# FLOWSCOPE: INTERPRETING GENERATION DYNAMICS IN CONTINUOUS FLOW LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Continuous Flow Language Models (CFLMs) generate text by integrating a learned velocity field from noise to the sequences of embeddings that represent texts, providing an alternative to discrete diffusion and autoregressive language models. However, the generation dynamics in the continuous embedding space are not mechanistically studied. This work presents the first mechanistic interpretability analysis of CFLMs, examining the generative process through its central driving force, the velocity field, with sparse dictionary learning. Two embedding CFLMs, LangFlow and ELF, are investigated. We analyze the models based on their distinct designs, use the noise-free per-step estimation of the output state as the proxy for the model's internal belief, and map the evolution of these beliefs along generation paths. Token identities commit progressively following the learned schedule, in a coarse-to-fine order across token types. The disentangled dictionary separates a small population of frequently activating features from specialized features whose activity increases as trajectories converge, and activation steering then verifies the mechanical causality in these flow models. By examining per-variant trajectory properties, we implement online stability stopping for LangFlow and static schedule truncation for ELF, reducing denoiser evaluations by 12–25% while achieving 0.86–0.94 paired-token agreement, with measurable quality trade-offs. The analysis further examines the dynamics of self-distilled ELF weights, noting observable shifts in their commit clock and feature activation profile.

## 1 INTRODUCTION

Continuous flow language models (CFLMs) represent a recent alternative to autoregressive (AR) Language Models and discrete Diffusion Language Models (DLMs) for generative language modeling. Continuous formulations include simplex-based Flow Map Language Models (Lee et al., 2026a) and embedding-space models such as LangFlow (Chen et al., 2026) and ELF (Hu et al., 2026). We study the latter, which refine continuous text embeddings from Gaussian noise. In this process, all token states are refined concurrently within a continuous space.

While AR models benefit from a well-developed mechanistic interpretability toolkit, including sparse autoencoders (SAEs) and activation steering (Bricken et al., 2023; Cunningham et al., 2024; Turner et al., 2023), and recent work has begun to study discrete DLMs (Wang et al., 2026a; Zhou et al., 2026), mechanistic interpretability of CFLMs remains largely unexplored. Key questions of the generation process remain unresolved: What information does each step encode across the trajectory? When and how are tokens decided? These questions are distinct from AR model interpretability, where sequential token decisions follow next-token probabilities: CFLMs update all positions simultaneously over $T$ steps, with bidirectional interaction and state refinement until the last step. The velocity outputs represent displacements, which we find require distinct handling from hidden states.

This work introduces FLOWSCOPE, a mechanistic interpretability study of CFLMs that directly analyzes the velocity field. Each CFLM state update carries the noisy state along its interpolant path toward the model's estimated terminal $\hat{\mathbf{x}}_t$. We therefore use the terminal estimate as a direct proxy for the model's current belief about the outcome. To interpret this belief, we re-center the velocity update at the origin. This removes the interpolant-state term and, after rescaling by the prediction coefficient, exposes the predicted terminal in embedding coordinates. We obtain this estimate directly



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.

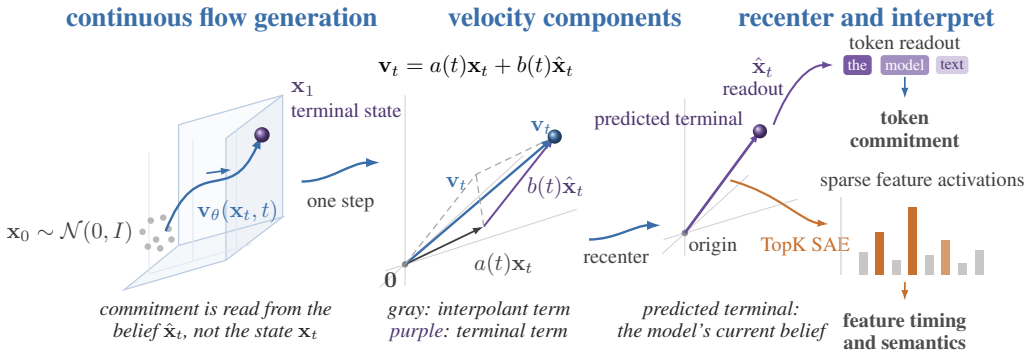


Figure 1: **FlowScope at one sampling step.** The velocity splits into an interpolant-state term and a predicted-terminal term; recentering isolates the predicted terminal, the model's current belief, which is read out as tokens and decomposed into sparse features.

from the model, as detailed in Section 4. The formulation gives us a common representation of the model's belief at each step, allowing us to investigate when token identities settle and what each step adds along the way. Through this lens, we find that token identities stabilize progressively while the beliefs continue to evolve. We track this process through token commitment timing, the point at which token readout from the model's belief becomes stable. Punctuation and function words stabilize earlier than frequent and rare content in the observed class averages, with an unresolved content tail extending into the final evaluations. A sparse dictionary trained on beliefs reveals a small broad population that activates across positions and a large specialized population whose activity rises around commitment and remains elevated afterward. The analysis is operationalized at inference. Belief steering increases target occurrences across the trajectory. Online stability stopping for LangFlow and a fixed, shortened schedule for ELF reduce denoiser evaluations by $12$–$25\%$ while maintaining $0.86$–$0.94$ token agreement with reference generation, exposing a practical trade-off between computation and changes to the generated output. The dictionary also transfers to modified checkpoints: an endpoint self-distilled student matches the base perplexity at half the evaluations. Analysis shows a re-timing of feature activation, with a overall retained profile.

This work contributes **(i)** A belief lens for embedding CFLMs: recentering the velocity exposes the predicted terminal as the model's per-step belief interpretable via the learned dictionary. **(ii)** A commitment-based temporal analysis of the generation trajectory: readout-robust commitment timing in LangFlow and feature activity along generation paths. **(iii)** Online stopping and static schedule truncation informed by trajectory measurements, with evaluation savings assessed alongside output agreement and generation quality. **(iv)** An analysis of endpoint self-distilled checkpoints with these instruments, which finds that the student meets its half-evaluation budget by advancing the commit clock and selectively re-timing content features over a largely inherited feature set.

## 2 Related Work

**Interpreting diffusion and flow LMs.** DLM-Scope (Wang et al., 2026a) trains SAEs on the hidden states of discrete diffusion LMs, while Zhou et al. (2026) map attribute-level commitment schedules of DLMs. In the continuous regime, flow matching (Lipman et al., 2023) uses velocity fields to transform noise into data. Our analysis uses the standard algebra involving interpolants and preconditioning for this type of generation (Karras et al., 2022; Albergo and Vanden-Eijnden, 2023). Unlike previous analyses of hidden activations using SAEs, we train the dictionaries on the predicted terminal state from the velocity field and treat their features as components of the model's current output estimate.

**Commitment and critical windows.** Li et al. (2026) and Kohut et al. (2026) exploit token-level early convergence to accelerate discrete DLM inference: Prophet commits the remaining tokens when a confidence-gap condition is satisfied, while JOT freezes the converged positions separately. We carry out a similar analysis by examining both confidence and stability along continuous-flow

trajectories. Previous studies of diffusion dynamics have found specific time intervals during which semantic structure emerges or becomes fixed (Raya and Ambrogioni, 2023; Biroli et al., 2024; Li and Chen, 2024). We investigate whether semantic and token commitment establish an analogous intervention window in continuous text flows.

**Intervention and inference control.** Activation steering affects the model's representations along directions that are related to the task or concept, while SAE features offer more disentangled directions for interpreting and controlling the model's behavior (Turner et al., 2023; Templeton et al., 2024). We apply the steering operation to the predicted terminal of a CFLM and compare it in early and late sampling windows. With respect to discrete DLMs, sampling acceleration has been investigated using fast solvers (Song et al., 2021; Lu et al., 2022; Sabour et al., 2024), distillation (Salimans and Ho, 2022; Song et al., 2023), velocity approximation (Jin et al., 2026; Gao et al., 2026), and early-exit methods based on confidence or trajectory statistics (Li et al., 2026; Kohut et al., 2026; Sun et al., 2026; Wang et al., 2026b; Lee et al., 2026b). In this work, we select each CFLM's stopping or step-skipping rule from its measured commitment dynamics.

## 3 PRELIMINARIES

**Continuous flow language models.** A CFLM represents a length-$L$ sequence by a continuous state $\mathbf{x}_t \in \mathcal{X}$ and learns a time-dependent velocity field $\mathbf{v}_\theta : \mathcal{X} \times [0,1] \to \mathcal{X}$ that defines an trajectory,

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}_\theta(\mathbf{x}, t), \quad \mathbf{x}(0) \sim \mathcal{N}(0, I), \tag{1}$$

whose endpoint $\mathbf{x}(1)$ approximates the data distribution. In practice, the flow is discretized into a finite sequence of updates, and the endpoint representation is mapped to tokens by a model-specific readout. LangFlow and ELF, the two models studied here, operate on continuous text embeddings.

**Predicted terminal and velocity.** LangFlow and ELF operate in embedding space and expose a prediction $\hat{\mathbf{x}}_t$ of the clean terminal state conditioned on the current interpolant $\mathbf{x}_t$. Their samplers convert this prediction into a velocity according to the interpolation schedule:

$$\mathbf{v}_t = a(t)\,\mathbf{x}_t + b(t)\,\hat{\mathbf{x}}_t, \tag{2}$$

where $a(t)$ and $b(t)$ are scalar functions determined by the model's path parameterization. We interpret $\hat{\mathbf{x}}_t$ as the model's current belief about the terminal output.

