# ELF: Embedded Language Flows

## Continuous denoising, shared final-step decoding

Paper: Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao,
Tianhong Li, Yoon Kim, Jacob Andreas, Kaiming He

MIT | Independent paper discussion

arXiv:2605.10938v2 June 26, 2026
Reading: October 14, 2026

Keep the trajectory continuous; recover tokens with the same network at the end.

# Does discrete output require token prediction at every step?

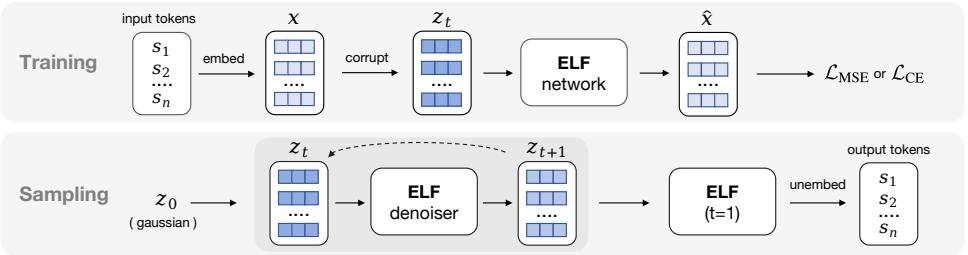
**Research question**

Is the gap between continuous and discrete language diffusion inherent, or partly a design problem?

| Design family | Interface to discrete text |
|---|---|
| Discrete diffusion | Denoise directly in token space. |
| Many continuous DLMs | Predict tokens during intermediate training steps. |
| Latent diffusion LMs | Use a separate latent-to-text decoder. |
| **ELF** | **Continuous flow; a shared final decoding mode.** |

Paper, Secs. 1–2, PDF pp. 1–3; App. A, Table 2, pp. 17–18. Families contain exceptions.

# One network handles both continuous and discrete recovery




- Train: encode text, corrupt embeddings, select denoise or decode mode.
- Generate: update the whole embedding sequence, then decode once.
- Frozen T5-small supplies 512-dimensional contextual training targets.

Paper, Fig. 3, PDF p. 3; Sec. 4, p. 6. Original figure.

# Flow Matching connects noise to clean embeddings

$$\mathbf{z}_t = t\mathbf{x} + (1-t)\boldsymbol{\epsilon}, \qquad t \in [0,1]$$

$$\mathbf{v} = \frac{d\mathbf{z}_t}{dt} = \mathbf{x} - \boldsymbol{\epsilon}$$

## State

$\mathbf{x}, \mathbf{z}_t, \boldsymbol{\epsilon} \in \mathbb{R}^{L \times d}$

OWT: $L = 1024$, $d = 512$.

$t = 0$: noise; $t = 1$: data.

## Parameterization

The network predicts $\mathbf{x}_\theta$.

$$\mathbf{v}_\theta = \frac{\mathbf{x}_\theta - \mathbf{z}_t}{1-t}$$

Velocity is computed from that prediction.

Paper, Sec. 3.1, PDF p. 4. Training uses scaled Gaussian noise; details in backup.

# Predict clean data, but train with a velocity loss

$$\mathcal{L}_{\mathrm{MSE}} = \mathbb{E}\|\mathbf{v}_\theta - \mathbf{v}\|^2 = \mathbb{E}\frac{\|\mathbf{x}_\theta - \mathbf{x}\|^2}{(1-t)^2}$$

- The objective is a time-weighted clean-embedding error.
- Clean prediction aligns the denoising and decoding tasks.
- In controlled ablations, **x**-prediction is stable from 512 to 1024 dimensions; **v**-prediction degrades at higher dimensions.

## State, bottleneck, and backbone

The flow state is 512-dimensional per token. A 128-dimensional input bottleneck feeds a 768-dimensional ELF-B backbone.

Paper, Eq. (1), p. 4; Fig. 11, p. 26; Table 7, p. 30.

