of revision.

Tables 12, 13, 14, 15, and 16 give the full per-corruption-rate breakdown underlying these averages for AG News, arXiv, LM1B, PubMed, and WikiText-103. The same two patterns hold at every individual corruption rate, so the averaged results are not driven by any single value of p.

I Additional Experimental Results

This appendix reports the exact numerical values underlying the results aggregated in the main text.

I.1 Exact Values for the Greedy-Tail Step Sweep

Table 10 reports the exact perplexity and entropy values underlying Figure 4, for every greedy-tail step count g and each of the three USDMs.

I.2 Exact Values Across Sampling-Step Budgets

Table 11 reports the exact perplexity and entropy values underlying Figure 7, for sampling-step budgets from 16 to 1024 under both standard ancestral sampling and greedy-tail sampling.

I.3 Exact Values for the MAUVE Greedy-Tail Sweep

Table 17 reports the exact PPL, entropy, and MAUVE values underlying Figure 3, for every greedy-tail step count $g \in \{0, 1, 3, 6, 9\}$ and each of the three USDMs, with and without CTR-Reg.

I.4 Per-Corruption-Rate Breakdown for the 524B-Token Checkpoint

Section F reports accuracy for the 524B-token DUO checkpoint averaged over corruption rates $p \in \{10, 20, 30, 40\}\%$ across six evaluation corpora. Table 18 gives the full per-corruption-rate breakdown underlying these averages, confirming that corrupted-token and clean-token accuracy remain close to each other at every individual corruption rate, not only on average.

I.5 Exact Values for the CTR-Reg Coefficient Ablation

Table 19 reports the exact PPL and entropy values underlying Figure 6, for DUO and UDLM at $\lambda \in \{0, 0.02, 0.1\}$ across every greedy-tail step count g.

I.6 Per-Corruption-Rate Breakdown for the CTR-Reg Coefficient Ablation

Table 1 reports corruption-probe accuracy averaged over corruption rates $p \in \{10, 20, 30, 40\}\%$ for $\lambda \in \{0, 0.02, 0.1\}$. Table 20 gives the full per-corruption-rate breakdown underlying the $\lambda = 0.1$ row across all six evaluation corpora, for DUO and UDLM.

Table 21 gives the per-corruption-rate breakdown of the copy rate for all three λ. The copy rate stays below 0.35% at every corruption rate, so the averages in Table 1 are not hiding a high-copy regime. For DUO it grows with p (e.g., 0.15%→0.30% on AG News at $\lambda = 0.1$), whereas for UDLM it is flat or slightly decreasing.

Table 10: **Exact PPL and entropy values underlying Figure 4.** For each of three uniform-state diffusion language models and each greedy-tail step count $g \in \{0, 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20\}$ (total step budget fixed at 512), we report perplexity (PPL, computed with GPT2-Large over 1024 generated samples; **bolded** value is the lower, i.e. better, of the two) and generation entropy for the NELBO-only baseline and for NELBO + **CTR-Reg**. Entropy values are colored green when they meet or exceed the real-OpenWebText reference entropy (4.83) and red when they fall short, matching Figure 4; a red entropy means the corresponding PPL was achieved via degenerate, low-diversity generation rather than genuine quality.