**Sparse autoencoders.** A TopK SAE (Gao et al., 2025) represents a vector $\mathbf{z} \in \mathbb{R}^d$ as a sparse nonnegative combination of learned dictionary directions,

$$\hat{\mathbf{z}} = \mathbf{b}_d + \mathbf{W}_d\,\mathbf{f}, \quad \mathbf{f} = \mathrm{ReLU}(\mathrm{TopK}(\mathbf{W}_e(\mathbf{z} - \mathbf{b}_d))), \tag{3}$$

where the encoder centers its input by the decoder bias, TopK retains the $k$ largest preactivations, and ReLU clips negative values to zero. At most $k$ features are positive, and training minimizes the reconstruction loss: $\|\mathbf{z} - \hat{\mathbf{z}}\|^2$.

## 4 METHODOLOGY

FLOWSCOPE studies LangFlow and ELF within a unified framework of embedding CFLMs. We use the predicted terminal as the model's current belief, interpret it with a sparse dictionary, and track how its content evolves. These trajectory measurements then support analyses of intervention timing and inference acceleration. We proceed with the methodology at the framework-level abstraction.

For a length-$L$ sequence, the interpolant $\mathbf{x}_t$, velocity $\mathbf{v}_t$, and predicted terminal $\hat{\mathbf{x}}_t$ lie in $\mathbb{R}^{L\times d}$. We collect $N$ trajectories with $S$ recorded evaluations, indexed by $s \in \{0, \ldots, S-1\}$ at sampler times $t_s$.

### 4.1 THE RELATION TO VELOCITY

The velocity describes how the current interpolant changes and combines contributions from both the interpolant and the predicted terminal. Under the parameterization in Equation (2), recentering removes the noisy interpolant component:

$$\mathbf{r}_t = \mathbf{v}_t - a(t)\mathbf{x}_t = b(t)\hat{\mathbf{x}}_t. \tag{4}$$

The remaining component is proportional to the predicted terminal. We therefore directly analyze $\hat{\mathbf{x}}_t$ in embedding coordinates.

### 4.2 BELIEF LENS

We train a TopK SAE on predicted-terminal vectors pooled across generations, steps, and token positions. Using predictions removes the sampler gain $b(t)$ present in the residual. Applied to the belief at generation $n$, evaluation $s$, and position $i$, the encoder yields the sparse code

$$\mathbf{f}_{s,i}^{(n)} = \mathrm{ReLU}\Big(\mathrm{TopK}\Big(\mathbf{W}_e(\hat{\mathbf{x}}_{t_s,i}^{(n)} - \mathbf{b}_d)\Big)\Big) \in \mathbb{R}^{d_{\mathrm{sae}}}, \tag{5}$$

whose $j$-th entry $f_{s,i,j}^{(n)}$ is the activation of feature $j$ and $\mathbf{d}_j$ is the $j$-th column of $\mathbf{W}_d$.

We interpret features by collecting their strongest activation sites and examining the context around them. To distinguish common features from specialized features, we measure the site prevalence of each feature over $N$ generations of $L$-token sequences:

$$\pi_j = \frac{1}{NL}\left|\left\{(n,i) : \exists s,\ f_{s,i,j}^{(n)} > 0\right\}\right|. \tag{6}$$

A feature is active if it has at least one strictly positive activation along the path. The distribution of $\pi_j$ over active features, examined in Section 5, determines how we partition features into broad and specialized populations. We sample a subset of activation contexts for each feature, and analyze their activation duration and timing.

### 4.3 STATE EVOLUTION AND TOKEN COMMITMENT

Let $q_{n,s,i}$ denote the model's own token distribution at evaluation $s$, and let $y_{n,s,i}$ denote its argmax. Token commitment is the first evaluation from which the readout agrees with the final recorded token at every remaining step:

$$c_{n,i} = \min\{s : y_{n,u,i} = y_{n,S-1,i} \text{ for every } u \geq s\}. \tag{7}$$

To examine how token content develops, we normalize this statistic by $S-1$ and compare commitment across final-token classes.

To understand the feature dynamics, we measure the relative steps from token commitment to the feature's activation step: $\delta = s - c_{n,i}$. Excluding an initial window of high-noise states, we compute the average activation rates of features within the broad and specialized populations and evaluate the rates across steps.

### 4.4 INTERVENTIONS AND INFERENCE ACCELERATION

We edit the predicted terminal along a unit token-embedding or SAE dictionary direction $\mathbf{d}$. At each position $i$ within a fixed evaluation window $\mathcal{W}$, we apply

$$\hat{\mathbf{x}}_{t_s,i} \leftarrow \hat{\mathbf{x}}_{t_s,i} + \alpha_s \mathbf{d}, \qquad \alpha_s = \alpha\,\nu_s, \quad s \in \mathcal{W}, \tag{8}$$

where $\nu_s$ is the mean belief norm over positions and sequences in the current generation batch. We compare early and late windows under the same initial noise, measuring target occurrences. The edited prediction propagates through each model's own channels.

Both acceleration rules exploit generation stabilization. For LangFlow, the token readout is free, so stabilization is tested online. Define $\mathbf{y}_s^{\mathrm{stop}} = (y_{n,s,1}, \ldots, y_{n,s,L}) \in \mathcal{V}^L$ for the current generation $n$, where $\mathcal{V}$ is the vocabulary. We stop when this entire token readout remains unchanged for $k_{\mathrm{stop}}$ consecutive comparisons:

$$\tau = \min\left\{s \geq k_{\mathrm{stop}} : \mathbf{y}_u^{\mathrm{stop}} = \mathbf{y}_{u-1}^{\mathrm{stop}} \text{ for } u = s - k_{\mathrm{stop}} + 1, \ldots, s\right\}. \tag{9}$$

For ELF, such token readout is unaffordable, so a fixed strategy is preferable: a shortened schedule that retains an initial portion of the original grid, and then jumps to the terminal time.

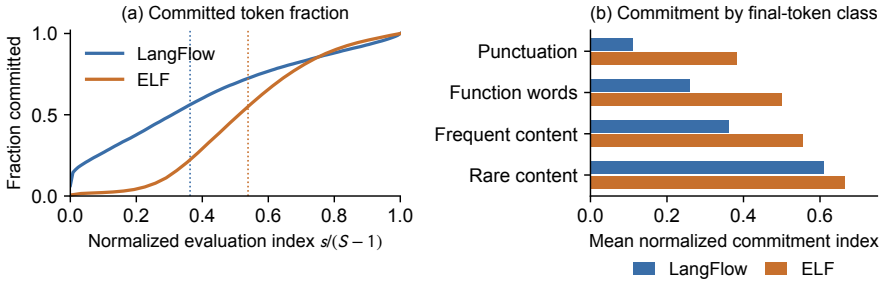


Figure 2: **Token commitment and its class order. (a)** Committed token fraction; dotted lines mark the mean commitment of $36.3\%$ for LangFlow and $53.9\%$ for ELF. **(b)** Punctuation and function words commit before frequent and rare content words in both models.

## 5 Experiments

We analyze $500$ unconditional trajectories per model: LangFlow's LM1B checkpoint with $d = 768$ and length $128$, whose $128$ sampling evaluations record the first $S = 127$ predictions while the final output evaluation is omitted from the cache, and ELF-B with $d = 512$, length $1024$, and $S = 32$ SDE evaluations. Token and feature analyses share these caches, with one TopK SAE of $16{,}384$ features at $k = 64$ trained on each model's predicted terminals. LangFlow beliefs follow its definition of state as the posterior-weighted mean token embedding. ELF beliefs are obtained as the output of forward evaluation. Implementation details and controls are in Section A; endpoint self-distillation details are in Section B.

### 5.1 Tokens commit in class order

We use token commitment to compare stabilization across positions and to align feature activities. As shown in Figure 2a, commitment exhibits a gradual progression across token positions, which bears a surprising resemblance to the discrete unmasking of DLMs. A flow model does not collapse to its output at a single moment or refine all token positions at a uniform pace. Instead, it settles position by position. In Figure 2b, the ordering of token commitment is shown: punctuation commits first, followed by function words, then frequent content words, with rare content words committing last, in both models. We read this as the process laying down formatting and scaffolding before it commits to lexical substance, a coarse-to-fine order that discrete models exploit elsewhere.

### 5.2 Specialized features arrive with commitment

In Figure 3a,b, the site-prevalence distribution of the dictionary is bimodal in both models: a small always-on population and a large occasion-driven one, separated by the histogram minimum. Table 1 shows the function of the two populations: Broad features are the plumbing of text, articles, and temporal markers that activate almost everywhere. Specialized features deliver specific semantics, such as rising prices and blogging, and activate sparsely on concentrated meanings. We further investigate feature activation and final token output when the models actively revise the semantics in Appendix Table 7.

The commit-aligned curves in Figure 3c,d show that broad-feature activity is flat around commitment, while specialized features rise to the commit point and remain elevated long after it, with roughly three-quarters of the specialized activation rate mass occurring after commitment in both models. As Table 1 shows that the specialized features encode more complex semantics, the late activation of specialized features aligns with the observation that content tokens commit later in the sequence. The analysis reinforces the hypothesis that CFLMs generate in a structure-then-content plan.

