Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution

SEDD: learning positive ratios to reverse token corruption

Paper by Aaron Lou, Chenlin Meng, and Stefano Ermon

Independent research reading presentation

ICML 2024 arXiv:2310.16834v3

Paper revision: June 6, 2024 Reading: October 14, 2026

The missing ingredient is a trainable discrete score

Research question

Can token diffusion compete with autoregressive language modeling while updating many positions together?

$$\underbrace{\frac{p_t(y)}{p_t(x)}}_{\text{reverse-process information}} \xrightarrow{\text{score entropy}} \underbrace{s_\theta(x,t)_y}_{\text{positive network output}}$$

- Discrete ratios replace the continuous score $\nabla_x \log p_t(x)$.
- Denoising makes ratio learning tractable with one network pass.
- The same objective connects to likelihood and flexible generation.

Paper, Secs. 1–3, PDF pp. 1–4.

Reverse jump rates require only probability ratios

For column probability vectors, $Q_t(y, x)$ is the forward rate $x \to y$:

$$\frac{dp_t}{dt} = Q_t p_t, \qquad Q_t(y,x) \geq 0 \ (y \neq x), \qquad \sum_y Q_t(y,x) = 0.$$

Time reversal

$$\bar{Q}_t(y,x) = Q_t(x,y)\frac{p_t(y)}{p_t(x)} \approx Q_t(x,y)s_\theta(x,t)_y, \quad y \neq x.$$

- Reverse probability flow: $p_t(x)\bar{Q}_t(y,x) = p_t(y)Q_t(x,y)$.
- Set each diagonal to minus the sum of its outgoing rates.
- Ratios are positive values; they need not sum to one.

Paper, Eqs. (1)–(3), PDF p. 2; Definition 3.5, PDF p. 3.

Score entropy penalizes missing positive mass

Write $r = p(y)/p(x)$ and let $s > 0$ be its prediction:

$$\boxed{\ell(s; r) = s - r + r \log(r/s)} \qquad \mathcal{L}_{\mathrm{SE}} = \mathbb{E}_{x \sim p} \sum_{y \neq x} w_{xy} \ell(s_\theta(x)_y; r).$$

$$\frac{\partial \ell}{\partial s} = 1 - \frac{r}{s}, \qquad \frac{\partial^2 \ell}{\partial s^2} = \frac{r}{s^2} > 0.$$

- For $r > 0$, the unique minimum is $s = r$ with zero loss.
- As $s \to 0$ with $r > 0$, the logarithmic penalty diverges.
- Consistency assumes full support, positive weights, and sufficient capacity.

Paper, Eq. (5), Proposition 3.2, PDF p. 3; proof, PDF p. 12.

Log outputs reveal why squared error can struggle

The network uses positive outputs. For the teaching parameterization $s = e^z$:

$$\frac{\partial \ell_{\mathrm{SE}}}{\partial z} = s - r, \qquad \frac{\partial \ell_{\mathrm{CSM}}}{\partial z} = s(s - r).$$

Example: true ratio $r = 0.2$, prediction $s = 0.001$

Score entropy gradient: -0.199

Squared-error gradient: -0.000199

- Squared error adds a small multiplicative factor when s is tiny.
- Positive outputs alone do not remove this optimization difference.

Teaching derivation from Eq. (5); exponentiated outputs: Appendix C.2, PDF p. 15.

Denoising replaces unknown ratios with known kernels

Sample $x_0 \sim p_0$ and $x_t \sim p_{t|0}(\cdot|x_0)$. Define

$$r_t(y; x_t, x_0) = \frac{p_{t|0}(y|x_0)}{p_{t|0}(x_t|x_0)}.$$

$$\mathcal{L}_{\mathrm{DSE}} = \mathbb{E}_{x_0,x_t} \sum_{y \neq x_t} w_{x_t y} \left[s_\theta(x_t, t)_y - r_t(y; x_t, x_0) \log s_\theta(x_t, t)_y \right].$$

- Averaging the conditional ratio over $x_0|x_t$ recovers $p_t(y)/p_t(x_t)$.
- Equal to score entropy up to a parameter-independent constant.
- One network input x_t supplies all required outputs.

Paper, Theorem 3.4 and Eq. (7), PDF p. 3; proof, PDF p. 12.

