GRAFT: RESTORING THE MISSING SAMPLING INTERFACE IN CONTINUOUS DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous diffusion language models generate a whole sequence in parallel, never drawing a single word at any intermediate step; only the final step turns embeddings into text. Models such as ELF now report autoregressive-level quality at a fraction of the sampling steps taken by discrete-token diffusion models, and this design principle—no step-wise discretization—is what their own ablations credit for the gain. We show that this same principle carries a hidden cost: a collapse of the lexical long tail. Measured against held-out human text, ELF's output matches the human rate on common words almost exactly, yet its rarest words appear at only $0.30\times$ the human rate, a gap that does not close from 105M to 652M parameters and that no sampler setting these models expose can reach. A same-size GPT-2, in contrast, spans zero to above the human rate by its decoding rule alone. We trace the difference to what continuous diffusion models lack: with no step that ever draws a discrete word, every step is instead driven by an average over the candidates still in play, and averaging a rare candidate together with the common word competing for its slot lands on the common word. An autoregressive model avoids this because its decoding rule samples one candidate rather than averaging over all of them, so the problem never arises there. We propose **GRAFT**, a training-free method that grafts this missing choice back onto a frozen model: while the model is still undecided, GRAFT commits the rare words the average would otherwise discard, then regenerates the sequence from scratch with those commitments held in place. Across three ELF scales and on LangFlow, an independently developed continuous dLLM, GRAFT restores the rare-word share of generated text from 0.42–$0.77\times$ the human rate to 0.82–$1.10\times$, a relative gain of 34–109%, and moves the entire word-frequency profile closer to human on every model tested, by $2.7\times$ on ELF-B and $12\times$ on LangFlow.

1 INTRODUCTION

Diffusion language models (dLLMs) generate all positions in parallel and refine them over a fixed number of steps, trading autoregression's $O(L)$ serial depth for $O(\text{steps})$ (Austin et al., 2021; Sahoo et al., 2024; Nie et al., 2025). dLLMs fall into two lines, distinguished by whether the diffusion process runs over discrete tokens (Lou et al., 2024; Nie et al., 2025; Bie et al., 2025) or over continuous embeddings (Li et al., 2022; Dieleman et al., 2022). Continuous diffusion language models historically lagged behind their discrete counterparts. Earlier approaches typically relied on either projecting the continuous state back to the token space at every denoising step (Li et al., 2022; Dieleman et al., 2022), or performing diffusion in the latent space of a separately trained autoencoder (Lovelace et al., 2023). ELF (Hu et al., 2026) removes both requirements: the entire denoising trajectory remains in the token-embedding space, and discretization is performed only once, at the final step. This design substantially improves efficiency, allowing ELF to outperform discrete diffusion baselines with only 32 sampling steps instead of 1024 and with roughly one-tenth of the training tokens. LangFlow (Chen et al., 2026) independently achieves comparable performance through a similar continuous-space formulation. Ablation studies in both works attribute much of these gains to the same key design choice: eliminating intermediate, step-wise discretization.

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.

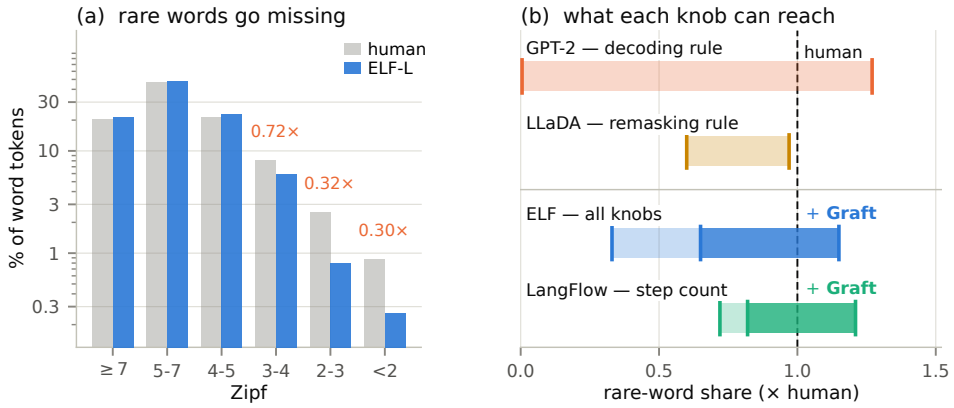

Figure 1: Lexical-tail coverage depends on whether a model samples discrete words during generation. **(a)** ELF-L matches human usage for common words but substantially underproduces rare ones, reaching only $0.72\times$, $0.32\times$, and $0.30\times$ of the human rate across the three rarest Zipf bands. **(b)** GPT-2 and LLaDA, which sample discrete words during decoding, reach or approach the human rare-word rate, while ELF and LangFlow remain below it. GRAFT restores the missing tail, raising ELF and LangFlow to $1.15\times$ and $1.21\times$ of the human rate, respectively.

