HOW TO GUIDE YOUR LANGUAGE FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a new method to guide flow matching models. Our approach, which we call **probe guidance**, uses the frozen internal states of an existing diffusion model to construct a guidance signal. This works using a similar principle as autoguidance, but eliminates the need for an additional forward pass at inference time and provides a reliable path to ensure that the weak and strong model share similar dynamics. We apply and benchmark this method on continuous diffusion language models, where probe guidance sets a new state-of-the-art performance on unconditional generation. When applied to a 1.7B diffusion language model, probe guidance consistently improves on multiple choice question answering benchmarks. Using our probes, we study the traditional autoguidance setting where the strong model is a weak checkpoint, and find that the weak model must come from a low-entropy region of training. These findings both provide a practical way to improve diffusion language models and shed light on the actual mechanism behind autoguidance, which is currently poorly understood.

1 INTRODUCTION

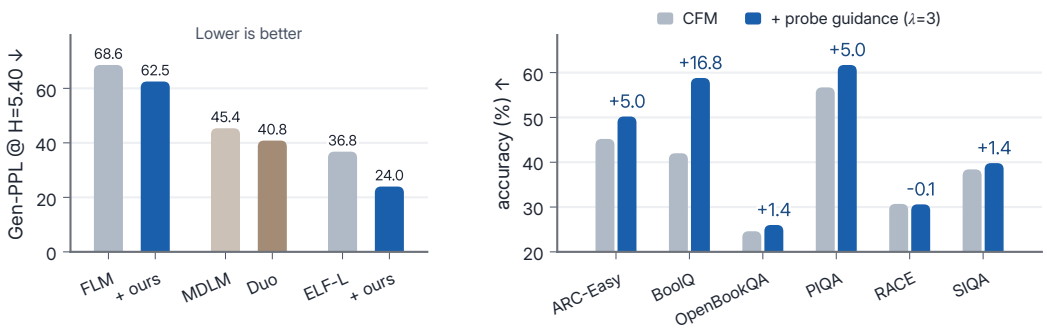
Diffusion and flow-matching models are the dominant paradigm for generation in continuous space, from images to videos to proteins (Esser et al., 2024; Ho et al., 2022; Lai et al., 2025; Wang et al., 2026; Abramson et al., 2024). Their success has inspired efforts to train diffusion language models (dLMs), which apply the principles of diffusion to text generation (Li et al., 2025; Shi et al., 2024; Sahoo et al., 2024) in discrete spaces. These approaches contrast with typical autoregressive models (Radford et al., 2019) by generating tokens simultaneously from a corrupted state, as opposed to one at a time in sequential order. While most language diffusion models are formulated over discrete space, a recent line of work studies *continuous* diffusion models that define an interpolant between a Gaussian prior and data as a continuous random variable (Chen et al., 2026; Lee et al., 2026; Davis et al., 2026; Roos et al., 2026; Hu et al., 2026).

These models inherit many of the advances in continuous diffusion. A particularly powerful technique is guidance, where the output of a diffusion model is adjusted at inference-time for better alignment with a given reward. The most widespread guidance method is classifier-free guidance (CFG), where the guiding signal comes from the difference between a class-conditioned model and an unconditional model (Ho & Salimans, 2022; Chung et al., 2025). More recent work introduced autoguidance, where the guiding signal comes from the difference between a primary model and weaker model (Karras et al., 2024).

Despite the importance of guidance in diffusion, it has been challenging to apply to dLMs. Some dLMs like ELF (Hu et al., 2026) use CFG with self-conditioning, where the conditioning signal is the model's own estimation of the posterior mean. This provides a knob to traverse the quality diversity tradeoff. However, although self-conditioning significantly improves dLM quality (Chen et al., 2026), the reasons why it works are poorly understood; as we show in our work, probe guidance consistently Pareto dominates self-conditioning. Moreover, because self-conditioning requires an additional unconditional pass, it adds training cost and makes scaling harder. FLM (Lee et al., 2026) uses autoguidance with a weak model obtained via isotropic dropout on the strong model; although this slightly improves performance, the weak model does not share the strong model dynamics, which is known to be important in autoguidance (Karras et al., 2024). In both cases, sweeping through guidance strength traces out a Pareto curve between quality and diversity.

This work introduces a new guidance method, which we call probe guidance, and studies its effects in the diffusion language model setting. The guidance signal is obtained from a lightweight MLP

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Unconditional generation. **(b)** Multiple choice evaluation.

Figure 1: Probe guidance improves both data space (FLM) and latent space (ELF) models on (a) unconditional generation and (b) multiple choice evaluation. On unconditional generation, probe guidance takes ELF-L down to a genPPL score of 23 at the OWT dataset entropy 5.40. In multiple choice evaluations, we find improvement or parity across categories.

probe ($\approx 2\%$ of model compute) trained on frozen hidden states inside a dLM. During inference, these probes are used to guide unconditional generation. In this work, we investigate the following:

Efficiency Compared to autoguidance, which requires a full additional pass of a weaker model, probe guidance requires $1.4-2.0\times$ fewer FLOPs. The probes can be trained using $\approx 1\%$ additional training compute.

Unconditional generation: Current continuous dLMs use CFG with self-conditioning or autoguidance to guide generations. Applied to latent models like ELF, probe guidance achieves a state-of-the-art GenPPL score of 23 at entropy 5.40. On data space models like FLM, probe guidance dominates the autoguided Pareto frontier.

Multiple choice question answering: Probe guidance consistently improves the performance of a 1.7B continuous dLM (Davis et al., 2026) across a suite of likelihood evaluations. As a supplementary contribution, we show how to adapt the likelihood estimator to make it amenable to guidance.

Guidance dynamics: Autoguidance has been challenging to achieve in language. We study the training dynamics of dLMs and find that both latent and data space dLM training show a period of rapid entropy increase. We demonstrate that autoguidance requires the weak model to come from the low-entropy region before the climbout.

Practically, probe guidance provides an efficient and reliable way to improve continuous diffusion language model performance. More generally, the use of frozen features provides provides a simple way to control the strength of the weak model and study the requirements for effective guidance.

2 Background and Related Work

Diffusion language models. Diffusion and flow matching models push noise to data by parameterizing an ODE or SDE through a neural network. Clean data is corrupted at varying levels of noise, then a network is trained to estimate the clean data conditioned on the current level of corruption and the noised data. Early work (Ho et al., 2020) formulated this process as a sequence of discrete transitions, while subsequent work on score matching (Song et al., 2020) and flow matching (Lipman et al., 2022) modeled the network as a continuous field between distributions. Explicitly, flow matching constructs a conditional interpolant by sampling $\boldsymbol{\epsilon} \sim p_{\text{prior}}$ and $\mathbf{x} \sim p_{\text{data}}$ and writing $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$, where the standard choice is the linear interpolant $\alpha_t = t, \sigma_t = 1 - t$. Given a neural parameterization of the velocity $\mathbf{v} = \dot{\mathbf{x}} = \mathbf{x} - \boldsymbol{\epsilon}$, one can sample from p_{data} by sampling $\mathbf{x}_0 \sim p_{\text{prior}}$ and running the ODE $d\mathbf{x} = \mathbf{v}\, dt$. A full intro to flow matching is in Appendix B.1. We follow recent work that parameterizes the velocity through the posterior mean (Li & He, 2026).

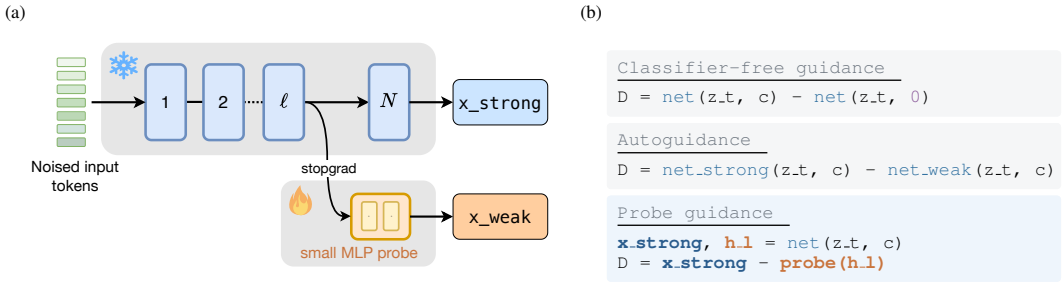


Figure 2: Overview of our method. (a) To train our probes, we detach hidden states from the frozen main model trunk and use an MLP to predict the flow target. This can be done during training, or with a short fine-tuning stage on a frozen backbone. (b) Classifier free guidance and autoguidance require two separate evaluations, whereas probe guidance only requires an evaluation of a small MLP on top of hidden states computed during the forward pass.

$$\mathbf{v} = \frac{\mathbb{E}[\mathbf{x}|\mathbf{x}_t, t] - \mathbf{x}_t}{1-t} = \frac{\mathbf{x}_\theta(\mathbf{x}_t, t) - \mathbf{x}_t}{1-t} \tag{1}$$

This formulation can be optimized using any Bregman divergence, which is particularly helpful in the discrete setting (see Appendix B).

