DISTILLSCOPE: BENCHMARKING FLOW-MAP DISTILLATION FOR LANGUAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Distillation promises fast language generation, but a recipe's apparent advantage can depend on how quality and computation are measured. We introduce `DistillScope`, a common training and evaluation framework for flow-map language-model distillation that examines how recipe comparisons change across quality criteria, training budgets, and inference settings. Semigroup distillation achieves lower external perplexity than Lagrangian distillation, while their ordering reverses under MAUVE. Denoising readout and length-extrapolation generation also favor different recipes. Accounting for measured training time reverses a recipe advantage observed at equal updates. Additional inference computation allows the same trained student to pass degeneration checks that it fails under tighter inference budgets. Initialization, objective ablations, and cross-environment comparisons delineate the scope of these findings. An exposure-matched sampling control and supporting analysis of composition and categorical readout clarify the distinctions between training allocation, map fidelity, and decoded quality. `DistillScope` makes these dependencies explicit, supporting recipe choices tied to a stated quality target and computational constraint.

1 INTRODUCTION

Diffusion language models generate text through iterative denoising, allowing multiple token positions to be updated in parallel. Discrete score-based and masked diffusion establish this approach for language modeling (Lou et al., 2024; Sahoo et al., 2024), and LLaDA extends masked denoising to large-scale pretraining and instruction tuning (Nie et al., 2025). Parallel updates create an opportunity for efficient generation, but every refinement round still requires a network evaluation. Distillation seeks to preserve useful generation while reducing that inference cost. The available distillation procedures impose different constraints on the student. Progressive distillation compresses teacher sampling steps (Salimans & Ho, 2022), consistency models learn predictions that agree along a denoising trajectory (Song et al., 2023), and flow map matching learns maps between noise levels (Boffi et al., 2025). Flow Map Language Models bring the latter formulation to text and derive and compare semigroup, Lagrangian, and Eulerian objectives (Lee et al., 2026). Their common map parameterization makes it possible to compare different ways of learning the same type of generative transition. We use this parameterization to study whether recipe advantages persist across evaluation and resource conventions.

Discrete-language distillation develops complementary approaches. Self-Distillation Through Time compresses denoising trajectories over multiple rounds (Deschenaux & Gulcehre, 2025); Di4C models dependencies between token positions (Hayakawa et al., 2025); and DiDi-Instruct uses divergence-based distillation (Zheng et al., 2026). Stochastic mask states, consistency across posterior bridges, and self-conditioned fixed-point flows offer further ways to learn few-step generators (Yoo et al., 2026b; Amin et al., 2026; Yoo et al., 2026a). These advances change targets, sampling processes, and training procedures. Their reported scores consequently describe complete systems at selected operating points. Interpreting those scores also requires specifying what constitutes better generation. Highly predictable samples can be repetitive (Holtzman et al., 2020), and criteria for generative models need not agree (Theis et al., 2016). Categorical-sampling precision can change diffusion outputs (Zheng et al., 2025). Generative Frontiers relates perplexity to distributional entropy and motivates evaluating them jointly (Pynadath et al., 2026), while MAUVE

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.

compares generated and reference text in feature space (Pillutla et al., 2021). These results establish the limitations of single-metric evaluation. The empirical question for distillation is which recipe comparisons change when the metric is changed, even after pronounced degeneration has been distinguished from other quality differences.

Resource accounting poses a parallel question. Equal optimizer updates give recipes the same training exposure at a fixed batch size, but a more expensive target can consume more time per update. Likewise, a one-step result does not describe a checkpoint that will be deployed with several evaluations. Controlled benchmarks in domain generalization and training algorithms show why selection and resource conventions belong to the procedure being compared (Gulrajani & Lopez-Paz, 2021; Kasimbeg et al., 2025). For flow-map distillation, the relevant object is a recipe evaluated under a stated quality criterion, training allocation, and inference budget.

We introduce DISTILLSCOPE to examine these dependencies within a common framework. The benchmark compares semigroup, Lagrangian, and Eulerian self-distillation with progressive-halving and damped fixed-point baselines. It evaluates fixed checkpoints across inference budgets, compares recipes under both equal updates and measured training time, and retains separate perplexity and distributional measurements. Entropy and repetition identify specified degeneration patterns. Denoising readout probes provide a separate measurement of the learned map. This design turns a collection of recipe scores into explicit comparisons with different scientific meanings.