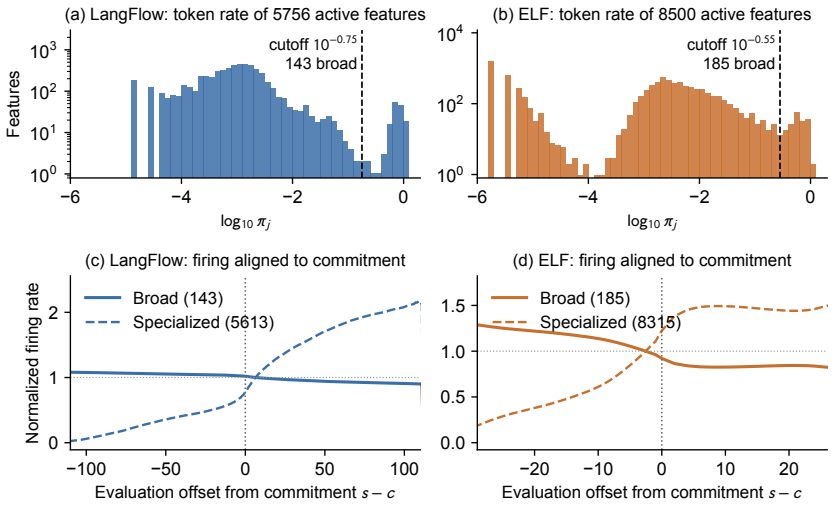


Figure 3: **Feature populations and commitment. (a, b) Activation frequency histogram**: Site-prevalence distributions separate broad and specialized features. **(c, d) Within-class normalized firing rate**: Commitment-aligned activation remains relatively steady for broad features and rises for specialized features. The activation rates are divided by their class means to compare trends.

### 5.3 The belief condenses onto the committed text

Following (Dreyer et al., 2025), we use the clarity score to monitor feature quality along the generation path. The score is the mean pairwise cosine similarity between CLIP embeddings. In CFLMs, we use the model's native token embedding. Table 2 computes it for two interpretation streams: the readout stream uses the token decoded at the activation step, and the final-text stream uses the token that eventually lands. Each feature is scored in the early, mid, and late thirds of the schedule and on either side of its positions' commitment.

ELF's outcome-token clarity climbs monotonically across windows: its early beliefs decode to repetition and fragments that occur in no landed text, and they sharpen onto exactly the tokens that survive. LangFlow's clarity is flat and high from the early window on: its decoded estimate is already close to its realized text, the same fact as its earlier commit clock read in a second coordinate. Before commitment, the readout stream is slightly more coherent than the other in both models: the belief is a marginally cleaner version of the text than the text's own history. Protocols for both readouts are in Section A.5.

Table 1: **Dictionary feature examples.** Strong activations identify recurring token contexts in both populations; labels are informal context descriptions. Duty is the average fraction of evaluations at which a feature is active, over the positions where it ever fires. Timing is the median normalized step of a feature's activation events, with quartiles.

| Model | Type | Label | duty | timing | Top sites |
|---|---|---|---|---|---|
| LangFlow | broad | averages and indices | 0.91 | 0.46 [0.22,0.72] | …dow jones industrial **average** slipped 1. 48 percent<br>…10 percent, on **average**. |
| LangFlow | specialized | price and sales rises | 0.35 | 0.64 [0.44,0.83] | …sweet crude for april delivery **rose** $ 1. 03 to $ 72<br>…sales **rose** about 9 percent |
| ELF | broad | dates and time | 0.56 | 0.42 [0.16,0.71] | …began in 1995 in **2003**<br>…Championship title for Liverpool City in **January** 2013 |
| ELF | specialized | blogging | 0.20 | 0.74 [0.58,0.87] | …reluctant about starting a **blog**<br>…a **blog** post contains good tools |

Table 2: **Per-feature clarity by stream and trajectory position.** Clarity is the mean pairwise cosine among the token-embedding directions of a feature's focal tokens and averaged over features with at least four scored tokens. Early, mid, and late are equal thirds of the normalized schedule $s/(S-1)$. Post-commit the two streams are identical by definition.

| Model | Stream | Early | Mid | Late | Pre-commit | Post-commit |
|---|---|---|---|---|---|---|
| LangFlow | readout | 0.62 | 0.54 | 0.57 | 0.48 | 0.55 |
| LangFlow | final text | 0.64 | 0.50 | 0.55 | 0.37 | 0.55 |
| ELF | readout | 0.31 | 0.55 | 0.69 | 0.26 | 0.64 |
| ELF | final text | 0.28 | 0.51 | 0.68 | 0.19 | 0.64 |

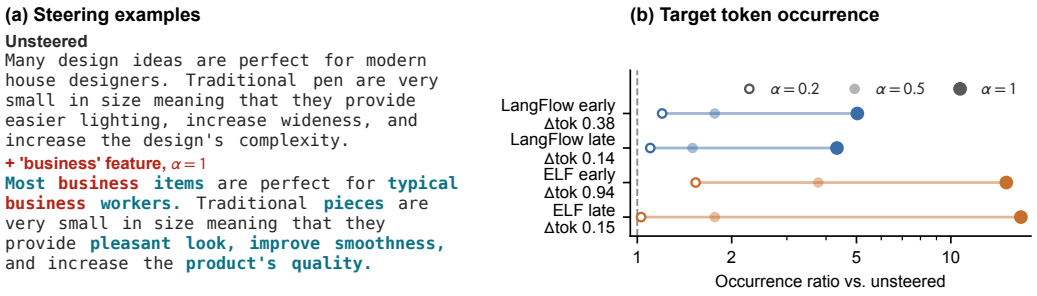


Figure 4: **Steering in early and late windows.** (a) One ELF generation and its steered pair (the 'business' feature, $\alpha$=1, late window; the pair with the largest target change among the 16 paired generations). Red marks the inserted target words, blue the other 11 changed words. (b) Pooled target occurrences relative to unsteered by window, at strengths $\alpha$ of the belief norm (open, half, filled: 0.2, 0.5, 1); row labels give each window's mean token-change rate.

### 5.4 Causal verification and steer timing matters

We verify the causal mechanism with alpha steering. Figure 4 carries the dose-response and an example pair. The commit clock affects steerability in different ways. Early edits introduce rewrites, move a large fraction of tokens, and raise repetition. For ELF, the early effect is destructive and does not even preserve the semantics of specific features. Late edits are substitutions. A small fraction of tokens moves; the target occurs at an elevated rate. The example in Figure 4a shows a successful late steering.

### 5.5 Readout stability buys cheaper sampling

Both acceleration rules in Table 3 instantiate one principle: once the readout stops changing, further evaluations buy nothing. The models differ only in what the test costs. LangFlow's readout arrives free with every evaluation, so it tests stabilization online and stops after three unchanged comparisons. ELF's decoder readout costs about as much as an evaluation, at 7.3 versus 8.0 ms per evaluation in component measurements, which, in contrast, is not a free lunch. Therefore, the early-stop schedule, which keeps the trajectory prefix and jumps to the terminal time, is favorable.

The results in Table 3 are consistent with the principle's predictions. Both rules display a slight tradeoff in generation capacity, yet the reductions in forward evaluations are significant. Notably, uniform respacing, which can approximate the sampling grid, fails to compete with naive truncation of the schedule, suggesting potentially less effective oscillation in the latter parts of ELF generation paths.

## 6 Self-distillation advances the commit clock

We advance the analysis on ELF with self-distillation (Zhang et al., 2019): our objective is a toy approach inspired by it, fine-tuning the checkpoint to predict its own outputs at more evaluation steps. This section formulates the intervention, sets up the FlowScope readouts on the students, and reports the results.

Table 3: **Inference acceleration.** Online stopping and fixed schedule reduction save denoiser evaluations with changes to the output. Ratios exclude decoder and trigger costs; agreement uses paired full-schedule outputs. uni-JSD is the unigram Jensen–Shannon divergence against the paired full-schedule outputs; for ELF the baseline's own split-half floor is $0.225$. MAUVE (Pillutla et al., 2021) compares each rule's outputs with the full schedule.

| Model | Rule | Eval. ratio | Token agree. | $\Delta$lp | uni-JSD | MAUVE |
|---|---|---|---|---|---|---|
| LangFlow | argmax-stability stopping, $k_{\text{stop}}$=3 | $1.14\times$ | 0.94 | $+0.04$ | 0.018 | 1.00 |
| ELF | fixed schedule, $24/32$ evaluations | $1.33\times$ | 0.86 | $-0.18$ | 0.222 | 0.999 |
| ELF | uniform respacing, $24/32$ evaluations | $1.33\times$ | 0.49 | $-0.19$ | 0.226 | 0.997 |

Table 4: **Feature advancement under self-distillation.** The three largest and three smallest advances on the SD-16 arm, base median minus student median on each arm's own normalized schedule. Contexts show the final text around the highest-activation site of each feature in the base corpus, with the activation site marked.

| Feature | advance | base→SD | events | Top site |
|---|---|---|---|---|
| 2526 | +0.768 | 0.97→0.20 | 5 | … **There** are a variety of apps |
| 3798 | +0.671 | 0.87→0.20 | 5 | … are on Saturday. The Premier **League** Champions are heading into their third |
| 3296 | -0.352 | 0.58→0.93 | 8 | … the suspect possessed a **fake** passport, but he hadn |
| 10707 | -0.351 | 0.52→0.87 | 5 | … To Go **O.C.:** Game of Thrones became one of |

### 6.1 Self-distillation formulation

A fixed teacher provides records $(\mathbf{x}, \mathbf{h}, t, y^*)$ holding an intermediate model input, its self-conditioning context, the stored time, and decoded endpoint tokens. The student minimizes

$$\mathcal{L} = \mathcal{L}_{\text{recipe}} + \lambda_{\text{sd}}\, \mathbb{E}_{(\mathbf{x},\mathbf{h},t,y^*)} \left[ \text{CE}(\text{dec}_\theta(\hat{\mathbf{x}}_\theta(\mathbf{x}, t; \mathbf{h})), y^*) \right]. \tag{10}$$

The endpoint term decodes the student's intermediate belief and averages cross-entropy over token positions, with gradients through prediction and decoding, while the recipe and distillation terms use separate minibatches. The unconditional student is fine-tuned from the released base checkpoint for $12{,}000$ steps at $\lambda_{\text{sd}} = 0.005$ under the original optimizer settings. Training details, the teacher-budget variant, and the strength check at tripled weight are in Section B.