Diffusion language models Diffusion models were generalized to discrete space using discrete state transitions (Austin et al., 2021). More recently, a number of methods have considered how continuous formulations like flow matching can be applied in the discrete setting. Embedding space methods like ELF, Diffusion-LM (Li et al., 2022), and CDCD (Dieleman et al., 2022) encode high dimensional vocabularies in compressed embeddings and perform diffusion in the embedding space. Simplex models like TESS (Mahabadi et al., 2024) and Bit-diffusion (Chen et al., 2022) form diffusion trajectories directly on the probability simplex.

Guidance Guidance can be broadly expressed in the following form, where the particular form of guidance is determined by the guidance vector $D(\mathbf{x}_t, t, c)$.

$$\mathbf{v}_{\text{guided}}(\mathbf{x}_t, t, c) = \mathbf{v}_\theta(\mathbf{x}_t, t, c) + \lambda\, \mathbf{D}(\mathbf{x}_t, t, c) \tag{2}$$

For example, classifier-free guidance (Ho & Salimans, 2022) sets $D(\mathbf{x}_t, t, c) = v_\theta(\mathbf{x_t}, t, c) - v_\theta(\mathbf{x_t}, t, \varnothing)$. Autoguidance (Karras et al., 2024) allows for unconditional sampling by setting $\mathbf{D}(\mathbf{x}_t, t, c) = \mathbf{v}_{\text{strong}}(\mathbf{x}_t, t, c) - \mathbf{v}_{\text{weak}}(\mathbf{x}_t, t, c)$, where $\mathbf{v}_{\text{strong}}$ and $\mathbf{v}_{\text{weak}}$ are strong and weak models. Our work is inspired by autoguidance, which requires that the strong and weak models have correlated dynamics (e.g., isotropically corrupting the model is insufficient to produce a good guidance signal). For this reason, autoguidance is commonly implemented using an early checkpoint of a main model, which fulfills this requirement but doubles the FLOPs at inference time. We discuss similar guidance works in Appendix A.

3 Method

Guidance using frozen states. Given a pretrained dLM, we extract hidden states $\mathbf{h}_\ell$, where ℓ indexes the layer. For a flow model trained with divergence $\mathcal{D}$ (e.g., a cross-entropy loss for FLM or an MSE loss for ELF), we train the probe using the following objective:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x},t,\boldsymbol{\epsilon}}\left[\sum_\ell \mathcal{D}\left(\mathbf{x}_\theta^w(\mathbf{x}_t, t, \ell), \mathbf{x}\right)\right] \qquad \mathbf{x}_\theta^w(\mathbf{x}_t, t) = \text{MLP}[\text{sg}(\mathbf{h}_\ell)] \tag{3}$$

where $\mathbf{x}$ is clean data. In other words, we train our probes (i.e., a weak version of the model) using the hidden states from the main model trunk. The stop-gradient removes the need for balancing multiple losses and prevents the weak path from corrupting the training dynamics of the main trunk. It also allows us to simultaneously train multiple probes on the same model trunk. During inference,



Figure 3: Real text examples using ELF-L and FLM-B. Unguided on left and with probe guidance on right. Errors highlighted in red. Guided ELF-L is notably more fluent than unguided. While FLM-B is a small model, so both guided and unguided models do not read as fluent English, it has fewer spelling mistakes and better sentence constructions (e.g., "the same to whom we usually see" $\rightarrow$ "the same guy".

we integrate using the guided velocity. Explicitly, let $\mathbf{x}_\theta^{\{s,w\}}$ represent the strong (s) and weak (w) outputs of the model. Then

$$\mathbf{x}_{\text{guided}} = \mathbf{x}_\theta^s(\mathbf{x}_t, t) + (\lambda - 1)\left(\mathbf{x}_\theta^s(\mathbf{x}_t, t) - \mathbf{x}_\theta^w(\mathbf{x}_t, t)\right) \tag{4}$$

and the velocity update is obtained using Equation 1. We adopt the convention in which $\lambda = 1$ recovers the unguided strong model and $\lambda > 1$ extrapolates along the guidance direction $\mathbf{x}_\theta^s - \mathbf{x}_\theta^w$; Figure 2 shows our training setup and compares the inference cost of probe guidance against CFG and autoguidance.

Training and inference Because training the probes does not affect the gradients of the main model trunk, the cost of training our probe guidance model is the gradient-free cost of running the model trunk up to the exit layer, plus the attention-free cost of training an MLP probe. As a result, the cost of training these probes measured via the number of training steps and the cost per training step is minimal. At inference time, the base hidden state is always computed for the primary model output, so the extra inference cost is only the additional MLP pass. This contrasts with standard autoguidance, which requires two forward passes: one for the strong model and one for the weak model (which is usually an equivalent compute version of the strong model). All of our experiments do prediction in $\mathbf{x}$ space and convert to velocity.

4 Unconditional probe guidance experiments

In this section, we demonstrate how probe guidance can improve unconditional generation across both latent and data space language diffusion models. We consider two main representative methods: ELF, a latent space model trained in a T5-latent space and FLM, a data space model that operates directly on raw tokens (Hu et al., 2026; Lee et al., 2026). In both cases, we use released checkpoints to train probes and mirror the inference settings described in the main paper (see Appendix C).

Datasets and evaluations We measure unconditional generation using generative perplexity (GenPPL) and unigram entropy. GenPPL computes the perplexity of generated sequences under a GPT2-L model (lower GenPPL means higher quality). Unigram entropy computes the entropy of the empirical distribution implied by the token counts. While prior works often report a single GenPPL value, (Pynadath et al., 2026) points out that these metrics are often misleading, and only

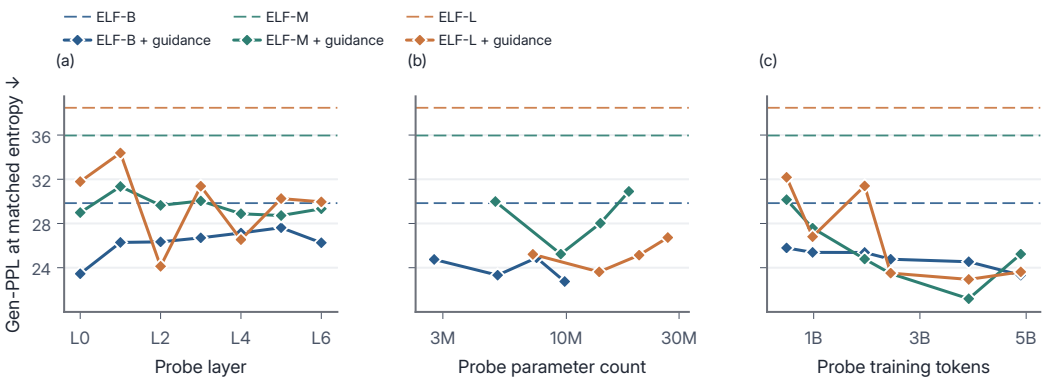


Figure 4: **Probe architecture ablations.** We study (a) which layer to attach the probe, (b) the strength of the probe in FLOPs, and (c) the amount of training per probe. We consider GenPPL at entropy $H = 5.2, 5.3, 5.4$ for ELF **B**, **M**, and **L** respectively. Unguided performance is dashed for each model size. We find remarkably steady performance across model sizes. All configurations beat the unguided baseline (dashed). Relatively small probes are still effective, and once the probe layer and size are fixed we find that performance is quite steady with respect to the amount of training. Unguided model baselines are dashed, FLM results in Appendix D

correspond to dominance in a restricted part of the phase space. To resolve these ambiguities, we plot the full Pareto curve to compare different metrics. To trace the Pareto curve, we vary self-conditioning strength for ELF. For FLM, we follow the prescription in Lee et al. (2026) where we guide the main model using a weaker model obtained by applying dropout to the weights. We perform all experiments using the OpenWebText (OWT) dataset and report GenPPL/entropy by generating 1000 samples.