The results reveal three consequential dependencies. At 100k updates and eight network evaluations, semigroup distillation has 14.8% and 17.0% lower geometric-mean perplexity than Lagrangian distillation under Qwen and GPT-2, across three runs per recipe. Yet a separate three-run comparison consistently favors Lagrangian distillation under MAUVE. A second reversal concerns cost: the fixed-point baseline's GPT-2 advantage at equal updates becomes a PSD advantage when training time is matched. Finally, the same Lagrangian checkpoints that fail one-step degeneration checks pass them at eight steps in the core budget sweep. The preferred recipe and the generation regime therefore depend on the comparison being made. Additional experiments test the reach of this account. Denoising readout and length-extrapolation generation can favor different recipes; initialization changes quality despite unchanged screen outcomes; and OpenWebText and masked-model comparisons expose limits to transferring the core results. A fixed interval mixture and its exposure-matched control provide a case study in separating coarse temporal coverage from finer allocation. Supporting analysis explains why continuous map error, categorical agreement, and distributional quality require distinct measurements. Together, these results motivate a benchmark whose output is a set of conditional comparisons that readers can use for their quality and resource requirements.

- **A common framework:** five recipes with explicit comparison conditions.
- **Quality criterion:** PSD leads on perplexity, while LSD leads on MAUVE.
- **Training cost:** matching time reverses FP's equal-update GPT-2 advantage.
- **Inference budget:** the same LSD checkpoints fail at one NFE and pass at eight.

2 THE DISTILLSCOPE BENCHMARK

DISTILLSCOPE compares how distillation recipes convert a training budget into generation at a specified number of inference steps. The basic evaluation unit is a trained checkpoint at an inference budget, with a defined sampler and scoring configuration. This makes both the learned map and its use at inference part of the comparison.

2.1 PROBLEM SETUP

Given a pretrained denoising teacher, we study the training of a student that generates text using K network evaluations. We compare distillation recipes across training and inference budgets, measuring degeneration and generation quality.

For a sequence of length L and vocabulary size V, a flow-map language model transports a continuous state $x \in \mathbb{R}^{L \times V}$ from noise toward token predictions (Boffi et al., 2025; Lee et al., 2026). Let $D_\nu(x, s, t)$ denote the student's predicted vocabulary distribution at each position for a jump from

transport time s to t. We use the parameterization

$$X^{\nu}_{s,t}(x) = \frac{1-t}{1-s}\, x + \frac{t-s}{1-s}\, D_\nu(x, s, t), \qquad 0 \le s \le t \le 1, \quad s < 1. \tag{1}$$

It gives the identity map at $s = t < 1$ and the student prediction at $t = 1$. Ideal flow maps satisfy the composition identity $X_{u,t} \circ X_{s,u} = X_{s,t}$; the recipes impose different training constraints on the approximation learned by the student. Model-conditioning times are related to transport times by the model's time reparameterization. The notation above suppresses that conversion, which is included when computing time derivatives.

Generation starts from a noise state x_0 and applies the student on a time grid $0 = t_0 < \cdots < t_K = 1$, with $x_{k+1} = X^{\nu}_{t_k, t_{k+1}}(x_k)$. Each jump requires one network function evaluation (NFE), so K is the inference budget. The core flow-map sampler is deterministic given x_0 and uses positionwise $\arg\max$ for the terminal readout. State arithmetic uses float64, with network forwards in fp32.

2.2 Five distillation recipes

The benchmark implements semigroup (PSD), Lagrangian (LSD), and Eulerian (ESD) self-distillation following Lee et al. (2026), together with progressive-halving (PH) and fixed-point (FP) baselines. PH is a local progressive-compression baseline motivated by Salimans & Ho (2022); Deschenaux & Gulcehre (2025); FP is a local damped self-consistency baseline. Their configured targets are distinct from native reproductions of those methods. Table 1 summarizes their target construction and loss.

Table 1: **The five benchmark recipes.** Targets are detached from gradient computation. Target construction, projection, and loss form are part of each recipe; the EMA copy is used for evaluation.

Recipe	Off-diagonal target construction	Target model and loss
PSD	Compose two shorter jumps into a long-jump target.	Online student; cross-entropy.
LSD	Correct the prediction at the transported state using a terminal-time derivative; apply entmax.	Online student; cross-entropy.
ESD	Correct the teacher prediction using a source-time derivative along the flow.	Teacher anchor; squared probability error.
PH	Compose shorter jumps from a frozen model over four rounds of increasing jump width.	Round-frozen student; cross-entropy.
FP	Refine a self-consistency target with four damped iterations ($\alpha = 0.3$).	Online student; cross-entropy.