# Final-step decoding is a second mode of the same network

$$\mathcal{L}_{\mathrm{CE}} = \mathbb{E}\big[\mathsf{CrossEnt}(W\mathbf{x}_\theta(\tilde{\mathbf{z}}, 1, \mathrm{decode}), \mathbf{s})\big]$$

| | Denoise: 80% | Decode: 20% |
|---|---|---|
| Corruption | One $t$ per sequence | One corruption level per token |
| Time condition | Sampled $t < 1$ | Fixed $t = 1$ |
| Objective | Velocity regression | Token cross-entropy |

- Corrupted decoder inputs teach recovery from imperfect embeddings.
- At inference: decode-mode forward pass, unembedding, then argmax.

Paper, Eq. (2), p. 4; App. C.1, pp. 21–23; Alg. 5, p. 24.

# Self-conditioning supplies guidance without an external prompt

- Feed the previous clean prediction back alongside the noisy state.
- Training: 50% predicted condition, 50% zero; stop its gradient.
- Inference: reuse the previous step's prediction.

$$\mathbf{v}_{\mathrm{cfg}}(\mathbf{z}_t \mid \mathbf{c}) = \omega \mathbf{v}(\mathbf{z}_t \mid \mathbf{c}) + (1-\omega)\mathbf{v}(\mathbf{z}_t \mid \varnothing)$$

## ELF learns self-conditioning CFG during training

The network takes the desired scale as input and predicts the guided field directly: one denoising forward pass per inference step.

Paper, Sec. 3.3, pp. 5–6; Eq. (5), p. 23. External-input CFG is different: it uses two predictions.

# Sampling updates the entire sequence before decoding

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \Delta t \frac{\hat{\mathbf{x}} - \mathbf{z}_t}{1 - t}$$

1. Start from Gaussian embeddings and a zero self-condition.
2. Follow a logit-normal time grid, with more resolution near noise.
3. Use deterministic ODE updates or stochastic noise re-injection.
4. At $t = 1$, switch modes and recover all tokens.

*N* denoising steps still require a final decoding forward pass.

Paper, Algs. 2, 5–6, pp. 5, 24. The "SDE" sampler is an SDE-inspired approximation.

# Evaluation measures fluency proxies and diversity together

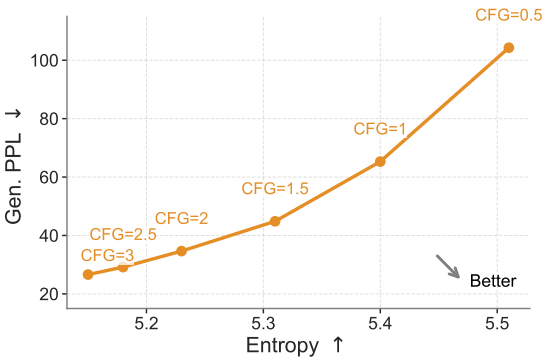
| Task | Data / length | Metric |
|---|---|---|
| Unconditional | OWT; 1024 tokens; 1,000 generated samples | GPT-2 Large PPL ↓; unigram entropy |
| Translation | WMT14 De-En; 64 source + 64 target | BLEU ↑ |
| Summarization | XSum; 1024 source + 64 target | ROUGE-1/2/L ↑ |

## What Gen. PPL means

Another LM scores generated text. It is not ELF's held-out likelihood. Repetitive text can score well, so entropy must remain visible.

Paper, Sec. 4, PDF p. 6; degeneration example in Table 5, p. 21.

# Guidance trades diversity for lower generative perplexity

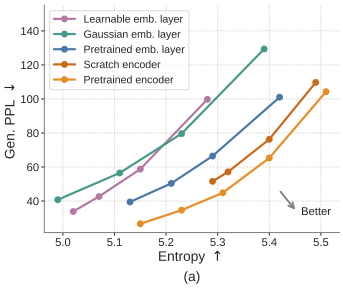


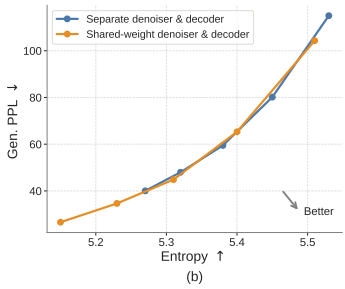

- Horizontal: entropy; vertical: Gen. PPL.
- Stronger guidance moves down and left.
- Compare frontiers, not isolated PPL minima.