### 6.2 Analysis setup

We cache $500$ generations of length $1024$ for the base and the student at $16$ and $8$ evaluations under guidance $3.0$, with seeds paired to the base cache. Commitment milestones follow each arm's own grid. Feature statistics encode the cached predictions with the frozen base belief dictionary: the student's own prevalence histogram carries no always-on cluster, so broad counts keep the base cutoff. Diversity metrics follow the acceleration suite on the final decodes.

### 6.3 Results

The self-distillation checkpoint matches the base perplexity at half the evaluations. The student buys its budget by advancing the commit clock. The base reaches half coverage around evaluation $17$, while the student reaches it at evaluation $6$ of its own $16$, and its clock stays ahead of the base's at every reported milestone, as shown in Table 5. Endpoint supervision teaches the model to settle on a token identity faster. More distillation configurations are discussed in Section B.

The distilled model inherits its plan and re-schedules it. In Table 5, the frozen base dictionary explains the student's beliefs at high FVE, indicating that the feature vocabulary is preserved. What distillation moves is timing, and Figure 5a shows the move is heterogeneous: broad features retain their normalized timing, while the specialized cloud is redrawn around an earlier center. Figure 5b grades the redraw by lateness: the advance grows with base timing, so later content moves earlier by more.

Table 5: **Self-distillation and its effect on the dynamics.** The student trained with two budgets against the base. Coverage is the percentage of each arm's own schedule. Post-commit share is the specialized-feature rate mass at offsets $s - c \geq 0$. rep-3 is the repeated-trigram fraction.

| Arm | Evals | PPL | 50% cov. | 90% cov. | Post-commit | Dict. FVE | rep-3 |
|---|---|---|---|---|---|---|---|
| base | 32 | 24.3 | 52.5% | 80.4% | 74.2% | 0.940 | 0.118 |
| SD | 16 | 24.7 | 39.2% | 80.8% | 75.0% | 0.923 | 0.126 |
| SD | 8 | 40.5 | 50.9% | 95.1% | 72.0% | 0.919 | 0.068 |

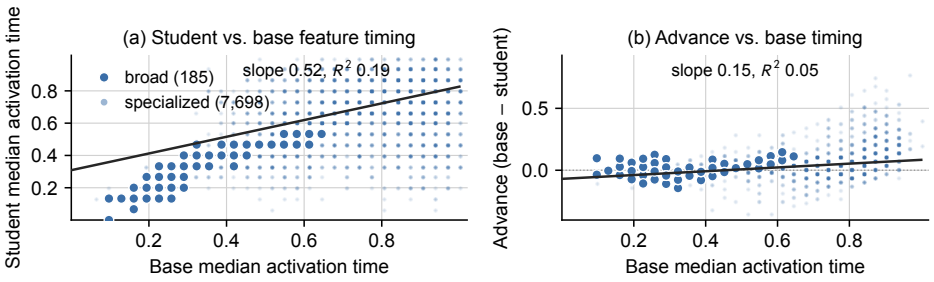


Figure 5: **Self-distillation re-times the specialized population.** **(a)** Median activation time per feature, student at 16 evaluations against the base, frozen base dictionary. Broad features lie on the identity line, slope 0.79; the specialized cloud lies below it, slope 0.52. **(b)** Timing advance, base minus student, against base timing, features with at least five events.

The extreme cases are listed in Table 4. The largest advancers are easy targets: endpoint surface features, sentence openers, and title anchors that the base fires only near the end. On the other hand, the features that slip towards the late end are harder to generate: rare abbreviations and unlikely word co-occurrences. Such observation resonates with the structure-then-content generation plans of the flow models. Since sentence structure planning occurs early and leaves room for correction, such features allow more headroom for acceleration.

16-step distillation retains the generation capacity, with perplexity and trigram repetition comparable to baseline. The distilled student's first readout becomes feature-rich, with an order of magnitude more active features than the base's, and it skips the noise phase that the base spends its early schedule in. Our analysis reveals that distillation facilitates structure formation in the model's belief.

## 7 Conclusion

We introduced FlowScope, a framework for interpreting embedding CFLMs through their predicted terminal beliefs, motivated by the velocity decomposition. Connecting token commitment with sparse feature dynamics reveals a temporally structured generation process: across LangFlow and ELF, punctuation and function words stabilize earlier than content words on average, while specialized feature activity rises around commitment and remains elevated afterward. Stable token identities therefore coexist with continued feature activity. This temporal view informs both intervention and inference: late belief edits increase target occurrences with fewer token changes, and model-specific stopping or schedule truncation reduces denoiser evaluations by 12–25% with measurable output trade-offs. Endpoint self-distillation further shows that the timing of belief formation can change under supervision. The ELF student approaches the base model's generative perplexity at half the evaluations, with earlier commitment and selective feature re-timing, while its beliefs remain well reconstructed by the frozen base dictionary. Together, these findings connect what a flow model predicts, when its predictions stabilize, and how that timing can be modified, making belief trajectories a useful basis for understanding and improving continuous text generation.

## AI USE STATEMENT

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis, and interpret results. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, assist with translation, clean and reformat dataset. And to generate synthetic data sets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI. In addition, SAE feature auto-interpretation involves LLM generation in its standard practice and original definition.

## ETHICS STATEMENT

This work focuses on methodological advances in language modeling and interpretability. This work does not involve human subjects, sensitive personal data, or applications that present immediate ethical concerns. We are not aware of any ethical issues specific to this work beyond those generally associated with machine learning research.

## REPRODUCIBILITY STATEMENT

We provide sufficient details in the main paper and appendix to reproduce our results, including methodological definitions, mathematical formulations, model configurations, and experimental hyperparameters. All pretrained model checkpoints used in this work are publicly available through their respective publications.

## REFERENCES


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*, 2026a.

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.

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.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, 2024.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Xu Wang, Bingqing Jiang, Yu Wan, Baosong Yang, Lingpeng Kong, and Difan Zou. DLM-scope: Mechanistic interpretability of diffusion language models via sparse autoencoders. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=Z9cuAhhIbU.

Hanhan Zhou, Shamik Roy, and Rashmi Gangadharaiah. Steering without breaking: Mechanistically informed interventions for discrete diffusion language models. *arXiv preprint arXiv:2605.10971*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Yi Liang, Soroush Vosoughi, and Shiwei Liu. Diffusion language model knows the answer before it decodes. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=g88nt4ieTG.

Zakhar Kohut, Severyn Shykula, Mykola Vysotskyi, Serhii Dmytryshyn, Dmytro Khamula, Michal Zakrzewski, Damian Rynczak, Jacek Małecki, Taras Rumezhak, and Volodymyr Karpiv. Just on time: Token-level early stopping for diffusion language models. *arXiv preprint arXiv:2602.11133*, 2026.

Gabriel Raya and Luca Ambrogioni. Spontaneous symmetry breaking in generative diffusion models. *Advances in Neural Information Processing Systems*, 36:66377–66389, 2023.

Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

Marvin Li and Sitan Chen. Critical windows: non-asymptotic theory for feature emergence in diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=nBGBzV4It3.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32211–32252. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Cheng Jin, Zhenyu Xiao, and Yuantao Gu. A-flops: accelerating diffusion models via adaptive flow path sampler. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i27.39397. URL https://doi.org/10.1609/aaai.v40i27.39397.

Huanlin Gao, Ping Chen, Fuyuan Shi, Ruijia Wu, Li YanTao, Qiang Hui, Youyuren, Ting Lu, Chao Tan, Shaoan Zhao, Zhaoxiang Liu, Fang Zhao, Kai Wang, and Shiguo Lian. Meancache: From instantaneous to average velocity for accelerating flow matching inference. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GMCyL7Xs9R.

Bohang Sun, Max Zhu, Francesco Caso, Jindong Gu, Junchi Yu, Philip Torr, Pietro Liò, and Jialin Yu. The path matters: Learning a token-commitment policy for diffusion language models. *arXiv preprint arXiv:2605.24697*, 2026.

Chengcheng Wang, Tingzhang Luo, Wenhao Li, Jianyuan Guo, and Chang Xu. Tacg: Trajectory-aware commit gating for diffusion language model decoding. *arXiv preprint arXiv:2607.03236*, 2026b.