All five recipes include a diagonal teacher-denoising cross-entropy term. The default batch allocates half its rows to this term, $1/64$ to the full-interval boundary $(s, t) = (0, 1)$, and the remainder to interior intervals. PH instead restricts interval width according to its training round. PSD and PH construct composition targets through the midpoint in model-conditioning time. Each recipe's off-diagonal target is detached, with gradients taken through the student prediction being fitted. The full objectives and time-coordinate conventions are specified in Appendix B.3.

2.3 Training and inference budgets

The core experiments use LM1B (Chelba et al., 2014), length-128 blocks, the BERT uncased vocabulary, and a global batch of 128. The reference student has 12 transformer blocks of width 768 and is initialized from the teacher, with a zero-initialized second time-conditioning pathway. We evaluate training budgets of 12k, 25k, 50k, and 100k optimizer updates and inference budgets of $\{1, 2, 4, 8\}$ NFE. The optimizer is AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$ and a 1,000-update warmup followed by a constant learning rate: 3×10^{-4} for PSD, LSD, ESD, and FP, and 10^{-3} for PH. The budget sweep uses raw weights at 12k and 25k updates and exponential moving average (EMA) weights at 50k and 100k, with decay 0.9999. At a fixed budget, the same checkpoint is evaluated across the NFE grid. The principal PSD/LSD quality comparison uses three training runs per recipe at 100k updates, EMA weights, and eight NFE. Run counts and any changes to the core setting accompany the corresponding results. Equal update budgets measure training exposure at fixed batch size; comparisons matched by training time are identified separately as the recipes differ in per-update cost.

Table 2: **DistillScope at a glance: different comparisons favor different recipes.** All panels use EMA weights and report run means or passing-run counts. A is separate from the geometric-mean cohort in Table 3. B reports equal-time PPL ratios to PSD; D evaluates generation at 256 tokens.

A Quality criterion
100k updates; 8 NFE; 3 runs/recipe

Recipe	Perplexity ↓		MAUVE ↑
	Qwen	GPT-2	
PSD	**163.9**	**204.1**	0.493
LSD	191.9	245.6	**0.580**

B Training-cost convention
8 NFE; 2 runs/recipe

Recipe	Equal updates ↓		Time ratios ↓	
	GPT-2	Qwen	GPT-2	Qwen
PSD	204.9	**165.6**	**1.000**	**1.000**
FP	**185.4**	172.2	1.071	1.077

C Screen-passing runs
100k updates; 2 runs/recipe

Recipe	1 NFE	4 NFE	8 NFE
PSD	2/2	2/2	2/2
LSD	0/2	2/2	2/2
ESD	0/2	0/2	0/2
PH	2/2	2/2	2/2
FP	0/2	2/2	2/2

D Readout versus length transfer
100k updates; 2 runs/recipe

Recipe	Cloze	256-token log-PPL	
	NLL ↓	GPT-2 ↓	Qwen ↓
PSD	**4.008**	5.662	5.506
LSD	4.157	5.829	5.671
PH	4.508	5.430	5.271
FP	6.488	**5.234**	**5.163**

2.4 Degeneration diagnostics and quality metrics

We characterize generated text through token statistics and complementary quality measures. For a generated block y, let $\widehat{p}_y(v)$ be its empirical token frequency and let $H(y) = -\sum_v \widehat{p}_y(v) \log \widehat{p}_y(v)$ be its unigram entropy in nats. Let $U_4(y)$ count its distinct four-grams. The repeated-four-gram fraction is $\mathrm{rep4}(y) = 1 - U_4(y)/(L-3)$ (Welleck et al., 2020). The core diagnostic evaluation uses 1,024 generated blocks. Writing bars for averages over blocks and H_{pool} for the entropy of the pooled token frequencies, an LM1B evaluation cell passes the degeneration screen when

$$\bar{H} \in [3.553,\ 4.594], \qquad H_{\mathrm{pool}} \le 9, \qquad \overline{\mathrm{rep4}} \le 0.008. \tag{2}$$