64-step ODE, ELF-B:
$\omega = 1$: 65.30/5.40
$\omega = 3$: 26.62/5.15
Values: PPL / entropy.

Paper, Fig. 4, p. 7; exact values from Table 10, p. 32. Original figure.

# Contextual embeddings help; shared decoding stays competitive



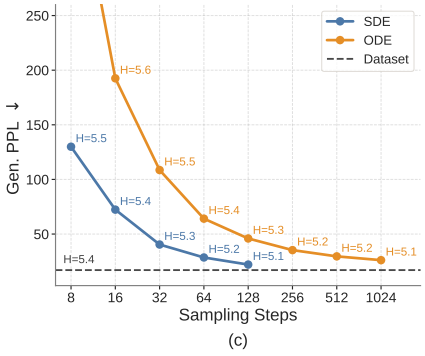




Frozen pretrained contextual representations give the best frontier among tested embeddings.

Sharing weights retains a similar frontier while removing a separate decoder stage.

Paper, Fig. 5a–b, PDF p. 7; Sec. 4.1, pp. 7–8. Complete original panels, cropped.

# Stochastic sampling is especially useful with few steps

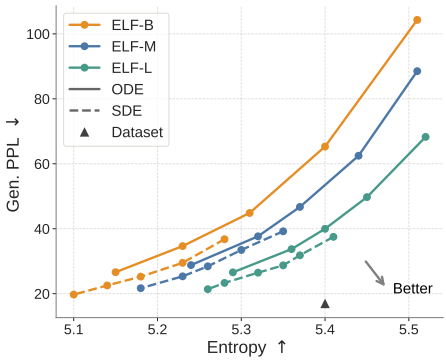



Paper, Fig. 5c, p. 7; Alg. 6, p. 24. Original panel, cropped.

- Same SC CFG scale: 1.
- Re-inject noise, move slightly toward the noisy regime, then denoise.
- Lower PPL at small step budgets; entropy also changes.

Error correction is the authors' proposed explanation.

# Larger models improve the quality–diversity frontier




ELF-B / M / L:
105M / 342M / 652M parameters.

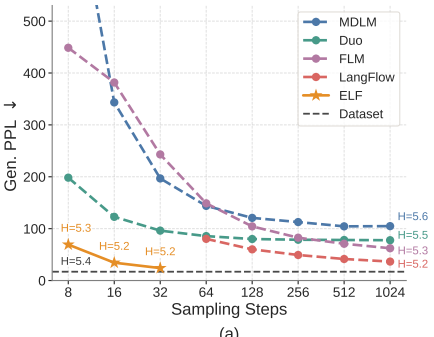
**A matched-entropy comparison:**
64-step SDE, entropy 5.28:

- B, CFG 0.5: PPL 36.77.
- L, CFG 3.0: PPL 23.31.

Training: 5 / 4 / 3 OWT epochs.

Paper, Fig. 6, p. 8; Tables 6 and 10, pp. 30, 32. Original figure.

# Undistilled ELF reaches Gen. PPL 24 with 32 steps



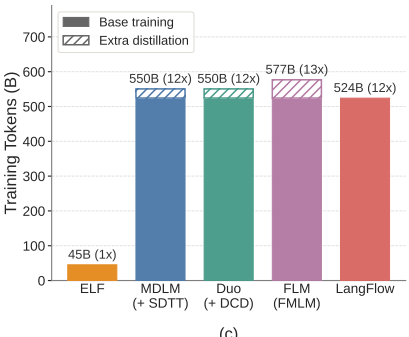

| Steps | PPL | Entropy |
|---|---|---|
| 8 | 67.32 | 5.14 |
| 16 | 33.66 | 5.16 |
| 32 | 24.08 | 5.15 |

32-step SE: $\pm 0.16$ PPL, $\pm 0.002$ entropy over six evaluation seeds.

ELF: 105M; baselines: about 170M. SC CFG=3; tuned SDE noise scale.

Paper, Fig. 7a, p. 8; means from Table 9, p. 31. Figure labels and table use different reporting detail.

# The training-token advantage has a specific accounting scope




- ELF: $5 \times 9.04\text{B} = \mathbf{45.2B}$ effective tokens.
- Undistilled baselines: **524.3B**, about $11.6\times$ as many.
- Counts exclude the T5 encoder's prior pretraining.
- Tokens do not account for every extra forward pass or teacher rollout.