Chia-Ming Lee, Shao-Kai Liu, Ming-Ching Chang, Xin Li, Yu-Lun Liu, and Chih-Chung Hsu. Commit locally, exit globally: Coordinating adaptive sampling and early exit in diffusion language models. *arXiv preprint arXiv:2607.28166*, 2026b.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, volume 2025, pages 26721–26754, 2025.

Maximilian Dreyer, Jim Berend, Tobias Labarta, Johanna Vielhaben, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. Mechanistic understanding and validation of large ai models with semanticlens. *Nature Machine Intelligence*, 7(9):1572–1585, Sep 2025. ISSN 2522-5839. doi: 10.1038/s42256-025-01084-w. URL https://doi.org/10.1038/s42256-025-01084-w.

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. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 110–119, 2016.

Linlu Qiu, Keya Hu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. Progressive distillation of ELF: Few-step generation for embedded language flows. https://linlu-qiu.github.io/assets/html/elf_pd.html, 2026.

# A Experimental Details

## A.1 Sampler coordinates and recorded velocities

**LangFlow.** Write $\hat{\mathbf{x}}_s := \hat{\mathbf{x}}_{t_s}$. Let $\gamma_s$ be the model's decreasing log-SNR coordinate, $\Delta\gamma_s = \gamma_{s+1} - \gamma_s$, and sp denote softplus. The implemented update is

$$w_s = \exp(\Delta\gamma_s/2), \qquad c_s = \exp\left(\frac{\mathrm{sp}(\gamma_s) - \mathrm{sp}(\gamma_{s+1})}{2}\right),$$
$$\mathbf{x}_{s+1} = w_s c_s \mathbf{x}_s + (1 - w_s)\sqrt{\mathrm{sigmoid}(-\gamma_{s+1})}\,\hat{\mathbf{x}}_s.$$

The recorded finite-step velocity is $(\mathbf{x}_{s+1} - \mathbf{x}_s)/\Delta\gamma_s$, so its coefficients in Equation (2) are

$$a_s = \frac{w_s c_s - 1}{\Delta\gamma_s}, \qquad b_s = \frac{(1 - w_s)\sqrt{\mathrm{sigmoid}(-\gamma_{s+1})}}{\Delta\gamma_s}.$$

These are finite-grid coefficients, not estimates needed to extract $\hat{\mathbf{x}}_s$. The implementation adds $10^{-12}$ to the velocity denominator. The belief is the native posterior's embedding expectation and is also used for self-conditioning. Of the 128 network evaluations, the first 127 predictions and update inputs are cached; the last output evaluation is not included in the commitment clock.

**ELF.** At a stochastic step with $h_s = t_{s+1} - t_s$, the sampler uses

$$\eta_s = \mathrm{clip}(1 - \gamma_{\mathrm{SDE}} h_s, 0, 1), \quad \widetilde{t}_s = \eta_s t_s,$$
$$\widetilde{\mathbf{x}}_s = \eta_s \mathbf{x}_s + (1 - \eta_s)\boldsymbol{\epsilon}_s, \qquad \boldsymbol{\epsilon}_s \sim \mathcal{N}(0, \sigma^2_{\mathrm{noise}} I),$$
$$\mathbf{v}_s = \frac{\hat{\mathbf{x}}_s - \widetilde{\mathbf{x}}_s}{\max(1 - \widetilde{t}_s, \epsilon_t)}, \qquad \mathbf{x}_{s+1} = \widetilde{\mathbf{x}}_s + (t_{s+1} - \widetilde{t}_s)\mathbf{v}_s.$$

Here $\gamma_{\mathrm{SDE}}$, $\sigma_{\mathrm{noise}}$, and the denominator floor $\epsilon_t$ are sampler/configuration parameters. The final step uses $\eta_s = 1$ without noise injection. At the perturbed input, $a_s = -1/\max(1 - \widetilde{t}_s, \epsilon_t)$ and $b_s = -a_s$. The cached tensors are $\widetilde{\mathbf{x}}_s$, the model prediction $\hat{\mathbf{x}}_s$, and $\mathbf{v}_s$, stored at reduced precision. Self-conditioning guidance combines predictions and velocities with the same affine weights, preserving this identity in exact arithmetic for the unconditional protocol. Noise enters before this forward evaluation. It is not an additive residual in the recorded drift identity. Empirical fit error therefore needs a cache/precision audit rather than an automatic attribution to SDE noise.

## A.2 Scale, protocols, and recipes

**Data and scale.** The cached belief and feature analyses use full unconditional generations recorded step by step. For LangFlow, we collect 500 generations at the released evaluation protocol (length 128, 128 sampling steps) with $S = 127$ recorded steps, giving 64,000 (sample, position) sites and 8.1M site-step observations. For ELF, we collect 500 generations of length 1024 with $S = 32$ recorded steps, giving 512,000 sites and 16.4M site-step observations. The dictionaries ($d_{\mathrm{sae}} = 16{,}384$, $k = 64$) are trained directly on cached $\hat{\mathbf{x}}_t$ rows: up to 3M rows subsampled per model. The loader stops collecting shards once it exceeds twice the row cap, then subsamples the collected pool. We use one dictionary per model and seed 0, with 30,000 Adam updates at learning rate $5 \times 10^{-4}$ and batch size 4096; decoder columns are renormalized after each update. The encoder centers its input by the learned decoder bias, selects the largest $k$ preactivations, and clips negative values to zero. The training-distribution reconstruction diagnostic, computed as one minus mean squared error divided by variance over flattened batch coordinates, is 0.997 for LangFlow and 0.955 for ELF. On the 128/128 LangFlow run, the unnormalized-residual dictionary uses approximately $2.5\times$ as many active atoms as the belief dictionary, based on the training evaluations, with dead fractions 0.1302 and 0.6537. This is consistent with the time-dependent residual gain increasing dictionary usage; the current comparison does not establish matched feature semantics or timing across these dictionaries. Generation-quality evaluations use $n = 1000$ generations at seed 42 under each model's official sampler and are scored by GPT-2-large at context length 1024; task evaluations use the full official validation splits with paired noise across systems. All experiments ran on NVIDIA B200, H200, and RTX Pro 6000 Blackwell GPUs.

Table 6: **Representative per-feature advancement under self-distillation.** The six largest and six smallest advances on the sd32 run, base median minus student median on each own normalized time. The features have at least five activations. Contexts show the final text around the highest-activation site of each feature in the base corpus, with the activation site marked.

| Feature | advance | base→SD | events | Top site |
|---|---|---|---|---|
| 2526 | +0.768 | 0.97→0.20 | 5 | … **There** are a variety of apps |
| 3798 | +0.671 | 0.87→0.20 | 5 | … are on Saturday. The Premier **League** Champions are heading into their third |
| 5085 | +0.636 | 0.90→0.27 | 5 | … loss at Old Man Stadium on **the** next day. They still have |
| 3554 | +0.606 | 0.81→0.20 | 23 | … units in Harris County by the **end** of the year. M |
| 9988 | +0.570 | 0.90→0.33 | 17 | … changes when it comes to the **light** app. As long as you |
| 10962 | +0.538 | 0.87→0.33 | 5 | … subject. If you have any **questions** about the subject, please let |
| 3296 | -0.352 | 0.58→0.93 | 8 | … the suspect possessed a **fake** passport, but he hadn |
| 10707 | -0.351 | 0.52→0.87 | 5 | … To Go **O.C.:** Game of Thrones became one of |
| 7188 | -0.314 | 0.42→0.73 | 6 | … Ben for taking these photos! **Finy** Photos is a |
| 14792 | -0.254 | 0.61→0.87 | 7 | … schools have moved from using the **online** term to abandoning the term |
| 15364 | -0.252 | 0.55→0.80 | 5 | … to really love our high quality **shooting** products! We've got |
| 13172 | -0.249 | 0.48→0.73 | 5 | … che Aven, authorities **said.** After a previous investigation the |

**Self-distillation measurement protocol.** The student of Section B (the $\lambda = 0.005$ run; the $\lambda = 0.015$ run for the strength check) is cached with 500 generations at length 1024 for 32, 16, and 8 evaluations, guidance 3.0, seeds paired with the base cache. Commitment milestones follow the Table 5 coverage convention on each arm's own grid. Feature statistics encode the cached predictions with the frozen base belief dictionary, and broad counts use the base 0.28 cutoff, because the student's own prevalence histograms carry no always-on cluster. Diversity metrics follow the acceleration suite on the final decodes. Per-feature timing maps against the base have slope 0.84 with $r = 0.90$ for broad features and slope 0.46 with $r = 0.41$ for specialized features at 32 evaluations, and 0.79 and 0.52 at 16. Advancement on the normalized grid, the base median minus the arm median per feature with at least five events, regressed on the base median gives slope 0.271 with $r = 0.39$ at 32 evaluations and slope 0.154 with $r = 0.23$ at 16, with the late quartile advancing 0.116 against 0.075 for the early quartile at 32 evaluations. Part of the positive slope is structural, since a feature cannot advance earlier than the start of the schedule; the time-warp slope above, which is free of that bound, agrees. The 8-evaluation arm is excluded from this reading because its grid resolves schedule fractions only in eighths. Advancement is also orthogonal to per-feature outcome clarity: across the 323 features with scored clarity cells, its correlations with early and late final-text clarity are $-0.01$ and $-0.10$, so the re-timing moves clocks, not definitions. Table 6 lists the extreme advancers with their activation contexts. At the matched 16-evaluation point every diversity metric is within one point of the base, while at 32 evaluations the repeated-trigram fraction rises from 0.118 to 0.224, and the first readout carries 1,500 to 2,900 active atoms against the base's 246.