These thresholds are calibrated from held-out text and reference student outputs and held fixed across the core LM1B grid. They describe aggregate token usage and repetition at the specified tokenizer and sequence length. We report screen status as a diagnostic alongside quality measurements; passing the screen does not establish semantic quality. Generative perplexity measures the predictability of generated text under an external autoregressive model. We use Qwen2.5-1.5B-Instruct and GPT-2-large as separate scorers (Yang et al., 2024; Radford et al., 2019). MAUVE (Pillutla et al., 2021) complements these scores by comparing generated and reference text distributions in feature space. The main MAUVE comparisons use 5,000 samples, GPT-2-large features, PCA retaining 90% of variance, and a fixed clustering seed. We retain each metric's ordering rather than combine them into a single score. For the principal PSD/LSD comparison, we average log perplexity across training runs and exponentiate to obtain geometric means. Training runs are the units of replication; samples and alternative scorers of the same outputs characterize each run.

3 Which conclusions change with the comparison?

We first establish the PSD–LSD perplexity difference, then examine how the recipe ordering changes across measurements and training-cost conventions. Finally, we vary inference computation while keeping each checkpoint fixed. These comparisons distinguish three questions: which quality criterion favors a recipe, which training allocation produces it, and at what inference budget it avoids the specified generation pathologies.

3.1 SEMIGROUP DISTILLATION HAS LOWER PERPLEXITY AT EIGHT NFE

Finding 1. PSD has lower geometric-mean perplexity under both scorers. The principal comparison uses three PSD and three LSD training runs, all evaluated at 100k updates with EMA weights and eight NFE. For each scorer, let Δ be the PSD mean log perplexity minus the LSD mean log perplexity. The ratio of geometric means is $\exp(\Delta)$, and the relative reduction is $1 - \exp(\Delta)$. Table 3 gives reductions of 14.8% under Qwen and 17.0% under GPT-2. The direction agrees across the two scoring models at this operating point. This establishes a replicated scorer-specific comparison, which provides the reference for examining the change under MAUVE.

The training-run count determines the resolution of this comparison. Enumerating the $\binom{6}{3} = 20$ run-label assignments gives a two-sided permutation value $p = 0.10$ for both scorers under exchangeability, the smallest attainable value for this absolute-difference test.

Table 3: **PSD relative to LSD.** 100k EMA, 8 NFE; three runs per recipe.

Scorer	Mean Δ log-PPL	PPL reduction
Qwen	-0.1600	14.8%
GPT-2	-0.1863	17.0%

3.2 CHANGING THE QUALITY CRITERION REVERSES THE COMPARISON

Finding 2. Agreement between perplexity scorers does not imply agreement across quality metrics. A separate comparison with three runs per recipe measures both generative perplexity and MAUVE at 100k updates, EMA weights, and eight NFE. PSD has lower perplexity under both scorers, whereas LSD has higher MAUVE (Figure 1). The arithmetic means are Qwen perplexity 163.9 versus 191.9, GPT-2 perplexity 204.1 versus 245.6, and MAUVE 0.493 versus 0.580 for PSD and LSD, respectively. This comparison supplies the cross-metric evidence separately from the geometric-mean effects in Table 3.

These metrics evaluate different properties. Generative perplexity measures how predictable a model's samples are to an autoregressive scorer, while MAUVE compares the generated and reference distributions through text features. The observed ordering therefore supports a scorer-specific advantage for PSD and a MAUVE-specific advantage for LSD. Reporting both preserves the distinction between these evaluation targets.

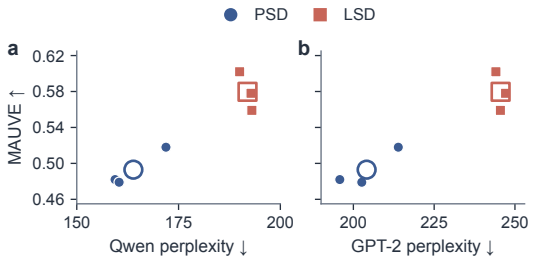

Figure 1: **Quality metrics favor different recipes.** 100k EMA, 8 NFE. Filled symbols show three runs per recipe; open symbols show means.

3.3 READOUT QUALITY DOES NOT DETERMINE LENGTH-TRANSFER ORDERING

Finding 3. Denoising readout and length-extrapolation generation favor different recipes.

Table 4: **Readout and length transfer.** Two runs per recipe, 100k EMA. Shaded bold entries mark row minima.