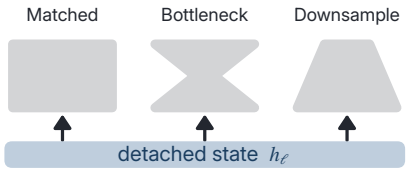
Training and inference. Our probe training mimics each respective model's training. For ELF, we use a logit-normal noise schedule with mean -1.5 and CFG strength sampled log-uniformly over [0.5, 5.0], and we generate samples using the recommended 32 or 64 step "SDE"[1] sampler. In our probe guidance curves, we set the self-conditioning CFG strength to 1.0 (no self-conditioning guidance) and sweep through the guidance strength. For FLM, we use the released -B model, which contains 170M parameters over the GPT-2 vocabulary (50k tokens). Due to the large number of parameters in the unembedding head, we share the unembedding head with the probe and backpropagate through without updating parameters. For both ELF and FLM, we train probes with the AdamW optimizer with learning rate 0.0003 and batch size 256.

4.1 Ablations

In this section, we systematically explore the design space of probe guidance. We center our investigation around the following axes: the probe position along the trunk (i.e. the depth at which the probe is attached), the connector between the main trunk and the probe, the compute in the probe itself, and the number of training FLOPs used to train the probe. Fundamentally, all of these axes are different ways of controlling how closely the strong and weak model align, but they each come with their own biases (for instance, a weak probe at the penultimate layer will be better than a very strong probe at an early layer).

Probe positioning: We first study the impact of probe positioning. This is important because early layers typically provide little contextual information, while late layers provide representations that already encode the main model trunk's prediction, so the weak and strong models collapse and the guidance direction vanishes. We find that placing the probes at early layers consistently leads to the best performance (measured by GenPPL at fixed entropy). This has two implications. First, it reduces the cost of training the probes, because the dominant cost (the main model forward pass) is restricted to the first few layers. Second, it suggests that a good weak model for guidance may

[1]Not a true SDE, it has different marginals than the ODE



ELF-L		FLM-B	
Method	Gen-PPL	Method	Gen-PPL
Baseline	*36.8*	*Baseline*	*79.3*
Downsample	**23.2**	Downsample	85.9
Matched width	**24.5**	Matched width	**65.2**
Bottleneck	30.7	Bottleneck	71.1

Figure 5: (Left) Schematic of our three connector designs. Tables report Gen-PPL at H=5.40 for ELF (center) and FLM (right).

require very little computation, which contrasts with current approaches where the weak model is equally expensive as the strong model (Karras et al., 2024).

Trunk-probe connector: We next study the transition between the trunk and the probe. Inspired by works that add a bottleneck layer akin to classical manifold learning (Li & He, 2026; Hinton & Salakhutdinov, 2006; Vincent et al., 2008), we consider three choices: matching the probe width to the model width, setting the probe width to the model width with an initial bottleneck layer, and setting the probe width *smaller* than the model width. These are depicted schematically in Figure 5.

While all three designs work well and outperform the base model, we find dynamics differ between ELF and FLM. Latent space models like ELF can benefit from aggressive downsampling, which we attribute in part to the fact that the weak model does not need to be particularly good, so a small amount of additional computation suffices. FLM, however, clearly benefits from matching widths. For this reason, we default to matched width and discuss further in Appendix D.

Probe FLOPs: Once the bottleneck is fixed, how should compute within the probe be allocated? Because our probes are MLPs and do not mix information across tokens, parameter count is directly proportionate to FLOPs. Figure 4b shows these sweeps. We consistently find that a small number of FLOPs is sufficient, and the optimal number of FLOPs is flat between three and seven percent of main model trunk compute. While we expect that the optimal probe design will be idiosyncratic to data and model architectures, the stability of these results is reassuring; it suggests that a small probe can provide a useful guidance signal.

Probe checkpoint: Finally, we study how the probe checkpoint evolves during training. We sweep guidance at different checkpoints and consistently observe a minimum after a few thousand steps. We find that training probes for $\sim 10^9$ tokens (about 10^4 steps on 8 H100 GPUs) suffices. By comparison, training the dLM trunk is typically $\sim 10^{11}$ or more tokens. Practically speaking, the relatively low number of steps required for good guidance is a desirable quality. Moreover, for a fixed probe position, we do not need to optimize the checkpoint; training longer does not significantly harm performance.

Finding. Probes should be placed at early layers, and trained for a short run of a few billion tokens over a few thousand steps. The probes themselves only need a small fraction of total model compute.

Probe guidance establishes a new Pareto frontier. Integrating these results reveals that probe guidance consistently establishes a new Pareto frontier across model sizes and model types, meaning the probe-guided models achieve higher quality for the same level of diversity. Pareto curves expressing the quality-entropy tradeoff are shown in Figure 6. At $H = 5.40$ (the approximate entropy of the OWT dataset), probe guidance improves GenPPL from $38 \to 23$ on ELF-L. The FLM-B base model improves Gen-PPL by 16 points with a lower inference overhead compared to dropout autoguidance. We show sample text in Figure 3, with further examples in Appendix G. At the same entropy, ELF-L is noticeably clearer with fewer spelling mistakes and grammatical errors.

5 Probe guidance improves Multiple Choice Question Answering

We next evaluate the effect of probe guidance on multiple choice question answering (MCQA), where the objective is to measure the zero-shot performance of a model on multiple choice questions. MCQA is important both as an evaluation metric for large models and as a step beyond simple

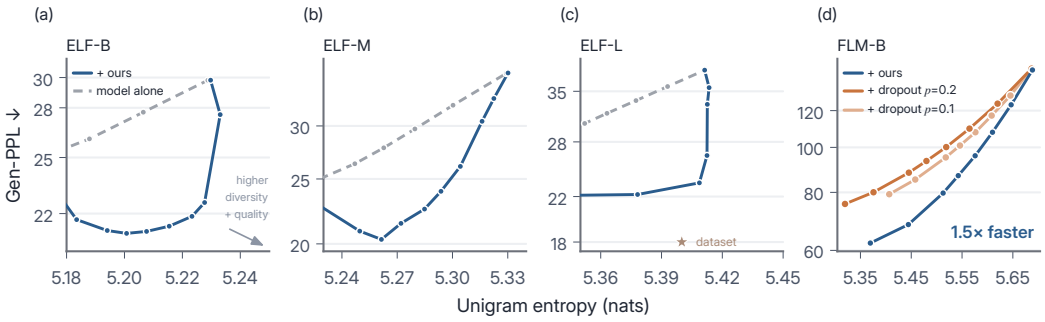


Figure 6: Across latent (a,b,c) and data space (d) released models, probe guidance consistently improves the generation quality gap. On ELF, each point represents a different CFG strength; on FLM, each point is a different autoguidance strength using a weak model obtained via dropout. Probe guidance consistently improves over the base performance. On FLM (d), probe guidance is $\approx 1.5\times$ cheaper than the prescribed dropout autoguidance and improves performance by nearly 20 genPPL points (see Appendix E.1 for discussion).

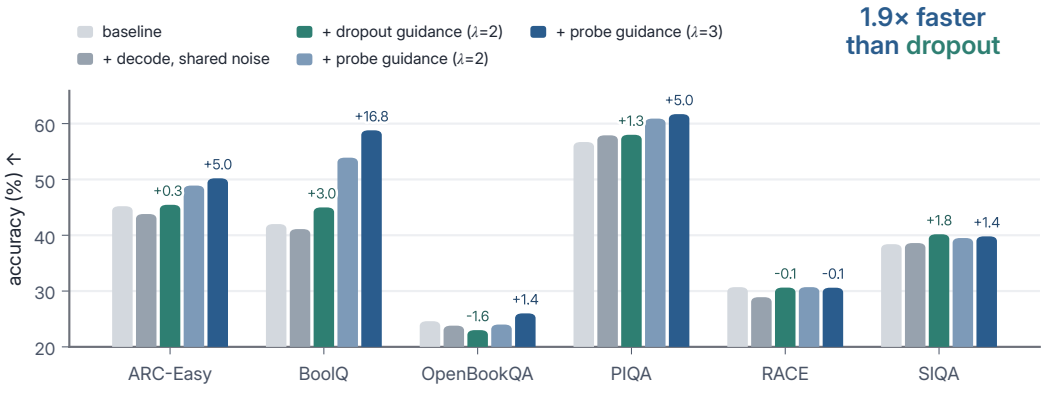


Figure 7: Performance on MCQA across benchmarks. We show the baseline reproduced performance of the flow matching model, performance with shared noise, and performance with guidance. Guidance improves MCQA across all benchmarks and closes the gap between continuous and discrete dLMs. As suggested in FLM, we include guidance based on dropout of the base model, which generally underperforms probe guidance and costs twice the FLOPs. Tabular results in Table 2.

unconditional generation. We build on (Davis et al., 2026; Roos et al., 2026), which scales a class of continuous dLMs called Categorical Flow Models (CFMs) to 1.7B parameters and trains on the Nemotron pretraining dataset (Su et al., 2025). We train probes using our prescription in Section 4. Figure 8a shows unconditional performance, where we develop the CFM Pareto curve by using dropout autoguidance. As with FLM, probe guidance Pareto dominates the base performance without incurring additional FLOP costs.