**Feature and semantic measurements.** Feature labels use the top 12 activation sites across generations, steps, and positions, with decoded belief context extending eight positions on either side of the activation site. For $A_j = \{(n, i) : \exists s, f^{(n)}_{s,i,j} > 0\}$, duty is

$$D_j = \frac{\sum_{(n,i)\in A_j} \sum_s \mathbf{1}[f^{(n)}_{s,i,j} > 0]}{S|A_j|}.$$

This is temporal persistence conditional on any activation, whereas $\pi_j = |A_j|/(NL)$ measures site prevalence. A specialized feature may therefore have high duty. Activation -step medians and the 25th and 75th percentiles are computed over all positive-activation events of a feature (every

generation, position, and evaluation at which it is active) and divided by $S-1$. Broad and specialized population curves average the commit-aligned activation rates defined in Section 4 and divide by their mean across offsets with nonzero rate. The equal-offset area at or after commitment sums a curve's normalized rates over $\delta \geq 0$ relative to all offsets, weighting offsets equally. It measures the area of the offset-rate curve. The fraction of all activation events after commitment is a different statistic. Final-token classes are assigned to individual decoded token IDs after removing tokenizer space markers, trimming whitespace, and lowercasing. Empty strings and strings composed entirely of ASCII punctuation are classed as punctuation. The fixed function-word lexicon below is checked next. Remaining tokens are content tokens: their frequencies are counts of final token IDs in the generated corpus, and the split is the median of these frequencies over content positions (including repeated occurrences); frequencies equal to or above the median are frequent. This classification is corpus-specific. It is neither a POS tagger nor an external training-corpus frequency rank.

**Prevalence partition.** We exclude zero-prevalence features before taking $\log_{10} \pi_j$. The histogram uses edges from $-6$ to $0.2$ in increments of $0.1$, smoothed by a normalized discrete Gaussian kernel of standard deviation $0.2$ over offsets $-0.6, -0.5, \ldots, 0.6$. The code locates a bulk peak below $-1$ and an upper peak above $-0.6$, then selects the lowest smoothed bin strictly between them. The population analyses use the resulting rounded prevalence cutoffs, $0.18$ and $0.28$. This specifies the fitted partition. Its stability across dictionaries or held-out generations is untested.

**Function-word lexicon.** The exact (whitespace-separated) list is:

> a an the and or but if while although though because since so nor yet for of in on at by to from with without within into onto upon about above below under over between among through during before after behind beyond near off out up down is am are was were be been being do does did doing have has had having will would shall should can could may might must not no nor this that these those i you he she it we they me him her us them my your his its our their mine yours hers ours theirs myself yourself himself herself itself ourselves themselves who whom whose which what when where why how there here then than as such both each few more most other some any all only own same very too also just even still ever never again once

**Readouts and timing conventions.** LangFlow constructs its predicted terminal as the embedding expectation of its DiT posterior. Projecting that terminal back onto the normalized embedding matrix (rows of unit length scaled by $\sqrt{d}$, the matrix the model uses for the expectation) returns the posterior filtered through the embedding Gram matrix: a much sharper distribution whose argmax agrees with the native argmax on $63\%$, $80\%$, and $98.5\%$ of positions at the quarter, middle, and end of generation. All LangFlow commitment measurements therefore use the native posterior. It is recorded for the cache's own $500$ generations by regenerating them with identical seeds and batch order; the reproduction was checked by exact agreement of the final projected tokens. Reruns on a different accelerator diverge within the $128$ evaluations and were discarded, and the endpoint check does not establish intermediate tensor equality, so the cached trajectories are not certified beyond their final tokens. The projection labels cached beliefs with tokens only in the state-versus-belief clock comparison, where both clocks use it. Steering outputs and the acceleration trigger use the full model logits. The 128-evaluation LangFlow sampler records only the first 127 predictions and update inputs; commitment is relative to the last recorded readout. For ELF's SDE cache, the recorded velocity is the drift at the perturbed forward input, not the total displacement between consecutive unperturbed states. Figure 2 and the timing plots divide recorded-step indices by $S-1$.

**Alignment exclusions.** The alignment cutoff $K$ is selected from the specialized population's activation profile. Let $M_{s,j} = \sum_{n,i} f^{(n)}_{s,i,j}$ and $M_j = \sum_s M_{s,j}$ be the activation mass of feature $j$ at evaluation $s$ and over the trajectory, respectively. For the specialized feature set $\mathcal{P}$, the implementation computes

$$A_s = \frac{\sum_{j\in\mathcal{P}} M_j \left(M_{s,j} / \sum_\ell M_{s,\ell}\right)}{\sum_{j\in\mathcal{P}} M_j}, \qquad K = \min\left\{ s : A_s \geq \frac{0.1}{S} \sum_u A_u \right\}.$$

Thus the cutoff uses activation magnitudes. It is $15$ evaluations for LangFlow and one for ELF. Both the activation counts and their support exclude evaluations $s < K$ and sites with $c_{n,i} < K$, retaining

$46{,}350/64{,}000$ LangFlow sites and $508{,}989/512{,}000$ ELF sites. The aligned curves describe these retained sites, excluding $27.6\%$ of LangFlow sites and approximately $0.6\%$ of ELF sites. The exclusion can disproportionately remove early-committing lexical classes; support correction alone does not remove this selection effect. This exclusion applies only to the alignment analysis; token commitment and feature coverage use the full caches.

### A.3 Robustness of the commit statistics

**Why commitment, not a confidence threshold.** A confidence statistic requires a top-1 threshold, and its value depends on the sharpness scale of the readout it is applied to. On LangFlow's native posterior, sweeping the threshold over $\{0.50, 0.90, 0.95, 0.99\}$ moves the mean crossing from $52.9\%$ to $78.2\%$ of the trajectory, with $4\%$ to $28\%$ of positions never crossing; the same thresholds applied to LangFlow's embedding projection place the mean confidence-threshold crossing near $6\%$ of the recorded trajectory. Commitment is threshold-free and moves by less than two points of the trajectory between these readouts, so the paper reports commitment as the primary probe.

**Robustness of the commit criterion.** In the cached analyses, both criteria use a readout of the belief $\hat{\mathbf{x}}_t$. Applying the same stability criterion to the state, with both clocks read through the belief projection, moves the mean commit point from $0.38$ to $0.73$ for LangFlow and from $0.54$ to $0.87$ for ELF, with state-based commitment later on $78\%$ and $94\%$ of positions, respectively. State-based stability therefore tracks convergence of the interpolant rather than changes in the prediction readout. Varying the evaluation grid changes the normalized step fractions. Across $11$ grids (step counts $16$–$256$, uniform and logit-normal placements, both models), the reported commitment coordinates are the more stable quantity. ELF's commitment coordinate spans $t = 0.146$–$0.160$ on the $32/64$-step grids, and LangFlow's $\gamma$ changes from $5.45$ to $5.74$ under refinement. Confidence -threshold crossings also vary with the grid. This empirical sensitivity check does not establish exact grid invariance or isolate the causes of between-model differences.

**Partition-threshold sensitivity.** The populations are split at the density minimum of the log token-rate histogram, at $\pi_j = 0.18$ for LangFlow and $0.28$ for ELF, giving $143$ and $185$ broad features. Restricting the specialized population to the rarest LangFlow features ($\pi_j \leq 10^{-4}$, at most six sites among 64k, 611 features) leaves $80\%$ of its summed support-corrected rates at or after commitment, so the post-commit weighting is not carried by the more common specialized features alone. At the earlier length-$64$ protocol this rarest tail reversed sign; at the released protocol it does not.

### A.4 Dictionary validation

**Held-out cache-consistency regression.** Jointly fitting $(a_t, b_t)$ at each recorded step on half the generations and evaluating on the other half, the two-term linear model explains the velocity with held-out $R^2 = 1.000$ at every step for LangFlow and mean $0.9964$ with minimum $0.9631$ for ELF. These scores use a per-dimension-centered denominator and an intercept in the fitted model. These regressions are empirical cache diagnostics, distinct from the exact sampler identities above. A fresh four-generation, length-$1024$ ELF audit evaluates the analytic identity on the exact forward tensors: all $32$ steps match the implemented float32 drift exactly. Recomputing from float16-stored inputs and predictions gives a maximum stepwise relative $L_2$ residual of $4.88 \times 10^{-4}$. This validates the fresh sampler path but does not isolate the larger historical regression residual. No belief inversion uses fitted coefficients.

**Variance explained by the velocity fit.** Over the full pooled data, the centered linear fit explains $0.985$ of ELF's velocity variance on $16.4$M rows, with the state term alone at $0.77$ and the prediction term alone at $0.14$, and $0.910$ of LangFlow's on $8.1$M rows, with the state term alone at $0.12$ and the prediction term alone at $0.90$. The dominant term therefore differs between the two models, motivating removal of the state component before comparing their predicted-terminal representations.