Paper, Fig. 7c, p. 8; Table 8, p. 31. Cost interpretation based on the stated counting formula.

# ELF leads the reported translation and summarization systems

| Model | De-En BLEU | XSum R1 | R2 | R-L |
|---|---|---|---|---|
| AR (99M) | 25.2 | 30.5 | 10.2 | 24.4 |
| MDLM (99M) | 18.4 | 33.4 | 11.6 | 25.8 |
| Duo (170M+35M) | 21.3 | 31.4 | 10.1 | 25.0 |
| E2D2 (99M) | 24.8 | 28.4 | 8.3 | 22.0 |
| **ELF (105M+35M)** | **26.4** | **36.0** | **12.2** | **27.8** |

- ELF: 64-step ODE, SC CFG=1, input-condition CFG=2.
- Input guidance uses conditional and unconditional predictions.
- Mixed literature / reproduced baselines; tokenizers and schedules differ.

Selected rows of Table 1, p. 9; protocol: Table 11, p. 32. All scores higher is better; full table in notes.

# Distillation makes one student step match a teacher interval

Teacher: multiple steps from $\mathbf{z}_t$ to $\mathbf{z}_r$. Student: one Euler update.

$$\mathbf{z}_r = \mathbf{z}_t + (r-t)\frac{\tilde{\mathbf{x}} - \mathbf{z}_t}{1-t} \quad \Longrightarrow \quad \tilde{\mathbf{x}} = \mathbf{z}_t + \frac{1-t}{r-t}(\mathbf{z}_r - \mathbf{z}_t)$$

$$\mathcal{L}_{\text{distill}} = \mathbb{E}\|\mathbf{x}_\theta(\mathbf{z}_t, t) - \tilde{\mathbf{x}}\|^2$$

- Fixed 64-step teacher throughout; retain the decoding CE objective.
- Student curriculum: $16 \to 8 \to 4 \to 2 \to 1$ steps, one epoch per round.
- Each new student starts from the preceding student's weights.

Paper, App. B, Eqs. (3)–(4), p. 19; Table 3, p. 20. First student initialized from the teacher.

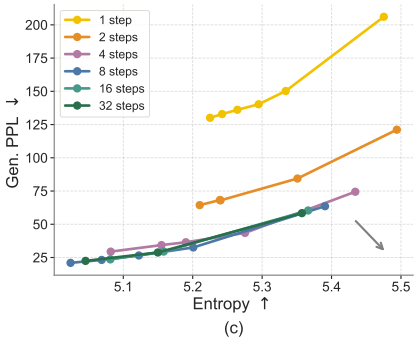
# ELF+PD improves few-step PPL, with a diversity trade-off

| | FMLM | | ELF+PD | |
|---|---|---|---|---|
| Steps | PPL ↓ | Entropy | PPL ↓ | Entropy |
| 1 | 168.30 | 5.17 | **136.10** | 5.26 |
| 2 | 133.29 | 5.25 | **68.25** | 5.24 |
| 4 | 111.31 | 5.26 | **34.33** | 5.16 |
| 8 | 86.50 | 5.36 | **23.18** | 5.07 |
| 16 | 63.63 | 5.29 | **22.12** | 5.06 |
| 32 | 45.09 | 5.25 | **21.32** | 5.04 |

All rows use the final-round student. Task training rises to about 90B tokens.
At 8 steps, lower PPL also comes with lower entropy.

Paper, Table 4 and Fig. 9, p. 20. Additional distilled baselines in backup.

# One-step generation remains substantially harder




- CFG-swept frontiers are close for 8–32 steps.
- Fewer than 8 steps lose quality.

**A revealing failure case**

Round 2, one step:
PPL $= 1.7$; entropy $= 0.64$.

An excellent PPL can reflect collapse.

Paper, Fig. 9c, p. 20 (only entropy above 5 shown); Table 5, p. 21. Original panel.

