

# IMPROVING FEW-STEP LANGUAGE FLOWS WITH UNTIED SELF-CONDITIONING

Bocheng Li<sup>1,2</sup>      Linli Xu<sup>1,2\*</sup>

<sup>1</sup>University of Science and Technology of China

<sup>2</sup>State Key Laboratory of Cognitive Intelligence

bcli@mail.ustc.edu.cn      linlixu@ustc.edu.cn

## ABSTRACT

Flow-matching language models refine all token positions in parallel and can trade sampling steps for latency, yet generation quality still degrades sharply with few sampling steps. We trace a source of this degradation to a train–inference mismatch in previous-prediction self-conditioning: during training, the self-conditioning input is computed from the current noisy state with no intervening solver step; during sampling, the solver folds the previous prediction into the latent before that same prediction reappears as the explicit self-conditioning input. This coupling, absent during training, creates redundancy that grows with step width. We show that the mismatch degrades both the self-conditioning input and the solver update, and derive a correction for each from the model’s own structure. From the frozen projection weights we identify directions along which the self-conditioning input is redundant with the latent and dampen them; from the solver’s integration structure we derive that a step-average prediction is needed and approximate it from prediction history, with scale set by offline trajectory statistics. The resulting sampler, Untied Self-Conditioning, requires no retraining and uses one evaluation per step. At 8 sampling steps on LangFlow, it reduces OpenWebText generative perplexity from 531 to 62 ( $8.6\times$ ); under an adapted Arena-Hard-Auto v2 protocol, its outputs are preferred in 96% of pairwise comparisons. On ELF-B it reduces generative perplexity from 71 to 43. Improvements hold from 8 to 256 sampling steps.

## 1 INTRODUCTION

Flow matching (Lipman et al., 2023) has enabled high-quality parallel generation in continuous domains, with recent progress extending to text (Li et al., 2022; Gong et al., 2023; Strudel et al., 2022; Gulrajani & Hashimoto, 2023; Hu et al., 2024; Chen et al., 2026; Hu et al., 2026). These models embed discrete tokens in a continuous latent space and refine all positions simultaneously, approaching autoregressive quality while admitting variable-cost sampling: fewer iterative steps directly reduce latency (Chen et al., 2026; Hu et al., 2026). Making each model evaluation as informative as possible is the central challenge for few-step generation.

A standard technique in these models is previous-prediction self-conditioning (Chen et al., 2023): at each step, the model receives its own clean prediction from the preceding step as additional input (Strudel et al., 2022; Karimi Mahabadi et al., 2024; Gulrajani & Hashimoto, 2023; Yuan et al., 2024; Liu et al., 2024a; Hu et al., 2026). Self-conditioning consistently improves generation quality, yet it introduces a train–inference mismatch: during sampling, the solver folds the previous prediction into the latent before that same prediction reappears as the explicit self-conditioning input, creating a coupling between the two input pathways that is absent during training and grows with step width. Prior work has studied train–inference discrepancies in the self-conditioning recurrence (Shabalin et al., 2025; Liu et al., 2024b; Cong et al., 2026) and proposed inference-time modifications to individual components of the sampling loop (Zhang et al., 2026; Lu et al., 2025; Sadat et al., 2026), but has not diagnosed the specific mechanism by which this coupling degrades the self-conditioning input or the solver’s own update.

---

\*Corresponding author.

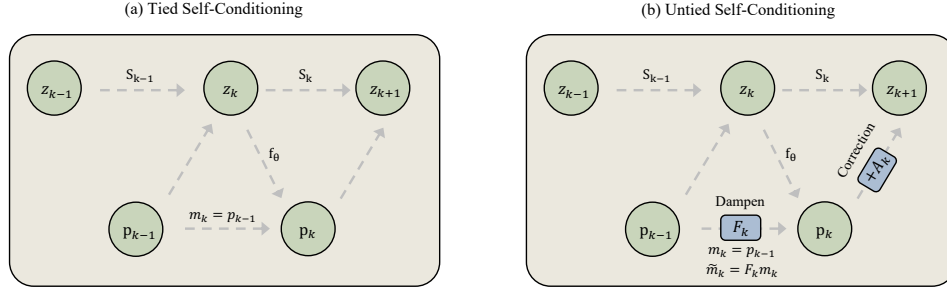


Figure 1: **Train-inference mismatch and its correction.** (a) Standard self-conditioning: the solver folds  $p_{k-1}$  into the latent  $z_k$ , so  $z_k$  already encodes  $p_{k-1}$ . Then  $m_k = p_{k-1}$  re-introduces the same information as an explicit input. During training, no solver step intervenes, so this redundancy does not arise. (b) Untied Self-Conditioning corrects both consequences: an overlap transform  $F_k$  dampens self-conditioning components the latent already represents, while a solver correction  $A_k$  shifts the prediction toward the step average. The raw prediction is stored for the next self-conditioning input.

In this work, we show that the solver-introduced coupling degrades few-step generation through two specific mechanisms, and derive a correction for each from the model’s own structure. First, the solver folds the previous prediction into the latent, so the explicit self-conditioning input re-introduces information the latent already carries; we show from the model’s frozen projection weights that this redundancy concentrates in directions shared by both input pathways. A controlled experiment confirms the causal role of coupling: varying only the dependence between successive steps while keeping each step’s marginal distribution fixed, the benefit of the self-conditioning input shrinks monotonically, and at strong coupling full self-conditioning becomes harmful (Section 5.3). Second, the single-point prediction at the start of each step departs from the average prediction needed for the exact solver update; we show that this gap is first order in step width and can be approximated from prediction history. Both effects compound in the few-step regime, explaining the sharp quality loss.

We introduce Untied Self-Conditioning, a training-free sampler that corrects both mechanisms with one model evaluation per step. For the self-conditioning input, an eigendecomposition of the frozen projection weights identifies the directions along which the input is redundant with the latent; a transform dampens those directions so the explicit input carries complementary information. For the solver, prediction history approximates the step-average prediction, with scale set by offline trajectory statistics. The corrected prediction enters only the solver; the raw prediction is stored for self-conditioning.

Across ELF and LangFlow, Untied Self-Conditioning improves generation quality in every tested configuration. At eight model evaluations it reduces LangFlow GenPPL on OpenWebText from 531 to 62 (8.6 $\times$ ) and on ELF-B from 71 to 43. At 32 evaluations on ELF-B it achieves a GenPPL of 21, approaching the data reference, with no retraining and 95% throughput retained.

Our contributions are:

- We show that self-conditioning can become harmful at strong coupling, even when each step’s prediction quality is unchanged. The cause is a train-inference mismatch: the solver couples the latent and self-conditioning pathways in a way that is absent during training. A controlled experiment confirms the causal role by varying only the coupling while preserving marginals.
- We derive the geometric structure of the resulting redundancy from the model’s frozen projection weights, and the solver’s approximation gap from its integration structure; each derivation yields a specific correction objective.
- Untied Self-Conditioning implements both corrections in a sampler that requires no retraining and uses one model evaluation per step. It improves generation quality at every tested NFE, with the largest gains in the few-step regime.

Section 2 reviews the sampling recurrence. Section 3 diagnoses the mismatch and derives a correction objective for each consequence. Section 4 assembles the complete sampler. Section 5 presents controlled experiments and generation results.

## 2 PRELIMINARIES

A language flow embeds discrete tokens into a continuous space and learns a velocity field  $v_\theta(z, t)$  that transports noise to clean data. Sampling integrates the induced ODE from a noise sample  $z_0$ ; in practice, a numerical solver discretizes the integral over a time grid  $0=t_0 < \dots < t_K=1$ . At each step  $k$ , the model produces a clean prediction  $p_k$  from which the solver computes the latent update. We report the total number of frozen-model evaluations as NFE.

Previous-prediction self-conditioning (Chen et al., 2023) supplies the model with its own clean prediction from the preceding step as an additional input. The sampling recurrence is

$$p_k = f_\theta(z_k, m_k, t_k), \quad z_{k+1} = S_k(z_k, p_k), \quad m_{k+1} = p_k, \quad (1)$$

where  $m_k$  is the *self-conditioning input*: the previous clean prediction carried forward from step  $k-1$ . During training, the model receives  $m = \text{sg}[f_\theta(z, 0, t)]$  computed from the current noisy state for a random half of examples, and zero otherwise (Chen et al., 2023; Strudel et al., 2022; Karimi Mahabadi et al., 2024; Gulrajani & Hashimoto, 2023; Yuan et al., 2024; Liu et al., 2024a; Hu et al., 2026). During sampling,  $m_k=p_{k-1}$  while  $z_k$  has already been advanced with  $p_{k-1}$ . The intervening solver step is the source of cross-step correlation analyzed in Section 3.1.

## 3 SOLVER COUPLING CREATES A TRAIN-INFERERENCE MISMATCH

During training, the model receives the current noisy state  $z$  and a self-conditioning input  $m$  computed from that same state through a detached forward pass. No solver step intervenes between  $z$  and  $m$ , so no earlier prediction has been folded into  $z$  by the solver. During sampling, the situation is different. The solver updates  $z_{k+1}$  as a weighted combination of  $z_k$  and  $p_k$ , folding the prediction directly into the latent. Then the next evaluation sees  $m_{k+1} = p_k$  as an explicit input alongside that updated latent. The model therefore receives  $p_k$  twice: once absorbed into  $z_{k+1}$  through the solver, once passed directly as  $m_{k+1}$ . This coupling between the two input pathways is absent during training and grows with step width.

The mismatch has two consequences. First, the self-conditioning input partly repeats what the latent already encodes; the redundancy concentrates in directions that both input projections share and can be quantified from the model’s frozen weights. Second, the solver evaluates the model once at the start of each step, but accurate integration requires the average prediction over the step; the gap grows at low step counts. Section 3.1 establishes the causal role of the coupling. Sections 3.2 and 3.3 derive a correction objective for each consequence.

### 3.1 CROSS-STEP CORRELATION REDUCES THE VALUE OF SELF-CONDITIONING

To confirm that the coupling itself, not the quality of individual predictions, drives the degradation, we first isolate its effect in a scalar Gaussian model. The useful coefficient on the self-conditioning input depends on how much of its information the latent already carries. Let  $Y \sim \mathcal{N}(0, \tau^2)$  be a clean target observed through two noisy channels:

$$Z = Y + \sigma_z \epsilon_z, \quad M = Y + \sigma_m \epsilon_m, \quad \text{Corr}(\epsilon_z, \epsilon_m) = \rho, \quad (2)$$

where  $\epsilon_z, \epsilon_m$  are zero-mean, unit-variance, jointly Gaussian, and independent of  $Y$ . The correlation  $\rho$  controls how much noise the two channels share, analogous to the solver coupling  $p_{k-1}$  into both  $z_k$  and  $m_k$ . Varying  $\rho$  leaves both marginals  $P(Y, Z)$  and  $P(Y, M)$  unchanged. Only the joint  $P(Z, M)$  changes.

The coefficient on  $M$  that minimizes prediction error decreases monotonically with  $\rho$ , crossing zero at a critical correlation that depends on the noise levels (Appendix A.1). Beyond this crossing, a fixed positive coefficient on  $M$  becomes harmful. So changing only the coupling between inputs can change whether the self-conditioning input helps or hurts prediction.

The sampling recurrence has this structure. The prediction  $p_{k-1}$  appears explicitly in  $m_k$  and also affects  $z_k$  through the solver. Section 5.3 confirms the predicted monotone decrease on a frozen language model.

### 3.2 THE SELF-CONDITIONING INPUT SHOULD REDUCE ITS OVERLAP WITH THE LATENT

Given that the coupling creates redundancy between the two input pathways, we now derive its geometric structure from the model’s frozen projection weights. The redundancy concentrates in directions that both input projections share; this structure determines which components of the self-conditioning input to dampen.

In self-conditioned flow-matching language models, the latent and self-conditioning input enter the network through a single learned projection that operates on their concatenation. This is equivalent to two linear maps whose outputs are summed. Let  $P_z$  and  $P_m$  project onto the row spaces of these effective projection matrices. On the self-conditioning subspace, define

$$K = P_m P_z P_m. \quad (3)$$

The eigenvectors  $q_i$  of  $K$  are the principal directions shared by the two row spaces, and the eigenvalues  $\rho_i^2 \in [0, 1]$  are the squared cosines of their principal angles. Concretely, a large  $\rho_i^2$  means that both the latent projection and the self-conditioning projection respond strongly to the same input component: any signal the self-conditioning input carries along that direction is already represented in the latent. Along such directions the explicit input is redundant.

For any self-conditioning input  $m$ , the weighted projection onto these shared directions is

$$\mathcal{E}(m) = \frac{1}{2} \|P_z P_m m\|^2 = \frac{1}{2} \langle m, K m \rangle. \quad (4)$$

This quantity is our overlap objective: smaller  $\mathcal{E}$  means less redundancy with the latent pathway.

The objective  $\mathcal{E}$  suggests a natural family of transforms: dampen  $m$  along high-overlap directions and leave the rest unchanged. The map  $(I - aK)$  with  $0 < a \leq 1$  does exactly this. It retains fraction  $1 - a\rho_i^2$  along direction  $q_i$ , so directions shared by both projections ( $\rho_i$  large) are dampened most, while directions unique to self-conditioning pass through. Section 4.1 instantiates this family.