**Time-resolved reconstruction diagnostics.** We split the cached representations into eight time bins and pre/post-commitment phases. For vectors $z_r \in \mathbb{R}^d$, the uncalibrated reconstruction score is

$$\mathrm{FVE} = 1 - \frac{\sum_r \|z_r - \widehat{z}_r\|_2^2}{\sum_r \|z_r - \overline{\mathbf{z}}_{\mathrm{cell}}\|_2^2},$$

where the cell mean is a vector with one mean per dimension. No gain or offset is fitted to the reconstructions. Counts refer to vectors, not scalar coordinates. For the belief dictionaries, earliest/latest pre-commit FVE is $0.9887/0.9799$ for LangFlow and $0.9555/0.8912$ for ELF; post-commit FVE ranges are $0.9922$–$0.9987$ and $0.9319$–$0.9558$, respectively. Mean positive activation counts across these cells are $63.22$–$64.00$ for LangFlow and $64.00$ for ELF. Scoring the same cells on $256$ fresh sequences per model (seed base $93000$, never used for dictionary training; eight predeclared evaluation sites spanning the recorded schedule and sixteen seeded interior positions per sequence) reproduces every well-conditioned cell with median change $0.006$ and maximum $0.023$; the largest change is a late-trajectory LangFlow cell. The pooled flattened-coordinate diagnostic is $0.985$ for LangFlow and $0.924$ for ELF on fresh sequences, against $0.997$ and $0.955$ on the training cache. The first-evaluation cell is ill-conditioned rather than failed: interior-position beliefs there are nearly constant, with cell variance five orders of magnitude below the final bin, and its negative score is reproduced in-sample at matched positions ($-2.23$ in-sample versus $-2.30$ on fresh sequences for LangFlow; $0.975$ versus $0.976$ for ELF) while its absolute error remains small (per-coordinate MSE $3.8 \times 10^{-5}$). The fresh rows exercise $4{,}820$ LangFlow and $5{,}559$ ELF features at least once, out of $4{,}096$ sampled sites per model, so features rarer than this sampling floor are not expected to appear. These diagnostics show neither an early reconstruction deficit nor a collapse in positive activation count , and their reconstruction scores transfer to fresh sequences. They do not establish uniform feature sensitivity or semantic validity. Gain removal by using $\hat{\mathbf{x}}_t$ does not standardize its norm over time.

A dictionary-free comparison gives a consistent result. Linear CKA on $4096$ paired sites shows that LangFlow's last recorded interpolant shares its covariance structure with the initial noise (CKA $0.99$ at step $0$), whereas the belief's similarity to that reference reaches $0.15$. On ELF, the belief is more similar to the last recorded interpolant than the current state from mid-trajectory at $0.58$ versus $0.33$, reaching $0.72$. These controls use cached forward inputs, not a separately recorded post-sampling endpoint; for ELF they include the SDE input perturbation. Pre-commit rows constitute $36\%$ of LangFlow and $52\%$ of ELF site-step observations.

### A.5 Convergence-curve protocol

**Judged divergence.** The corpora cover every feature with positive activation mass on the $500$-generation analysis caches, $5{,}756$ for LangFlow and $8{,}500$ for ELF. On the cohorts active in all three windows, different-concept rates are $44.2\%$, $9.4\%$, and $6.7\%$ early to late for ELF against floors near $16\%$, $7\%$, and $5\%$, and $13.1\%$, $3.9\%$, and $0.8\%$ for LangFlow against a late floor of $8.4\%$. Each feature is described twice from its activation-ranked context sites, once from the belief readout at the activation step and once from the final decode at the same positions (LangFlow: last recorded native forecast; ELF: cache terminal decode), stream-blind; a blind judge reports same_concept, related, different, or insufficient_evidence. The repeat floor uses two descriptions of a single stream written with independent seeds, separately for both streams. Windows are schedule thirds; a feature enters a window cohort only with at least four sites there. Mechanical and failed generations are excluded by boolean flags. The run comprises $262{,}048$ generations; all pass structural prompt/hash/seed checks, $0.097\%$ failed to finish, and instruction violations of $0.6\%$ are excluded by flag. Writer and judge are one model family, so shared biases cannot be excluded; this is a convergence measurement, not semantic validation of features.

**Clarity.** For a multiset of focal tokens with unit token-direction vectors $u_i$ in the model's own readout space (LangFlow: the posterior-expectation embedding; ELF: decoder-head rows), clarity is $\big(\|\sum_i u_i\|^2 - n\big)/(n(n-1))$, the mean pairwise cosine. Cells use features with at least four scored tokens; per-cell counts for Table 2 are, for the readout stream, LangFlow $717/4695/5006$ early/mid/late and $386/5343$ pre/post, ELF $1072/5414/5958$ and $1800/6077$; for the final-text stream, LangFlow $723/4694/5006$ and $398/5343$, ELF $1162/5432/5959$ and $1854/6077$. Sites are activation-ranked rather than a random sample of activations, and the corpora are drawn from the $500$-generation caches that also trained the dictionaries. These are descriptive association statistics, and the fit evaluation is the held-out reconstruction reported above. Embedding cosine has a small shared floor from mutually similar common words, roughly $0.02$–$0.05$ here; all reported values stand far above it.

### A.6 STEERING AND ACCELERATION DETAILS

**Steering protocol.** Ten content-word targets were fixed before any steering run, with $n = 100$ paired generations per target under shared noise, generated in five batches of twenty. LangFlow LM1B receives direct prediction edits along unit token-embedding directions at zero-based evaluation indices $4 \leq s < 20$ or $108 \leq s < 120$ of its 128-evaluation schedule. The per-step strength is $\alpha_s = \alpha\,\nu_s$ as in Equation (8), for $\alpha \in \{0.2, 0.5, 1\}$. The modified prediction enters both the integration update and the next self-conditioning input; the early and late windows contain 16 and 12 edits, respectively. Pooled target occurrences are 36, 53, and 151 (early) and 33, 45, and 130 (late) against 30 unsteered. At $\alpha = 1$ the ratios to baseline are $5.0$ with a $95\%$ interval of 3.7–6.4, and $4.3$ with 3.8–5.2, from $10{,}000$ paired bootstrap resamples of the five generation batches, jointly across the fixed targets. The early count exceeds the late count for 2 of ten targets. The norm-scaled dose couples samples within each batch; with only five resampling units, these intervals are fragile. Mean token change rates are $0.38$ (early) and $0.14$ (late) at $\alpha = 1$. Random-direction controls draw each direction from an independent generator. The earlier absolute-strength protocol ($\alpha_s = \alpha \in \{1, 2, 5\}$ embedding units, the same windows) gave 164 early and 33 late occurrences at $\alpha = 5$. Mean belief norms are approximately 3–7 early and 25 late, so this absolute edit has a smaller relative magnitude late. The difference between protocols shows sensitivity to the strength convention, without isolating belief scale from the other differences between windows.

For ELF, we steer 60 dictionary features at sequence length $1024$ and compare with 30 random-direction controls. Selection uses the 30 highest-usage and 30 context-selected exemplars from the activation analysis; under the density-minimum cutoff the former are all broad and the latter all specialized. Neither selection is a representative sample of its population. A label containing a function word does not determine population membership: membership is defined by the measured site prevalence $\pi_j$. Labels summarize activation contexts. They do not name exclusive concepts. ELF edits enter the next self-conditioning prediction after the current state update, rather than directly modifying that update. The prediction hook uses $\alpha_s = \alpha\,\nu_s$ as for LangFlow, $\alpha \in \{0.2, 0.5, 1\}$, 16 paired generations per feature and strength, and zero-based windows $1 \leq s < 5$ or $27 \leq s < 30$ of the 32-evaluation schedule. The evaluated string is the most frequent marked token string in the feature's saved activation contexts, stripped of surrounding whitespace. Occurrences use case-sensitive, nonoverlapping substring counts in decoded text, without a word-boundary constraint. Thus these counts are lexical proxies and may include occurrences inside longer words; they are not independently validated conceptual applicability labels. For the selected specialized features the pooled counts are $1{,}033$, $2{,}542$, and $10{,}116$ (early) and $692$, $1{,}189$, and $11{,}258$ (late) against $673$ unsteered, with increases for 15, 20, 21 and 10, 20, 20 of 30 features. Mean token change rates are $0.80$, $0.91$, $0.94$ early and $0.02$, $0.05$, $0.15$ late; early random-direction rates are $0.77$, $0.88$, $0.93$. The repeated-trigram fraction is computed on whitespace-tokenized text and averaged over generations for each direction. Its median across the selected specialized features is $0.31$ early and $0.09$ late at $\alpha = 1$, against $0.0912$ unsteered. For the 28 broad features with nonempty labels, pooled labeled-string counts increase by $34\%$ early and $102\%$ late at $\alpha = 1$, but only 12 and 7 features show increases. The pooled count therefore does not describe a uniform response across broad features.