# The paper's own examples reveal semantic errors

| Example | Intended content | Generated error |
|---|---|---|
| Translation | Representatives in 26 US states | Turns states into countries |
| XSum | Calf-muscle injury; return expected in August | Changes the injury to a bone problem |
| 8-step PD | A passage about education | Repeats loosely connected claims despite PPL 19.22 |

**Reading assessment**

Reported metric gains coexist with factual and semantic failures. PPL, entropy, BLEU, and ROUGE do not fully measure fidelity.

Paper, App. F.3, pp. 34–37; F.4, pp. 38–39. Examples paraphrased; assessment is ours.

# ELF makes a strong case for continuous language generation

1. **Interface:** continuous denoising and final token recovery can share one network.
2. **Mechanism:** clean-data prediction, self-conditioning CFG, and stochastic sampling work well together.
3. **Evidence:** strong quality–diversity–step trade-offs on OWT; gains on translation and summarization.
4. **Next test:** include encoder/teacher compute, measure end-to-end latency, and evaluate semantic fidelity at few steps.

Synthesis of Secs. 3–4 and Apps. B–F. Evaluated model scale: 105M–652M; conditional targets: 64 tokens.

# Primary source and discussion guide

**Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He.**

*ELF: Embedded Language Flows*. MIT, arXiv:2605.10938v2.
Revised June 26, 2026.

arxiv.org/abs/2605.10938v2
github.com/lillian039/ELF

- Main discussion: approximately 25 minutes.
- Following slides: equations, implementation settings, and comparison details.

Original figures are reproduced from the authors' arXiv v2 source (CC BY 4.0). Tables are selected/reformatted.

# Backup: training-time guidance changes the regression target

$$\mathbf{v}_{\text{target}} = \mathbf{x} - \boldsymbol{\epsilon} + \left(1 - \frac{1}{\omega}\right)(\mathbf{v}_c^{\text{cfg}} - \mathbf{v}_0^{\text{cfg}}),$$
$$\mathcal{L} = \|\mathbf{v}_\theta^{\text{cfg}} - \text{stopgrad}(\mathbf{v}_{\text{target}})\|^2.$$

- $c$: self-conditioning prediction; 0: null self-conditioning.
- Both fields come from clean-embedding predictions via $(\mathbf{x}_\theta - \mathbf{z}_t)/(1-t)$.
- Without self-conditioning, regress to $\mathbf{x} - \boldsymbol{\epsilon}$ instead.
- Sample $\omega \in [0.5, 5]$; supply it as a control input.
- At $\omega = 1$, the target correction disappears.

Paper, Eq. (5), p. 23; Alg. 3, p. 22. Training uses extra predictions; inference uses one guided prediction.

# Backup: model and corruption settings

## ELF-B

12 layers; hidden size 768; 12 heads.
512-dimensional T5-small embeddings.
128-dimensional input bottleneck.
4 time + 4 CFG + 4 mode tokens.

## OWT optimization

Muon, LR 0.002, batch 512.
5 epochs; EMA 0.9999.
No weight decay; constant LR after warmup.

Paper, App. C.1, pp. 21–23; Tables 6–7, p. 30.

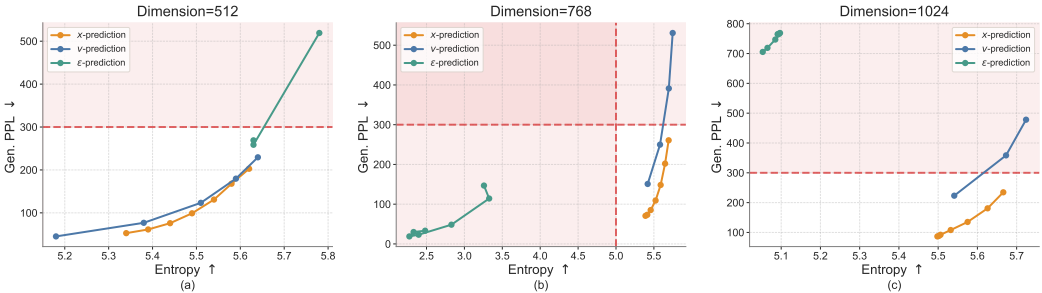
## Denoising branch

$t = \sigma(u)$, $u \sim \mathcal{N}(-1.5, 0.8^2)$.
One $t$ per sequence. Noise scale 2.