### 3.3 THE SOLVER SHOULD APPROXIMATE THE STEP-AVERAGE PREDICTION

The second consequence of the mismatch concerns the solver itself. The solver evaluates the model once at the start of each step  $[t_k, t_{k+1}]$ , but the true velocity field changes throughout the step. The exact update requires a weighted average  $\bar{p}_k$  of predictions over the step interval (Appendix A.3); a single evaluation provides only the initial point  $p_k$ .

The gap  $C_k^* = \bar{p}_k - p_k$  is first order in step width  $h_k$ , and its contribution to the latent update is second order in  $h_k$ : negligible with many steps but substantial at low NFE. Appendix C.7 verifies these orders on high-resolution trajectories.

Computing  $\bar{p}_k$  exactly requires multiple evaluations per step. A weaker condition suffices: if a nonzero direction  $B_k$  is applied with scale  $\eta > 0$ , the change in squared correction error is

$$\|C_k^* - \eta B_k\|^2 - \|C_k^*\|^2 = \eta^2 \|B_k\|^2 - 2\eta \langle C_k^*, B_k \rangle. \quad (5)$$

The endpoint error is proportional to this quantity (Appendix A.3), so any direction  $B_k$  positively correlated with  $C_k^*$  improves the update for a range of scales  $\eta$ .

Computing  $\bar{p}_k$  from past predictions avoids extra evaluations. An exponential moving average tracks recent output:

$$E_{k+1} = (1 - \alpha)E_k + \alpha p_k, \quad E_0 = 0. \quad (6)$$

The residual  $H_k = p_k - E_k$  estimates recent change in prediction and serves as the primary correction direction. Section 4.2 refines it with a second direction from solver-weighted history, and normalizes the result using offline trajectory statistics. Appendix C.7 evaluates the resulting direction against high-resolution step averages.

## 4 UNTIED SELF-CONDITIONING

Section 3 diagnosed a train–inference mismatch with two consequences: the self-conditioning input carries redundancy that concentrates in specific directions of the projection geometry (Section 3.2), and the solver’s single-point evaluation departs from the step-average prediction (Section 3.3). We now implement a correction for each within a single sampling loop.

A transform derived from the overlap structure dampens the self-conditioning input along redundant directions, and a history-based correction shifts the solver toward its step average. Both share one model evaluation per step; offline trajectory statistics set the correction’s mean and scale.

#### 4.1 SELF-CONDITIONING TRANSFORM

The goal is to reduce  $\mathcal{E}(m)$  from Section 3.2 at each step. Principal directions  $Q$  and squared cosines  $\rho_i^2$  are computed once from frozen weights; the transform retains fraction  $\ell_{k,i}$  along direction  $q_i$ :

$$\begin{aligned} F_k &= I - Q(I - \text{Diag}(\ell_{k,1}, \dots, \ell_{k,r}))Q^\top, \\ \widehat{m}_k &= F_k m_k. \end{aligned} \quad (7)$$

Directions with larger  $\rho_i^2$  receive stronger dampening, following the  $(I - aK)$  form that reduces the overlap  $\mathcal{E}$  at every step (Section 3.2). The base exponent  $\lambda_R^{(0)}$  is derived from the model’s response geometry (Appendix C.4);  $\eta_R \geq 0$  scales it, giving effective exponent  $\lambda_R = \lambda_R^{(0)} \eta_R$ . The dampening strength adapts across steps: it is strongest when the solver has folded a large fraction of the prior prediction into the latent, and weakens when the raw and corrected running averages disagree (Section 4.3). At the first step no prior prediction exists, so the input passes unchanged. Appendix B.1 gives the full retention schedule.

#### 4.2 SOLVER-STEP CORRECTION

**Solver-weighted history.** Each solver step  $S_k(z_k, q_k)$  depends linearly on the prediction  $q_k$  it receives (before correction  $q_k = p_k$ ; with correction  $q_k = p_k + A_k$ , see Section 4.3). We track the running weighted average of predictions that entered the solver:

$$O_{k+1} = (1 - \kappa_k) O_k + \kappa_k q_k, \quad O_0 = 0, \quad (8)$$

where  $\kappa_k \in [0, 1]$  is the solver’s update weight (Appendix A.2).

**Correction direction.** We construct a direction that correlates with  $C_k^*$  from two signals available in the current trajectory. The first is the EMA residual  $H_k = p_k - E_k$  from Section 3.3, which captures recent prediction change. The second compares the current prediction with the running average  $O_k$ :

$$G_k = \begin{cases} \|H_k\| \frac{p_k - O_k}{\|p_k - O_k\|} & \text{if } \|H_k\|, \|p_k - O_k\| > \epsilon, \\ 0 & \text{otherwise} \end{cases}. \quad (9)$$

The combined correction direction is

$$U_k = H_k + \eta_O \Delta u_k (H_k - G_k), \quad (10)$$

where  $\Delta u_k \in [0, 1]$  is a normalized step-width feature (Appendix B.3) and  $\eta_O \geq 0$  weights the disagreement between the two estimates. The two estimates reflect different prediction streams:  $E_k$  accumulates the raw prediction  $p_k$  that enters self-conditioning, while  $O_k$  accumulates the corrected prediction  $p_k + A_k$  that enters the solver. Their per-token cosine distance  $\bar{\delta}_k$  quantifies cross-pathway agreement and also gates the self-conditioning dampening strength (Section 4.3).

**Normalizing with offline statistics.** The direction  $U_k$  comes from the current trajectory, but its scale is arbitrary. We normalize it to match statistics precomputed from stored trajectories: a per-step token mean  $\mu_k$  and centered RMS  $\sigma_k$ , estimated on disjoint trajectory banks (Appendix B.4). Let  $C_{\text{tok}}$  center a tensor by subtracting its per-position mean. The normalized correction is

$$\begin{aligned} \Pi_k(U_k) &= \mu_k + \sigma_k \frac{C_{\text{tok}} U_k}{\text{RMS}(C_{\text{tok}} U_k)}, \\ A_k &= \eta_T \Pi_k(U_k). \end{aligned} \quad (11)$$

This replaces the scale and mean of  $U_k$  while preserving its centered direction (Appendix B.5). Appendix C.7 evaluates the resulting direction against high-resolution step averages.

### 4.3 STATE UPDATE

One evaluation produces  $p_k = f_\theta(z_k, \tilde{m}_k, t_k)$ ; each correction applies to its own pathway:

$$\begin{aligned} z_{k+1} &= S_k(z_k, p_k + A_k), & m_{k+1} &= p_k, \\ O_{k+1} &= (1 - \kappa_k)O_k + \kappa_k(p_k + A_k), & E_{k+1} &= (1 - \alpha)E_k + \alpha p_k. \end{aligned} \quad (12)$$

The raw  $p_k$  updates self-conditioning and the EMA; the corrected  $p_k + A_k$  advances the solver and its state. During training the model receives raw predictions as self-conditioning input; routing the correction there would create a second mismatch (Section 5.5).

The untied routing produces two running averages,  $E_k$  from the raw pathway and  $O_k$  from the corrected pathway, whose agreement connects both corrections. The self-conditioning dampening adapts via

$$a_k = \kappa_{k-1}(1 - \bar{\delta}_{k-1}), \quad (13)$$

where  $\bar{\delta}_{k-1}$  is the cosine distance between  $H_{k-1}$  and  $G_{k-1}$ . When the estimates agree, dampening is strong; when they disagree, it relaxes ( $a_0=0$ ). Algorithm 1 summarizes the loop.

## 5 EXPERIMENTS

We first show that Untied Self-Conditioning improves generation quality at every tested NFE, then verify the diagnosed mismatch and each derived correction through controlled experiments that isolate cross-step coupling, overlap-aligned directions, and correction placement.

### 5.1 EXPERIMENTAL SETUP

**Data and generation.** ELF-B, ELF-M, and ELF-L are evaluated on OpenWebText with NFE ranging from 8 to 64. LangFlow is evaluated on OpenWebText and LM1B with NFE from 8 to 256. OpenWebText generations contain 1,024 tokens; LangFlow LM1B generations contain 128.

**Evaluation.** We report generative perplexity (GenPPL) under GPT-2 Large and unigram entropy  $H$ . GenPPL measures the quality of generated text;  $H$  measures diversity. Section 5.2 additionally reports pairwise LLM-judge preference and per-sample external-evaluator likelihood as complementary quality signals. All main results average six seeds, each with 1,024 generations; controlled studies use smaller sample counts noted in each subsection. Appendix C.1 gives the full metrics, uncertainty units, and protocol.

**Matched conditions.** Every comparison keeps the pretrained model, base solver, time grid, tokenizer, and decoding procedure fixed. Matched configurations share the initial latent, time grid, and random stream. The overlap basis and offline trajectory statistics are computed once from frozen model weights and shared across all configurations.

### 5.2 GENERATION QUALITY IMPROVES AT EVERY TESTED NFE

Untied Self-Conditioning improves generation quality relative to the official sampler in every tested configuration (Tables 3, 4). At 8 NFE, Untied Self-Conditioning reduces LangFlow GenPPL on OpenWebText from 531.2 to 61.6 ( $8.6\times$ ) and on LM1B from 264.0 to 73.0 ( $3.6\times$ ). On ELF-B at 8 NFE it reaches 42.66 versus the official sampler’s 70.63. At 32 NFE on ELF-B it reaches 21.0 without additional training. Figure 3(a) places the results in system-level context.

Figure 2 annotates unigram entropy at every matched point: on LangFlow, Untied Self-Conditioning moves entropy closer to the corpus reference; on ELF, entropy is preserved.

At 8 and 16 NFE on LangFlow, every one of 6,144 paired samples has lower GenPPL (Appendix C.12). On ELF-B at 8 NFE, varying each coefficient independently over a  $[\times\frac{1}{2}, \times 2]$  range always improves GenPPL (Appendix C.11). Across six workloads the sampler retains 95.5% geometric-mean throughput (Appendix C.10).

**Algorithm 1** Untied Self-Conditioning sampling. The corrected prediction updates the solver states  $(z, O)$ ; the raw prediction updates the self-conditioning states  $(m, E)$ .

**Input:** frozen model  $f_\theta$ , base solver  $S_k$ , time grid  $\{t_k\}_{k=0}^K$

**Precomputed:** overlap basis  $Q$ , cosines  $\{\rho_i^2\}$ , base transform exponent  $\lambda_R^{(0)}$ , cosine distance  $\bar{\delta}_k$ , per-step correction mean  $\mu_k$  and centered RMS  $\sigma_k$

**Hyperparameters:**  $\eta_R, \eta_T, \eta_O$  (nonnegative strength coefficients);  $\alpha \in (0, 1]$  (EMA weight on the current prediction)

**Initialize:**  $m_0, E_0, O_0 \leftarrow 0$ ;  $\kappa_{-1} \leftarrow 0$ ,  $\bar{\delta}_{-1} \leftarrow 0$

**for**  $k = 0, \dots, K-1$  **do**

1.  $\tilde{m}_k \leftarrow F_k m_k$  *self-conditioning transform*
2.  $p_k \leftarrow f_\theta(z_k, \tilde{m}_k, t_k)$  *one model evaluation*
3.  $H_k \leftarrow p_k - E_k$ ;  $G_k \leftarrow \|H_k\| / \|p_k - O_k\|$ ; construct  $U_k$  by equation 10
4. Compute  $A_k$  by equation 11 *solver correction*
5.  $z_{k+1} \leftarrow S_k(z_k, p_k + A_k)$ ;  $O_{k+1} \leftarrow (1 - \kappa_k)O_k + \kappa_k(p_k + A_k)$  *corrected  $\rightarrow$  solver*
6.  $m_{k+1} \leftarrow p_k$ ;  $E_{k+1} \leftarrow (1 - \alpha)E_k + \alpha p_k$  *raw  $\rightarrow$  self-conditioning*

**end for; return**  $z_K$

The loop uses exactly NFE evaluations: one per solver step for ELF; NFE−1 before the solver updates plus one final evaluation for LangFlow (Appendix A.2).

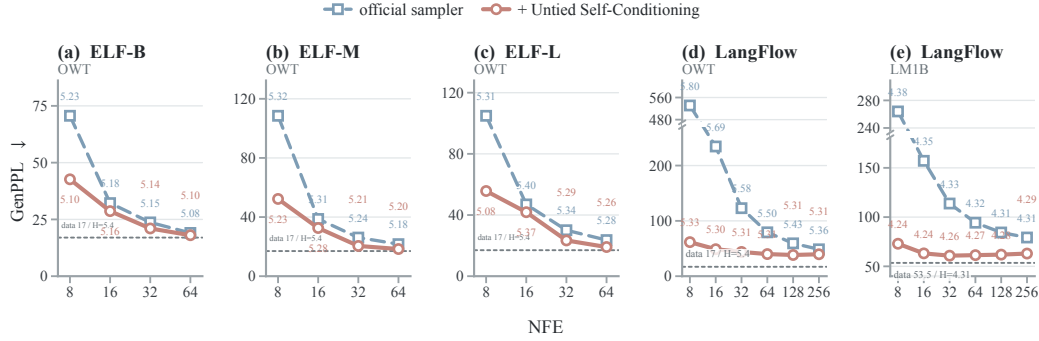


Figure 2: **GenPPL versus NFE; every curve point annotates entropy.** Panels (a)–(e) compare the official sampler (dashed) with Untied Self-Conditioning (solid) for each model–dataset pair. Dotted lines mark data references taken from Hu et al. (2026) (OWT) and Yoo et al. (2026) (LM1B).