**Acceleration accounting.** Denoiser-evaluation ratios use each run's recorded full-schedule count as the numerator. LangFlow uses $100$ paired generations from the LM1B checkpoint at length $128$ with $128$ evaluations. Generated tokens are decoded with the BERT tokenizer, keeping special tokens as LangFlow's own generative-perplexity script does, and retokenized for GPT-2 scoring. The online stop requires three consecutive unchanged comparisons, corresponding to four identical consecutive native-logit argmax sequences, and emits the current tokens without another forward. If the rule never fires, the full schedule and final readout are completed. ELF uses $48$ paired generations at length $512$, in batches of four with seeds $3000$ plus the batch's starting sequence index. Its shortened grid is $(t_0, \ldots, t_{23}, t_{32})$: 23 baseline intervals followed by one enlarged terminal step, with a decoder pass on the resulting state. The evaluation ratio is $32/24 = 1.33$. No wall-clock ratio is measured. A matched-budget baseline regenerates the official sampling grid with $24$ intervals and uses the same initial seeds and batch layout; it is labeled uniform respacing in the saved artifacts. Agreement and quality scores use these same paired outputs. For ELF, decoded text excludes special tokens and is retokenized for GPT-2 with a maximum of $1024$ tokens; mean log probability changes from $-3.125$ to $-3.306$ nats per token. Beyond agreement, every arm is scored against its paired full-schedule outputs on repetition, distinct-$n$ (Li et al., 2016), marginal divergence, and MAUVE (Pillutla et al.,

2021). Unigram repetition is $0.281 \rightarrow 0.292$ for LangFlow stopping and $0.613 \rightarrow 0.604$ for ELF truncation and $0.613 \rightarrow 0.590$ for uniform respacing; distinct-1 is $0.282 \rightarrow 0.272$, $0.138 \rightarrow 0.141$, and $0.138 \rightarrow 0.150$. The ELF divergences use disjoint paired-sequence halves: unigram JSD is $0.222$ (truncation) and $0.226$ (uniform) against a split-half floor of $0.225 \pm 0.006$, and bigram JSD is $0.657$ and $0.662$ against $0.657$; both rules sit at the floor, a descriptive statement about marginal token distributions rather than paired fidelity. LangFlow's suite records no split-half floor; its unigram JSD is $0.018$. MAUVE is $0.9996$, $0.9992$, and $0.9974$ at $n = 100$ and $n = 48$ generations per side, sample sizes at which MAUVE estimates are noisy. All GPT-2 scores in this subsection use the base GPT-2 model on retokenized text, distinct from the GPT-2-large scorer used for the distillation tables; perplexity transforms of these log probabilities are therefore not comparable across subsections.

**Paired agreement uncertainty.** Percentile bootstrap intervals use $10{,}000$ resamples of generation seed groups: $100$ individual LangFlow generations, and $12$ ELF batches of four generations. The $95\%$ intervals are $[0.924, 0.953]$ for LangFlow stopping, $[0.850, 0.867]$ for ELF fixed truncation, and $[0.467, 0.508]$ for ELF uniform respacing. Exact-sequence agreement is $40/100$ for LangFlow, with a generation-bootstrap $95\%$ interval of $[0.30, 0.50]$, and $0/48$ for both ELF schedules. The all-zero ELF empirical bootstrap is degenerate, so we report its observed count without treating it as a bound on population reliability. These intervals quantify agreement with the saved baseline outputs, not general generation quality. The preserved schedule prefix shares stochastic updates with the full ELF run; reusing a seed across different grids does not couple SDE increments at matched times. No corresponding uncertainty estimate is claimed for the evaluation ratio or GPT-2 log probability.

**Trigger and decoder timing.** Separate component benchmarks use synchronized repetitions on one GPU at batch size 1. LangFlow's forward pass already computes the vocabulary logits. The benchmark measures the additional argmax and sequence-equality check on those logits; the forward timing includes logit production, while the isolated check is not a measurement of the full stopping loop. This check costs $0.034$ ms, or $0.46\%$ of a $7.5$ ms model step at length $128$, using $200$ repetitions. ELF's decoder costs $7.3$ ms over $50$ repetitions at length $1024$, compared with an $8.0$ ms denoiser evaluation over $200$ repetitions. This decoder cost is $3.8\%$ of $24$ denoiser evaluations in that benchmark, not an end-to-end overhead measurement for the length-$512$ quality experiment.

### A.7 Additional revision controls

**Common-support alignment.** With the original dictionaries and partitions fixed, we recompute alignment without the initial-site exclusion and compare each eligible site with itself over offsets $-w, \ldots, -1$ and $1, \ldots, w$. Offset zero is excluded from this paired contrast. Table 7 reports the change in the number of positive specialized features per site-step, with $1000$ generation-bootstrap resamples conditional on the fitted dictionary. All-site equal-offset area at or after commitment is $85.9\%$ for LangFlow and $75.9\%$ for ELF, compared with $84.2\%$ and $74.2\%$ under the original cutoffs. Sites without both windows cannot enter the paired analysis.

Table 7: **Same-site alignment controls.** The placebo permutes commitment within generation and position quartile, without matching lexical class. Its positive contrast shows that an overall time trend contributes to the aligned rise; neither contrast identifies a causal commitment effect.

| Model | $w$ | Eligible sites | Paired change (95% interval) | Placebo |
|---|---|---|---|---|
| LangFlow | 4 | 50,749 | $0.952$ $[0.923, 0.984]$ | $0.570$ |
| ELF | 2 | 490,602 | $4.922$ $[4.860, 4.981]$ | $2.858$ |

Because TopK nearly fixes the total positive count, the broad/specialized changes largely describe redistribution within the sparse code. These controls retain the historical LangFlow readout pairing and do not resolve its intermediate cache-alignment limitation.

**Fresh endpoint-inclusive recording.** A separate 32-sequence LangFlow run (eight seeds, batches of four) jointly records all $128$ native readouts and belief snapshots at sixteen positions. The final output evaluation changes the retrospective commitment index at $1.81\%$ of positions compared with the first $127$ readouts. This run is a fresh endpoint sensitivity check. It does not reproduce or replace the historical $500$-sequence analysis. Replaying the three-comparison trigger uses $118.6$ evaluations

on average at $0.964$ token agreement in this small sample; these descriptive values are not a new calibrated acceleration benchmark.

### A.8 Scope

**Scope of each result.** Commit timing and the readout-robustness comparison are evaluated on LangFlow and ELF. Sparse-dictionary analyses use LangFlow and ELF. Word-targeted steering uses LangFlow. Inference acceleration uses an online trigger for LangFlow and a static shortened schedule for ELF. Endpoint self-distillation uses ELF.

## B Exploratory endpoint self-distillation

### B.1 Objective

The objective and the main recipe appear in Section 6; the endpoint term decodes the student's intermediate belief and supervises it against the teacher's decoded endpoint tokens. This appendix records the training settings, the teacher-budget variant, the quality ladder against progressive distillation, and the strength check.

### B.2 Recipe

The unconditional SD student is fine-tuned from the released base checkpoint for $12{,}000$ steps with the model's original recipe plus the endpoint term of Equation (10) ($\lambda_{\text{sd}} = 0.005$, effective batch $512$, EMA $0.9999$). The recipe uses Muon with auxiliary Adam, a resolved learning rate of $0.002$, and $300$ warmup updates. Each microbatch contains $16$ data sequences and $8$ cached trajectory records, with gradients accumulated over $32$ microbatches.

The main ladder uses $1000$ full $32$-step teacher trajectories, with four intermediate states and their self-conditioning context stored per trajectory. The student prediction is decoded by a separate decoder-mode forward at terminal time, with gradients through both forwards. Separate teacher-budget experiments use a $64$-step teacher truncated at $75\%$, whose target tokens are decoded from the prediction at the cut; this variant is not the teacher used for the main ladder. All cells of Table 8 were evaluated in one job on a single node because generative-perplexity offsets across GPU families are checkpoint-dependent.

### B.3 Results

Inference truncation changes how an existing model is sampled. Endpoint self-distillation instead tests whether the model can learn to predict its own outcomes with fewer steps. Table 8 compares the base ELF checkpoint, endpoint self-distillation (SD), and the released progressive-distillation checkpoint (PD) (Qiu et al., 2026), using $1000$ generations per cell. The $32$-evaluation SD cell improves perplexity to $18.1$ while doubling the repeated-trigram rate; we read that arm as overfitting, and the main text keeps the matched $16$-evaluation operating point.

At $8$, $16$, and $32$ steps, SD reaches generative perplexities of $40.5$, $24.7$, and $18.1$, close to the base model's $38.6$, $24.3$, and $17.2$ at twice the budget; entropies are likewise close but not identical. SD achieves this without requiring token agreement with a particular teacher trajectory. PD has lower default-setting perplexity at $8$ and $16$ steps, together with lower entropy. At a nearby entropy of $5.16$, the $16$-step comparison favors SD at $24.36$ with entropy $5.164$ and guidance $6.0$, over PD at $29.16$ with entropy $5.162$ and guidance $1.5$. This operating-point comparison does not establish dominance over the full guidance frontier.

Table 8: **ELF generation quality across sampling budgets.** Endpoint self-distillation approaches the base model's perplexity at twice the sampling budget; progressive distillation is stronger at $T \leq 16$ but at lower entropy. Cells are GPT-2-large perplexity / token entropy; sampling settings are in the text.

| **Model** | $T$=8 | $T$=16 | $T$=32 | $T$=64 |
|---|---|---|---|---|
| ELF base | 84.8 / 5.36 | 38.6 / 5.25 | 24.3 / 5.16 | **17.2** / 5.05 |
| +SD (ours) | 40.5 / 5.29 | 24.7 / 5.18 | **18.1** / 5.09 | — |
| +PD (Qiu et al., 2026) | **22.6** / 5.06 | **22.0** / 5.06 | 21.7 / 5.05 | — |