		NELBO		NELBO + CTR-Reg	
Model	**Greedy-Tail Steps (g)**	**PPL (↓)**	**Entropy**	**PPL (↓)**	**Entropy**
DUO	0	**105.37**	5.18	105.86	5.19
	1	110.88	4.96	**64.67**	5.09
	2	118.86	4.87	**56.93**	5.08
	3	112.91	4.72	**53.32**	5.06
	4	102.94	4.57	**51.16**	5.06
	5	91.46	4.39	**49.57**	5.05
	6	76.15	4.24	**49.17**	5.05
	8	55.18	3.94	**47.49**	5.05
	10	**42.01**	3.70	47.41	5.05
	12	**32.79**	3.48	47.37	5.05
	14	**27.37**	3.34	47.15	5.06
	16	**22.49**	3.19	46.78	5.05
	18	**19.65**	3.09	46.44	5.04
	20	**18.90**	3.05	47.07	5.04
UDLM	0	95.64	5.18	**93.28**	5.17
	1	81.10	4.96	**58.58**	5.08
	2	93.13	4.92	**51.39**	5.06
	3	91.31	4.82	**47.74**	5.05
	4	91.90	4.76	**46.15**	5.06
	5	87.46	4.67	**44.31**	5.05
	6	83.52	4.58	**43.93**	5.04
	8	73.10	4.40	**43.09**	5.04
	10	59.01	4.20	**41.91**	5.04
	12	48.85	4.02	**42.96**	5.04
	14	**41.34**	3.86	42.60	5.05
	16	**33.10**	3.70	42.38	5.04
	18	**28.73**	3.57	42.02	5.04
	20	**25.67**	3.49	42.61	5.04
SEDD	0	**193.99**	5.22	194.59	5.24
	1	**98.52**	4.91	111.68	5.07
	2	**90.80**	4.80	97.11	5.03
	3	**79.82**	4.68	90.34	5.02
	4	**72.69**	4.58	89.00	5.01
	5	**65.24**	4.46	87.19	5.00
	6	**59.26**	4.35	86.20	5.00
	8	**51.52**	4.16	85.67	5.00
	10	**48.34**	3.95	84.32	5.00
	12	**46.28**	3.72	84.70	5.00
	14	**44.47**	3.46	82.63	4.99
	16	**43.29**	3.18	85.74	5.00
	18	**41.66**	2.98	85.08	5.00
	20	**41.40**	2.78	84.03	5.00

Table 11: **Exact PPL and entropy values underlying Figure 7.** For each of three uniform-state diffusion language models and each total sampling-step budget from 16 to 1024, we report perplexity (PPL, computed with GPT2-Large; **bolded** value is the lower, i.e. better, of the two) and generation entropy for the NELBO-only baseline and for NELBO + **CTR-Reg**, under standard ancestral sampling (DUO's default sampler includes one trailing greedy step) and under greedy-tail sampling (final three steps fixed to be greedy). Entropy values are colored green when they meet or exceed the real-OpenWebText reference entropy (4.83) and red when they fall short. Under greedy-tail sampling, the NELBO-only baseline's lower PPL is consistently paired with red (sub-threshold) entropy, indicating the gain comes from reduced diversity rather than genuine quality; CTR-Reg keeps entropy green while matching or improving PPL.

		Standard Ancestral Sampling				**Greedy-Tail Sampling (final 3 steps)**			
		NELBO		NELBO + CTR-Reg		NELBO		NELBO + CTR-Reg	
Model	**Steps**	**PPL**	**Ent.**	**PPL**	**Ent.**	**PPL**	**Ent.**	**PPL**	**Ent.**
DUO	16	147.50	4.94	**80.53**	5.08	126.75	4.56	**63.63**	5.04
	32	124.84	4.97	**71.23**	5.09	122.84	4.69	**56.90**	5.06
	64	117.41	4.97	**67.88**	5.10	119.18	4.71	**55.10**	5.06
	128	111.73	4.97	**66.25**	5.09	118.33	4.73	**54.03**	5.07
	256	111.39	4.97	**65.27**	5.10	114.16	4.72	**52.86**	5.06
	512	110.88	4.96	**64.67**	5.08	112.91	4.72	**53.32**	5.06
	1024	110.61	4.96	**64.06**	5.09	112.74	4.72	**52.77**	5.06
UDLM	16	124.61	5.15	**124.43**	5.15	105.39	4.68	**56.73**	5.00
	32	105.20	5.16	**103.89**	5.16	98.77	4.78	**51.42**	5.03
	64	**97.16**	5.16	97.80	5.16	95.27	4.81	**49.33**	5.04
	128	**92.71**	5.16	93.71	5.16	92.24	4.82	**48.34**	5.05
	256	**93.57**	5.17	95.02	5.17	93.51	4.82	**47.89**	5.06
	512	95.64	5.17	**93.28**	5.18	91.31	4.82	**47.74**	5.05
	1024	**94.98**	5.17	95.67	5.18	91.96	4.82	**47.61**	5.05
SEDD	16	302.62	5.17	**292.20**	5.17	77.43	4.00	**73.43**	4.80
	32	237.53	5.19	**234.26**	5.20	**84.05**	4.47	84.85	4.89
	64	**206.40**	5.20	206.46	5.21	**82.15**	4.61	87.01	4.96
	128	201.27	5.22	**201.23**	5.22	**79.86**	4.65	90.03	5.00
	256	191.58	5.21	**190.13**	5.22	**79.19**	4.66	89.38	5.00
	512	**193.99**	5.22	194.59	5.23	**79.82**	4.68	90.34	5.02
	1024	**194.34**	5.23	198.93	5.23	**80.83**	4.68	93.38	5.03