**Pairwise LLM-judge evaluation.** Figure 3(b) provides a direct quality comparison independent of perplexity using an adapted Arena-Hard-Auto v2 protocol (Li et al., 2025). Each NFE pairs 1,024 generations from the official sampler and Untied Self-Conditioning on the same latent; each pair is judged blind in both presentation orders by DeepSeek V4 Pro, yielding 8,192 judgments in total. Untied Self-Conditioning wins 96.3% of non-tie comparisons at 8 NFE (Arena score 79.2, 95% CI [77.7, 80.8]) and retains a clear advantage at 64 NFE (77.4%, Arena score 59.7, CI [58.1, 61.2]). Appendix C.2 gives the full protocol.

Appendix C.9 additionally compares Untied Self-Conditioning with training-free sampling methods under matched conditions; Untied Self-Conditioning outperforms all tested alternatives. The following controlled experiments verify the diagnosed mismatch and each derived correction independently. The step-average scaling experiment (Appendix C.7) additionally verifies that the correction  $C_k^*$  is first order in step width and its latent contribution is second order, matching Section 3.3.

### 5.3 DOES CORRELATION ACROSS SAMPLING STEPS CHANGE A FIXED SELF-CONDITIONING UPDATE?

This experiment directly tests the causal mechanism diagnosed in Section 3.1: does the coupling itself, not the quality of individual predictions, change the value of self-conditioning? The experiment in Figure 4 preserves the marginal distribution at every NFE and changes only the correlation between successive steps. Five correlations from 0 to 1 are tested at 8 and 16 NFE.

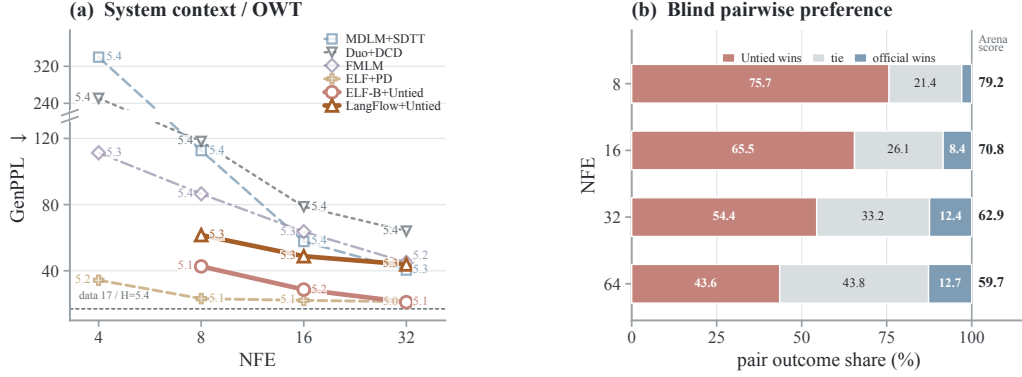


Figure 3: **System context and pairwise LLM-judge preference.** (a) System-level OpenWebText context; external curves taken from Hu et al. (2026). (b) Pair outcomes on LangFlow OpenWebText from 1,024 comparisons per NFE. Each bar partitions Ours wins, ties, and Official wins; labels at the right endpoint give the Arena score.

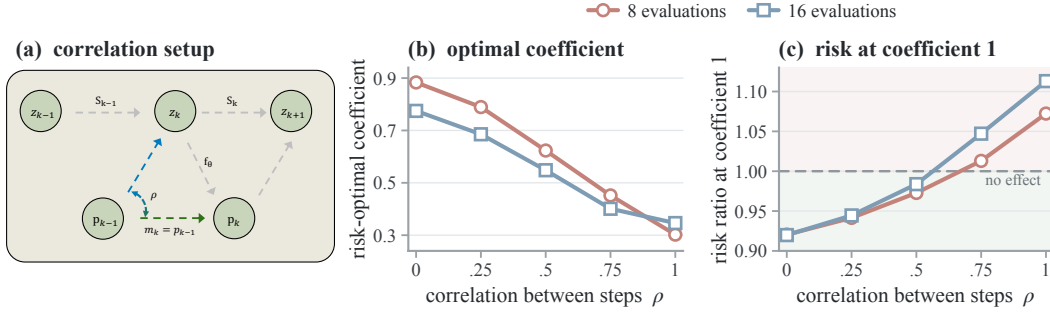


Figure 4: **Correlation across sampling steps changes the value of a fixed self-conditioning update.** (a) The experiment varies only the dependence  $\rho$  between the carried prediction  $m_k = p_{k-1}$  and the next latent  $z_k$ , while preserving both marginals. The MSE-optimal interpolation coefficient along one frozen self-conditioning-induced prediction direction decreases with  $\rho$  (b); consequently, a fixed coefficient of one crosses from helpful to harmful (c).

The MSE-optimal coefficient decreases monotonically from 0.88 to 0.30 at 8 NFE and from 0.78 to 0.35 at 16 NFE. At high correlation, full self-conditioning becomes harmful: the MSE ratio (coefficient one vs. zero) rises from 0.92 to 1.07 and from 0.92 to 1.11, respectively. The useful coefficient therefore depends on the joint distribution along the sampling trajectory, confirming that the solver-introduced correlation drives the effect. Appendix C.5 reports the full protocol and controls.

#### 5.4 WHICH SELF-CONDITIONING DIRECTIONS SHOULD BE RETAINED?

Dampening high-overlap directions outperforms matched alternatives. Table 8 (Appendix C.6) holds eigenvalues fixed and varies only their assignment to input directions, with the solver correction disabled. The overlap-aligned, reverse, and random transforms share the same eigenvalues and mean strength; they differ only in which directions receive those eigenvalues. The overlap-aligned assignment from Section 3.2 most strongly reduces  $\mathcal{E}(m)$  and outperforms official self-conditioning by 0.15 GPT-2 NLL as well as both matched controls.

Table 1: Two  $2 \times 2$  tests on ELF-B at 8 NFE with 1,024 matched generations from seed 0 (GenPPL / unigram entropy): (a) state assignment; (b) method components.

|                      | (a) State assignment |                     | (b) Components |                |                     |
|----------------------|----------------------|---------------------|----------------|----------------|---------------------|
|                      | Solver: $p_k$        | Solver: $p_k + A_k$ | SC transform   | Correction off | Correction on       |
| Next SC: $p_k$       | 67.27 / 5.23         | <b>45.63 / 5.14</b> | Off            | 74.18 / 5.27   | 50.04 / 5.16        |
| Next SC: $p_k + A_k$ | 157.77 / 5.35        | 334.19 / 5.33       | On             | 67.27 / 5.23   | <b>45.63 / 5.14</b> |

### 5.5 WHERE SHOULD THE SOLVER CORRECTION BE APPLIED?

The solver is the correct destination for the history-based correction. Table 1 applies the same correction either in the solver update, in the next self-conditioning input, or in both places, while fixing the self-conditioning transform. Applying it in the solver reaches 45.63 GenPPL compared with 67.27 without the correction under the same self-conditioning transform. Adding the correction only to the self-conditioning input raises GenPPL to 157.77, worse than no correction at all.

This ordering holds in five additional settings. Panel (b) isolates the two components: the solver correction provides the larger individual gain, and the complete sampler outperforms either component alone. Appendix C.9 reports the additional settings and direction controls.

## 6 RELATED WORK

**Self-conditioned language flows.** Continuous diffusion and flow models for text embed discrete tokens into a continuous space and refine all positions in parallel (Li et al., 2022; Strudel et al., 2022; Gulrajani & Hashimoto, 2023; Han et al., 2023; Karimi Mahabadi et al., 2024; Tae et al., 2025; Gong et al., 2023; Yuan et al., 2024; Liu et al., 2024a; Yang et al., 2026; Hu et al., 2024; Chen et al., 2026; Hu et al., 2026). Previous-prediction self-conditioning (Chen et al., 2023) supplies the model with its own clean prediction from the preceding step; variants appear across embedding, simplex, and contextual formulations (Strudel et al., 2022; Karimi Mahabadi et al., 2024; Gulrajani & Hashimoto, 2023; Yuan et al., 2024; Liu et al., 2024a; Hu et al., 2026). We study models in which a frozen network receives the current latent and a previous clean prediction through separate linear input projections. ELF and LangFlow are two instances of this interface.

**Training-time modifications to self-conditioning.** Several studies diagnose train–inference mismatch in the self-conditioning recurrence and address it through modified training (Shabalin et al., 2025; Liu et al., 2024b; Cong et al., 2026). Other methods expose the model to its own intermediate states during training (Savinov et al., 2022), specialize through post-training (Cardei et al., 2026), or distill recurrent self-conditioning iterations (Yoo et al., 2026). All require modifying model weights. Our method keeps the model frozen and constructs a transform from the geometry of the input projections.

**Inference-time changes to self-conditioning.** Analog Bits studies an exponential moving average of previous predictions and a self-guidance rule that uses two model evaluations (Chen et al., 2023). ACE estimates a repetition-associated direction from generated trajectories and subtracts it from the self-conditioning input (Zhang et al., 2026). These methods modify the self-conditioning input independently of the solver update.

**Solvers and multi-step methods.** Advanced solvers reuse past model evaluations to improve numerical integration of the probability-flow ODE (Zhang & Chen, 2023; Lu et al., 2025; Zhao et al., 2023). Separately, several methods aggregate past clean predictions to stabilize or refine the current prediction supplied to the solver (Wizadwongsa et al., 2024; Qian et al., 2024; Li & zeng, 2026; Sadat et al., 2026; Li et al., 2026). In self-conditioned models, the same prediction enters both the solver and the next network evaluation. These methods improve the numerical update but do not account for the self-conditioning pathway.

**Offline model statistics for solver design.** DPM-Solver-v3 precomputes evaluator-free empirical model statistics to instantiate a model-dependent solver parameterization (Zheng et al., 2023). Offline

trajectory statistics have proved useful for training-free solver design more broadly. These methods target numerical integration without modifying the self-conditioning pathway.

In summary, prior methods modify individual components of the sampling loop in isolation. This work identifies that the solver introduces coupling between the two pathways that is absent during training, and derives both corrections from the model’s projection geometry and integration structure.

## 7 CONCLUSION

We diagnosed a train–inference mismatch in self-conditioned flow sampling: the solver introduces coupling between the latent and self-conditioning pathways that is absent during training. A controlled experiment confirmed the causal role of this coupling: when it is strong, a fixed self-conditioning coefficient crosses from helpful to harmful. From the model’s frozen projection weights we derived the geometric structure of the resulting redundancy, and from the solver’s integration structure we derived the step-average approximation gap. Untied Self-Conditioning corrects both in a training-free sampler with one model evaluation per step. Across ELF and LangFlow, it improves generation quality at every tested NFE; with eight evaluations, it reduces LangFlow GenPPL on OpenWebText from 531 to 62, with gains confirmed by pairwise LLM-judge preference and per-sample external-evaluator likelihood.

## REFERENCES

- Michael Cardei, Huu Binh Ta, and Ferdinando Fioretto. Simple self-conditioning adaptation for masked diffusion models, 2026. URL <https://arxiv.org/abs/2604.26985>.
- Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=3itjR9QxFw>.
- Yuxin Chen, Chumeng Liang, Hangke Sui, Ruihan Guo, Chaoran Cheng, Jiaxuan You, and Ge Liu. Langflow: Continuous diffusion rivals discrete in language modeling, 2026. URL <https://arxiv.org/abs/2604.11748>.
- Dat Nguyen Cong, Tung Kieu, and Hoang Thanh-Tung. FastDiSS: Few-step match many-step diffusion language model on sequence-to-sequence generation. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17572–17592, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.870. URL <https://aclanthology.org/2026.findings-acl.870/>.
- Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023. URL [https://openreview.net/forum?id=jQj\\_rLVXsj](https://openreview.net/forum?id=jQj_rLVXsj).
- Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 16693–16715. Curran Associates, Inc., 2023. doi: 10.52202/075280-0730. URL [https://proceedings.neurips.cc/paper\\_files/paper/2023/file/35b5c175e139bfff5f22a5361270fce87-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/35b5c175e139bfff5f22a5361270fce87-Paper-Conference.pdf).
- Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11575–11596, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.647. URL <https://aclanthology.org/2023.acl-long.647/>.
- Keya Hu, Linlu Qiu, Yiyang Lu, Hanhong Zhao, Tianhong Li, Yoon Kim, Jacob Andreas, and Kaiming He. Elf: Embedded language flows, 2026. URL <https://arxiv.org/abs/2605.10938>.