However, the same design choice that makes continuous dLLMs efficient also introduces a hidden lexical cost: collapse of the vocabulary tail (Figure 1a). Here, rarity is defined by corpus frequency rather than model probability; proper names, technical terms, and other low-frequency words together account for more than a tenth of human text. On held-out text from ELF's training distribution, common words appear at nearly the human rate, while the rarest occur at only $0.30\times$ that rate. This gap persists from 105M to 652M parameters (Table 1) and is not closed by sampler tuning. When a rare word is lost, it is typically replaced by a more common word or a fabricated *non-word*, reducing lexical fidelity. For example, one ELF summary reads "New Barcelona president Neyll has resigned," producing a fluent sentence with an invented name (§4). Yet such failures are easily missed by standard metrics: GPT-2-based generative perplexity and unigram entropy. The former can favor text that avoids rare words, while the latter is too coarse to reveal a deficit concentrated in the lexical tail.

The cause lies in a missing step in the continuous dLLM pipeline. Autoregressive models first predict a vocabulary distribution and then apply a decoding rule—such as greedy, top-k, or nucleus sampling— to select a discrete word. We refer to this distribution-to-word operation as a *sampling interface*. Continuous dLLMs lack such an interface before the final step: their denoising head instead predicts continuous embeddings by regression throughout the trajectory, while the decoding head is consulted only once at the end. Because regression is driven toward the conditional mean, competing candidates are averaged rather than sampled; when a rare word competes with a more probable common word, the resulting embedding tends to move toward the common one. By the time the decoding head is finally read, each position has already narrowed to only 1.02–1.05 effective candidates, leaving little room for temperature scaling, sampling, or beam search to recover the lost tail. In short, rare-word probability survives inside the model, but the generation process never turns it into a discrete choice.

Can this missing sampling interface be restored at inference time, without retraining the model?

We answer yes with **GRAFT**, a training-free method that restores a discrete choice while rare-word alternatives are still present. Part-way through generation, GRAFT reads the decoding head and identifies rare words that retain substantial probability but are losing to more common competitors. It then restarts generation from noise with these words fixed from the first step, allowing the remaining sequence to form around them. Repeating this read-and-restart procedure accumulates a set of committed rare words, after which a final generation produces the output. Sub-word constraints and an anti-repetition correction (Zhang et al., 2026) mitigate the artifacts introduced by fixing words.

Our contributions are threefold. First, we identify a lexical long-tail collapse in continuous dLLMs, showing that the effect persists across model scales and sampler settings when measured against

a same-distribution human baseline (§4). Second, we uncover its mechanism: rare words are lost when a more common competing word accumulates sufficient probability, and the competitor's probability alone strongly predicts which rare words will be suppressed (§5). Third, we introduce **GRAFT**, a training-free method that restores the missing lexical tail. Across three ELF scales and an independent LangFlow implementation, GRAFT raises the rare-word share to 0.82–$1.10\times$ the human rate, moves the overall word-frequency distribution closer to human—by $2.7\times$ on ELF-B and $12\times$ on LangFlow—and brings ELF-L's ζ to within 0.010 of the human value (§7.2–§7.5).

2 Related work

Diffusion language models fall into two lines, distinguished by whether the diffusion process runs over discrete tokens or over continuous embeddings.

The discrete line began with D3PM, which first carried diffusion to discrete tokens, using an absorbing mask state as the corruption process (Austin et al., 2021); MDLM then reduced masked diffusion to a weighted cross-entropy that trains as easily as a language model (Sahoo et al., 2024); and LLaDA scaled the recipe to 8B parameters and autoregressive-level quality (Nie et al., 2025). Every model in this line draws a token from a categorical distribution at each unmasking step, so remasking order and temperature come as decoding knobs for free.

The continuous line diffuses in embedding space instead. Diffusion-LM opened it, but had to round the embeddings back to the vocabulary at every step to continue (Li et al., 2022); the variants that followed kept some per-step anchor to the vocabulary, and none caught up with autoregression at equal compute. ELF removed these anchors entirely, keeping the whole trajectory in embedding space and discretizing only at the last step, and reached autoregressive-level quality in 32 steps (Hu et al., 2026); LangFlow reached the same point independently, by a related route (Chen et al., 2026). The anchors this line removed, however, were also its only place to draw a word. This paper measures what that removal costs, and puts the draw back before generation begins.

We list here only the most representative work in each line; a fuller review is given in Appendix F.

3 Preliminaries

The models. We study ELF (Hu et al., 2026) and LangFlow (Chen et al., 2026), two independently developed continuous dLLMs. ELF represents a sentence as $x \in \mathbb{R}^{L\times d}$ and learns a flow from noise to data. Under *x-prediction*, given a noised state z_t at time t, the model predicts the clean embedding $\hat{x}(z_t, t)$ over $\sim$32 denoising steps. A single network has two output heads: the denoising head, trained with MSE, drives every step of the trajectory, while the decoding head, trained with token-level cross-entropy through a shared unembedding, is used only at the final step. LangFlow follows the same overall design with $\sim$256 steps.