Table 12: **Per-corruption-rate results on AG News.** Full breakdown of Table 9 by corruption rate $p \in \{10, 20, 30, 40\}\%$ on AG News, for each of three uniform-state diffusion language models, comparing NELBO alone (black) against NELBO + **CTR-Reg** (purple). The **bolded** value in each pair is the higher of the two; *Avg* rows report the corruption-rate-averaged values used in Table 9.

Model	Loss	Corruption Rate (p)	Corrupted-Token Acc. (%)	Clean-Token Acc. (%)	Overall Acc. (%)
DUO	NELBO	10%	**52.6**	49.1	49.5
		20%	45.6	44.1	44.4
		30%	38.0	38.2	38.2
		40%	29.6	31.4	30.7
		Avg	41.5	40.7	40.7
	+ CTR-Reg	10%	52.5	**80.3**	**77.5**
		20%	**46.0**	**75.8**	**69.9**
		30%	**38.4**	**70.3**	**60.7**
		40%	**30.2**	**63.5**	**50.2**
		Avg	**41.8**	**72.5**	**64.6**
SEDD	NELBO	10%	**43.9**	56.2	54.9
		20%	**37.4**	49.6	47.1
		30%	**30.1**	41.7	38.2
		40%	**22.9**	33.1	29.0
		Avg	**33.6**	45.1	42.3
	+ CTR-Reg	10%	43.0	**82.4**	**78.4**
		20%	36.4	**75.8**	**67.9**
		30%	29.1	**68.6**	**56.7**
		40%	22.0	**60.8**	**45.3**
		Avg	32.6	**71.9**	**62.1**
UDLM	NELBO	10%	**52.3**	53.4	53.3
		20%	**46.4**	49.0	48.4
		30%	**39.7**	43.6	42.5
		40%	**32.9**	37.8	35.8
		Avg	**42.8**	46.0	45.0
	+ CTR-Reg	10%	51.9	**77.6**	**75.0**
		20%	46.2	**72.7**	**67.4**
		30%	39.5	**65.8**	**57.9**
		40%	32.7	**56.7**	**47.1**
		Avg	42.6	**68.2**	**61.9**

Table 13: **Per-corruption-rate results on arXiv.** Full breakdown of Table 9 by corruption rate $p \in \{10, 20, 30, 40\}\%$ on arXiv, for each of three uniform-state diffusion language models, comparing NELBO alone (black) against NELBO + **CTR-Reg** (purple). The **bolded** value in each pair is the higher of the two; *Avg* rows report the corruption-rate-averaged values used in Table 9.