- Vincent Hu, Di Wu, Yuki Asano, Pascal Mettes, Basura Fernando, Björn Ommer, and Cees Snoek. Flow matching for conditional text generation in a few sampling steps. In Yvette Graham and Matthew Purver (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 380–392, St. Julian’s, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-short.33. URL <https://aclanthology.org/2024.eacl-short.33/>.
- Rabeeh Karimi Mahabadi, Hamish Ivison, Jaesung Tae, James Henderson, Iz Beltagy, Matthew Peters, and Arman Cohan. TESS: Text-to-text self-conditioned simplex diffusion. In Yvette Graham and Matthew Purver (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2347–2361, St. Julian’s, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.144. URL <https://aclanthology.org/2024.eacl-long.144/>.
- Shigui Li and Delu zeng. DiFA: Inference-time forward-process alignment for diffusion models. In *Forty-third International Conference on Machine Learning*, 2026. URL <https://openreview.net/forum?id=YGyWLM00EW>.
- Tiancheng Li, Weijian Luo, Zhiyang Chen, Liyuan Ma, and Guo-Jun Qi. Self-guidance: Boosting flow and diffusion generation on their own. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(1):781–791, 2026. doi: 10.1109/TPAMI.2025.3611831.
- Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34209–34231. PMLR, 13–19 Jul 2025. URL <https://proceedings.mlr.press/v267/li25h.html>.
- Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 4328–4343. Curran Associates, Inc., 2022. doi: 10.52202/068431-0313. URL [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/1be5bc25d50895ee656b8c2d9eb89d6a-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/1be5bc25d50895ee656b8c2d9eb89d6a-Paper-Conference.pdf).
- 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>.
- Pan Liu, Xiaohua Tian, and Zhouhan Lin. Enable fast sampling for Seq2Seq text diffusion. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8495–8505, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.497. URL <https://aclanthology.org/2024.findings-emnlp.497/>.
- Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. Text diffusion with reinforced conditioning, 2024b. URL <https://arxiv.org/abs/2402.14843>.
- Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, June 2025. ISSN 2731-5398. doi: 10.1007/s11633-025-1562-4. URL <http://dx.doi.org/10.1007/s11633-025-1562-4>.
- Yurui Qian, Qi Cai, Yingwei Pan, Yehao Li, Ting Yao, Qibin Sun, and Tao Mei. Boosting diffusion models with moving average sampling in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8911–8920, June 2024. URL [https://openaccess.thecvf.com/content/CVPR2024/html/Qian\\_Boosting\\_Diffusion\\_Models\\_with\\_Moving\\_Average\\_Sampling\\_in\\_Frequency\\_Domain\\_CVPR\\_2024\\_paper.html](https://openaccess.thecvf.com/content/CVPR2024/html/Qian_Boosting_Diffusion_Models_with_Moving_Average_Sampling_in_Frequency_Domain_CVPR_2024_paper.html).

- Seyedmorteza Sadat, Farnood Salehi, and Romann M. Weber. HiGS: History-guided sampling for plug-and-play enhancement of diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=cyQUZDMpg3>.
- Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=T0GpzBQ1Fg6>.
- Alexander Shabalin, Viacheslav Meshchaninov, Egor Chibulatov, Vladislav Lapikov, Roman Kim, Grigory Bartosh, Dmitry Molchanov, Sergey Markov, and Dmitry Vetrov. Tencdm: Understanding the properties of the diffusion model in the space of language model encodings, 2025. URL <https://arxiv.org/abs/2402.19097>.
- Robin Strudel, Corentin Tallec, Florent Alth  , Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and R  mi Leblond. Self-conditioned embedding diffusion for text generation, 2022. URL <https://arxiv.org/abs/2211.04236>.
- Jaesung Tae, Hamish Ivison, Sachin Kumar, and Arman Cohan. TESS 2: A large-scale generalist diffusion language model. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21171–21188, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1029. URL <https://aclanthology.org/2025.acl-long.1029/>.
- Suttisak Widadwongsa, Worameth Chinchuthakun, Pramook Khungurn, Amit Raj, and Supasorn Suwajanakorn. Diffusion sampling with momentum for mitigating divergence artifacts. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=HXc5aXeoc8>.
- Zhihan Yang, Wei Guo, Shuibai Zhang, Subham Sekhar Sahoo, Yongxin Chen, Arash Vahdat, Morteza Mardani, and John Thickstun. Continuous diffusion scales competitively with discrete diffusion for language, 2026. URL <https://arxiv.org/abs/2605.18530>.
- Jaehoon Yoo, Wonjung Kim, Floor Eijkelboom, Chanhyuk Lee, Nicholas M. Boffi, Seunghoon Hong, and Jinwoo Kim. Self-conditioned flow map language models via fixed-point flows, 2026. URL <https://arxiv.org/abs/2607.00714>.
- Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. Text diffusion model with encoder-decoder transformers for sequence-to-sequence generation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 22–39, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.2. URL <https://aclanthology.org/2024.naacl-long.2/>.
- Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=Loek7hfb46P>.
- 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, 2026. URL <https://arxiv.org/abs/2607.00588>.
- Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 49842–49869. Curran Associates, Inc., 2023. doi: 10.52202/075280-2170. URL [https://proceedings.neurips.cc/paper\\_files/paper/2023/file/9c2aale456ea543997f6927295196381-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/9c2aale456ea543997f6927295196381-Paper-Conference.pdf).

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 55502–55542. Curran Associates, Inc., 2023. doi: 10.52202/075280-2423. URL [https://proceedings.neurips.cc/paper\\_files/paper/2023/file/ada8de994b46571bdcd7eef2d3f9cff-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/ada8de994b46571bdcd7eef2d3f9cff-Paper-Conference.pdf).

## A DERIVATIONS AND PROOFS

### A.1 GAUSSIAN CONDITIONING CALCULATION

Let  $T = \tau^2$ ,  $a = \sigma_z$ , and  $b = \sigma_m$ . Gaussian conditioning on  $(Z, M)$  gives the coefficient on  $M$  as

$$b_\rho = \frac{T(a^2 - \rho ab)}{\Delta_\rho}, \quad \Delta_\rho = (T + a^2)(T + b^2) - (T + \rho ab)^2 > 0. \quad (14)$$

Neither marginal  $P_{Y,Z}$  nor  $P_{Y,M}$  depends on  $\rho$ . For  $0 < a < b$ , direct differentiation gives

$$\frac{db_\rho}{d\rho} = -\frac{Tab[(b^2 - a^2)(T + a^2) + a^2b^2(\rho - a/b)^2]}{\Delta_\rho^2} < 0. \quad (15)$$

Thus  $b_\rho$  vanishes at  $\rho = a/b$  and changes sign there. At  $\rho = 1$ , the shared-noise observations permit exact cancellation,  $Y = (bZ - aM)/(b - a)$ . The conditional fusion rule can therefore change sign while both feature–target marginals remain fixed.

### A.2 EXACT AVERAGE OF PAST PREDICTIONS USING THE SOLVER COEFFICIENTS

For ELF, distinguish the time  $s_k$  of the model evaluation (after any re-noising) from the target time  $t_{k+1}$ . Write the latent at that evaluation as

$$z_k = (1 - s_k)\nu_k + s_k O_k, \quad (16)$$

where  $\nu_k$  contains the complementary noise component and  $O_k$  is the normalized contribution of predictions previously supplied to the solver. Suppose the solver uses  $q_k$  to advance the latent from  $s_k$  to  $t_{k+1}$ , with

$$\beta_k = \frac{t_{k+1} - s_k}{1 - s_k}. \quad (17)$$

Substitution into the affine solver update gives

$$O_{k+1} = \frac{(1 - \beta_k)s_k O_k + \beta_k q_k}{t_{k+1}} = (1 - \kappa_k)O_k + \kappa_k q_k, \quad \kappa_k = \frac{\beta_k}{t_{k+1}}. \quad (18)$$

The identity  $(1 - \beta_k)s_k + \beta_k = t_{k+1}$  makes the two coefficients sum to one. The coefficient of  $q_k$  in the full latent is  $\beta_k$ , whereas  $\kappa_k = \beta_k/t_{k+1}$  is its update share inside normalized  $O_{k+1}$ . With  $O_0 = 0$  and  $s_0 = 0$ , the first normalized update has  $\kappa_0 = 1$  and  $O_1 = q_0$ , even though the full-latent coefficient is only  $\beta_0 = t_1$ . Re-noising changes  $s_k$  but preserves this algebraic separation.

For a LangFlow transition from negative log-SNR  $\gamma_t$  to  $\gamma_s$ , the clean-prediction component after dividing by the signal coefficient obeys

$$O_s = e^{(\gamma_s - \gamma_t)/2} O_t + \left(1 - e^{(\gamma_s - \gamma_t)/2}\right) q_t. \quad (19)$$

Thus equation 8 holds with  $\kappa = 1 - e^{(\gamma_s - \gamma_t)/2}$ . Additive stochastic terms belong to the complementary component of the latent and do not enter  $O_k$ .

### A.3 WEIGHTED PREDICTION OVER A SOLVER STEP AND STEP-SIZE ORDERS

Hold the self-conditioning input fixed between two model evaluations, and let  $y(u)$  denote the normalized latent coordinate driven by the clean prediction. For self-conditioned language flows whose solver updates the latent affinely in the clean prediction, a change of time coordinate gives

$$\frac{dy}{du} = p(u) - y(u), \quad p(u) = f_\theta(z(u), m_k, t(u)). \quad (20)$$

Here  $u$  is the increasing solver time coordinate (ELF uses  $u = -\log(1 - t)$  in its active region; LangFlow uses  $u = -\gamma/2$ ) and  $h_k = u_{k+1} - u_k > 0$  is the width of step  $k$  in this coordinate. Multiplying equation 20 by  $e^u$  gives

$$\frac{d}{du} (e^u y(u)) = e^u p(u). \quad (21)$$

Integration over  $[u_k, u_k + h]$  yields

$$y(u_k + h) = e^{-h} y(u_k) + \int_0^h e^{-(h-s)} p(u_k + s) ds. \quad (22)$$

Defining the step-average prediction

$$\bar{p}_k(h_k) = \frac{1}{1 - e^{-h_k}} \int_0^{h_k} e^{-(h_k-s)} p(u_k + s) ds, \quad (23)$$

the exact endpoint is

$$y(u_k + h_k) = e^{-h_k} y(u_k) + (1 - e^{-h_k}) \bar{p}_k(h_k). \quad (24)$$

If the prediction path is three times differentiable,

$$p(u_k + s) = p_k + s \dot{p}_k + \frac{s^2}{2} \ddot{p}_k + O(s^3). \quad (25)$$

Substitution into equation 23 gives

$$C_k^*(h) = a_1(h) \dot{p}_k + \frac{a_2(h)}{2} \ddot{p}_k + O(h^3), \quad (26)$$

where

$$\begin{aligned} a_1(h) &= \frac{h - 1 + e^{-h}}{1 - e^{-h}} = \frac{h}{2} + O(h^2), \\ a_2(h) &= \frac{h^2 - 2h + 2 - 2e^{-h}}{1 - e^{-h}} = \frac{h^2}{3} + O(h^3). \end{aligned} \quad (27)$$

Since  $1 - e^{-h} = h + O(h^2)$ , the true correction is first order and its contribution to the endpoint is second order. For any candidate  $B_k$ , substituting  $p_k + \eta B_k$  for the interval prediction in equation 24 gives

$$y_{k+1}^{(\eta B)} - y_{k+1}^* = (1 - e^{-h_k})(\eta B_k - C_k^*). \quad (28)$$

The squared endpoint error therefore differs from equation 5 by the positive factor  $(1 - e^{-h_k})^2$ , preserving the sign of the improvement criterion.

The same weighted-average representation holds for a general scalar affine flow

$$\frac{dy}{du} = a(u)y(u) + b(u)p(u). \quad (29)$$

If

$$\varphi(v, u) = \exp\left(\int_u^v a(s) ds\right), \quad (30)$$

variation of constants weights  $p(u + s)$  by  $\varphi(u + h, u + s)b(u + s)$ . Normalizing this weight gives the corresponding step-average prediction whenever its integral is nonzero. ELF uses  $u = -\log(1 - t)$  on finite active intervals; the LangFlow update after dividing by the signal coefficient uses  $u = -\gamma/2$  and  $h = (\gamma_t - \gamma_s)/2$ . The ELF terminal endpoint has an infinite unregularized log-clock width; the sampler uses the bounded interval feature defined in Appendix B.3 for this step.

#### A.4 CORRECTION FOR A FINITE, ZERO-INITIALIZED EMA

For an affine sequence in the model-evaluation index,  $p_i = p_0 + iv$ , with  $E_{i+1} = (1 - \alpha)E_i + \alpha p_i$ ,  $E_0 = 0$ , and  $\alpha \in (0, 1]$ , let  $r = 1 - \alpha$ . Direct summation gives  $w_n p_n - E_n = L_n v$ , where  $w_n = 1 - r^n$  and

$$L_n = \alpha \sum_{j=1}^n j r^{j-1} = \frac{1 - (n+1)r^n + nr^{n+1}}{\alpha}. \quad (31)$$

The raw residual satisfies  $p_n - E_n = r^n p_n + L_n v$ , so the bias-corrected increment  $\hat{v} = (w_n p_n - E_n)/L_n$  recovers  $v$  exactly. For  $\alpha \in (0, 1)$  and  $n \geq 1$ , the lag  $L_n$  is strictly positive and increases monotonically toward  $1/\alpha$ ; at  $\alpha = 1$ ,  $L_n = 1$ . This identity concerns increments per model evaluation. Converting them to derivatives in solver time also requires the time-grid spacing and prediction curvature. The sampler uses the raw residual as a history-based direction.

## B METHOD DETAILS

### B.1 PRINCIPAL-ANGLE DIRECTIONS FOR SELF-CONDITIONING

Let  $W_z$  and  $W_m$  be the effective frozen input-projection matrices for the latent and self-conditioning inputs. Orthonormal row-space bases define projectors  $P_z$  and  $P_m$ . On the self-conditioning row space,

$$P_m P_z P_m Q = Q \text{Diag}(\rho_1^2, \dots, \rho_r^2), \quad 0 \leq \rho_i \leq 1. \quad (32)$$

The basis  $Q$  and squared cosines  $\rho_i^2$  are computed once from frozen model weights.