Word-frequency metrics, measured against human text. We compare model outputs with a human reference to measure the lexical-tail deficit and the overall word-frequency distribution. *Zipf frequency* is the log-scale corpus frequency of a word (van Heuven et al., 2014), with larger values indicating more common words. The *tail* is the fraction of word tokens with Zipf <4, further split into $[3, 4)$, $[2, 3)$, and <2 to measure the depth of the deficit. A *non-word* is a string with no lexicon entry (Zhang et al., 2026), which may replace a lost rare word. *Unigram entropy* measures lexical diversity and helps show why standard diversity checks can miss a tail-specific collapse. A sample's *Zipf profile* is the distribution of lexicon words across six Zipf bands (Appendix A). We compare this profile with the human reference using *JSD*, where 0 indicates an exact match. Finally, the *Zipf coefficient* ζ is the magnitude of the slope fitted to the rank–frequency curve; larger ζ indicates stronger concentration on a small set of common words.

4 The collapse: rare words missing, no metric noticing

This section quantifies the lexical-tail collapse introduced above. Table 1 groups word tokens from $1{,}000$ unconditional samples at each ELF scale by Zipf frequency and compares them with held-out human text from the same corpus. The problem is not that the text is broadly poor, but that ELF

Table 1: Word-frequency profile of unconditional generation (1,000 samples each) against held-out human text from the training distribution. High-frequency buckets match human almost exactly; the deficit is confined to the tail, and the rarest bucket does not move from 105M to 652M parameters.

Zipf bucket (% of word tokens)	ELF-B	ELF-M	ELF-L	Human	ELF-L / Human
≥ 7	20.60	20.62	21.02	20.38	$1.03\times$
$[5, 7)$	53.07	51.59	48.13	46.20	$1.04\times$
$[4, 5)$	20.81	20.69	22.64	21.27	$1.06\times$
$[3, 4)$	3.14	4.93	5.82	8.12	$\mathbf{0.72}\times$
$[2, 3)$	0.44	0.68	0.80	2.51	$\mathbf{0.32}\times$
< 2	0.27	0.24	0.26	0.86	$\mathbf{0.30}\times$
non-word (no entry)	1.67	1.25	1.35	0.66	$\mathbf{2.05}\times$
types / 100k	2,949	3,288	3,505	5,329	$0.66\times$

overuses common words and underuses rare ones: common-word statistics closely match human text, while the deficit is concentrated in the tail. The missing mass shifts partly to mid-frequency words and partly to non-words, whose rate is about $2\times$ the human level. This raises three questions: whether the deficit shrinks with scale or tuning, whether it depends on the task, and whether standard metrics would detect it.

The deficit does not disappear with scale or sampler tuning. It remains as the model grows from 105M to 652M parameters, and none of the sampler knobs we test closes the gap. Its severity also depends on how much rare content must be generated rather than copied. Translation, which can copy rare tokens from the source, is largely unaffected. Summarization, which must generate names, fabricates proper nouns absent from both source and dictionary at 3.53%, compared with 0.82% in human references. Unconditional generation, where all content is self-produced, is the most affected. Details of this task ordering are given in Appendix A.

Two simpler explanations can be ruled out before turning to the mechanism: undertraining and insufficient sampler tuning. Undertraining cannot explain the effect, since ELF-B saw $5\times$ as many training tokens as GPT-2 small at comparable scale. Nor is the issue simply the choice of decoding setting. The same GPT-2 checkpoint moves from $0.00\times$ to $1.27\times$ the human tail by changing only its decoding rule, whereas ELF's full sweep over steps, guidance, temperature, and model scale reaches only 0.33–$0.65\times$ (Figure 1b; Appendix E.6). The key difference is therefore not training scale, but whether the model exposes a word-level sampling interface.

Standard metrics also fail to reveal the problem. GPT-2-based generative perplexity can favor text that avoids rare words because GPT-2 assigns them lower probability. On ELF-B, perplexity *improves* from 44.5 to 19.0 as the number of steps increases from 16 to 64, even while 4-gram repetition rises $12\times$ (Appendix E.6). Unigram entropy is similarly insensitive: it appears normal even though ELF's Zipf slope is -1.227, compared with -0.986 for human text, because the statistic is dominated by common words (§3). Thus, a deficit concentrated in the lexical tail can remain hidden under standard evaluations, extending to the lexical axis what Zheng et al. (2025) and Zhang et al. (2026) observed for low-diversity text.

5 Root cause: the average inside the denoising head

The lexical tail disappears from the output, yet no sampler setting recovers it (§4). We trace this failure to the denoising head itself: because it is trained by regression, it predicts an average over competing word candidates, and that average is pulled toward a more probable common word. Three findings support this mechanism; Appendix B rules out eight alternative explanations.

Finding 1: The rare words are already gone before the decoding head reads them. The decoding head maps embeddings to words, so it is the natural first suspect. If it were biased toward common words, it could itself produce the observed tail collapse. We test this using embeddings that still contain genuine human text: 128 held-out OpenWebText documents are encoded with ELF's frozen T5 encoder and fed into ELF-B without added noise, using the same readout as at the end of generation. Across $124{,}212$ positions, the decoding head recovers 100.00% of the original words (Table 4, Appendix B). Thus, whenever a rare word is still present in the embedding, the decoding head can recover it. The tail must therefore already be absent from the embeddings that reach the final readout.