Diffusion weighting yields a likelihood upper bound

Choose $w_{x_t y} = Q_t(x_t, y)$ and integrate over time:

$$\mathcal{L}_{\mathrm{DWDSE}}(x_0) = \int_0^T \mathbb{E}_{x_t|x_0} \sum_{y \neq x_t} Q_t(x_t, y)\, \ell(s_\theta(x_t, t)_y;\, r_t(y; x_t, x_0))\, dt.$$

$$\boxed{-\log p_0^\theta(x_0) \leq \mathcal{L}_{\mathrm{DWDSE}}(x_0) + D_{\mathrm{KL}}(p_{T|0}(\cdot|x_0) \,\|\, p_{\mathrm{base}})}$$

- Proof idea: path-space KL bounds endpoint KL by data processing.
- Training can drop constants; likelihood evaluation needs the full bound.
- Reported diffusion perplexities are Monte Carlo estimates of upper bounds.

Paper, Theorem 3.6, Eqs. (9)–(10), PDF pp. 3–4; Appendix A, pp. 12–13.

Sparse token corruption avoids an exponential output

A sequence has d positions and vocabulary size n: $|\mathcal{X}| = n^d$.

$$s_\theta(\boldsymbol{x}, t)_{i,a} \approx \frac{p_t(x^1, \ldots, a, \ldots, x^d)}{p_t(\boldsymbol{x})} \qquad \text{output shape } d \times n.$$

- Independent forward token jumps connect only Hamming-distance-one states.
- Zero transition weights remove all other ratios from the objective.
- A bidirectional Transformer still models dependencies across the sequence.

$$p^{\text{seq}}_{t|0}(\boldsymbol{x}_t|\boldsymbol{x}_0) = \prod_{i=1}^{d} p^{\text{tok}}_{t|0}(x^i_t|x^i_0).$$

Paper, Sec. 3.3, Eqs. (11)–(14), PDF p. 4.

Absorb and Uniform provide closed-form corruption

Let $\alpha(t) = e^{-\bar{\sigma}(t)}$, with $\bar{\sigma}(t) = \int_0^t \sigma(u)\, du$.

Uniform

$$p_{t|0}^{\text{tok}}(\cdot|a) = \alpha\delta_a + (1-\alpha)\frac{\mathbf{1}}{n}.$$

Corrupt into random tokens.

Start generation from uniform noise.

Absorb

$$p_{t|0}^{\text{tok}}(\cdot|a) = \alpha\delta_a + (1-\alpha)\delta_{\text{MASK}}.$$

Corrupt into an absorbing MASK.

Start generation from all MASK.

Uniform uses the appendix's $1/n$ generator scaling. Absorb gives the strongest combined likelihood and generation results.

Paper, Eqs. (15)–(16), PDF p. 4; Appendix C.1, PDF p. 15; Tables 1–3.

Training needs one noisy sequence and one forward pass

1. Draw clean text $\boldsymbol{x}_0$ and a random time t.
2. Corrupt each token with the known transition kernel.
3. Run a time-conditioned bidirectional Transformer on $\boldsymbol{x}_t$.
4. Weight local denoising losses by $Q_t^{\mathrm{tok}}(x_t^i, a)$ and update.

Architecture and optimization

RoPE; adaLN-zero time conditioning; exponentiated score outputs.
Batch 512; learning rate 3×10^{-4}; warmup 2000 iterations;
gradient norm clipped to 1; EMA 0.9999.

Paper, Algorithm 1, PDF p. 14; Appendix C.2–C.4, PDF pp. 15–16.

Tweedie sampling improves parallel reverse updates

Euler τ-leaping samples all positions independently using local reverse rates. The alternative uses $\Delta\bar{\sigma} = \bar{\sigma}(t) - \bar{\sigma}(t - \Delta t)$:

$$\widetilde{P}_i(a|\boldsymbol{x}_t) = \left[e^{-\Delta\bar{\sigma}Q^{\mathrm{tok}}} s_\theta(\boldsymbol{x}_t, t)_i\right]_a \left[e^{\Delta\bar{\sigma}Q^{\mathrm{tok}}}\right]_{x_t^i, a}.$$

- Insert the current token's self-ratio, $s_{i,x_t^i} = 1$.
- With exact scores, these are the correct reverse token marginals.
- Their product minimizes forward KL among independent token updates.
- Learned scores may require nonnegative clipping and normalization.

Paper, Theorems 4.1–4.2, Eqs. (18)–(20), PDF p. 5; Algorithm 2, p. 15.

Conditional ratios reuse the same unconditional network

Fix tokens $\boldsymbol{y}$ outside the generated positions Ω:

$$\frac{p_t(\boldsymbol{z}'|\boldsymbol{y})}{p_t(\boldsymbol{z}|\boldsymbol{y})} = \frac{p_t(\boldsymbol{z}' \oplus_\Omega \boldsymbol{y})}{p_t(\boldsymbol{z} \oplus_\Omega \boldsymbol{y})}.$$

Sampling rule

Place the prompt in the sequence. Predict scores for the whole sequence, but update only unfilled positions at each step.