Model	Loss	Corruption Rate (p)	Corrupted-Token Acc. (%)	Clean-Token Acc. (%)	Overall Acc. (%)
DUO	NELBO	10%	57.2	59.8	59.5
		20%	52.0	55.8	55.1
		30%	45.9	50.8	49.3
		40%	38.2	44.4	41.9
		Avg	48.4	52.7	51.5
	+ CTR-Reg	10%	**58.2**	**84.5**	**81.8**
		20%	**53.2**	**81.8**	**76.1**
		30%	**47.2**	**78.4**	**69.0**
		40%	**40.0**	**73.7**	**60.2**
		Avg	**49.7**	**79.6**	**71.8**
SEDD	NELBO	10%	**53.1**	62.6	61.7
		20%	**47.2**	57.9	55.8
		30%	**40.1**	51.9	48.4
		40%	**31.9**	44.4	39.4
		Avg	**43.1**	54.2	51.3
	+ CTR-Reg	10%	53.8	**87.5**	**84.0**
		20%	47.0	**83.4**	**75.9**
		30%	39.1	**78.2**	**66.2**
		40%	30.4	**71.6**	**54.7**
		Avg	42.6	**80.2**	**70.2**
UDLM	NELBO	10%	54.2	56.7	56.4
		20%	50.1	53.6	52.9
		30%	45.4	49.9	48.5
		40%	**40.0**	45.7	43.4
		Avg	47.4	51.5	50.3
	+ CTR-Reg	10%	**54.6**	**79.9**	**77.4**
		20%	**50.2**	**76.2**	**71.0**
		30%	**45.4**	**71.2**	**63.5**
		40%	40.0	**64.6**	**54.8**
		Avg	**47.6**	**73.0**	**66.7**

Table 14: **Per-corruption-rate results on LM1B.** Full breakdown of Table 9 by corruption rate $p \in \{10, 20, 30, 40\}\%$ on LM1B, for each of three uniform-state diffusion language models, comparing NELBO alone (black) against NELBO + **CTR-Reg** (purple). The **bolded** value in each pair is the higher of the two; *Avg* rows report the corruption-rate-averaged values used in Table 9.

Model	Loss	Corruption Rate (p)	Corrupted-Token Acc. (%)	Clean-Token Acc. (%)	Overall Acc. (%)
DUO	NELBO	10%	44.9	43.8	43.9
		20%	38.7	39.3	39.1
		30%	32.0	34.1	33.5
		40%	25.0	28.2	26.9
		Avg	35.2	36.3	35.8
	+ CTR-Reg	10%	**46.4**	**77.7**	**74.5**
		20%	**40.1**	**73.1**	**66.5**
		30%	**33.4**	**67.6**	**57.4**
		40%	**26.3**	**61.0**	**47.1**
		Avg	**36.5**	**69.9**	**61.4**
SEDD	NELBO	10%	**38.1**	53.5	52.0
		20%	**32.0**	46.8	43.9
		30%	25.9	39.0	35.0
		40%	19.8	30.6	26.3
		Avg	28.9	42.5	39.3
	+ CTR-Reg	10%	37.8	**82.7**	**78.2**
		20%	32.0	**76.2**	**67.3**
		30%	**25.9**	**69.3**	**56.3**
		40%	**19.9**	**61.9**	**45.1**
		Avg	28.9	**72.5**	**61.7**
UDLM	NELBO	10%	**47.8**	48.3	48.3
		20%	**42.0**	43.8	43.4
		30%	**35.8**	38.6	37.8
		40%	**29.4**	32.9	31.5
		Avg	**38.8**	40.9	40.3
	+ CTR-Reg	10%	47.0	**75.6**	**72.8**
		20%	41.2	**70.6**	**64.7**
		30%	35.0	**63.8**	**55.2**
		40%	28.7	**54.8**	**44.4**
		Avg	38.0	**66.2**	**59.3**

Table 15: **Per-corruption-rate results on PubMed.** Full breakdown of Table 9 by corruption rate $p \in \{10, 20, 30, 40\}\%$ on PubMed, for each of three uniform-state diffusion language models, comparing NELBO alone (black) against NELBO + **CTR-Reg** (purple). The **bolded** value in each pair is the higher of the two; *Avg* rows report the corruption-rate-averaged values used in Table 9.