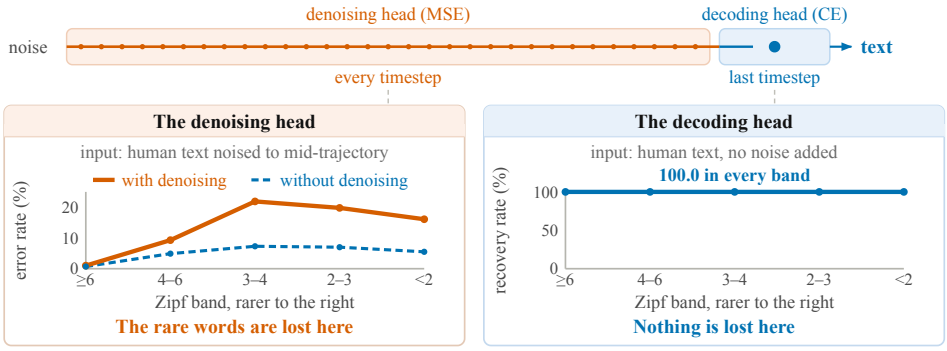

Figure 2: Where rare words are lost. ELF uses a denoising head (MSE) at every timestep and a decoding head (CE) only at the final readout. **Right:** on clean held-out human embeddings, the decoding head recovers all words across 124,212 positions (Finding 1). **Left:** on noised text at an intermediate generation level, errors increase sharply only after denoising and primarily on rare words: on ELF-B, the gap grows from 1.0% vs. 0.7% in the commonest band to 21.9% vs. 7.3% in $[3, 4)$ (Finding 2).

Finding 2: The rare words are lost inside the denoising head, and specifically because they are rare. With the final decoder ruled out, we next examine the denoising head, which drives every generation step. We noise 96 held-out OpenWebText documents to a level encountered part-way through generation and feed the same input to both heads. Because the heads share the same underlying network and receive the same information, their training objectives are the main difference (Appendix B). Across roughly 75,000 positions and all three ELF scales, their error rates are similar on common words, but on the rarest words the denoising head is wrong 2–2.9× as often (Table 7, Appendix E.4). This is not a general weakness: on high-frequency words, the denoising head is at least as accurate. Its excess error is concentrated in the lexical tail, showing that rare words are lost inside the denoising process.

Finding 3: The rare words are lost when a more common competitor already carries enough probability. If denoising averages over candidate words, then the probability of the strongest competitor should predict whether a rare word survives. We test this on 200 held-out OpenWebText documents, noised as above. For each position, decoding the noised embedding directly gives q, the probability assigned to the strongest competing word; applying the denoising head first and then decoding reveals which word survives. Among 4,036 rare positions on ELF-B, only 7% lose their word in the lowest-q decile, compared with 92% in the highest-q decile (Appendix E.7). Moreover, when a rare word is lost, that same competitor replaces it 31.8% of the time. The strongest competitor's probability therefore predicts both whether a rare word is lost and, often, which word replaces it.

6 GRAFT: COMMIT THE RARE WORDS THE AVERAGE WOULD LOSE

GRAFT follows from §5: the model loses a rare word by averaging it away during generation, so we find the rare words it is about to lose and hand them back to it as given before it generates again. A word is handed over through the model's own conditioning input, and we call handing one over in this way *committing* it. ELF accepts known words at arbitrary positions: it starts the generation with each given word already in place, holds it there through every denoising step, and forms the rest of the sequence around it, as it does around a prompt. LangFlow has no such input, and we emulate one by writing the given word's embedding into its position at every step (Appendix A). The procedure has four steps, of which the first three repeat.

Step 1: generate, and stop part-way. A round starts a generation from fresh noise and runs it only as far as a point we call the *read time*; the rest of that trajectory is never run. The read time is one denoising step inside the short window where positions still hold several candidates: earlier, the sequence has too little shape for those candidates to mean anything; later, each position has already narrowed to the 1.02–1.05 candidates of §1 (where this window sits in each model: Appendix E.3).

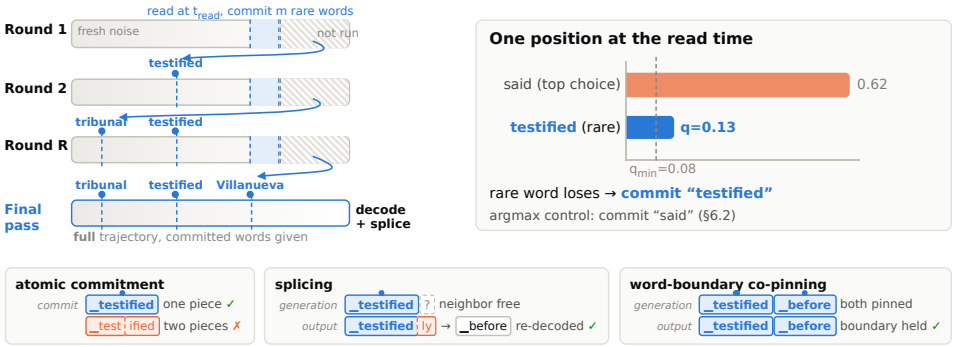

Figure 3: GRAFT commits rare words from the first denoising step and uses restarts to preserve them. **Left:** each of the R rounds starts from fresh noise with all previously committed words fixed, runs only to the read time, and commits m additional rare words; a final full trajectory then generates the output with all commitments held in place. **Top right:** at the read time, GRAFT commits a rare word that retains substantial probability but loses to the top choice, while the argmax control commits that top choice instead. **Bottom:** three rules prevent committed words from creating non-words.