- Prompts can be prefixes, suffixes, or disjoint spans.
- In Absorb, an unmasked prompt token also identifies its clean token.
- Same-time conditional ratios alone do not justify every corruption scheme.

Paper, Sec. 4.2, Eqs. (21)–(22), PDF p. 5; Algorithm 3, p. 15. Last two points: interpretation of the absorbing kernel.

Three metrics measure different objects

Metric	Evaluated object	Interpretation
BPC / model PPL ↓	Real data under the trained model	Density modeling; diffusion reports upper bounds
Generative PPL ↓	Generated samples under GPT-2 large	Typicality under an external evaluator; annealing-sensitive
MAUVE ↑	Generated and reference text sets	Distribution-level quality and coverage

A low generative PPL is not a measurement of factual accuracy.

Paper, Secs. 5.2–5.3, PDF pp. 5–8; Appendix C.6–C.8, p. 16.

The three likelihood settings use different protocols

- **text8:** character modeling; prior D3PM-style architecture and splits.
- **1BW training:** matched tokenization and size from DiffusionBERT; includes a separately trained autoregressive Transformer.
- **GPT-2 zero-shot:** SEDD trained on OpenWebText; public GPT-2 trained on WebText.

Comparable scale, with parameter differences

Non-embedding parameters: SEDD about 90M / 320M versus GPT-2 about 86M / 304M. SEDD also separates input and output word matrices.

Paper, Sec. 5.1–5.2, PDF pp. 5–6; Appendix C.2 and C.5, pp. 15–16.

SEDD improves the non-autoregressive text8 result

Model	BPC ↓
Autoregressive	1.23
Discrete Flow (AR base)	1.23
MAC	≤ 1.40
BFN	≤ 1.41
D3PM Absorb	≤ 1.45
SEDD Uniform	≤ 1.47
SEDD Absorb	$\mathbf{\leq 1.39}$

- Best reported non-autoregressive value in this comparison.
- 0.06 BPC below D3PM Absorb's bound.
- Still 0.16 BPC above the autoregressive result.

Paper, Table 2, PDF p. 6; selected rows. Diffusion values are upper bounds.

On 1BW, Absorb approaches the autoregressive baseline

Model	PPL ↓
Autoregressive	31.98
D3PM Absorb	$\leq$ 77.50
Diffusion-LM	$\leq$ 118.62
BERT-Mouth	$\leq$ 142.89
DiffusionBERT	$\leq$ 63.78
SEDD Uniform	$\leq$ 40.25
SEDD Absorb	$\leq$ **32.79**

- Reported bound is 57.7% below D3PM Absorb.
- 48.6% below DiffusionBERT.
- Gap to the AR value: 0.81 PPL.

This is training on 1BW, distinct from zero-shot 1BW evaluation.

Paper, Table 3, PDF p. 6. Percentages calculated from reported values; bound tightness can differ.

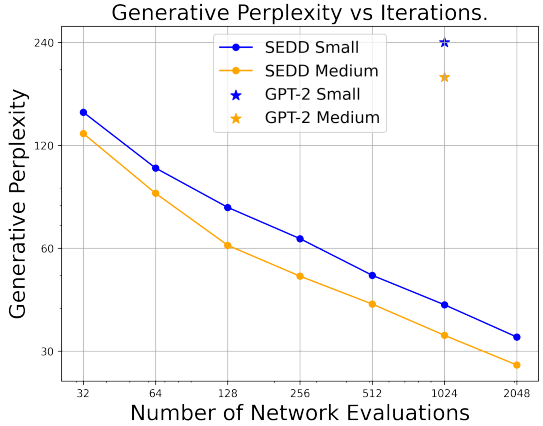
SEDD Absorb leads on 3 of 5 zero-shot columns

	Small		Medium	
Dataset	GPT-2	SEDD	GPT-2	SEDD
LAMBADA	**45.04**	$\leq$ 50.92	**35.66**	$\leq$ 42.77
WikiText2	42.43	$\leq$ **41.84**	31.80	$\leq$ **31.04**
PTB	138.43	$\leq$ **114.24**	123.14	$\leq$ **87.12**
WikiText103	41.60	$\leq$ **40.62**	31.39	$\leq$ **29.98**
1BW	**75.20**	$\leq$ 79.29	**55.72**	$\leq$ 61.19

No sliding window. WikiText2 uses its train split. GPT-2 values were recomputed except for 1BW.