Model	Loss	Corruption Rate (p)	Corrupted-Token Acc. (%)	Clean-Token Acc. (%)	Overall Acc. (%)
DUO	NELBO	10%	56.3	58.2	58.0
		20%	50.3	53.6	53.0
		30%	43.4	48.0	46.6
		40%	35.0	40.9	38.5
		Avg	46.3	50.2	49.0
	+ CTR-Reg	10%	**57.3**	**83.8**	**81.2**
		20%	**51.6**	**80.9**	**75.1**
		30%	**44.9**	**77.1**	**67.4**
		40%	**36.9**	**71.8**	**57.8**
		Avg	**47.7**	**78.4**	**70.4**
SEDD	NELBO	10%	**50.7**	61.4	60.3
		20%	**44.0**	56.2	53.8
		30%	**36.2**	49.5	45.6
		40%	**27.8**	41.4	36.0
		Avg	**39.7**	52.1	48.9
	+ CTR-Reg	10%	50.2	**86.5**	**82.8**
		20%	42.9	**81.6**	**73.9**
		30%	34.6	**75.7**	**63.4**
		40%	26.0	**68.3**	**51.4**
		Avg	38.5	**78.0**	**67.9**
UDLM	NELBO	10%	53.3	55.8	55.5
		20%	48.4	52.0	51.3
		30%	43.0	47.7	46.3
		40%	37.0	42.7	40.4
		Avg	45.4	49.6	48.4
	+ CTR-Reg	10%	**54.1**	**80.4**	**77.8**
		20%	**49.0**	**76.4**	**70.9**
		30%	**43.3**	**70.9**	**62.6**
		40%	**37.1**	**63.4**	**52.9**
		Avg	**45.9**	**72.8**	**66.0**

Table 16: **Per-corruption-rate results on WT-103.** Full breakdown of Table 9 by corruption rate $p \in \{10, 20, 30, 40\}\%$ on WT-103, for each of three uniform-state diffusion language models, comparing NELBO alone (black) against NELBO + **CTR-Reg** (purple). The **bolded** value in each pair is the higher of the two; *Avg* rows report the corruption-rate-averaged values used in Table 9.

Model	Loss	Corruption Rate (p)	Corrupted-Token Acc. (%)	Clean-Token Acc. (%)	Overall Acc. (%)
DUO	NELBO	10%	56.5	57.5	57.4
		20%	49.8	52.9	52.3
		30%	42.2	46.9	45.5
		40%	33.6	39.3	37.0
		Avg	45.5	49.1	48.0
	+ CTR-Reg	10%	**57.8**	**87.0**	**84.1**
		20%	**51.3**	**83.6**	**77.2**
		30%	**44.1**	**79.3**	**68.7**
		40%	**35.5**	**73.1**	**58.1**
		Avg	**47.2**	**80.8**	**72.0**
SEDD	NELBO	10%	**50.2**	66.2	64.6
		20%	**43.5**	60.2	56.9
		30%	**35.3**	52.5	47.4
		40%	**26.8**	43.4	36.8
		Avg	**39.0**	55.6	51.4
	+ CTR-Reg	10%	49.2	**88.3**	**84.4**
		20%	42.7	**83.1**	**75.1**
		30%	34.3	**77.0**	**64.2**
		40%	26.0	**69.6**	**52.3**
		Avg	38.1	**79.5**	**69.0**
UDLM	NELBO	10%	**57.1**	58.9	58.7
		20%	**51.2**	54.7	54.0
		30%	**45.0**	49.8	48.4
		40%	**38.3**	43.7	41.5
		Avg	**47.9**	51.8	50.7
	+ CTR-Reg	10%	56.6	**84.2**	**81.5**
		20%	50.5	**80.3**	**74.3**
		30%	44.5	**74.6**	**65.6**
		40%	37.8	**66.5**	**55.0**
		Avg	47.4	**76.4**	**69.1**

Table 17: **Exact PPL, entropy, and MAUVE values underlying Figures 3.** For each of three uniform-state diffusion language models and each greedy-tail step count $g \in \{0, 1, 3, 6, 9\}$ (applied after 500 ancestral sampling steps), we report perplexity (PPL, computed with GPT2-Large; **bolded** value is the lower, i.e. better, of the two), generation entropy, and MAUVE score (**bolded** value is the higher, i.e. better, of the two) for the NELBO-only baseline and for NELBO + **CTR-Reg**. Entropy values are colored green when they meet or exceed the real-OpenWebText reference entropy (4.83) and red when they fall short. As g increases, the NELBO-only baseline's entropy drops below threshold and MAUVE collapses even as its PPL appears to improve, indicating reduced diversity rather than genuine quality; CTR-Reg keeps entropy green and MAUVE high while matching or improving PPL.