Step 2: read the decoding head, once. At that single point, the decoding head—the only part of the model that names words at all (§3)—ranks the whole vocabulary at every position, and GRAFT reads out this full ranking. This is the only time the decoding head is consulted before the end of a generation.

Step 3: select the rare words that are losing, and commit them. A position qualifies when its strongest rare candidate—among the 512 most likely words there—carries probability above $q_{\min}$=0.08 but is not the top choice; a rare word that is already winning needs no rescue. A qualifying position must also pass the rules of below. Among all qualifying positions, GRAFT ranks them by that probability and commits the m highest, each to its own strongest rare candidate.

Step 4: restart, and finish on the last round. The generation of Step 1 is then abandoned rather than finished. The next round returns to fresh noise, with the new commitments added to the given words, and Steps 1 through 3 run again; after R rounds, a final generation—with all committed words now given—runs to completion and produces the text (Figure 3; Algorithm 1 in Appendix A). We use R=4 and m=8, so a sample carries up to 32 committed words in total; that total is what we call the *dose*. This costs R partial generations plus one full one—between two and three full trajectories on ELF, depending on where its sampler places the denoising steps—and, with overhead, 2.2× the wall clock of the model's own stock sampler (Appendix E.1).

GRAFT's intervention is not entirely free of side effects, and each requires a targeted remedy.

Committing words can disrupt nearby text. Although GRAFT commits whole words, the model operates on sub-word pieces, and a word may span multiple positions. A commitment fixes only the position it occupies, while neighboring positions are still generated freely. If the following piece was originally meant to complete the displaced word, the two may merge into a non-word. We prevent this with three rules. First, *atomic commitment* restricts commitments to single-piece words placed only at positions that already contain a single-piece word, avoiding partial overwrites. Second, *splicing* re-decodes the position immediately after a commitment at output and requires it to begin a new word, preventing leftover fragments from attaching to the committed word. Third, *word-boundary co-pinning* fixes that following position to a word-beginning piece from the model's own draft throughout generation, so the model knows from the first step that a new word begins there.

The lexicon used to define "rare" introduces a second issue. It may contain strings that are not valid English words, as well as words that are rare in English but common in another language, either of which could artificially inflate the measured tail. We therefore apply two gates. A candidate may be committed only if it is *attested* in a held-out slice of human text, and it is rejected if it is more than 2.5 Zipf points more common in any of thirteen other languages than in English (Appendices E.8 and E.12). These gates affect only which words GRAFT may commit, not how the metric counts them, and both are enabled in the main results of Table 2.

A third side effect is increased repetition. Unlike the local sub-word artifacts above, this effect is distributed across the generated text. It matches the self-conditioning feedback identified by Zhang et al. (2026), where repetition accumulates along a dominant direction in the state fed back to the model. We apply their correction at λ=0.5, counteracting this direction at each denoising step. The same direction appears in both models, while larger doses sharply degrade generation quality (Appendix D).

These remedies leave one design question: why must GRAFT restart from noise rather than intervene on an ongoing trajectory? We tested two cheaper alternatives, and both fail. The first discretizes the sequence at a single intermediate denoising step and continues generation from the resulting words. Early in the trajectory, positions still support several candidates and are not yet mutually coherent, so independently selected sub-word pieces do not compose reliably; later, when discretization becomes safe, the text has already largely settled and the intervention has little effect. The second alternative applies a softer intervention over several steps, repeatedly moving selected positions toward the rare word's embedding. Small moves are undone by subsequent denoising and the sampler returns to its original output. Strong moves can preserve the target word, but sharply increase non-words and local corruption. In our experiments, only a word supplied before the first denoising step persists reliably, and only in this regime does the identity of the committed word matter. This motivates GRAFT's restart-and-commit design; the corresponding measurements are reported in Appendix D.

7 EXPERIMENTS

We test whether committing the rare words that would otherwise be lost restores the lexical tail. We first describe the models, metrics, and dose setting, then report the main results, metric-validity checks, alternative explanations, and ablations.

We evaluate ELF-B/M/L (105M/342M/652M official checkpoints, 32 sampling steps) and LangFlow (652M, 256 steps), all frozen, with unconditional generation and n=200 samples per arm unless noted. Both model generation and the human reference come from OpenWebText, ELF's training corpus, at the same 1024-token sequence length. Thus, all statistics are measured against a same-distribution human baseline. We do not report generative perplexity because, as shown in §4, it rewards the lexical deficit we measure.

Table 2 reports six metrics. *Tail %* is the share of rare word tokens, and $\times$*human* expresses it relative to the human rate of 11.88%. *Non-word %* controls for fabricated strings: a tail gain is meaningful only if it is not obtained by increasing non-words, whose human rate is 1.14% (Appendix A).