Because diffusion models do not natively provide a tractable likelihood, evaluating zero-shot performance requires a likelihood proxy. For an interpolant $I_t = \alpha_t x_1 + (1-\alpha_t)\epsilon, \epsilon \sim \mathcal{N}(0,1)$, (Davis et al., 2026) suggests the following proxy

$$-\log p_\theta(x) \;\le\; \mathbb{E}_{t\sim U[0,1],\, x_0\sim\mathcal{N}(0,I)}\left[\frac{2\dot{\alpha}_t\,\alpha_t}{(1-\alpha_t)^3}\,\ell_{\mathrm{CE}}\big(e_x,\,\pi_\theta(x_t,t)\big)\right] \;+\; \ell_{\mathrm{CE}}\big(x,\,\pi_\theta(e_x,1)\big) \quad (5)$$

The first term is the signal-to-noise weighted time integral of the denoising cross entropy, and the second term accounts for the final readout. In practice, the weighting of the first term is unstable due to the $1/(1-\alpha_t)$ factor; (Davis et al., 2026) therefore uses an unweighted sum over cross entropies, and drops the decoding term

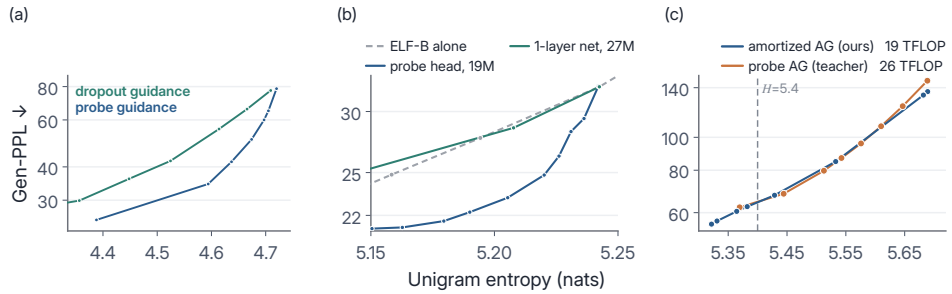


Figure 8: (a) Unconditional CFM performance with probe guidance. (b) Autoguidance with an ℓ layer model vs probe guidance at layer ℓ. (c) Amortizing the probe into FLM provides equal performance as manual guidance.

Guidance shifts away from the trained distribution, which amplifies errors in the final decoding step. For this reason, we construct a new guided scoring mechanism in Equation 6 which maintains the unweighted cross entropy integral but keeps the final decoding term. See Appendix F for discussion.

$$S_\theta(x) \;=\; -\,\mathbb{E}_{t\sim U[0,1],\, x_0\sim\mathcal{N}(0,I)}\Big[\ell_{\mathrm{CE}}\big(x_j,\, \tilde{\pi}_\theta(x_t,t)\big)\Big] - \ell_{\mathrm{CE}}\big(x_j,\, \tilde{\pi}_\theta(e_x,1)\big) \tag{6}$$

Here, $\tilde{\pi}$ represents the *guided* probability estimation. We evaluate the guided model using our new estimator on a standard suite of multiple choice question answering tasks across reasoning and reading comprehension: ARC-e (Clark et al., 2018), BoolQ (Clark et al., 2019), SIQA (Zadeh et al., 2019), RACE (Lai et al., 2017), OBQA (Mihaylov et al., 2018), and PIQA (Bisk et al., 2020), consistent with other large diffusion model methods (Davis et al., 2026; Sahoo et al., 2026). In Figure 7, we measure the effect of probe guidance on these metrics, as well as the new scoring estimator that we introduce. Adding guidance consistently improves all metrics, including +16.8 points on BoolQ and +1.4 on the challenging SIQA benchmark. The addition of guidance brings CFM close to parity with leading discrete diffusion models like Duo and MDLM, see Table 2.

6 Adapting Autoguidance to Language

Despite the importance of guidance in computer vision, applying similar methods in language has proven challenging. The reasons why both autoguidance and CFG work are still poorly understood (Bradley & Nakkiran, 2024; Li et al., 2026). Although motivated by the gamma sharpened distribution ($\tilde{p} = p^\gamma \;\to\; \nabla \log \tilde{p} = \gamma \nabla \log p$), it can be shown that CFG does not preserve the gamma sharpened marginals. Indeed, to our knowledge, no method to date has successfully applied standard autoguidance (i.e., with a weak checkpoint) to continuous language diffusion models. In this section, we show how studying probe guidance leads to insights for standard autoguidance.

How does probe guidance compare with autoguidance? We first contrast probe guidance against the standard autoguidance setting where the weak model is an early checkpoint (Karras et al., 2024). We train ELF-B and FLM-B models from scratch and guide the model using early checkpoints. Figure 9 shows both genPPL during training and autoguided curves. Perplexity starts off low in a degenerate entropy region, where the model repeats phrases continuously resulting in pathologically low diversity. This is followed by a steep climb to high entropy and a smooth descent as the model continuously improves.

In our probe guidance experiments, we found that early probes provided the best guidance signal, even though the probe generations were quite poor. This suggested that the correct way to guide a dLM was with a low-entropy model. We hypothesized that the optimal guidance checkpoint for traditional autoguidance would immediately precede the entropy climbout, and that the effect of guidance is to pull *away* from this low entropy region. We validated this hypothesis on both ELF and FLM (Figures 9 and 13). Interestingly, we observe identical behavior for both approaches. Despite the fact that these two models operate on very different spaces (i.e. latent space for ELF and discrete data space for FLM) and that the entropy climbout occurs after a different number

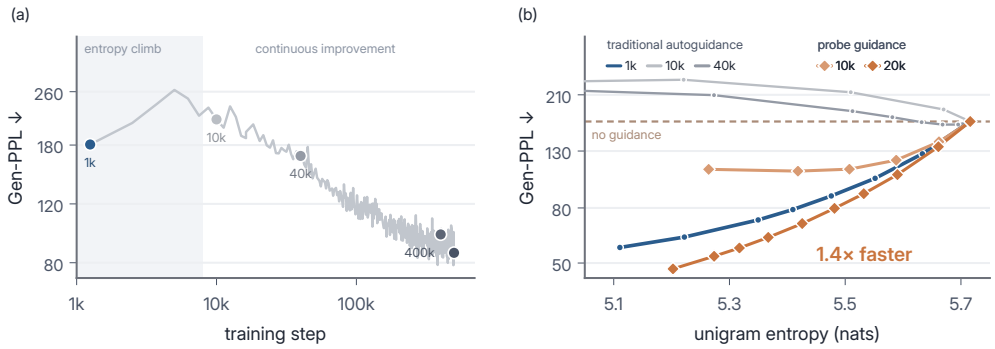


Figure 9: Effect of autoguidance using weak checkpoints on FLM (ELF in Appendix). We make two observations. First, autoguidance only succeeds with a weak checkpoint preceding the entropy climbout (in **blue**). Later checkpoints (**in gray**) all fail at guiding. Second, an advantage of **probe guidance** is that once a probe is positioned, the dynamics are comparatively stable.

of training tokens: low entropy checkpoints work for traditional autoguidance, while every post entropy climbout checkpoint collapses. This empirical finding may account for why autoguidance has been difficult to apply in the language model setting. When training a DiT for 100k steps, a 30k step checkpoint would provide a reasonable guidance signal. In language, this occurs well past the entropy climbout for both ELF and FLM.

Finding. Autoguidance for dLMs should use weak models focused on the low-entropy, low genPPL region. Weak models from after the entropy climb that are more similar to the strong model in dynamics consistently lead to overguidance.

Probe guidance is not equivalent to a small transformer. (Karras et al., 2024) studies developing a weak model by both taking an earlier checkpoint and reducing model size. For a probe anchored at layer ℓ, we next study how the guided dynamics compare to guidance using a separately trained transformer of depth ℓ. Figure 8 compares guiding an ELF-B model using a single layer transformer against using a probe after one layer; the probe traces out a much stronger Pareto frontier, while the small transformer can barely guide the larger model.

Amortization. Although the FLOP increase in running the additional probe is small, it may still be undesirable in some settings. We find that probe guidance can be easily distilled into a new model, requiring less than 10^4 steps. Figure 8 shows that we can capture the full probe guidance benefit.