		NELBO			NELBO + CTR-Reg		
Model	g	**PPL**	**Ent.**	**MAUVE**	**PPL**	**Ent.**	**MAUVE**
DUO	0	**104.96**	5.187	0.701	105.57	5.188	**0.713**
	1	108.47	4.950	0.414	**64.15**	5.087	**0.760**
	3	113.02	4.720	0.180	**52.70**	5.057	**0.804**
	6	77.57	4.250	0.142	**48.96**	5.054	**0.805**
	9	48.56	3.816	0.166	**47.69**	5.049	**0.807**
UDLM	0	94.33	5.177	0.743	**93.57**	5.175	**0.744**
	1	80.90	4.960	0.470	**57.94**	5.082	**0.790**
	3	91.84	4.822	0.220	**47.38**	5.050	**0.809**
	6	84.36	4.580	0.125	**44.15**	5.050	**0.810**
	9	65.13	4.290	0.141	**44.15**	5.052	**0.828**
SEDD	0	**195.09**	5.228	**0.478**	196.24	5.235	0.451
	1	**98.37**	4.915	0.266	111.08	5.067	**0.471**
	3	**79.27**	4.673	0.155	90.53	5.017	**0.540**
	6	**60.40**	4.362	0.146	86.58	5.007	**0.560**
	9	**48.95**	4.055	0.152	84.56	5.000	**0.576**

Table 18: **Per-corruption-rate results for the official DUO checkpoint (524B tokens).** Full breakdown of the **524B-token DUO** row of Table 7 by corruption rate $p \in \{10, 20, 30, 40\}\%$, across all six evaluation corpora. In each row, the **bolded** value is the higher of corrupted-token vs. clean-token accuracy; *Avg* rows report the corruption-rate-averaged values used in Table 7.

Dataset	Corruption Rate (p)	Corrupted-Token Acc. (%)	Clean-Token Acc. (%)	Overall Acc. (%)
OWT	10%	**71.0**	69.6	69.7
	20%	**65.5**	65.2	65.3
	30%	58.3	**59.7**	59.3
	40%	48.1	**51.4**	50.1
	Avg	60.7	**61.5**	61.1
AG News	10%	**61.8**	58.7	59.0
	20%	**54.4**	53.2	53.4
	30%	45.2	**46.8**	46.3
	40%	34.0	**38.4**	36.7
	Avg	48.8	**49.3**	48.9
arXiv	10%	**65.2**	63.7	63.8
	20%	59.6	59.6	59.6
	30%	51.9	**54.1**	53.5
	40%	41.0	**45.8**	43.9
	Avg	54.4	**55.8**	55.2
LM1B	10%	**51.6**	50.7	50.8
	20%	44.5	**45.4**	45.3
	30%	36.3	**38.9**	38.2
	40%	27.0	**30.8**	29.3
	Avg	39.9	**41.5**	40.9
PubMed	10%	**65.6**	62.5	62.8
	20%	**59.3**	58.0	58.2
	30%	50.8	**51.8**	51.5
	40%	38.6	**42.4**	40.9
	Avg	53.6	**53.7**	53.4
WT-103	10%	**64.9**	64.5	64.6
	20%	58.6	**60.0**	59.7
	30%	49.6	**53.0**	51.9
	40%	36.1	**41.3**	39.2
	Avg	52.3	**54.7**	53.9

Table 19: **Exact PPL and entropy values underlying Figure 6.** For DUO and UDLM and each greedy-tail step count $g \in \{0, 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20\}$ (total step budget fixed at 512), we report perplexity (PPL, GPT2-Large, 1024 generated samples; **bolded** value is the lowest of the three) and generation entropy for $\lambda = 0$ (NELBO-only), $\lambda = 0.02$ (**our default**), and $\lambda = 0.1$ (**blue**). Entropy values are colored green when they meet or exceed the real-OpenWebText reference entropy (4.83) and red when they fall short, matching Figure 6.