At step  $k$ , the transform uses

$$\hat{a}_k = \kappa_{k-1}(1 - \bar{\delta}_{k-1}), \quad (33)$$

where  $\kappa_{k-1}$  is the solver’s update weight from equation 8 and  $\bar{\delta}_{k-1}$  is the average clipped cosine distance defined in Appendix B.4. The state initialization  $\kappa_{-1} = 0$ ,  $\bar{\delta}_{-1} = 0$  makes  $F_0 = I$ . Let  $\lambda_R^{(0)} \geq 0$  be the fixed base transform exponent,  $\lambda_R = \lambda_R^{(0)} \eta_R$ ,  $r \geq 1$ , and  $0 < \epsilon_{\text{fp}} \leq 1$ . Then

$$\begin{aligned} b_{k,i} &= \max\{1 - \hat{a}_k \rho_i^2, \epsilon_{\text{fp}}\}, \\ g_k &= \left( \prod_{i=1}^r b_{k,i} \right)^{1/r}, \\ \ell_{k,i} &= (g_k b_{k,i})^{\lambda_R}. \end{aligned} \quad (34)$$

The resulting transform is

$$F_k = I - Q(I - \text{Diag}(\ell_{k,1}, \dots, \ell_{k,r}))Q^\top. \quad (35)$$

It is symmetric, preserves the orthogonal complement of  $Q$ , and multiplies principal direction  $i$  by  $\ell_{k,i}$ . The floor  $\epsilon_{\text{fp}}$  prevents complete suppression of any direction. The common factor  $g_k^{\lambda_R}$  adds a uniform contraction within the principal subspace; because  $0 < b_{k,i} \leq 1$  and  $b_{k,i}$  is nonincreasing in  $\rho_i^2$ , the transform satisfies the contraction conditions in Section 3.2. Setting this factor to one isolates the direction-dependent dampening in the experiment reported in Appendix C.7.

### B.2 INPUT-PROJECTION GEOMETRY ACROSS MODELS

Table 2 reports the effective input-projection matrices used to compute the principal-angle basis. For ELF, the latent and self-conditioning inputs pass through a shared bottleneck projection before the first transformer block, producing effective matrices of shape  $128 \times 512$  (rank 128 in a 512-dimensional input space). Because the row spaces are 128-dimensional subspaces of  $\mathbb{R}^{512}$ , the principal-angle spectrum is non-trivial: squared cosines range from near zero to 0.92 with a well-spread distribution that is stable across checkpoints.

For LangFlow, the self-conditioning projection is  $768 \times 1536$  and splits into two  $768 \times 768$  blocks. Both are full rank, so  $P_z = P_m = I$  and all squared cosines equal one. The self-conditioning transform therefore applies isotropic dampening in this architecture.

Table 2: Effective input-projection geometry. ELF projections pass through a 128-wide bottleneck, producing a well-spread principal-angle spectrum; LangFlow projections are full rank and the transform applies isotropic dampening. Canonical correlations  $\rho_i$  are the cosines of the principal angles between the two row spaces.

| Model    | Shape   | Rank | $\rho_{\min}$             | $\rho_{\text{med}}$ | $\rho_{\max}$ | Mean $\rho_i^2$ | $ \rho_i \geq 0.9 $ | $ \rho_i \leq 0.1 $ |
|----------|---------|------|---------------------------|---------------------|---------------|-----------------|---------------------|---------------------|
| ELF-B    | 128×512 | 128  | 0.001                     | 0.594               | 0.959         | 0.391           | 18                  | 14                  |
| ELF-M    | 128×512 | 128  | 0.001                     | 0.581               | 0.950         | 0.381           | 17                  | 13                  |
| ELF-L    | 128×512 | 128  | 0.004                     | 0.577               | 0.952         | 0.375           | 13                  | 13                  |
| LangFlow | 768×768 | 768  | full rank; all $\rho_i=1$ |                     |               |                 |                     |                     |

### B.3 PREDICTION HISTORY AND SOLVER-STEP WIDTH

The EMA uses weight  $\alpha \in (0, 1]$  on the current prediction, with  $E_0 = 0$ . The history residual used by the sampler is

$$H_k = \mathbf{1}[\sigma_k > 0](p_k - E_k), \quad (36)$$

and is active on steps with nonzero precomputed statistics. The zero-initialized EMA introduces a startup bias that decays as  $(1-\alpha)^k$ ; a bias-corrected variant  $\hat{v}_k = (w_k p_k - E_k)/L_k$  exists (Appendix A.4), but the sampler uses the raw residual since the bias vanishes after a few steps.

A second direction estimate compares  $p_k$  with the running weighted average  $O_k$  and is rescaled samplewise to match  $\|H_k\|_F$ :

$$G_k = \begin{cases} \|H_k\|_F \frac{p_k - O_k}{\|p_k - O_k\|_F} & \text{if } \|H_k\|_F, \|p_k - O_k\|_F > \epsilon, \\ 0 & \text{otherwise,} \end{cases} \quad (37)$$

as in equation 9. The history-based correction direction is

$$U_k = H_k + \eta_O \Delta u_k (H_k - G_k). \quad (38)$$

It reduces to  $H_k$  when the two estimates agree or the step width vanishes. For ELF, let  $t_k^{\text{start}}$  be the official solver-grid start before any re-noising and define

$$c_\epsilon(t) = \begin{cases} -\log(1-t), & t \leq 1-\epsilon, \\ -\log \epsilon + \frac{t-(1-\epsilon)}{\epsilon}, & t > 1-\epsilon, \end{cases} \quad \tilde{c}_\epsilon(t) = \frac{c_\epsilon(t)}{c_\epsilon(1)}. \quad (39)$$

The feature used by the sampler is  $\Delta u_k = \tilde{c}_\epsilon(t_{k+1}) - \tilde{c}_\epsilon(t_k^{\text{start}})$ . LangFlow uses  $\Delta u_k = (\gamma_k - \gamma_{k+1})/(\gamma_{\max} - \gamma_{\min})$ . Both lie in  $[0, 1]$  on their sampling grids. The exact solver-time widths are  $-\log(1-t_{k+1}) + \log(1-s_k)$  and  $(\gamma_k - \gamma_{k+1})/2$ , respectively (Appendix A.3).

### B.4 EMPIRICAL STATISTICS FROM REFERENCE CORRECTIONS

The offline statistics are computed from a fixed reference recurrence that applies a correction during the middle portion of sampling and ramps smoothly. The specific constants below define this reference and are not varied per configuration. Let  $s_k^{\text{ref}}$  be a step-dependent activation schedule,  $g_T^{(0)}$  a fixed base scale, and  $\lambda_k^{\text{ref}} = 3.5 g_T^{(0)} s_k^{\text{ref}}$ . On each reference trajectory, the correction is

$$\begin{aligned} \tilde{H}_k &= \lambda_k^{\text{ref}} (p_k - E_k), \\ \tilde{G}_k &= \begin{cases} \|\tilde{H}_k\|_F \frac{p_k - O_k}{\|p_k - O_k\|_F}, & \|\tilde{H}_k\|_F, \|p_k - O_k\|_F > \epsilon, \\ 0, & \text{otherwise,} \end{cases} \\ \tilde{U}_k &= \tilde{H}_k + \Delta u_k \tilde{\delta}_k \odot (\tilde{H}_k - \tilde{G}_k). \end{aligned} \quad (40)$$

The activation schedule is  $s_k^{\text{ref}} = \mathbf{1}[0.40 < r_k \leq 0.95] \sqrt{(r_k - 0.40)/0.55}$  for normalized sampling progress  $r_k$ ; inactive tokens are zeroed throughout. During generation, Equation 38 uses  $H_k = p_k - E_k$  on every step with  $\sigma_k > 0$ . For active token  $n$ ,  $\tilde{\delta}_{k,n}$  is  $\min\{1, 1 - \cos(\tilde{H}_{k,n}, \tilde{G}_{k,n})\}$  when

both vectors are nondegenerate and zero otherwise. The reference recurrence supplies  $p_k + \tilde{U}_k$  to the solver and stores the raw  $p_k$  for the next self-conditioning input.

For a correction tensor  $V$  on active positions  $I$ , let  $M_I(V)$  be its token mean and  $C_IV = V - M_I(V)$  its centered component. Two disjoint frozen trajectory banks  $\mathcal{B}_\mu$  and  $\mathcal{B}_\sigma$  define

$$\begin{aligned}\bar{\delta}_k &= \mathbb{E}_{\xi \sim \mathcal{B}_\mu} \left[ \frac{1}{|I_k^\xi|} \sum_{n \in I_k^\xi} \tilde{\delta}_{k,n}^\xi \right], \\ \mu_k &= \mathbb{E}_{\xi \sim \mathcal{B}_\mu} [M_{I_k^\xi}(\tilde{U}_k^\xi)], \\ \sigma_k^2 &= \frac{\sum_{\xi \in \mathcal{B}_\sigma} \|C_{I_k^\xi} \tilde{U}_k^\xi\|_F^2}{\sum_{\xi \in \mathcal{B}_\sigma} |I_k^\xi| D},\end{aligned}\tag{41}$$

where  $D$  is the prediction width. Thus  $\mu_k$  and  $\sigma_k$  summarize the correction used by the fixed reference recurrence. The true correction  $C_k^*$  enters separately through the endpoint condition in equation 5. As with empirical model statistics in training-free solvers (Zheng et al., 2023), the update form is fixed analytically and the frozen model and grid supply its coefficients. Here  $\mathbb{E}_{\xi \sim \mathcal{B}}$  denotes a uniform empirical average over the stored bank.

### B.5 MATCHING THE EMPIRICAL MEAN AND RMS

For active positions  $I_k$ , let  $C_{\text{tok}}$  center over  $I_k$  and  $\text{RMS}_{I_k}$  compute the root-mean-square over active positions and width. If  $\text{RMS}(C_{\text{tok}} U_k) > 0$ , the unscaled projection  $\Pi_k(U_k)$  in equation 11 satisfies

$$M_{I_k}(\Pi_k) = \mu_k, \quad \text{RMS}_{I_k}(C_{\text{tok}} \Pi_k) = \sigma_k.\tag{42}$$

Among all tensors sharing these two moments, the centered component uniquely maximizes alignment with  $C_{\text{tok}} U_k$  (by Cauchy–Schwarz). The outer strength  $\eta_T$  scales both moments, so the final correction  $A_k$  has mean  $\eta_T \mu_k$  and centered RMS  $\eta_T \sigma_k$ . Inactive positions receive zero correction. If the centered direction has zero norm, the implementation retains only  $\eta_T \mu_k$ .

The affine endpoint condition in equation 5 now applies with  $B_k = \Pi_k(U_k)$ . Whenever  $\langle C_k^*, \Pi_k(U_k) \rangle > 0$ , a nonempty interval of positive  $\eta_T$  strictly improves the endpoint obtained from  $p_k$  alone.

To separate the two contributions, write  $N_k = |I_k|$  and  $\bar{C}_k^* = M_{I_k}(C_k^*)$ . When the centered target is nonzero,  $\theta_k$  denotes the angle between  $C_{\text{tok}} C_k^*$  and  $C_{\text{tok}} U_k$ . Orthogonality of the token-mean and centered components gives the exact decomposition

$$\begin{aligned}\langle C_k^*, \Pi_k(U_k) \rangle_F &= N_k \langle \bar{C}_k^*, \mu_k \rangle + \sigma_k \frac{\langle C_{\text{tok}} C_k^*, C_{\text{tok}} U_k \rangle_F}{\text{RMS}_{I_k}(C_{\text{tok}} U_k)} \\ &= N_k \langle \bar{C}_k^*, \mu_k \rangle + \sigma_k \sqrt{N_k D} \|C_{\text{tok}} C_k^*\|_F \cos \theta_k.\end{aligned}\tag{43}$$

The first term depends on how well the precomputed mean  $\mu_k$  aligns with the true correction’s mean; the second depends on the angle  $\theta_k$  between the centered correction direction and the centered target. Frozen reference trajectories set  $\mu_k$  and  $\sigma_k$ ; the current trajectory determines  $\theta_k$  through  $U_k$ . The endpoint is strictly improved whenever  $2 \langle C_k^*, \Pi_k(U_k) \rangle_F > \eta_T \|\Pi_k(U_k)\|_F^2$ .

## C ADDITIONAL EXPERIMENTAL DETAILS AND RESULTS

### C.1 EVALUATION PROTOCOL

Every matched comparison uses the same frozen checkpoint, initial latent, base solver, time grid, sequence length, and number of model evaluations. Each configuration starts from the same random stream; generation and evaluation are run separately to avoid interference.

ELF uses its official random logit-normal grid. Its SDE noise scale is 2.0, 2.0, 1.5, and 1.0 at 8, 16, 32, and 64 NFE. The official self-conditioning scale is 3.0 for all evaluated NFE. Evaluation uses four

ranks and official seeds 0–5; each seed produces 1,024 generations of length 1,024. The independent seed run is the unit for the reported standard error.

LangFlow uses its official time grid, obtained from quantiles of the proposal distribution between  $1 - 10^{-5}$  and  $10^{-5}$ , and seeds 0–5. Each seed produces 1,024 generations per setting. OpenWebText samples have length 1,024 and LM1B samples have length 128.