Paper, Table 1 and Sec. 5.2.3, PDF p. 6; Appendix C.6, p. 16. PPL lower is better; bold compares reported values.

More reverse steps improve Absorb sample quality



- 1000 samples, each 1024 tokens; GPT-2 large evaluator.
- 32 steps: comparable or better than unannealed GPT-2.
- 2048 steps: reported 6–8× lower generative PPL.

Both axes are logarithmic. No temperature scaling.

Paper, original Fig. 1(a) and Sec. 5.3.1, PDF p. 7; Appendix C.7, p. 16.

Fewer network calls do not directly specify latency

Autoregressive inference

About 1024 sequential token steps.

KV cache reuses previous states; each new step has a different cost from a full-sequence pass.

SEDD inference

32–2048 full-sequence updates.

No growing KV cache; each step revisits the sequence.

- $1024/32 = 32$: a network-evaluation ratio.
- Authors observed similar runtime at about 100 SEDD steps.
- Removing KV cache allowed about 4–6$\times$ larger batches in their setup.

Paper, Sec. 6, PDF pp. 8–9. Runtime and batch observations use the authors' unoptimized implementation.

Conditional quality is competitive without annealing

Method	Annealing	MAUVE ↑
GPT-2	Nucleus .95	.955
GPT-2	None	.802
SSD-LM	Threshold .95	.919
SSD-LM	None	.312
SEDD Standard	None	**.957**
SEDD Infill	None	.942

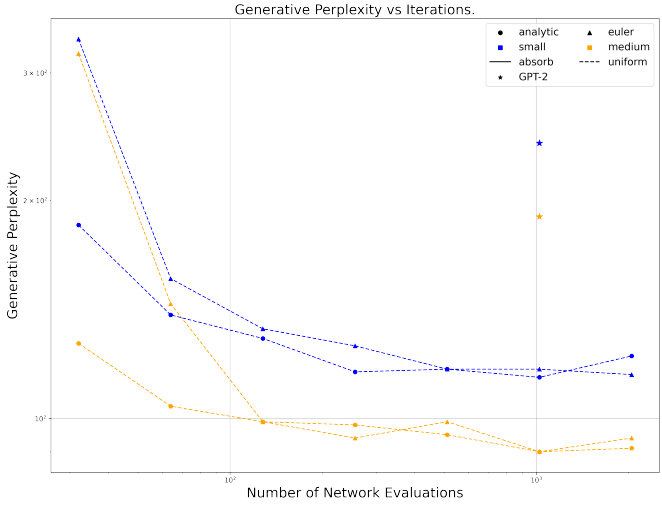
All models: medium size.

- Standard: 50 prefix tokens, generate 50.
- Infill: 25 prefix + 25 suffix tokens, generate 50.
- 5000 generated / 1000 reference samples.

The .002 lead over nucleus GPT-2 is small; no uncertainty interval is reported.

Paper, Table 5, PDF p. 8; Appendix C.8, PDF p. 16.

Uniform generation saturates as reverse steps increase



Original Fig. 2, PDF p. 17. Blue/orange: small/medium; circles/triangles: Tweedie/Euler.

Squared error causes a large optimization failure

Loss ablation at GPT-2 scale

Replace score entropy with squared error while retaining the Q_t weights. The authors report 3–4$\times$ higher likelihood loss, corresponding to approximately 10,000$\times$ higher perplexity.

- The exponential relation between NLL and perplexity explains the very different multipliers.
- This supports the importance of loss geometry when learning ratios.
- Separately, Uniform plateaus; the main generation gains come from Absorb.

Paper, Appendix D.1–D.2, PDF pp. 17–18. The loss ablation is reported as an aggregate observation without per-setting values.

SEDD aligns ratios, loss geometry, and generation

1. **A suitable target:** local positive ratios determine the reverse continuous-time chain.
2. **A scalable objective:** denoising score entropy learns these ratios and yields a likelihood bound.
3. **Useful generation controls:** parallel reverse steps and prompts at arbitrary positions.

Practical tradeoff

Absorb is strongest in these GPT-2-scale experiments. Full-sequence iteration cost, fixed output length, and scaling remain consequential design questions.

Paper, Secs. 3–7, PDF pp. 3–9.

Paper and resources

Aaron Lou, Chenlin Meng, and Stefano Ermon

Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution.
ICML 2024. arXiv:2310.16834v3, revised June 6, 2024.

- Versioned arXiv paper
- Author code repository
- Original figures: Fig. 1(a), PDF p. 7; Fig. 2, PDF p. 17.