		$\lambda = 0$ **(NELBO)**		$\lambda = 0.02$		$\lambda = 0.1$	
Model	**Greedy-Tail Steps (g)**	**PPL (↓)**	**Entropy**	**PPL (↓)**	**Entropy**	**PPL (↓)**	**Entropy**
DUO	0	**105.37**	5.18	105.86	5.19	109.03	5.19
	1	110.88	4.96	**64.67**	5.09	73.24	5.13
	2	118.86	4.87	**56.93**	5.08	67.82	5.11
	3	112.91	4.72	**53.32**	5.06	64.56	5.11
	4	102.94	4.57	**51.16**	5.06	63.52	5.10
	5	91.46	4.39	**49.57**	5.05	63.26	5.12
	6	76.15	4.24	**49.17**	5.05	64.81	5.12
	8	55.18	3.94	**47.49**	5.05	61.84	5.11
	10	**42.01**	3.70	47.41	5.05	63.19	5.11
	12	**32.79**	3.48	47.37	5.05	62.77	5.13
	14	**27.37**	3.34	47.15	5.06	61.12	5.11
	16	**22.49**	3.19	46.78	5.05	62.65	5.11
	18	**19.65**	3.09	46.44	5.04	63.03	5.11
	20	**18.90**	3.05	47.07	5.04	64.32	5.13
UDLM	0	95.64	5.18	**93.28**	5.17	95.91	5.18
	1	81.10	4.96	**58.58**	5.08	67.42	5.12
	2	93.13	4.92	**51.39**	5.06	62.16	5.12
	3	91.31	4.82	**47.74**	5.05	58.97	5.11
	4	91.90	4.76	**46.15**	5.06	58.09	5.11
	5	87.46	4.67	**44.31**	5.05	57.59	5.11
	6	83.52	4.58	**43.93**	5.04	57.75	5.11
	8	73.10	4.40	**43.09**	5.04	58.05	5.12
	10	59.01	4.20	**41.91**	5.04	57.35	5.11
	12	48.85	4.02	**42.96**	5.04	57.53	5.11
	14	**41.34**	3.86	42.60	5.05	57.74	5.11
	16	**33.10**	3.70	42.38	5.04	57.96	5.11
	18	**28.73**	3.57	42.02	5.04	57.60	5.11
	20	**25.67**	3.49	42.61	5.04	58.35	5.11

Table 20: **Per-corruption-rate results at $\lambda = 0.1$ (blue).** Full breakdown of the $\lambda = 0.1$ row of Table 1 by corruption rate $p \in \{10, 20, 30, 40\}\%$, across all six evaluation corpora, for DUO and UDLM. *Avg* rows report the corruption-rate-averaged values used there.