7 Conclusions

We have introduced a new guidance method for diffusion language models that steers a frozen backbone with a lightweight MLP probe trained on its own detached hidden states. Our approach provides a straightforward way to inherit backbone dynamics from a larger model, requires a small amount of additional training and inference compute, and achieves state-of-the-art performance on unconditional generation. This method is effective across dLMs (latent and data space) and improves on large scale question-answering tasks. Although we study it in the language case, it is domain agnostic and targets the unconditional distribution, offering a further advantage over CFG. Studying the probes sheds further light on the standard autoguidance setting, where we have identified a key criterion for guiding dLMs.

Despite the strength of this approach, continuous dLMs still underperform standard autoregressive language models, and the reasons why guidance works at all remain not fully understood. These are both important areas for future research. We believe probe guidance provides a powerful tool as we bring the controllability and efficiency of modern diffusion models into natural language paradigms.

AI USE STATEMENT

In this work, we have used generative AI tools to generate code for experiments, for figure generation, and to explore hypotheses in developing our likelihood estimator. We have not used generative AI to create synthetic datasets, formulate mathematical claims, or interpret results. The remainder of the required disclosure tasks do not apply to this work. We have reviewed AI assisted work – in particular, by manually inspecting samples and evaluations – and take responsibility for the final content of this work.

ETHICS STATEMENT

As with other advances in language modeling, improved generative quality and efficiency can enable both beneficial and harmful applications. We do not see specific ethical concerns that need to be highlighted.

REPRODUCIBILITY STATEMENT

Although we train FLM and ELF models for our autoguidance runs, our primary results on probe guidance use publicly available checkpoints. We include all hyperparameters required to reproduce our experiments in Table 1.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 51304–51323, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

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

Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. Cfg++: Manifold-constrained classifier free guidance for diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 30824–30850, 2025.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Oscar Davis, Anastasiia Filippova, Pierre Ablin, Victor Turrisi, Amitis Shidani, Marco Cuturi, and Louis Béthune. Scaling categorical flow maps. *arXiv preprint arXiv:2605.07820*, 2026.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Zixuan Fu, Chong Wang, Lanqing Guo, Kailai Zhou, Jiahao Nie, and Bihan Wen. A frozen pixel-space diffusion model can guide itself with its own samples. *arXiv preprint arXiv:2607.29122*, 2026.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

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

Jerry Y Huang, Justin Lin, Sheel Shah, Kartik Nair, and Nicholas M Boffi. How to guide your flow: Few-step alignment via flow map reward guidance. *arXiv preprint arXiv:2604.27147*, 2026.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024.

Chieh-Hsin Lai, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, 2025.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 785–794, 2017.

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

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, 2025.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

Xiang Li, Rongrong Wang, and Qing Qu. Towards understanding the mechanisms of classifier-free guidance. *Advances in Neural Information Processing Systems*, 38:62835–62889, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Rabeeh Karimi Mahabadi, Hamish Ivison, Jaesung Tae, James Henderson, Iz Beltagy, Matthew E Peters, and Arman Cohan. Tess: Text-to-text self-conditioned simplex diffusion. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2347–2361, 2024.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. Generative frontiers: Why evaluation matters for diffusion language models. *arXiv preprint arXiv:2604.02718*, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Daan Roos, Oscar Davis, Floor Eijkelboom, Michael Bronstein, Max Welling, Ismail Ilkan Ceylan, Luca Ambrogioni, and Jan-Willem van de Meent. Categorical flow maps. *arXiv preprint arXiv:2602.12233*, 2026.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Subham Sekhar Sahoo, Jean-Marie Lemercier, Zhihan Yang, Justin Deschenaux, Jingyu Liu, John Thickstun, and Ante Jukic. Scaling beyond masked diffusion language models. *arXiv preprint arXiv:2602.15014*, 2026.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2459–2475, 2025.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Yuyang Wang, Jiarui Lu, Navdeep Jaitly, Joshua Susskind, and Miguel Angel Bautista. Simplefold: Folding proteins is simpler than you think. In *International Conference on Learning Representations*, volume 2026, pp. 32027–32054, 2026.

Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. Social-iq: A question answering benchmark for artificial social intelligence. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8799–8809. IEEE, 2019.

Xingyu Zhou, Qifan Li, Xiaobin Hu, Hai Chen, and Shuhang Gu. Guiding a diffusion transformer with the internal dynamics of itself. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11536–11545, 2026.

A Related work

A variety of works have studied different forms of guidance. We highlight two related approaches here. Internal guidance (Zhou et al., 2026) takes a similar approach where the weak model is read from an internal state, but allows backpropagation into the main model trunk. This approach demonstrates strong performance on ImageNet generation, but (1) requires that the loss value for the weak branch be carefully tuned, because it affects the main trunk, and (2) requires training a whole model from scratch for each run. Moreover, it is not clear how to scale; successful internal guidance at the ℓth layer for a small model does not translate to a larger model. By training small probes on top of a trunk, probe guidance makes it easier to systematically explore the full design space and reason about scaling, as we do in our ablations.

Concurrent work (Fu et al., 2026) also studies using frozen states. This focuses primarily on pixel-space generation, and the probes involve additional attention layers with inter-token communication. We focus primarily on the diffusion language model setting with attention-free probes.

While our version of probe guidance does work on images, the diffusion language model setting is particularly interesting because there is no classifier-free guidance equivalent. CFG is exceptionally powerful in image modeling and difficult to beat, but is somewhat specific to ImageNet. In autoregressive modeling, models benefit from strong unconditional performance from an initial pretraining stage, which suggests that finding unconditional guidance methods may be an important step in building strong dLMs.

A separate direction studies reward guidance or reward alignment. Here, the goal is to align samples to some external reward function $R(x)$. Works like FMRG (Huang et al., 2026) and Universal Guidance (Bansal et al., 2024) adjust the sampling trajectory during inference using the signal from an independent reward. Despite similar nomenclature, the methods target distinct but overlapping goals. Guidance methods like classifier-free guidance and autoguidance are primarily concerned with image quality, while reward guidance methods usually have an external reward function and either explicitly or implicitly sample from the reward-tilted distribution.

B Mathematical formulations

B.1 Overview of flow matching

x-prediction and the velocity map. For the linear interpolant $\mathbf{x}_t = t\,\mathbf{x} + (1-t)\,\boldsymbol{\epsilon}$, the marginal velocity is $\mathbf{v} = \mathbf{x} - \boldsymbol{\epsilon} = (\mathbf{x} - \mathbf{x}_t)/(1-t)$. Parameterizing the network to predict the posterior mean $\mathbf{x}_\theta(\mathbf{x}_t, t) \approx \mathbb{E}[\mathbf{x} \mid \mathbf{x}_t, t]$ and substituting recovers Equation 1. Because the regression target is clean data $\mathbf{x}$, the objective may be written with any Bregman divergence $D_\psi(\mathbf{x}_\theta, \mathbf{x})$: the squared error ($\psi = \frac{1}{2}\|\cdot\|^2$) for the latent ELF model, and the cross-entropy (ψ the negative entropy, over the one-hot simplex) for FLM. In both cases the training target is the clean datapoint, which stabilizes optimization over large vocabularies.

Guiding in $\mathbf{x}$-space equals guiding in velocity. Since the velocity is an affine function of the prediction, any affine combination of $\mathbf{x}$-predictions maps to the *same* affine combination of velocities. For our guided field,

$$\mathbf{v}_{\text{guided}} = \frac{\mathbf{x}_{\text{guided}} - \mathbf{x}_t}{1-t} = \frac{\mathbf{x}_\theta^s + (\lambda-1)(\mathbf{x}_\theta^s - \mathbf{x}_\theta^w) - \mathbf{x}_t}{1-t} = \mathbf{v}^s + (\lambda-1)(\mathbf{v}^s - \mathbf{v}^w). \tag{7}$$

Constructing the guided prediction in $\mathbf{x}$-space and converting is therefore identical to applying the guidance directly to the velocity, which is why we build the guided field in $\mathbf{x}$-space throughout. At $\lambda = 1$ this reduces to $\mathbf{v}^s$, the unguided strong model.

Any Bregman divergence optimizes flow matching. When our prediction target is simply $\mathbb{E}[\mathbf{x}|\mathbf{x}_t, t]$, we can use any Bregman divergence to optimize the model. This follows the known fact that the minimizer of a Bregman divergence is the posterior mean. This becomes relevant because it justifies models like FLM, which use the cross-entropy as a loss function (the Bregman divergence corresponding to the negative entropy).

C EXPERIMENTAL DETAILS

Backbones. We train probes on publicly released checkpoints without modifying them: ELF-{B,M,L}-owt (an embedded T5-latent flow; 105M/342M/652M parameters) and FLM-B (a 170M-parameter one-hot flow over the GPT-2 vocabulary). The backbone is frozen throughout and its hidden states are detached before the probe.