To measure the full word-frequency distribution, we report *JSD* and $|\Delta\zeta|$. JSD compares the complete Zipf profile with the human reference, while $|\Delta\zeta|$ measures the difference in Zipf concentration from the human value ζ=0.971; lower is better for both. Finally, *Types* counts distinct generated words and provides a general measure of vocabulary breadth.

7.1 SETTING THE DOSE

The dose—the number of words GRAFT commits—controls a trade-off: too few commitments leave the tail short, while too many overshoot the human tail and distort the overall frequency profile (Figure 4, Appendix E.11). We use up to m=8 qualifying positions per round for R=4 rounds at all three ELF scales. This single untuned setting places ELF-B just below the human tail, ELF-M near it, and ELF-L slightly above it.

The same dose overshoots on LangFlow because its stock sampler already reaches 0.77$\times$ the human tail, compared with 0.42$\times$ for ELF-B. We therefore select a LangFlow-specific dose using JSD on one seed and evaluate the selected setting on two held-out seeds. The chosen dose commits none from $[3, 4)$ and at most one each from $[2, 3)$ and below per round, bringing LangFlow to 1.03$\times$ the human tail and the closest frequency profile in Table 2. We use this held-out selection only where the fixed dose overshoots; all ELF results remain untuned (Appendix E.10).

Table 2: Main results on *unfiltered* output (n=200 per arm). GRAFT rows and the stock rows of ELF-L and LangFlow average three seeds; the remaining rows are single arms. The LangFlow *tuned* row uses the held-out dose of §7.1; no ELF arm is tuned. *Best-of-4* is the equal-budget control of §7.4. Bold marks the value closest to human within each model.

	lexical tail, and its control			distance to human profile		
	tail %	×human	non-word %	JSD ↓	$\lvert\Delta\zeta\rvert$ ↓	types
HUMAN REFERENCE	11.88	1.00	1.14	—	—	17 726
ELF-B (105M)						
Stock	5.01	0.42	2.64	.0134	.182	11 795
Stock, best-of-4	5.97	0.50	1.79	.0101	.175	11 482
Argmax control	5.31	0.45	2.42	.0133	.209	12 124
GRAFT	**9.76**	**0.82**	2.57	**.0049**	**.130**	**13 829**
ELF-M (342M)						
Stock	5.75	0.48	1.97	.0090	.138	14 111
GRAFT	**12.02**	**1.01**	3.11	**.0079**	**.066**	**15 695**
ELF-L (652M)						
Stock	7.65	0.64	2.67	.0052	.058	15 478
GRAFT	**13.05**	**1.10**	2.88	**.0046**	**.010**	**17 106**
LangFlow (652M, independent implementation)						
Stock	9.16	0.77	3.28	.0025	**.011**	18 896
Argmax control	9.75	0.82	3.47	.0059	.104	**18 572**
GRAFT	14.81	1.25	3.58	.0058	.063	18 981
GRAFT (tuned)	**12.23**	**1.03**	3.79	**.0002**	.045	18 940

7.2 RESULTS

GRAFT restores the lexical tail without simply creating fabricated words (Table 2, Figure 1b). Across the four models, the rare-word share rises from 0.42–$0.77\times$ the human rate to 0.82–$1.10\times$, gains of 3.1–6.3 percentage points. Meanwhile, non-word rates rise by at most 1.1 points and even fall on ELF-B ($2.64 \rightarrow 2.57$), indicating that the added vocabulary is largely recovered rather than fabricated. The broader frequency profile improves as well. JSD to the human profile decreases on all four models, by $2.7\times$ on ELF-B and $12\times$ on LangFlow, while ELF-L's ζ comes within 0.010 of the human value. LangFlow is the one exception on ζ: its stock sampler is already within .011 of human, and no GRAFT dose improves on it, so we make the ζ claim only for ELF (Appendix E.11).

7.3 METRIC VALIDITY

For tail % to be a valid measure of lexical reach, rather than an easily gamed number, two premises need to hold, and a further check confirms the resulting gains are not an artifact of sample quality.

The first premise is that a rare word counted in the tail is used *appropriately*, not merely inserted: GPT-2-large, used nowhere in the sampler, rates the committed words no less apt in context than rare words occurring naturally in human text (Appendix E.12). The second is that a word is only counted as rare if it is genuinely rare in English, not merely rare because it belongs to another language: every tail figure is discounted for words that are rare in English but common elsewhere (Appendix E.8).

As a further check on validity, Appendix E.9 re-reads every arm after discarding samples whose non-word or 4-gram repetition rate exceeds the human sample's own 95th percentile, following Zheng et al. (2025) and Zhang et al. (2026). The gains reported below survive this filter, confirming they are not driven by a handful of degenerate samples. We note, however, that no system, ours included, closes the gap to the human non-word rate (2.0–3.8% against 1.14%).

7.4 ALTERNATIVE EXPLANATIONS

The title names what this section rules out: two alternative explanations, besides the identity of the committed word, that could account for the gains reported above.