Measurement	PSD	LSD	PH	FP
A Cloze NLL ↓				
$\tau = 0.25$	**4.008**	4.157	4.508	6.488
$\tau = 0.5$	**3.183**	3.243	3.710	4.035
B 256-token log-PPL, 8 NFE ↓				
GPT-2	5.662	5.829	5.430	**5.234**
Qwen	5.506	5.671	5.271	**5.163**

The capability comparison evaluates two runs per recipe at 100k updates with EMA weights. Cloze probes predict clean reference tokens from corrupted states; the generation probe produces free-running samples at 256 tokens, twice the training length. PSD has lower cloze NLL than FP at both corruption levels, whereas FP has lower length-256 generative log-perplexity under both scorers (Table 4). All four students pass the degeneration screen in this length-transfer comparison.

The reversal concerns denoising and length extrapolation, rather than two measurements of in-distribution generation. The teacher's own eight-step length-256 outputs are degenerate, so their low perplexity is not a quality reference. These

results show that readout NLL alone does not determine which student gives lower perplexity under the tested length shift. A separate six-corpus readout suite evaluates PSD at 135,576 updates and the other recipes at 100k (Appendix D.3).

3.4 Training cost changes the comparison

Finding 4. **Equal training updates and equal training time yield different comparisons.** In the two-run PSD/FP comparison at 100k updates, EMA weights, and eight NFE, FP has lower GPT-2 perplexity (185.4 versus 204.9), while PSD has lower Qwen perplexity (165.6 versus 172.2). PSD is cheaper per training update. Matching measured training time on fixed hardware therefore allocates 135,576 updates to PSD for the cost of 100k FP updates. At that allocation, the point estimates favor PSD under both scorers, with FP perplexity 7.1% higher under GPT-2 and 7.7% higher under Qwen (Figure 2; Appendix D.2). Equal updates measure equal data exposure, whereas matched time measures performance under the training-cost constraint. NFE specifies inference evaluations, not wall-clock generation latency.

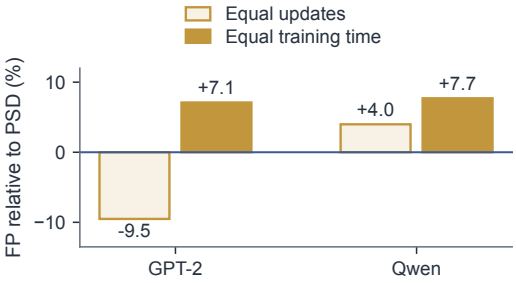

Figure 2: **Training-cost conventions change the comparison.** Two EMA runs per recipe, 8 NFE; PSD is the denominator.

3.5 Additional inference steps change the generation regime

Finding 5. **One-step failure can coexist with eight-step recovery in the same student.** Across the LSD budget sweep, the checkpoints at 12k, 25k, 50k, and 100k updates fail the degeneration screen at one NFE and pass it at eight (Table 5A). Each row in A compares the same weights at the two inference budgets. This behavior is also observed in the 12k recipe grid, with three runs per recipe: all three LSD runs fail at one NFE and pass at eight. Additional composition steps therefore change the observed generation regime without further training.

Intermediate inference budgets expose variation within this contrast. At 12k updates and two NFE, one of the three LSD runs passes the screen. In the budget sweep, the 25k checkpoint at four NFE has a repeated-four-gram fraction of 0.00844, just above the 0.008 cutoff. These observations locate the transition within the evaluated settings without implying a monotone boundary. The sweep uses raw weights at 12k and 25k and EMA weights at 50k and 100k; the within-checkpoint NFE comparisons hold the weight choice fixed.

Table 5: **Inference budget changes observed screen outcomes.** A: one LSD checkpoint at each budget. B: two EMA runs per recipe at 100k updates.

A LSD checkpoints

Updates	Weights	1 NFE	8 NFE
12k	Raw	Fail	Pass
25k	Raw	Fail	Pass
50k	EMA	Fail	Pass
100k	EMA	Fail	Pass

B Passing runs at 100k

Recipe	1 NFE	4 NFE	8 NFE
PSD	2/2	2/2	2/2
LSD	0/2	2/2	2/2
ESD	0/2	0/2	0/2
PH	2/2	2/2	2/2
FP	0/2	2/2	2/2

The five-recipe comparison shows distinct profiles at 100k updates (Table 5B). PSD and PH pass at one, four, and eight NFE in both runs. LSD and FP fail at one NFE but pass at four and eight; ESD fails at all three inference budgets. More inference computation thus changes the outcome for some recipes within this grid, while others retain their initial screen status. The screen also identifies why very low generative perplexity requires context: the one-step FP checkpoint can attain Qwen perplexity 1.0 with mean within-block entropy approximately zero. Such a value describes highly predictable, degenerate output rather than evidence of improved generation.