Probe architecture. Each probe is an MLP: N residual blocks of the form $\mathbf{x} + W_2\,\mathrm{SiLU}(W_1\mathbf{x})$ with inner expansion factor e, and an output projection to the target space (the T5 latent for ELF; the shared, frozen unembedding for FLM). Unless noted, N=3 and e=4 ($\approx$ 15M parameters on ELF-L, a 2.4% forward-FLOP overhead). The width and depth ablations vary e and N respectively; because the probe is a pure MLP, both its FLOPs and its parameter count scale as $N \times e$.

Training. Probes are optimized with AdamW (learning rate 3×10^{-4}, weight decay 0.01 on weight matrices only), a 1000-step linear warmup to a constant learning rate, gradient-norm clipping at 1.0, and global batch size 256, for 20k steps on OpenWebText packed to length 1024. Only the probe updates; the frozen backbone contributes no gradient. Unless stated otherwise the weak endpoint used for guidance is the step-10k checkpoint (see the weak-model ablation).

Inference. All ELF samples use a 32-step SDE sampler with γ=1.5 and EMA weights. For the CFG baselines we sweep the self-conditioning CFG scale; for the guidance curves we fix the self-conditioning CFG scale to 1.0 and sweep the guidance scale λ. We report Gen-PPL (scored by GPT-2-Large) and mean unigram entropy over 1000 generations at seed 42. Real-data reference points are computed with the identical metric code on held-out OpenWebText.

C.1 FROM SCRATCH TRAINING DETAILS

We also train FLM and ELF models from scratch for our autoguidance experiments. We identically follow the settings from the source papers in training, detailed below. Unless explicitly noted otherwise, the results in this paper are probes on the released checkpoints, not the trunks trained from scratch. See Table 1 for detailed hyperparameters.

FLM: We train models at -B, -M, and -L scale. We use the standard FLM setup, which is a DiT backbone with adaLN time conditioning, dropout 0.1, and untied input and output embeddings. We optimize with AdamW (learning rate 3×10^{-4}, weight decay 0, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$) under a constant schedule with 2500 warmup steps, a global batch of 512 sequences ($\approx$!5.2×10^5 tokens per step), gradient clipping at 1.0, bf16 precision, and an EMA of the weights with decay 0.9999.

B is trained for 500k steps ($\approx$ 30 epochs). Although we set M and L to train for 1M steps ($\approx$ 60 epochs), we observed instabilities after approximately 800k steps, so we used the last stable checkpoint. We use 'torch.compile' in training and DDP across 8×H100 nodes. We eval using a 256-step Euler ODE solver. We briefly explored training Muon variants (learning rate 0.02 on 2-D parameters, 3×10^{-4} on embeddings, the output head, and 1-D parameters); all of them collapsed to low entropy early and recovered only to high entropy at high perplexity. Thus, all reported FLM results use the AdamW runs.

ELF: We train ELF at -B, -M, and -L scales. We follow the setup in (Hu et al., 2026): in short: we train with Muon at learning rate 2×10^{-3}, weight decay 0, global batch size 512 sequences, 0.5 epochs of warmup, bf16 precision, EMA 0.9999, and 5 epochs, approximately 95k steps.

D ADDITIONAL ABLATIONS AND EXPERIMENTS

This section describes select additional ablations, some of which are referenced in the main text.

Choice of connector We motivate the connector choice as follows. The bottleneck connector is a plausible design choice because bottleneck layers are known to be helpful in diffusion modeling, particularly when the output space is high dimensional Hu et al. (2026); Li & He (2026). The downsampling layer is advantageous because it means most of the probe computation can be done in a lower dimensional space. In our experiments, bottleneck connections were clearly harmful.

Table 1: **Training hyperparameters.** Depth/width/heads are listed as B/M/L. FLM sequences are packed to full length, ELF sequences are padded. Probe widths are matched to the backbone hidden size, so the probe's input projection is an identity; the ELF probe projects to the T5-small dimension (512) while the FLM probe reuses the backbone's own frozen vocabulary head.

(a) ELF from scratch

Parameter	Value
Depth	12/24/32
Width	768/1056/1280
Heads	12/16/16
Encoder	T5-small
Bottleneck	128
Seq. length	1024
Time sched.	Logit normal
sc-CFG range	$[0.5, 5]$
Optimizer	Muon
Learning rate	$2 \cdot 10^{-3}$
Weight decay	0
Global batch	512
Warmup	0.5 epoch
Epochs	5 ($\approx$95k steps)
EMA	0.9999
Precision	bf16

(b) FLM from scratch

Parameter	Value
Depth	12/24/36
Width	768/1024/1280
Heads	12/16/20
Cond. width	128
Dropout	0.1
Tokenizer	GPT-2
Vocab	50258
Seq. length	1024
Loss	softcap. CE
Softcap	30
Optimizer	AdamW
Learning rate	$3 \cdot 10^{-4}$
Weight decay	0
β_1, β_2	0.9, 0.999
ϵ	10^{-8}
LR sched.	const.+warmup
Warmup	2500
Global batch	512
Micro batch	16/8/4
Grad. clip	1.0
Steps	500k/1M/1M
EMA	0.9999
Precision	bf16

(c) Probe training

Parameter	ELF	FLM
Backbone	frozen	frozen
Blocks n_ℓ	3	3
Expansion e	4	4
Width	matched	matched
Input proj.	identity	identity
Output head	$\to 512$	shared
Loss	v-MSE	softcap. CE
Optimizer	AdamW	AdamW
Learning rate	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$
Global batch	256	256
Grad. clip	1.0	1.0
Steps	20k	20k

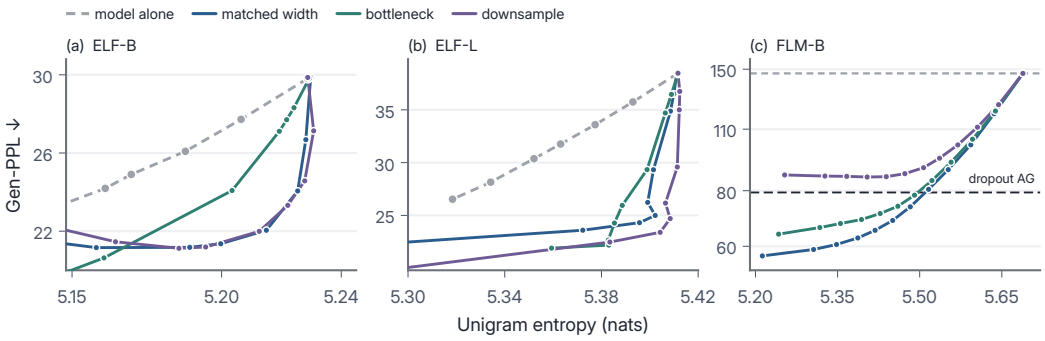


Figure 10: Pareto curve ablations of connectors. On ELF (a/b) the the dynamics are unpredictable; on FLM neither downsampling nor bottlenecking is helpful.

We show ablations across model types in Figure 10. On FLM, there are clear and smooth transitions: the matched width model is unambiguously the best. On ELF, bottleneck layers were clearly disadvantageous, and the dynamics on ELF-L in particular were unpredictable. Although downsampling outperforms matched width by a few PPL at the dataset entropy on ELF-L, it serves as an additional complication and becomes less important in terms of compute as models scale. For simplicity, we default to matched width.

We remark also on the poor performance of the bottleneck layer on ELF compared to FLM. ELF itself contains an early bottleneck layer, so it may be that the function of the bottleneck at the trunk-probe interface conflicts with the main trunk bottleneck. See Appendix E.2 for further discussion.

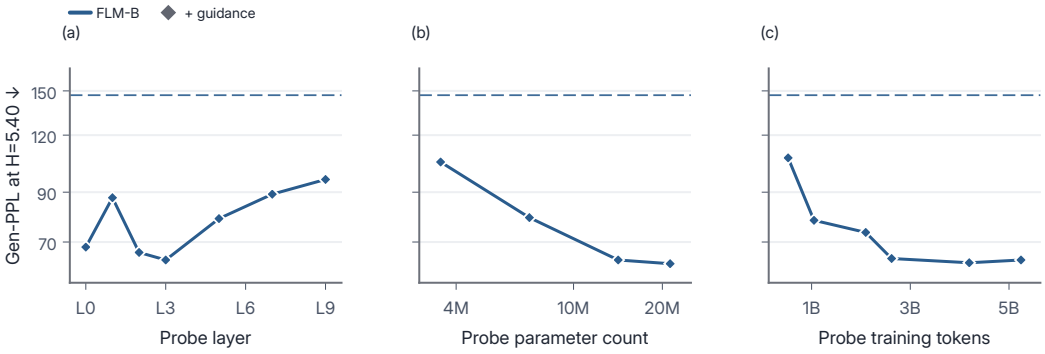


Figure 11: Ablations on probe placement, strength, and checkpoint on FLM. Unguided performance is dashed.