## Decoding branch

$p_i = \sigma(u_i)$, $u_i \sim \mathcal{N}(0.8, 0.8^2)$.
One $p_i$ per token; time input stays 1.
Noise scale 5 on OWT, 1 for conditional tasks.

# Backup: clean prediction is robust at higher dimensions




- Controlled setup: frozen encoder/decoder, adaLN-Zero, 64-step ODE.
- Red regions: entropy below 5 or PPL above 300.
- **v**-prediction is competitive at 512 dimensions, weaker at 768/1024.

Paper, App. D.1, pp. 25–26; Fig. 11, p. 26. Original figure.

# Backup: several implementation choices matter

| Choice | Observation in the paper |
|---|---|
| Bottleneck: 32/128/512 | 32 often sacrifices entropy; 512 has worse PPL; 128 balances both. |
| Denoising probability | 0.8 gives the best tested frontier. |
| Conditioning | In-context tokens slightly improve quality; 148M $\rightarrow$ 105M parameters. |
| Optimizer | Tuned Muon improves over tuned AdamW, especially with SDE sampling. |
| Time grid / noise | Logit-normal helps few-step sampling; noise scale changes the frontier. |

Paper, App. D.2–D.6, Figs. 12–16, pp. 26–29.

# Backup: the stochastic update uses the original state

$$\alpha = 1 - \gamma \Delta t, \qquad t_{\text{back}} = \alpha t, \qquad \mathbf{z}_{\text{back}} = \alpha \mathbf{z}_t + (1-\alpha)\boldsymbol{\epsilon}'$$

$$\hat{\mathbf{x}} = \text{net}_\theta(\mathbf{z}_{\text{back}}, t_{\text{back}}, \text{denoise})$$

$$\mathbf{v} = \frac{\hat{\mathbf{x}} - \mathbf{z}_t}{1 - t}, \qquad \mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \Delta t\, \mathbf{v}$$

- Perturb the evaluation point; update from the original $\mathbf{z}_t, t$.
- $\gamma = 0$ recovers the ODE update.
- This is the paper's practical SDE-inspired rule, not an exact SDE solver.

Paper, Alg. 6, p. 24; App. C.2, pp. 24–25. Self-conditioning omitted here as in Alg. 6.

# Backup: distilled systems across all reported step budgets

| Steps | MDLM+SDTT | Duo+DCD | FMLM | ELF+PD |
|---|---|---|---|---|
| 1 | 1260.60 / 5.26 | 5743.90 / 6.02 | 168.30 / 5.17 | 136.10 / 5.26 |
| 2 | 877.22 / 5.34 | 891.16 / 5.41 | 133.29 / 5.25 | 68.25 / 5.24 |
| 4 | 339.73 / 5.38 | 250.86 / 5.37 | 111.31 / 5.26 | 34.33 / 5.16 |
| 8 | 112.66 / 5.41 | 118.21 / 5.41 | 86.50 / 5.36 | 23.18 / 5.07 |
| 16 | 57.74 / 5.39 | 78.74 / 5.43 | 63.63 / 5.29 | 22.12 / 5.06 |
| 32 | 40.41 / 5.34 | 63.98 / 5.40 | 45.09 / 5.25 | 21.32 / 5.04 |

Each cell: Gen. PPL / entropy.

ELF+PD settings:
1–8 steps: $\gamma = 1.5$, SC CFG=2.5.
16–32 steps: $\gamma = 2.0$, SC CFG=2.0.

Paper, Table 4, p. 20; App. B.2, p. 19. Values from final-round student.

# Backup: conditional protocols qualify the comparison

- ELF keeps source embeddings clean and prepends them to targets; drop the source condition with 10% probability during training.
- Table 1 includes a 35M encoder for ELF and Duo.
- WMT14 mostly uses literature results; XSum mostly uses reproduced baselines.
- Reproduced MDLM uses two 32-token blocks; E2D2 uses 8-token blocks; Duo uses 1,000 sampling steps.
- Fig. 17's XSum axis says ROUGE-1 near 27.8, whereas Table 1 reports R1=36.0 and R-L=27.8. Use Table 1 for numeric metric identity.

Paper, Table 1, p. 9; App. C.1, p. 23; Fig. 17, p. 28; Table 11, p. 32. Last point is a reading observation.