ELF computes generative perplexity with frozen GPT-2 Large after applying its official filter to remove empty decoded samples, then retokenizes decoded text for unigram entropy. LangFlow computes entropy from generated token IDs and perplexity after decoding. Official and Untied Self-Conditioning configurations use the same evaluator convention in each comparison.

Our ELF integration reproduces the official sampling code exactly: under matched model weights and random streams, all intermediate and final tensors agree elementwise across ranks. The LangFlow integration is verified analogously on closed-loop generation outputs.

## C.2 PAIRWISE LLM-JUDGE EVALUATION PROTOCOL

The pairwise evaluation adapts the Arena-Hard-Auto v2 protocol (Li et al., 2025) for unconditional text generation. Every NFE in  $\{8, 16, 32, 64\}$  produces 1,024 text pairs by matching official and Untied Self-Conditioning generations on the same latent and sample index (LangFlow OpenWebText, seed 42, 1,024-token sequences). Each pair is presented twice to the judge with the answer order exactly reversed, yielding 8,192 judgments.

The judge (DeepSeek V4 Pro, thinking enabled) receives neither method names nor NFE labels. It scores grammatical fluency, discourse coherence, semantic content, and degeneration artifacts, then returns a five-level verdict:  $A \gg B$ ,  $A > B$ ,  $A = B$ ,  $B > A$ ,  $B \gg A$ . Significant verdicts ( $\gg$ ) contribute three binary observations to the Arena score; slight verdicts ( $>$ ) and ties contribute one.

Pair-level outcomes merge the two games per pair: a pair is a candidate win if the weighted score exceeds 0.5, an official win if below 0.5, and a tie otherwise. The 95% confidence interval uses 10,000 paired bootstrap resamples. No judgments were excluded or reassigned.

## C.3 COMPLETE ENDPOINT RESULTS

Tables 3 and 4 give the absolute GenPPL, uncertainty, and unigram entropy values corresponding to Figure 2.

Table 3: Untied Self-Conditioning reduces GenPPL at every ELF NFE; the gap is largest at 8 NFE. Entries are mean  $\pm$  standard error over six seeds, each with 1,024 samples.

| Model | NFE | GenPPL $\downarrow$ |                | $H$            |                |
|-------|-----|---------------------|----------------|----------------|----------------|
|       |     | Official            | Ours           | Official       | Ours           |
| ELF-B | 8   | 70.6 $\pm$ 1.7      | 42.7 $\pm$ 2.3 | 5.23 $\pm$ .02 | 5.10 $\pm$ .01 |
|       | 16  | 32.2 $\pm$ 0.8      | 28.7 $\pm$ 1.0 | 5.18 $\pm$ .01 | 5.16 $\pm$ .01 |
|       | 32  | 23.7 $\pm$ 0.3      | 21.0 $\pm$ 0.2 | 5.15 $\pm$ .01 | 5.14 $\pm$ .01 |
|       | 64  | 19.0 $\pm$ 0.2      | 18.0 $\pm$ 0.3 | 5.08 $\pm$ .01 | 5.10 $\pm$ .01 |
| ELF-M | 8   | 108.5 $\pm$ 7.4     | 52.3 $\pm$ 3.5 | 5.32 $\pm$ .03 | 5.23 $\pm$ .00 |
|       | 16  | 38.7 $\pm$ 1.4      | 32.6 $\pm$ 1.5 | 5.31 $\pm$ .01 | 5.29 $\pm$ .01 |
|       | 32  | 26.1 $\pm$ 0.2      | 20.5 $\pm$ 0.2 | 5.24 $\pm$ .00 | 5.21 $\pm$ .00 |
|       | 64  | 21.7 $\pm$ 0.2      | 18.4 $\pm$ 0.1 | 5.18 $\pm$ .01 | 5.20 $\pm$ .00 |
| ELF-L | 8   | 104.9 $\pm$ 5.3     | 55.7 $\pm$ 3.2 | 5.31 $\pm$ .10 | 5.08 $\pm$ .12 |
|       | 16  | 46.9 $\pm$ 1.7      | 41.9 $\pm$ 2.1 | 5.40 $\pm$ .01 | 5.37 $\pm$ .01 |
|       | 32  | 30.1 $\pm$ 0.3      | 23.4 $\pm$ 0.3 | 5.34 $\pm$ .00 | 5.29 $\pm$ .01 |
|       | 64  | 23.6 $\pm$ 0.2      | 19.1 $\pm$ 0.1 | 5.28 $\pm$ .00 | 5.26 $\pm$ .00 |

Table 4: Untied Self-Conditioning improves LangFlow quality at every NFE on both datasets, reducing OpenWebText GenPPL from 531 to 62 at 8 NFE. Entries are mean  $\pm$  standard error over six seeds, each with 1,024 samples (OpenWebText: 1,024 tokens; LM1B: 128 tokens).

| Data | NFE | GenPPL $\downarrow$ |                | $H$            |                |
|------|-----|---------------------|----------------|----------------|----------------|
|      |     | Official            | Ours           | Official       | Ours           |
| OWT  | 8   | 531.2 $\pm$ 3.4     | 61.6 $\pm$ 0.2 | 5.81 $\pm$ .00 | 5.33 $\pm$ .00 |
|      | 16  | 234.5 $\pm$ 0.8     | 48.8 $\pm$ 0.1 | 5.69 $\pm$ .00 | 5.30 $\pm$ .00 |
|      | 32  | 123.0 $\pm$ 0.5     | 43.8 $\pm$ 0.1 | 5.58 $\pm$ .00 | 5.31 $\pm$ .00 |
|      | 64  | 79.3 $\pm$ 0.3      | 40.0 $\pm$ 0.1 | 5.50 $\pm$ .00 | 5.31 $\pm$ .00 |
|      | 128 | 59.4 $\pm$ 0.1      | 38.3 $\pm$ 0.1 | 5.43 $\pm$ .00 | 5.31 $\pm$ .00 |
|      | 256 | 48.6 $\pm$ 0.2      | 39.6 $\pm$ 0.1 | 5.36 $\pm$ .00 | 5.31 $\pm$ .00 |
| LM1B | 8   | 264.0 $\pm$ 1.3     | 73.0 $\pm$ 0.1 | 4.38 $\pm$ .00 | 4.24 $\pm$ .00 |
|      | 16  | 157.1 $\pm$ 0.6     | 63.2 $\pm$ 0.2 | 4.35 $\pm$ .00 | 4.24 $\pm$ .00 |
|      | 32  | 113.8 $\pm$ 0.7     | 60.8 $\pm$ 0.1 | 4.33 $\pm$ .00 | 4.26 $\pm$ .00 |
|      | 64  | 94.3 $\pm$ 0.3      | 61.4 $\pm$ 0.1 | 4.32 $\pm$ .00 | 4.27 $\pm$ .00 |
|      | 128 | 84.4 $\pm$ 0.1      | 61.9 $\pm$ 0.2 | 4.31 $\pm$ .00 | 4.28 $\pm$ .00 |
|      | 256 | 79.3 $\pm$ 0.2      | 63.0 $\pm$ 0.2 | 4.31 $\pm$ .00 | 4.29 $\pm$ .00 |

Table 5: Word-level 4-gram repetition rate (word Rep-4) for LangFlow endpoints. Entries are mean  $\pm$  standard error over six seeds.

| Data | NFE | Word Rep-4 ( $\times 10^{-3}$ ) |                 |
|------|-----|---------------------------------|-----------------|
|      |     | Official                        | Ours            |
| OWT  | 8   | 0.11 $\pm$ .01                  | 1.82 $\pm$ .04  |
|      | 16  | 0.51 $\pm$ .02                  | 3.59 $\pm$ .10  |
|      | 32  | 1.54 $\pm$ .06                  | 5.33 $\pm$ .16  |
|      | 64  | 3.95 $\pm$ .08                  | 7.78 $\pm$ .19  |
|      | 128 | 8.38 $\pm$ .19                  | 11.92 $\pm$ .31 |
|      | 256 | 16.91 $\pm$ .41                 | 18.51 $\pm$ .53 |
| LM1B | 8   | 0.07 $\pm$ .01                  | 0.47 $\pm$ .02  |
|      | 16  | 0.18 $\pm$ .02                  | 0.67 $\pm$ .04  |
|      | 32  | 0.34 $\pm$ .04                  | 0.66 $\pm$ .05  |
|      | 64  | 0.57 $\pm$ .05                  | 0.87 $\pm$ .06  |
|      | 128 | 0.81 $\pm$ .05                  | 1.11 $\pm$ .07  |
|      | 256 | 1.11 $\pm$ .10                  | 1.34 $\pm$ .09  |

#### C.4 PRECOMPUTATION AND INFERENCE-TIME SETTINGS

The principal-angle directions for the latent and self-conditioning input projections are computed once from the frozen input weights. The per-step statistics in this subsection use only latent and prediction tensors. For ELF, finite differences measure the first-order model responses to perturbing the self-conditioning input and the prediction supplied to the solver. Their  $2 \times 2$  Gram matrix determines the base transform exponent  $\lambda_R^{(0)}$  and the reference correction scale  $g_T^{(0)}$ . For LangFlow, the full-rank projection geometry (Appendix B.2) gives  $\lambda_R^{(0)} = g_T^{(0)} = 1$ ; the coefficients in Table 6 are effective strengths directly. For ELF, denote the response Gram matrix by  $G$ , the total strength by  $s_0$ , and the relative solver weight by  $\varphi$ . Define  $v = (1 - \varphi, \varphi)^\top$  and  $\mathbf{1} = (1, 1)^\top$ . The resulting scales are

$$\chi = \sqrt{\frac{\mathbf{1}^\top G \mathbf{1}}{v^\top G v}}, \quad \lambda_R^{(0)} = s_0 \chi (1 - \varphi), \quad g_T^{(0)} = s_0 \chi \varphi. \quad (44)$$

We set  $s_0 = \varphi = 0.75$ , fixing the relative first-order response norm of the two interventions.

The empirical per-step statistics in equation 41 use independent trajectory banks. ELF combines four sets of 1,024 trajectories for each model and NFE. Disjoint halves estimate the token mean and average clipped cosine distance on one side and the centered RMS on the other. LangFlow uses 128 independent latents with a disjoint 64/64 split. These quantities summarize the correction in the

fixed reference recurrence. High-resolution step averages are used only in the separate diagnostics of Appendix C.7.

Table 6 lists the inference-time coefficients for all evaluated configurations. Lower NFE generally uses stronger correction, as fewer steps amplify the solver-introduced coupling (Section 5.3). On a single H100 GPU, the offline precomputation for one ELF-B NFE takes 128 seconds at 8 NFE and 401 seconds at 64 NFE. LangFlow precomputes all evaluated NFE in a single pass of 496 seconds (OWT) and 180 seconds (LM1B).

### C.5 CHANGING CORRELATION WHILE PRESERVING EACH NOISE-LEVEL DISTRIBUTION

The coupling experiment uses 256 fresh OpenWebText examples with frozen ELF-B at 8 and 16 NFE. For each example and correlation in  $\{0, 0.25, 0.5, 0.75, 1\}$ , it constructs successive standardized Gaussian corruptions with the specified correlation. This preserves the Gaussian marginal at every evaluation. Within each correlation setting, the unmodified and modified model evaluations receive exactly the same current latent, time, and noise.

Denote the base prediction by  $v_{\text{base}}$  and the prediction after applying one fixed self-conditioning transform by  $v_{\text{sc}}$ . For the fixed target  $y$ , define  $d = v_{\text{sc}} - v_{\text{base}}$  and  $e = v_{\text{base}} - y$ . The reported coefficient is the exact empirical quadratic-risk optimum along this single direction,

$$\alpha^*(\rho) = -\frac{\sum \langle e, d \rangle}{\sum \|d\|^2}, \quad v(\alpha) = v_{\text{base}} + \alpha d. \quad (45)$$

This coefficient measures interpolation along the fixed self-conditioning-induced direction and is used only in this diagnostic. Table 7 reports the continuous optimum and the risk of applying the modification with coefficient one. The optimum decreases monotonically at both NFE, while the same modification applied with coefficient one crosses from lower to higher risk than the base model. The paired  $\rho = 0$  minus  $\rho = 1$  differences in the optimum are 0.58 at 8 NFE and 0.43 at 16 NFE, with 95% intervals  $[0.55, 0.62]$  and  $[0.40, 0.46]$ .

Controls using a preliminary prediction computed from the current noisy state have risk ratios near 1.02; controls without self-conditioning have ratios within  $10^{-4}$  of 1. The quadratic risk decomposition is exact to within  $10^{-8}$ , confirming that the reported coefficients are not affected by numerical artifacts.

### C.6 CONTROL TRANSFORMS WITH THE SAME EIGENVALUES

The experiment in Table 8 disables the solver correction and uses 256 fresh ELF-B generations at 8 NFE. Every configuration supplies the raw prediction to the solver, stores it for the next model evaluation, and makes eight model evaluations. The overlap-aligned, reverse, and random transforms have exactly the same set of eigenvalues and mean strength. Their only difference is the assignment of these eigenvalues to the principal directions.