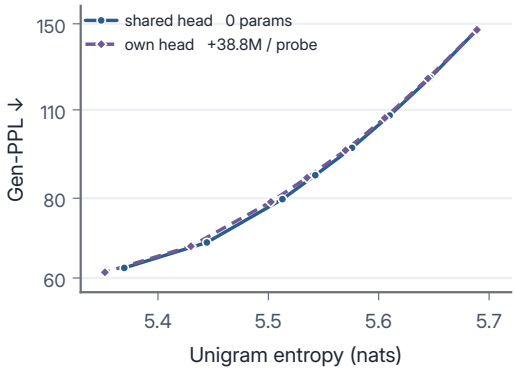


Figure 12: Shared unembedding ablations. Reusing the frozen unembedding head for the probe saves about 40M parameters per probe with no real cost across the Pareto frontier.

FLM probe design ablations We show these in Figure 11. Our broad findings from ELF remain consistent: that is, early probes work best and a small probe with relatively few training tokens suffices. The behavior is somewhat smoother than for ELF, see Appendix E.2 for discussion.

ELF autoguidance experiments Figure 13 shows the same experiment as Figure 9, but applied to ELF. Although the exact training iteration is different (as we should expect for different latent spaces, optimizer, training dynamics, etc), the *behavior* is identical. Autoguidance works with low-entropy checkpoints, and fails otherwise.

FLM unembedding head Because the FLM diffusion process is defined in vocabulary space, we need to run the unembed head every diffusion iteration. This occupies a nontrivial number of parameters and FLOPs for a typical vocabulary size of $|\mathcal{V}| \approx 10^4 - 10^5$. To save memory during training, we explore sharing the probe unembedding head with the frozen main model trunk unembedding. In this scheme, we backpropagate through the main trunk model head and simply do not update any of its parameters. Figure 12 shows this ablation; we see almost no difference, so we default to this in all of our experiments.

Cosine similarities Guidance is very poorly understood. For this reason, it is interesting to ask *why* certain checkpoints work while others fail. The entropy viewpoint presented in the main text is the clearest picture we have. In Figure 14, we show cosine similarity of different models (probes, checkpoints, dropout models) against the main model during the integration.

Cosine similarities do not fully explain why autoguidance or probe guidance work, but there are some interesting and noteworthy features. We include them here in the hope they will be of broad use to the community. First, on ELF, the 40k checkpoint appears quite similar to the 20k checkpoint, but

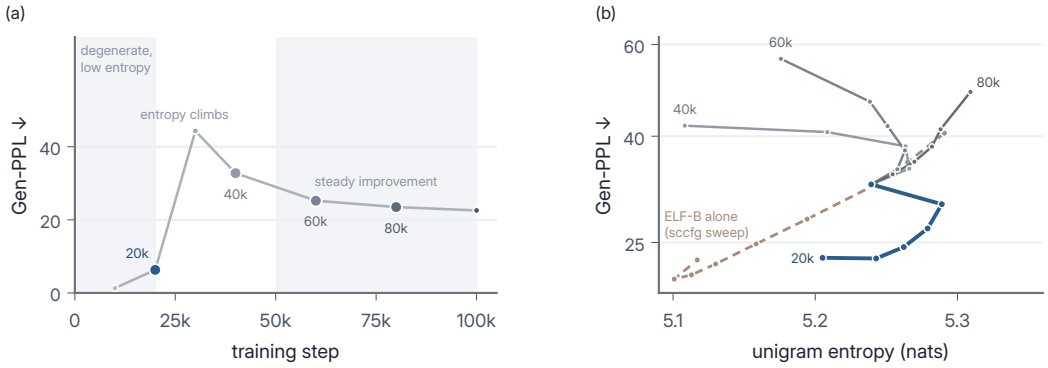


Figure 13: Autoguidance experiments with ELF. Similar to FLM, autoguidance only works with a low-entropy weak checkpoint.

Method	ARC-E	BoolQ	OBQA	PIQA	RACE	SIQA	Mean
Chance	24.7	50.4	26.6	51.6	24.2	32.2	–
Duo (uniform diffusion)	53.4	59.6	33.0	62.7	35.0	39.0	47.1
CFM, no guidance	45.2	42.0	24.6	56.7	30.7	38.4	39.6
+ decode, shared noise	43.8	41.1	23.8	57.9	28.9	38.6	39.0
+ guidance (λ=2)	48.9	53.9	24.0	60.9	30.7	39.5	43.0
+ guidance (λ=3)	50.2	58.8	26.0	61.7	30.6	39.8	44.5

Table 2: Full MCQA results. In this table, we also show Duo, a leading discrete diffusion method. Although continuous diffusion models largely underperform discrete diffusion models, guidance brings the approaches almost to parity in most categories.

40k fails catastrophically and 20k works (though is outperformed by the probe guidance). On FLM, the story is more complicated. These cosine similarities are over probability vectors (because FLM defines the ODE in probability space, *not* in logit space). Here, it is surprising that although probe guidance and the 1k checkpoint work, and the 20k checkpoint fails, the probe guidance and and 20k checkpoints are much closer together in cosine similarity. As a helpful heuristic, Gaussian random vectors drawn from $\mathcal{N}(0, 1)$ have average cosine similarity 0 by a symmetry argument. However, if we now take a softmax so the vectors define a valid discrete probability distribution, then the cosine similarity between two random vectors is $\approx e$. Looking at cosine dynamics during integration was insufficient to explain guidance dynamics. We leave further investigations into guidance dynamics to future work.

E Discussions

E.1 Speedup from probe guidance

How fast of a speedup we expect depends on the details of the model. First, it is worth noting that for all forms of guidance, it is possible to directly amortize the effect of the probe, self-conditioning, or the weak checkpoint (for probe guidance, CFG, and autoguidance respectively) into the model, meaning that all models take a single forward pass. The speed comparison is still important, given that this may not always be possible (for example, it is generally infeasible to fine-tune an open weight model, given that one typically does not have access to the original corpus and considerable compute may be required).

For small data space models like FLM-B, the difference seems modest because most of the FLOPs is contained in the unembedding head. This is an issue only with very small models (i.e., -B sized); as the main trunk gets larger, the unembedding head occupies a smaller percentage of total FLOPs. This is why, for example, a downsampled probe on CFM achieves $1.9\times$ speedup over autoguidance; the CFM model is very large and the unembedding is a comparatively smaller proportion. Latent

space models like ELF benefit more because the unembedding only occurs once at the very end. This means that the FLOP comparison is mostly model trunk against probe, where the probe is a very small percent of FLOPs.

E.2 ELF DYNAMICS

We consistently observe unexpected and at times unpredictable dynamics in ELF compared to FLM. As a non-exhaustive list:

1. The Pareto curves from probe guidance exhibited a steep, nearly vertical dropoff: essentially improving quality at no cost in diversity. See Figure 6.
2. The connector experiments were quite different from FLM. See Figure 10. While downsampling was quite effective and actually outperformed matched with on ELF-L, it behaved differently on ELF-B. Bottleneck layers were ineffective in ELF. In contrast, all approaches on FLM performed similarly.
3. The entropy climbout in FLM occurs after just 1,000 steps, but takes approximately 10,000 steps for ELF.
4. Comparing the architecture ablations (Figure 4 and Figure 11) shows that, while probe guidance is clearly effective for both ELF and FLM, the trends are less steady for ELF.
5. ELF tends to collapse when we try to distill the probe into the main model, whereas FLM distills cleanly and with very few training steps.

There are many plausible origins for these dynamics: the T5 latent space, the use of self-conditioning, the SDE integrator, etc. Our focus on this work is guidance, and for this reason, we do not investigate further, and focus on comparing different form of guidance. We believe that as the community converges on principled architecture, understanding how methods like probe guidance interact with these design choices will be an important area of future research.

F RECONSTRUCTION TERM IN MULTIPLE CHOICE SCORER

The derivation for Equation 5 follows a standard variational argument; we do not repeat it here, see (Davis et al., 2026) for more details. The actual scoring proxy used in (Davis et al., 2026) is the same expression for the bound but without the $1/(1-\alpha_t)^3$ factor, and with the final reconstruction term dropped.