The first is that the gain comes from the act of committing rather than from the word committed: restarting, holding words fixed, and the sub-word repairs might each lift the tail on their own, regardless of which word is chosen. To separate the two, on ELF-B and LangFlow we run an argmax

Table 3: Ablation of the three commitment rules on ELF-B. Each row adds one rule to the previous setting; *Fidelity* is the fraction of committed words preserved in the final output. The last row also changes the sampler's step distribution, so its 0.5-point tail reduction reflects both effects. The full GRAFT setting further includes the two gates and the anti-repetition correction.

	tail %	non-word %	fidelity %
stock	4.54	2.35	—
+ rare commitment, no rule	7.92	4.14	86.7
+ atomic commitment	8.86	2.71	87.4
+ splicing	9.61	2.47	100.0
+ word-boundary co-pinning	9.10	1.78	100.0

control that shares the GRAFT arm's positions, mechanics, dose, and compute, but commits the model's top choice at each position instead of the rare word. This control is null on every column of Table 2: on ELF-B the tail moves only from 5.01 to 5.31, and ζ moves in the wrong direction (.209 against .182); on both models, the bootstrap interval of its tail gain includes zero (Appendix E.9). The gain therefore rests on which word is committed, not on the act of committing.

The second alternative is compute: because the method resembles best-of-K selection, the gain could simply reflect spending more compute per sample. The four restarts of a GRAFT sample run only to the read time, and its measured wall clock is 2.2× the stock sampler's, so we compare against the equal-budget baseline: the best of four stock samples. At that budget, GRAFT still leads by a wide margin on both models: on ELF-B, best-of-4 reaches 0.50× the human tail against the tuned GRAFT's 1.00×; on LangFlow, 0.86× against 1.03× (Table 2). Nor does this gap close with still more compute: the stock sampler catches up on LangFlow only at best-of-16, and on ELF-B not at all (Appendix E.4).

7.5 ABLATION STUDY

Table 3 adds the three commitment rules of §6 to the stock ELF-B sampler one at a time, to separate two questions the rules could in principle answer: where the tail gain comes from, and what the rules themselves are for. The first row answers the first question. Committing rare words with no rule at all already lifts the tail from 4.54 to 7.92—three quarters of the eventual gain—so the tail comes from the act of commitment itself, not from any of the three rules. This bare commitment is not free, however: the non-word rate rises from 2.35 to 4.14, and 13% of the committed words reach the output as some other string rather than the word intended. The remaining rows answer the second question: the three rules exist to repair that fabrication, not to grow the tail further. Atomic commitment and splicing, added together, bring every committed word through to the output intact, taking fidelity to 100%. Word-boundary co-pinning then drives the non-word rate down further, to 1.78—below the stock sampler's own 2.35—at a small cost of 0.5 points of tail.

In sum, the tail comes from committing the rare word; the three rules are not where it comes from, but they are what bring fabrication down, in the end below the stock sampler's own rate. The two gates and the anti-repetition correction are ablated separately in Appendices E.1 and E.12.

8 LIMITATIONS AND CONCLUSION

We show that continuous diffusion language models suffer from a lexical long-tail collapse caused by the absence of an intermediate sampling interface: rare words remain represented while positions are uncertain, but are progressively washed out by conditional-mean denoising before the final decoding step. GRAFT restores this missing interface at inference time by identifying losing rare-word candidates, committing them, and regenerating around those commitments. Without any training, it raises the rare-word share from 0.42–0.77× to 0.82–1.10× the human rate across three ELF scales and an independent implementation, at 2.2× the wall-clock cost. Several directions remain open, including training-time mechanisms that preserve rare-word mass without restarts, richer use of the decoding distribution, extension to broader conditional tasks and data distributions, and a formal analysis of conditional-mean readout in embedding space together with human evaluation.

REPRODUCIBILITY STATEMENT

All models are public checkpoints (ELF-B/M/L, LangFlow OWT); the human baseline is held-out OpenWebText. The protocol (bars, bootstrap, clean-PPL) is specified in §7 and Appendix A, which also gives GRAFT's exact procedure as Algorithm 1; defaults are in §6; probe code, per-arm generation outputs, per-commitment paired logs (position, committed token, counterfactual token, probabilities, entropy, final token), the human reference sample and the statistics implementations will be released publicly on publication.

ETHICS STATEMENT

This work diagnoses and mitigates a text-generation defect; it does not introduce new data collection or human subjects. Two considerations. First, the defect itself has a safety face: models that fabricate proper nouns at $4\times$ the human rate produce plausible-looking misinformation (invented people, places, organizations); restoring real rare words reduces one fabrication channel but does not verify factuality—GRAFT commits words the model supports, not words that are true. The attestation gate of §6, on in every arm of Table 2, commits only words *attested in held-out human text*, which removes the channel by which the repair could itself invent proper nouns (2.1% of ELF and 0.8% of LangFlow commitments before the gate, 0% after); measured on XSum, GRAFT has no detectable effect on unsupported-entity rates in either direction (Appendix E.5). Second, increasing rare-word coverage can surface offensive or private strings memorized from the training corpus at a higher rate; our probes measured no verbatim-memorization pathology in these models (Appendix D), but deployments should keep standard output filtering in place.