Dataset	Model	Corruption Rate (p)	Corrupted-Token Acc. (%)	Clean-Token Acc. (%)	Overall Acc. (%)
AG News	DUO	10%	52.6	89.7	86.0
		20%	45.9	86.8	78.6
		30%	38.3	82.9	69.5
		40%	29.9	78.0	58.7
		Avg	41.7	84.3	73.2
	UDLM	10%	53.2	88.2	84.7
		20%	47.3	84.4	77.0
		30%	40.5	79.2	67.6
		40%	33.5	72.4	56.8
		Avg	43.7	81.0	71.5
arXiv	DUO	10%	57.4	91.0	87.6
		20%	52.3	89.5	82.0
		30%	46.1	87.5	75.1
		40%	38.8	84.8	66.4
		Avg	48.6	88.2	77.8
	UDLM	10%	54.9	87.9	84.6
		20%	50.1	85.0	78.0
		30%	45.1	81.7	70.7
		40%	39.6	77.5	62.3
		Avg	47.4	83.0	73.9
LM1B	DUO	10%	45.5	87.7	83.5
		20%	39.0	84.7	75.6
		30%	32.3	80.8	66.3
		40%	25.3	75.9	55.7
		Avg	35.5	82.3	70.2
	UDLM	10%	47.0	86.1	82.2
		20%	41.0	82.1	73.9
		30%	34.7	76.7	64.1
		40%	28.3	69.8	53.2
		Avg	37.7	78.7	68.3
PubMed	DUO	10%	56.7	90.2	86.9
		20%	50.8	88.6	81.1
		30%	44.0	86.4	73.7
		40%	35.9	83.3	64.3
		Avg	46.8	87.1	76.5
	UDLM	10%	54.3	87.6	84.2
		20%	48.6	84.5	77.3
		30%	42.8	80.9	69.4
		40%	36.6	76.1	60.3
		Avg	45.6	82.3	72.8
WT-103	DUO	10%	57.3	93.2	89.6
		20%	50.4	91.3	83.1
		30%	42.9	88.8	75.0
		40%	34.3	85.1	64.8
		Avg	46.2	89.6	78.1
	UDLM	10%	57.1	92.0	88.5
		20%	51.0	89.2	81.6
		30%	44.7	85.6	73.3
		40%	37.8	80.2	63.3
		Avg	47.7	86.8	76.7
OWT	DUO	10%	63.9	94.8	91.8
		20%	58.5	93.2	86.3
		30%	51.9	90.9	79.2
		40%	44.2	87.6	70.2
		Avg	54.6	91.6	81.9
	UDLM	10%	63.8	94.0	90.9
		20%	58.8	91.7	85.1
		30%	53.0	88.6	77.9
		40%	46.6	84.2	69.2
		Avg	55.5	89.6	80.8

Table 21: **Per-corruption-rate copy rate.** Full breakdown of the copy-rate rows of Table 1 by corruption rate p, for DUO and UDLM at $\lambda = 0$ (NELBO-only), 0.02 (**our default**), and 0.1 (**blue**), in %. *Avg* rows report the corruption-rate-averaged values used there.

		DUO			UDLM		
Dataset	p	0	**0.02**	**0.1**	0	**0.02**	**0.1**
AG News	10%	0.004	0.03	0.15	0.007	0.03	0.08
	20%	0.004	0.04	0.18	0.010	0.03	0.09
	30%	0.004	0.05	0.23	0.005	0.03	0.09
	40%	0.004	0.06	0.30	0.005	0.03	0.09
	Avg	0.004	0.05	0.22	0.007	0.03	0.09
arXiv	10%	0.004	0.05	0.19	0.005	0.04	0.12
	20%	0.005	0.06	0.21	0.006	0.04	0.12
	30%	0.005	0.06	0.25	0.006	0.03	0.11
	40%	0.006	0.07	0.31	0.007	0.02	0.09
	Avg	0.005	0.06	0.24	0.006	0.03	0.11
LM1B	10%	0.002	0.040	0.18	0.004	0.03	0.13
	20%	0.003	0.050	0.22	0.004	0.04	0.13
	30%	0.003	0.057	0.25	0.005	0.03	0.12
	40%	0.004	0.063	0.29	0.005	0.02	0.11
	Avg	0.003	0.05	0.24	0.005	0.03	0.12
PubMed	10%	0.004	0.05	0.17	0.005	0.04	0.12
	20%	0.005	0.06	0.20	0.006	0.04	0.12
	30%	0.006	0.06	0.24	0.007	0.03	0.11
	40%	0.006	0.07	0.29	0.007	0.03	0.10
	Avg	0.005	0.06	0.23	0.006	0.04	0.11
WT-103	10%	0.008	0.04	0.15	0.004	0.03	0.12
	20%	0.006	0.04	0.23	0.006	0.03	0.11
	30%	0.005	0.05	0.25	0.009	0.03	0.11
	40%	0.006	0.08	0.33	0.010	0.03	0.10
	Avg	0.006	0.05	0.24	0.007	0.03	0.11
OWT	10%	0.003	0.03	0.12	0.002	0.026	0.09
	20%	0.003	0.03	0.14	0.003	0.020	0.09
	30%	0.003	0.04	0.17	0.003	0.020	0.09
	40%	0.003	0.04	0.20	0.004	0.020	0.08
	Avg	0.003	0.04	0.16	0.003	0.02	0.09