For a self-conditioning input  $m$ , define

$$\mathcal{E}(m) = \frac{1}{2} \sum_{n \in I_k} \sum_i \rho_i^2 \langle m_n, q_i \rangle^2, \quad (46)$$

where  $q_i$  is principal direction  $i$ . Table 8 reports the cosine similarity between the self-conditioning modification and  $-\nabla \mathcal{E}$ , the decrease in  $\mathcal{E}$  divided by the modification RMS, and both endpoint metrics. Table 9 repeats the three eigenvalue assignments with the solver correction active. The overlap-aligned assignment gives the highest cosine similarity, the largest normalized decrease in  $\mathcal{E}$ , and the lowest two likelihood measures. The configuration with solver correction but no self-conditioning transform separates the gain due to the transform in this matched comparison.

### C.7 REFERENCE AVERAGES OVER ONE SOLVER STEP

Subdividing each solver step into 32 substeps provides a high-resolution reference for the step-average prediction. Across 8, 16, and 32 NFE, the correction  $C_k^*$  scales as first order in step width (fitted exponent 1.04, 95% CI  $[1.02, 1.05]$ ) and its contribution to the latent update scales as second order

Table 6: Inference-time coefficients (Algorithm 1). “—” marks the base value 1 (unchanged).

| System           | NFE     | $\eta_R$ | $\eta_T$ | $\eta_O$ | $\alpha$ |
|------------------|---------|----------|----------|----------|----------|
| ELF-B            | 8       | 3.3      | 1.7      | —        | 0.7      |
|                  | 16      | 2.3      | 0.5      | —        | 0.8      |
|                  | 32      | 2.3      | 1.1      | —        | 0.8      |
|                  | 64      | 1.5      | —        | 1.5      | 0.8      |
| ELF-M            | 8       | 2.3      | 1.5      | 6.0      | 0.8      |
|                  | 16      | 1.3      | 0.9      | —        | 0.8      |
|                  | 32      | 1.7      | 1.7      | 0.7      | —        |
|                  | 64      | 3.3      | 2.5      | 0.8      | 0.8      |
| ELF-L            | 8       | 2.3      | 2.0      | 2.0      | 0.5      |
|                  | 16      | 3.0      | 0.8      | 1.5      | 0.8      |
|                  | 32      | 2.5      | 2.2      | —        | —        |
|                  | 64      | 3.3      | 3.3      | 0.5      | 0.8      |
| LangFlow<br>OWT  | 8       | 2.5      | 1.9      | —        | —        |
|                  | 16, 32  | 1.3      | 1.7      | —        | —        |
|                  | 64, 128 | 1.3      | 1.9      | —        | 0.8      |
|                  | 256     | 1.3      | 1.3      | —        | 0.8      |
| LangFlow<br>LM1B | 8       | 2.5      | 2.5      | —        | —        |
|                  | 16      | 1.3      | 2.5      | —        | 0.9      |
|                  | 32, 128 | 0.6      | 2.5      | —        | 0.8      |
|                  | 64, 256 | 1.3      | 2.5      | —        | 0.8      |

Table 7: Changing temporal correlation while preserving the marginal distribution at each noise level.  $\alpha^*$  is the risk-optimal interpolation coefficient along one fixed self-conditioning-induced prediction direction; the ratio compares prediction risk at coefficient one with the base model.

| $\rho$ | 8 NFE      |            | 16 NFE     |            |
|--------|------------|------------|------------|------------|
|        | $\alpha^*$ | risk ratio | $\alpha^*$ | risk ratio |
| 0.00   | 0.88       | 0.92       | 0.78       | 0.92       |
| 0.25   | 0.79       | 0.94       | 0.69       | 0.94       |
| 0.50   | 0.62       | 0.97       | 0.55       | 0.98       |
| 0.75   | 0.45       | 1.01       | 0.40       | 1.05       |
| 1.00   | 0.30       | 1.07       | 0.35       | 1.11       |

(fitted exponent 2.02, 95% CI [2.01, 2.04]), matching the predicted orders from Section 3.3. The EMA-based estimate from past predictions outperforms equal-RMS permutations at every NFE.

The scaling experiment uses frozen ELF-B at 8, 16, and 32 NFE. Every setting contains 128 fresh trajectories. The first 64 fit one nonnegative coefficient for each estimate derived from prediction history; the remaining 64 evaluate normalized MSE and equal-RMS permuted controls. Confidence intervals use 4,000 sample bootstrap resamples.

Table 10 isolates the self-conditioning transform on 256 fresh ELF-B samples at 16 NFE with the solver correction disabled. Direction-dependent dampening outperforms both the official transform and uniform scaling, confirming that aligning dampening with the principal-angle spectrum improves generation quality.

Table 8: **The overlap-aligned transform outperforms matched direction assignments.** Every solver correction is disabled. Overlap-aligned, reverse, and random use the same retention eigenvalues and differ only in their assignment to principal directions. Alignment is the cosine with the steepest decrease in the overlap objective  $\mathcal{E}$ .

| Configuration   | Alignment $\uparrow$ | $\Delta\mathcal{E}$ / RMS $\downarrow$ | GenPPL $\downarrow$ |
|-----------------|----------------------|----------------------------------------|---------------------|
| Official        | —                    | —                                      | 64.27               |
| Overlap-aligned | <b>0.94</b>          | −40.35k                                | <b>55.64</b>        |
| Reversed        | 0.51                 | −22.94k                                | 73.95               |
| Random          | 0.74                 | −32.15k                                | 59.59               |

Table 9: Self-conditioning transforms with the solver correction active. The three transformed configurations use the same eigenvalues, assigned to principal directions in different orders.  $\mathcal{E}$  is defined in equation 4.

| Configuration                     | Alignment $\uparrow$ | $\Delta\mathcal{E}$ / RMS $\downarrow$ | GenPPL $\downarrow$ |
|-----------------------------------|----------------------|----------------------------------------|---------------------|
| Official SC, no solver correction | —                    | —                                      | 65.23               |
| Official SC, solver correction    | —                    | —                                      | 38.17               |
| Overlap-aligned                   | <b>0.94</b>          | −53.62k                                | <b>34.93</b>        |
| Reversed                          | 0.54                 | −32.52k                                | 40.21               |
| Random                            | 0.75                 | −43.69k                                | 36.92               |

Table 10: Self-conditioning controls on 256 ELF-B samples at 16 NFE with the solver correction disabled. The overlap-aligned transform dampens principal direction  $i$  by  $\max(1 - \hat{a}_k \rho_i^2, \epsilon_{\text{fp}})$  before any global scaling.

| Configuration                                | GenPPL $\downarrow$ | GPT-2 NLL $\downarrow$ | $H$  |
|----------------------------------------------|---------------------|------------------------|------|
| Official                                     | 29.44               | 3.39                   | 5.13 |
| Overlap-aligned dampening, no global scaling | <b>26.56</b>        | <b>3.28</b>            | 5.06 |
| Overlap-aligned dampening, method scaling    | 27.42               | 3.32                   | 5.08 |
| Uniform scaling                              | 29.90               | 3.40                   | 5.13 |

### C.8 WHERE TO APPLY THE SOLVER CORRECTION ACROSS CHECKPOINTS

The main experiment on where to apply the solver correction uses 1,024 matched ELF-B generations at 8 NFE. A second experiment applies the same fixed solver correction to five additional model and NFE settings with 64 samples each and no parameter changes. As shown in Table 11, applying the correction in the solver outperforms the official sampler on both paired likelihoods. Adding the same correction to the next self-conditioning input underperforms the solver assignment.

Table 11: Where to apply the solver correction across checkpoints. The first two differences apply it only in the solver update and compare with the official sampler. The last also adds it to the next self-conditioning input and compares with applying it only in the solver.

| Model | NFE | GPT-2 $\Delta_{\text{solver}}$ | OWT $\Delta_{\text{solver}}$ | OWT $\Delta_{\text{next SC}}$ |
|-------|-----|--------------------------------|------------------------------|-------------------------------|
| ELF-B | 16  | −0.07                          | −0.16                        | +3.00                         |
| ELF-M | 8   | −0.38                          | −0.31                        | +2.51                         |
| ELF-M | 16  | −0.17                          | −0.18                        | +3.45                         |
| ELF-L | 8   | −0.27                          | −0.25                        | +1.63                         |
| ELF-L | 16  | −0.14                          | −0.14                        | +2.75                         |

### C.9 ADDITIONAL ABLATIONS

The remaining tables use ELF-B, 8 NFE, the official SDE noise scale 2.0, official seed 0, and 1,024 matched generations per configuration.

Table 12: Controls for the solver correction on ELF-B with eight model evaluations.

| Solver correction                               | GenPPL ↓     | $H$  |
|-------------------------------------------------|--------------|------|
| History-based direction with mean/RMS scaling   | <b>45.63</b> | 5.14 |
| Common component only                           | 57.21        | 4.98 |
| History-based direction permuted across samples | 49.27        | 4.99 |
| Random direction with equal RMS                 | 71.45        | 5.02 |
| Official (no correction)                        | 74.18        | 5.27 |

Table 12 changes the correction direction. A random equal-RMS direction nearly recovers official GenPPL, confirming that the correction gain requires the specific history-based direction. Removing the sample-specific component raises GenPPL, and permuting it across samples reduces the gain further.

Table 13 compares Untied Self-Conditioning with training-free sampling methods under matched conditions on ELF-B at 8 NFE. Untied Self-Conditioning reaches 45.63 GenPPL versus 56.90 for HiGS (Sadat et al., 2026) and 74.18 for the official sampler. Combining HiGS with ACE (Zhang et al., 2026) yields 65.84, above Untied Self-Conditioning’s solver correction alone (50.04; Table 1b). The paired improvement over HiGS with ACE is 0.36 GPT-2 NLL, with 95% interval [0.35, 0.38].

#### C.10 THROUGHPUT

Throughput is measured with the same compiled model, batch size, number of model evaluations, and generated outputs for the official and modified sampling loops. Both loops are deterministic and produce outputs identical to those used in the generation experiments. Table 14 reports total samples per second and their ratio. The ratio ranges from 0.91 to 0.98, with geometric mean 0.96: the sampler retains over 95% of the baseline throughput without requiring additional model evaluations, since all auxiliary quantities are derived from tensors already available in the sampling loop.

Table 14: Sampling throughput under matched computation. Official and Untied Self-Conditioning use the same compiled model, batch size, NFE, and generated outputs. Retained is method throughput divided by official throughput.

| Model                   | NFE | Official (samples/s) | Untied Self-Conditioning (samples/s) | Retained     |
|-------------------------|-----|----------------------|--------------------------------------|--------------|
| ELF-B                   | 8   | 72.98                | 67.59                                | 92.6%        |
|                         | 16  | 39.39                | 35.93                                | 91.2%        |
| ELF-M                   | 8   | 22.20                | 21.47                                | 96.7%        |
|                         | 16  | 11.78                | 11.38                                | 96.6%        |
| ELF-L                   | 8   | 10.42                | 10.21                                | 98.0%        |
|                         | 16  | 5.53                 | 5.42                                 | 98.1%        |
| Geometric mean retained |     |                      |                                      | <b>95.5%</b> |

Table 13: Matched sampler comparison on frozen ELF-B at 8 NFE with 1,024 matched generations. All methods use the same model, time grid, and initial latent. Momentum (Wizadwongsa et al., 2024), HiGS (Sadat et al., 2026), and ACE (Zhang et al., 2026) are described in Section 6.

| Sampler                  | GenPPL ↓     | $H$  |
|--------------------------|--------------|------|
| Official                 | 74.18        | 5.27 |
| Momentum                 | 74.72        | 5.28 |
| ACE                      | 115.43       | 5.43 |
| HiGS                     | 56.90        | 5.18 |
| HiGS + ACE               | 65.84        | 5.24 |
| Untied Self-Conditioning | <b>45.63</b> | 5.14 |

### C.11 PARAMETER SENSITIVITY

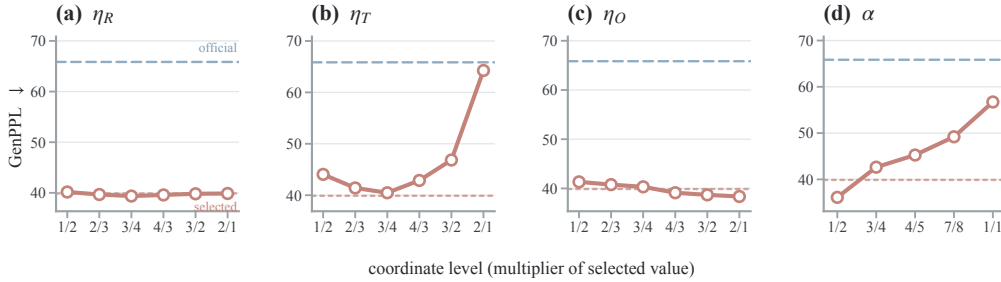


Figure 5: One-at-a-time sensitivity of GenPPL to each coefficient on ELF-B at 8 NFE. Each panel varies one coefficient while holding the other three at the values in Table 6. The dashed blue line marks the official sampler; the dotted red line marks the Table 6 value. All 24 evaluated points reduce GenPPL relative to the official sampler. The curves for  $\eta_R$  and  $\eta_O$  are nearly flat across the full  $[\times\frac{1}{2}, \times 2]$  range; doubling  $\eta_T$  gives the highest GenPPL among the tested variants.

To assess coefficient robustness, we run a frozen one-at-a-time sensitivity study on ELF-B at 8 NFE using 256 independent paired samples that are disjoint from the reported evaluation data. The coefficient grid and evaluation protocol were fixed before any generations were run; all 24 evaluated points are reported without selection.