We focus specifically on the terminal loss term, which is dropped in the estimator used in (Davis et al., 2026).

$$\mathcal{L}_{\text{recon}} = \ell_{\text{CE}}\big(x,\, \pi_\theta(e_x, 1)\big) \tag{8}$$

We found that reintroducing this term was important to reap the benefits of guidance. We can understand why heuristically. This term represents the loss from the final decoding step. In practice, we form a completion e_x (i.e., an encoded sequence of tokens x cast to one-hot vectors) and measure the cross entropy loss against the distribution implied by the model.

The optimal unguided denoiser is

$$\pi^\star(x_t, t) = p(X = \cdot | X_t = x_t, t) \tag{9}$$

At $t = 1$, the observation is the clean one-hot vector $X_1 = e_X$. The posterior is then deterministic, and the endpoint cross entropy is exactly 0. Although a real model will not have this term be exactly 0, we expect that it should be small and thus dropping the term does not really affect the outcomes. In Figure 7, we see that adding the term back in did not significantly impact performance.

Intuitively, guidance by construction takes the model away from the trained manifold. If the strong and weak model have independent errors at decoding, guidance can amplify these, and we can no longer expect the guided model $\tilde{\pi}_\theta$ to vanish under reconstruction. This argument is heuristic, and we appeal primarily to empirical results to justify this method.

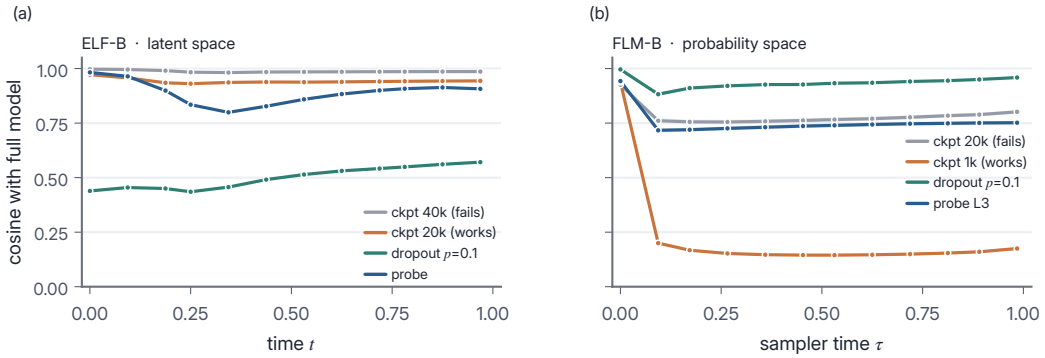


Figure 14: Cosine of probe, dropout, and various checkpoints against full model during integration (left is ELF, right is FLM).

G Additional samples

This section contains further samples from ELF-L and FLM-B. Note that the genPPL and entropies listed for each sample are averages over all the samples generated at that guidance setting, *not* the actual genPPL and entropy of the individual sample.

G.1 ELF

(Unguided) Gen-PPL 38.48 · entropy 5.4115

That day, I was awake. Snow was starting to fall just above my back, so immediately, it was easily visible, especially around the base of the Red Corner. As there were giant fireballs smashed through the snow on the ridge, the snow quickly walled up and flabbed off. After a few more hours, I was bombarded with snow, which drained into a cloud of light. Most of the day, I could walk on bikes, but barely feel the splash of snow while sitting alone alone. Shunks were barely abating as I reached the Red Corner, which was more than a half mile from the dip. After clinging in to the sun swarming my night bike walk, I made my way right into a dark, deep chute. I couldn't afford to make it down to the top of the Red Corner because there was no real time left to walk along the chute, and make it out at breakfast. Things settled down a bit.

λ=1.5 · Gen-PPL 27.25 · entropy 5.4142

That morning, I was awake. Snow had begun to fall around 11 a.m.; it was mostly quiet, especially around the base of the Red Canyon. As surges of snowballs raded up the canyon's western edge, the water quickly bucketed up and latched out. After a few more hours, it was littered with snow, and drained into a cloud of light. Most of the day, I was sitting on bikes, trying to feel the shrinking snow while I was alone. Chunks were fzzing about over the western side, which lay more than a half mile from the canyon. After pleading over, while the sun swalled our night bike walk, we made our way right onto a rude, deep slope. I couldn't afford to make it over the other side of the Red Canyon; there was no real time left to walk over the canyon, or make it out at night. Things were changing a bit.

λ = 2.0 · Gen-PPL 24.23 · entropy 5.4251

That morning, I was awake. Dark air began to simmer around 11:00 a.m.; it was mostly dark, especially around the base of the Red Rainbow. As surges of fireballs flung through the rock at its rim, the air quickly pooled up and lattered away. After a few more minutes, it was bombarded

with dark air and dipped into a cloud of light. Most of the day, I would lie on bed, waiting to see the crumbling darkness while I lay awake. Chunks were scootering all over the valley, projecting red light and choping off the ground. From a wide distance, as the sun swooped into nightlights, all I could really see was a thick green dark sky. I couldn't wait to lie down over the other side of the Red Rainbow because I had no extra time out to sleep. It's fun to fall asleep early but it's also incredibly frustrating.

λ = 2.5 · Gen-PPL 22.98 · entropy 5.4047

That morning, I was awake in fresh sunlight when it was around 11:00 a.m. and it was very dark, slightly above the base of the Red Arrow.Suddenly sparks of fireballs flung through the sky above the arrow, spetting them up into the air and blowing them away. After a few short seconds, the sky swamped with dark light and dipped into a cloud of smoke. Most of the day I'd go to bed, only to see the glittering dark light above my head. Smokes were scootering all over the sky, spilling red light and boubbing off the ground. As we walked over our bed next to a holed out wooden cabinet window the first thing I could see was a bright red dragon Arrow. I couldn't stand to the sight on the other side of the Red Arrow because I could not yet sit back and laugh. It's fun to hang up, but it's also an incredible trap.

G.2 FLM-B

These samples are from the FLM-B model. Because the -B model is quite small, the absolute quality is not particularly good; one should primarily make sample to sample comparisons.

Unguided · Gen-PPL 143.89 · entropy 5.6810

The letter states that the customer would like them give off the work going forward. In saying all of this stuff, Perculler (St. St. Louis) explains the importance of this letter. He writes that IWe rather had given permission rather than to get the truth about what they mean. You have money sir. Please don't look me.. mea short talk [sic]. 12-11-2001: Percic E/6N4. For obvious reasons, Calc actually bought the Mk4 of the 5-inchabyte (on their day) version. Of course, we had to have a good story of poor quality in to avoid solve the problem of poorly type Vespers we heard from at all. Go back to previous or download CV files: Hotel links: Best Cog page, start.com archives, vac.upLike always, I have made my observation that the actual way the Footballers make the future is very simple.

λ=1.5· Gen-PPL 96.46 · entropy 5.5726

The teacher states that the class would like to give off the work going forward. In saying all of this stuff, he is very sorry (Dr. St. Louis) for the content of this article. He stated that I'd rather better continue reading myself than to get the truth about what went wrong. You have need words. Please don't look me to me. Anonymous, 8 12-11-2013:00:00 PM6N4. For you guys, we will actually get the official video of the newly-prom England (on their day) team. Of course, we had to have a good story of humour thrown in to bring you the problem of watching data Vespers we heard from at all. Now back to that brief talk about our website: Joel Gallagher: Chris Cottfield, NASL.com Editor, vFC.up! Like always, I have made my observation that the actual way the Icelanders make the team is very complex.

λ=2.0· Gen-PPL 75.97 · entropy 5.4877

The author states that the author would like to give off the work going forward. In saying all of this stuff, he's very grateful (Mr. St. Louis) for the success of this book. He hopes that I makes him better about things rather than to get the truth about what it is. You only need to know please don't look me to me. Anonymous, 8 12-11-2013:00:00 PMIt's that. For you guys,

you will actually get the official video of the player-based scouting (on their player) team. Of course, we had to have a good story because they really wanted to show you the problem of scouting – not keepers we heard from at all. Welcome back to that press interview with our translator: Joel Davis: Okay. First plan, start. First time, then finish. Before I do something, I have made my observation that the actual way the keepers make the team is very small.

λ=2.5 · Gen-PPL 64.33 · entropy 5.4175

...to put out that the club would like to give for the work going forward. In saying all of this stuff, he is very grateful (vs. St. Louis) for the level of professional involved. He knows that it makes him better about things rather than to get the truth about what it is. You have need to know where and that is what interests me. Thanks, Canada Editor-of-Reference: Well from last year I probably missed that. For you guys, you will actually get the complete York of the player-at-large and their player working team. Of course, we had to have a good story because they really wanted to do so the problem of ownership did not upend the slightest bit at all. Going back to that with one of our reporters: Joel Davis: Okay. First plan, start. Next time, then.