AI USE STATEMENT

Generative AI tools were used substantially in this work, in the following required-disclosure categories: assisting with the design of experiments and providing feedback on methods (probe and control-arm design, dose and window selection); implementing methods (probe and sampler code); supporting analysis and interpreting results (statistical readouts, mechanism synthesis); and drafting and editing this manuscript, including its structure. Experiment execution was supervised and all judgments of experimental validity, retraction decisions, and final claims were reviewed and verified by the authors. AI tools were not used to generate synthetic training data, propose mathematical propositions or proofs, or fabricate any reported measurement. The authors take full responsibility for the content of this paper.

REFERENCES

Robin Algayres, Charles-Éric Saint-James, Mahi Luthra, Jiayi Shen, Dongyan Lin, Youssef Benchekroun, Rashel Moritz, Juan Pino, and Emmanuel Dupoux. LongTail-Swap: Benchmarking language models' abilities on rare words. *arXiv preprint arXiv:2510.04268*, 2025.

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

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021.

Iskander Azangulov and Leo Zhang. Sampling from flow language models via marginal-conditioned bridges. *arXiv preprint arXiv:2605.13681*, 2026.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Lin Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, Jianfeng Tan, Lanning Wei, Ji-Rong Wen, Yipeng Xing, Xiaolu Zhang, Junbo Zhao, Da Zheng, Jun Zhou, Junlin Zhou, Zhanchao Zhou, Liwang Zhu, and Yihong Zhuang. LLaDA2.0: Scaling up diffusion language models to 100B. *arXiv preprint arXiv:2512.15745*, 2025.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

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

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

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

Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and limitation of diffusion language model. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2502.09622.

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. Empowering diffusion models on the embedding space for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2505.13447.

Ishaan Gulrajani and Tatsunori B. Hashimoto. Likelihood-based diffusion language models. In *Advances in Neural Information Processing Systems*, 2023.

Florian Handke, Dejan Stančević, Felix Koulischer, Thomas Demeester, and Luca Ambrogioni. The entropic signature of class speciation in diffusion models. In *International Conference on Machine Learning*, 2026. arXiv:2602.09651.

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

Pascal Jutras-Dubé, Patrick Pynadath, Jeremy Lu, Yuan Gao, and Ruqi Zhang. Sticky jump diffusions: A unifying view of masked, continuous, and hybrid diffusion. *arXiv preprint arXiv:2607.10951*, 2026.

Kelly Kendro, Jeffrey Maloney, and Scott Jarvis. Do LLMs produce texts with "human-like" lexical diversity? *arXiv preprint arXiv:2508.00086*, 2025.

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

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

Bocheng Li and Linli Xu. Improving few-step language flows with untied self-conditioning. *arXiv preprint arXiv:2608.22244*, 2026.

Marvin Li and Sitan Chen. Critical windows: Non-asymptotic theory for feature emergence in diffusion models. In *International Conference on Machine Learning*, 2024.

Na Li, Yuchen Jiao, Changxiao Cai, and Gen Li. ConvergeFlow: Language flow with provable convergence to token embeddings. *arXiv preprint arXiv:2608.23551*, 2026.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems*, 2024.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems*, 2022.

Zachary C. Lipton and Jacob Steinhardt. Troubling trends in machine learning scholarship. *Communications of the ACM*, 62(3):45–53, 2019.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.

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

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, 2025.

Lavanya Nigam, Ishaan Bansal, Aryan Sood, Vidit Aggarwal, and Gaurav Kumar Nayak. Lost in interpolation: Why predictive feedback fails in diffusion language models. *arXiv preprint arXiv:2608.06529*, 2026.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, 2025.

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 *Advances in Neural Information Processing Systems*, 2021.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, 2024.

Alexander Shabalin, Simon Elistratov, Viacheslav Meshchaninov, Ildus Sadrtdinov, and Dmitry Vetrov. Why Gaussian diffusion models fail on discrete data and how to prevent it? In *Conference on Language Modeling*, 2026. arXiv:2604.02028.

Junzhe Shen, Jieru Zhao, Ziwei He, and Zhouhan Lin. CoDAR: Continuous diffusion language models are more powerful than you think. *arXiv preprint arXiv:2603.02547*, 2026.

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.

Robyn Speer. wordfreq: a library for looking up the frequencies of words. Zenodo software release, 2022.

Walter J. B. van Heuven, Paweł Mandera, Emmanuel Keuleers, and Marc Brysbaert. SUBTLEX-UK: A new and improved word frequency database for British English. *Quarterly Journal of Experimental Psychology*, 67(6):1176–1190, 2014.

Chuling Wen, Weijie Liang, and Jian Lu. Conditional total correlation and the serial depth of adaptive parallel sampling. *arXiv preprint arXiv:2608.25505*, 2026.

Longxuan Yu, Yunshu Wu, Yu Fu, Siheng Xiong, Rob Brekelmans, Hui Liu, Yue Dong, and Greg Ver Steeg. DSL-LLaDA: Scaling continuous denoising to 8B masked diffusion LMs. *arXiv preprint arXiv:2606.01024*, 2026.

Shuai Zhang, Zijie Chen, Hongliang He, Lun Du, and Zhenzhong Lan. Low perplexity is repetition: A one-dimensional self-conditioning attractor in continuous diffusion LMs. *arXiv preprint arXiv:2607.00588*, 2026.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *International Conference on Learning Representations*, 2025.