Every evaluated point improves GenPPL over the paired official sampler. The self-conditioning coefficient  $\eta_R$  varies GenPPL by less than 0.8 over the full  $[\times\frac{1}{2}, \times 2]$  range (39.4–40.2), and  $\eta_O$  varies by 3.0 (38.4–41.3). The solver coefficient  $\eta_T$  is more sensitive: doubling it raises GenPPL to 64.3, still below the official sampler’s 65.8. Reducing  $\eta_T$  to  $\times\frac{1}{2}$  yields 44.0. GenPPL increases monotonically from 36.1 at  $\alpha = \frac{1}{2}$  to 56.7 at  $\alpha = 1$ . Figure 5 visualizes all four coefficient sweeps.

### C.12 PER-SAMPLE QUALITY ON LANGFLOW

Table 15: Paired-sample NLL comparison on LangFlow OpenWebText (six seeds, 1,024 samples per seed per NFE). “Higher NLL” counts samples where Untied Self-Conditioning has higher GPT-2 NLL than the matched official sample.

| NFE | Paired samples | Ours higher NLL | Fraction | Mean $\Delta$ NLL |
|-----|----------------|-----------------|----------|-------------------|
| 8   | 6,144          | 0               | 0.00%    | −2.16             |
| 16  | 6,144          | 0               | 0.00%    | −1.57             |
| 32  | 6,144          | 2               | 0.03%    | −1.03             |
| 64  | 6,144          | 15              | 0.24%    | −0.69             |
| 128 | 6,144          | 127             | 2.07%    | −0.44             |
| 256 | 6,144          | 693             | 11.28%   | −0.20             |

Table 15 reports per-sample NLL comparisons on LangFlow OpenWebText. At 8 and 16 NFE, every one of the 6,144 paired samples has lower NLL under Untied Self-Conditioning.

### C.13 ILLUSTRATIVE TEXT COMPARISONS

We reproduce one matched pair for each model family to illustrate the decoded behavior. LangFlow examples come from the reported six-seed generations; ELF examples come from the disjoint 256-pair bank used in Section C.11. Each pair uses the same initial latent and sample index. The boxes contain the complete decoded generations, including end-of-text markers when present. Non-ASCII characters are transliterated for typesetting; no span is truncated.

**LangFlow–OpenWebText example.** 64 NFE, sample 599,  $\Delta\text{NLL} = -1.66$ . The Ours generation removes the repeated phrase “second place in second place” and sustains article-like segments for longer.

#### Official

<|endoftext|>

All of that in mind, of course, was that calling grand-peat—where Monaco start the Elite line at home—would improve to second place in second place overall in 2017. Ratings were spoiled from the first rankings, because Pabennis tried 19 points outrightly, and so far away to his best for competitive titles, but we quickly never predict the end of a steep, hot year within Giga Champions and one of this year’s McFOR favourites.

So for traveling to Valencia the rest of the year—every, for example—that’s my question of why, despite Sevan had so many promising moments in 2013, every player still felt the same cliché, wondrous pace about his own country, when in the Multiverse, it was Fernandez as a good roommate, not just my own wife and teenagers who had a dedicated, tactical approach.

For as long I went, the confidence I’d found throughout the entire season (that optimism happened now just as soon as I can imagine) was just real for the ideal year for anything. Valencia is there, we’re ahead, and it still feels bad to see it coming yet quickly.<|endoftext|>DSP Spaces, the Jaaharabad Airways Ltd, is building the Qatar port and will connect it to Ahmedabad Airport for the cross-border services.

The port will begin at Mokintabad International Airport and will then connect to Panurbiba National Airport.

Although the airport will be closed, the port will also connect back to Disa, Ag-a-Palad and a public waychard known as Zone Na.

The Panurbiba Center will also serve the so-called Lid Ban log-airlets and outlets as part of the project.

The Lid Ban Center will also serve the conservation of water provided for the port’s special water deck.

Besides the airport, DSP will move around the two-lane branches with an M/6 system and serve up to 2 hundred passengers. The campus or 251 passengers.

Commenting at the project, DSP Chairman, Gindas Kumar, said the project would be “donural home for Ahmedabad people.”

“With the Qatar port terminal, Zone Na will be Ahmedabad’s major link at the border. All investment investments will be made into this airport to make a much smoother environment for Indians,” said Kumar.

“The port will be extended from Zone Na to the Ahmedabad Air location without further delays for shuttle operators to no services from Ahmedabad International Airport. This will be a good haven for noncooperative passenger services and the port will have a fast link to its Qatar border.”

Zone Na Director of visual Group Joan Engmave told reporters that “The port, the arch from Mecca to Danda, was handed over to Qatar, and the frame will soon be out.”

Refaj Airport, Ahmedabad Airport and Matinadh Airport are the new Ahmedabad airports to take advantage of the Qatar proper. The project is another off-use airport such as the Public Satellite airport and well-lands.<|endoftext|>Two weeks ago I waited for Nigel Farage’s half-day residence near Queen Square in Goldstone Paul’s, came on a passport “thats our rise toward the independence vote” and sealed the decision to “born living Europeans that I didn’t tell voters”. “To assure you that this time round, or any other time of year, I must abide by the clear code of fact that I pass along for the visit,” cracked alily. The passport had turned me over to the City. I stood confronting myself on Yorkshire Street, in Goldstone Paul’s, suddenly mentioning Nigel Farage’s MPs as if he had decided to call them “friends” in “Hab that Wh nearby”. I was

deeply informed of this distinction and certainly obliged to hypothesise about it from a source I would only suspect at Ukip if unspecified. It is still unsurprising, though, that the case of Ukip in general was derived from his previously made Southampton speculation. And there are some recent facts to examine.

But what Ukip's main milons, or the compassionate people themselves have told us about their spin, Nigel Farage's change implies an overt and strengthened leadership has almost nothing to do with people. Indeed it is true that many times net immigration left a devaluus stock of workers and brings in desperate people to reject it for access. The standards proposed by EU states concerning net immigration have made it clear that we should never let so many migrants to leave, because of the interests of the migrant and internally rather than the overseas taxpayers; which is precisely why it is common interest that Ukip's extreme leadership has then rightly sent us up into Europe. That past national day, in France, Mr Farage periodically holed himself and European countries<|endoftext|>

### Ours

<|endoftext|>

All of that in mind, of course, was a telling no-brainer, because Monaco won the Champions League in 2014 and slipped back to second place in the Champions League in 2017. Ratings were spoiled from the first round, because the Bavarian crashed 19 points outrightly, and so nobody seemed to know what for sure, because instead they struggled to predict the end of a very, big year within Borussia Dortmund and one of the world's top F1 runners.

So for me anyway, the out of the year (2016, for example) wasn't my question of why, the Bavarian had so many great moments in 2016, every league move, the same flirting, the aerrolic event, etc, but for the Bavarian, it was at last a great day, not just my own wife and children who had a great, great experience.

Despite how close I was, the passion I'd seen throughout the entire season (that has happened here just as often as I can imagine) was just personal for the sake of the thing. Monaco is special, they're strong, and it would be bad to see it come true...<|endoftext|>Hamid Karzai, the Haqqani Taliban headliner, indicated that the United States would have to bury Mr. Karzai for the cease-fire talks.

The new peace agreement, an apparent move by Mr. Karzai to meet with the United States, has been in force after the disputed elections of Sunday. The military has also engaged in an effort to preserve Mr. Karzai's authority holding hold as early as Fallujah.

But on the Islamic State's announcement, Mr. Hamid Karzai, president of the American Parliament, said the U.N. "absolutelyprised" that the United States would have to intervene.

### Photo

Mr. Karzai added that the U.N. announcement about the renewed peace talks should allow the United States to take further steps and exchange its obligations as well as with the United Nations.

Advertisement Continue reading the main story

Mr. Karzai, who had called the peace talks "stalled" after Afghan fighters had rallied hard to take the coup push against Mr. Karzai, said he didn't know if the United States was trading in or if Karzai was able to bury the peace agreements just after carrying out the reconciliation.

Yol. Karzai signed off a deal to rein Talibanate Mr. Karzai's forces to keep Mr. Karzai in power. But Mr. Karzai has also formally cred the agreement and says he won't sign a final peace agreement.

Newsletter Sign Up Continue reading the main story Please verify you're not a robot by clicking the box. Invalid email address. Please re-enter. You must select a newsletter to subscribe to. Sign Up You will receive emails containing news content , updates and promotions from The New York Times. You may opt-out at any time. You agree to receive occasional updates and special offers for The New York Times's products and services. Thank you for subscribing. An error has occurred. Please try again later. View all New York Times newsletters.

Mr. Karzai said the agreement was "a good step toward the peace process" and called the peace agreement "a step solely that I didn't talk about."

On a day that saw conditions in rural Afghanistan triple-levels of temperature, the United States assessed a clear line of terms and amred along to Mr. Karzai's presidency. The decision was turned head over to the United States. Heads of Peace on Sunday dismissed reports that Mr. Sharif's forces had asked Mr. Karzai for talks to prevent his departure. A statement on Sunday said the Afghan government had "confirmed that Mr. Karzai's forces had been in force."

Advertisement Continue reading the main story

Mr. Karzai issued a statement hours after he announced the agreement, rejecting the fact that although the talks had ended, "extraordinary changes had been made for the recent developments."

The Afghan government's main said that, although the United States would resume talks before the parties signed, the agreement's effect dissolved in October and urged Mr. Karzai to stop giving discussions. But it also insisted that the United States would reach a bargaining gap between the parties before talks in Kabul and to compromise its efforts to pass a peace decree by the end of escalating disputes.

"We will never really allow both sides to negotiate," Mr. Karzai's spokesman said. "You don't negotiate a bad relationship when there's clear security."

Mr. Karzai and his troops were, in fact, supporters of both the United States and NATO.

**ELF-B example.** 8 NFE,  $\Delta\text{NLL} = -2.04$ . The Ours generation recovers from token-level collapse into connected sentences about a sustained topic.

#### Official

In conclusion conclusion,,,,, conclusion,, maxim maxim. >> din., din, din, din., am, >>,,,,, >>,, din din,,,,, din. din din,, din,,,,, din.. summarize,,,,,

#### Ours

In this article, a panel asked to determine the tank of the tank that is holding and the amount of tank that the tank is holding. Specifically the insights are determined by the size of the size of the tank tank, the location of the tank containers are holding and temperature, of the forced pressures of the tank held. The study is as described in the last part of the paper. In this article, again, calculations computes different features of tank storage, including 11 percent of the tank, half percent size included, 3/4 of the tank capacity, and the total amount of tank. If you want to use an increase in the tank, what is the total that you need? What can you use to determine how much tank you need? In this case, when you create a package so that once the maximum maximum capacity, the maximum can be estimated as 330 million USD or less because there are 20 storage stations to deliver the total, you will need to add an additional amount of storage per 20 storage containers to deliver the total. As example, if you create a platform tank, it will deliver the total of about 20 tanks all in a day. If you want to create the tank in a specific location, you would also need to create a platform tank. You have to have the tank that stores the containers holding the tank. The platform platform also has a locked pit which dumps the garbage and the device to hold out tank to the tank. This way the whole tank must be in a tight underground pit of the water, and then the tank tank must be filled in water so that it can run properly. As long as there is not enough water to hold the tank, the decay is required to get the perfect tank of the tank. But the major advantage of a platform holding is you use it to store the storage and water and to transport the data flows to foreign systems with a specific storage point. First and foremost, you ensure that the platform storage is exactly that is fit for the system you need. Platform holding is a staple of easily defined and easily integrated into your platform platform. Once you have a platform platform ready, you can create your own platform platform. Platform holding brings in some of the most advanced storage technologies available, and it includes countless different features and technical details that you can use. There are plenty of apps that go to you of how you want to launch your own platform platform, and you can use them often when you need platform holding. The fundamentals of platform holding are one of the fundamental principles of platform holding. The features of platform holding are easily defined and include many different elements, including popular software, programable design tools, intelligent design plugins, color charts, color tables, graphs, data, and customized background effects. They can be specially customized and implemented in custom graphics, some that adapt custom design features for customized applications, such as charts and maps. They can also be specially customized with apps used to create user-based playlists. If you want an tank that is a platform platform, there are features to be used. For example, I remember my reading an avatar book. I remember I reading the Oldboy game, when I was 16, and I was screaming and the kid started to think and I was kill me, and then he pulled her on the floor and shot me down. I didn't want to see her, but I was 13, and I didn't want to look at that much. So I'm shut to sleep, and waste my life fighting and take the life, and let her go. I mean, she's really insane, and I can't stand her. And then I'm all out of love! So this is a great comic book in my life, and I have so much to share. I feel really wasted for reading, too. While there is still more to read, and I'm getting really excited, I'll make some some cool ideas that I find worth reading and reading. If you'd like to share your insight, tell us your thoughts and inspiration, or let us take part. The chance to explore it today, you we'll get to explore all of the beautiful locales where you'll want to hang together, walk the streets, walk the streets, and enjoy amazing games and competitions. Next, you'll have a new sport, our global video game, where you can raise your children, inspire family and friends, and show their faith and heart. In great respect to Andrew Berger, Tettarian ansientist, is a surgid and author of writing the Prophet Creid and War Naraous Abiq Abiq is a

Jewish man who has been out of his own home in Iraq-occupied Islamic Arabia, for for the purpose of the killings of the Taliban while he undertakes the military operations in the nearby Iraq Arab militants forces.