Main presentation: approximately 25 minutes.
Backup slides: derivations, protocol details, and formula interpretation.

Backup: the denoising identity is an average identity

Under the appropriate support conditions,

$$\mathbb{E}_{x_0|x_t=x}\left[\frac{p_{t|0}(y|x_0)}{p_{t|0}(x|x_0)}\right] = \sum_{x_0} \frac{p_0(x_0)p_{t|0}(x|x_0)}{p_t(x)} \frac{p_{t|0}(y|x_0)}{p_{t|0}(x|x_0)} = \frac{p_t(y)}{p_t(x)}.$$

- The conditional target varies with the sampled clean sequence.
- Its posterior average gives the marginal ratio required for generation.
- The objective uses only computable corruption probabilities.

Expanded teaching derivation of Theorem 3.4; Paper, PDF pp. 3 and 12.

Backup: why the product of marginals is KL-optimal

Fix the current sequence x and write the exact reverse joint as $P(z|x)$. Restrict the sampler to $q(z|x) = \prod_i q_i(z_i|x)$:

$$D_{\mathrm{KL}}\left(P \,\middle\|\, \prod_i q_i\right) = C - \sum_i \mathbb{E}_{z_i \sim P_i} \log q_i(z_i|x).$$

Each term is minimized by $q_i = P_i$, the true reverse token marginal.

- Applying $e^{-\Delta\bar{\sigma}Q}$ only along coordinate i reverses noise on that coordinate while leaving the others noisy.
- Bayes' rule then supplies $P_i(z_i|x)$ from local ratios.
- The independent product can still lose within-step correlations.

Teaching explanation of Theorems 4.1–4.2, PDF p. 5; proof, pp. 13–14.

Backup: absorbing scores contain a masked-token posterior

For masked position i and visible context c, let $\alpha = e^{-\bar{\sigma}(t)}$. Independent absorbing corruption gives

$$s^*_{i,a} = \frac{\alpha}{1-\alpha}\, p_0(X^i_0 = a|c).$$

The sample loss at this position, up to constants, is

$$\sigma(t)\left[\sum_a s_{\theta,i,a} - \frac{\alpha}{1-\alpha}\log s_{\theta,i,x^i_0}\right].$$

If $s_{\theta,i,a} = \frac{\alpha}{1-\alpha}\pi_{\theta,i,a}$ with normalized π_θ, this becomes a weighted cross entropy.

The ratio scale calibrates the reverse jump rates across noise levels.

Teaching derivation from Eqs. (10), (12), (16), PDF p. 4; not an additional paper experiment.

Backup: small-model diffusion baselines remain behind

Dataset	D3PM	PLAID	Uniform	Absorb
LAMBADA	93.47	57.28	65.40	**50.92**
WikiText2	77.28	51.80	50.27	**41.84**
PTB	200.82	142.60	140.12	**114.24**
WikiText103	75.16	50.86	49.60	**40.62**
1BW	138.92	91.12	101.37	**79.29**

All entries are perplexity upper bounds. D3PM and PLAID were retrained using the authors' small-model architecture and training settings.

Paper, Table 1, PDF p. 6; Appendix C.5.3, PDF p. 16.

Backup: numerical conventions affect implementation

- **Uniform scaling:** the appendix scales the generator by $1/n$; then the retention factor is $e^{-\bar{\sigma}}$.
- **Likelihood endpoints:** the absorbing prior has tiny non-MASK mass to avoid infinite prior KL at finite noise.
- **Evaluation:** likelihood estimates use 1000 random time points.
- **Loss weights:** retain $Q_t(x_t, y)$ from Eq. (10); Absorb trains relevant reverse edges at masked positions.

Paper, Eq. (10), PDF p. 4; Algorithm 1, p. 14; Appendix C.1 and C.6, pp. 15–16.

Backup: two formula descriptions need care

Score entropy is generally asymmetric

$\ell(2;1) = 1 - \log 2$, but $\ell(1;2) = -1 + 2\log 2$.
Use nonnegativity and convexity in the predicted score; the p. 3 remark's symmetry claim does not follow from Eq. (5).

The printed log-linear schedule conflicts with the prose

Appendix C.1 prints $-\log(1-(1-\epsilon t)) = -\log(\epsilon t)$.
A schedule consistent with increasing linear corruption is $-\log(1-(1-\epsilon)t)$, derived here from the stated intent.

Formula interpretation based on Paper, PDF pp. 3, 5, and 15. The second expression is an inferred correction, not a verbatim paper formula.