4 CONTROLS AND TRANSFER OF THE CORE FINDINGS

4.1 INITIALIZATION CHANGES QUALITY WITHOUT CHANGING SCREEN STATUS

Finding 6. Initialization can change quality without changing screen status.

At the reference width and depth, native fresh initialization preserves the observed PSD and LSD screen outcomes in all eight recipe–NFE cells of the control. Yet fresh PSD has eight-NFE Qwen perplexity 407.4 versus 125.1 for the teacher-initialized reference, and MAUVE 0.054 versus 0.456 (Table 6); each fresh recipe has one run. The benchmark records initialization alongside architecture because passing the same screen can coexist with substantially different quality scores. The PH capacity comparison changes both architecture and initialization, including the quality of its first frozen target; it is reported separately in Appendix D.5.

Table 6: **PSD initialization control.** 12k raw, 8 NFE; one fresh run versus the teacher-initialized reference.

Initialization	Qwen PPL ↓	MAUVE ↑
Teacher	125.1	0.456
Fresh	407.4	0.054

4.2 REMOVING THE DERIVATIVE CORRECTION DOES NOT REMOVE ONE-STEP DEGENERATION

Finding 7. The derivative correction is not necessary for the observed LSD one-step failure. A factorial study varies derivative-term removal, the distribution of sampled times, and batch composition at 12k updates on LM1B. None of the four derivative-zero configurations passes the one-NFE screen. Across all 15 training runs, including the additional clipping and sampling controls, every run fails at one NFE and passes at eight (Table 7). Thus removing this term alone does not explain the endpoint contrast. This intervention does not identify which remaining ingredient causes it. The intermediate operating points remain sensitive to training allocation. Under the late-time sampling warp, both runs fail at two NFE and pass at four and eight. Intermediate-NFE outcomes also vary across runs of the base configuration. The result supports reporting the full inference profile of an ablation, rather than only its one-step score (Appendix D.6).

Table 7: **Screen outcomes under the tested objective ablations.** LM1B, 12k updates, raw checkpoints. The rows overlap: derivative-zero and late-warp runs are included in the full study, and their counts must not be added as independent replications. The first row counts derivative-zero configurations; the other rows count training runs.

LSD intervention	Comparison	Screen-passing outcome
Derivative correction set to zero	1 NFE	0/4 configurations
Full factorial and additional controls	1 / 8 NFE	0/15 / 15/15 runs
Late-time sampling warp	2 / 4 NFE	0/2 / 2/2 runs

4.3 ONE-STEP PROFILES CHANGE ACROSS TEXT ENVIRONMENTS

Finding 8. One-step screen outcomes differ between the evaluated LM1B and OpenWebText settings.

The continuous-flow OpenWebText comparison trains PSD, PH, and LSD for 10k updates on 1,024-token blocks, with two runs per recipe and raw checkpoints. At one NFE, both PSD runs and both PH runs fall below the OWT entropy floor of 4.6687 nats; their mean within-block entropies range from 3.862 to 4.146. LSD has one run below that floor and one narrowly above it, at 4.703 (Figure 3). At two NFE, all six runs pass the full screen. These settings change corpus, tokenizer, block length, and training budget together; the contrast does not isolate sequence length. Low-NFE outcomes therefore need to be measured within each environment. The environment-specific screen and full NFE profile remain necessary parts of the benchmark result (Appendix D.7).

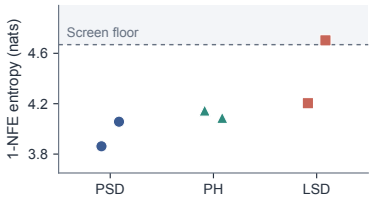

Figure 3: **OWT, 1 NFE.** 10k raw; 1,024 tokens; two runs/recipe.

4.4 MATCHED EXPOSURE LEAVES A GAP BETWEEN EVALUATED PROCEDURES

Finding 9. At matched token exposure, the evaluated PSD run remains behind released SDTT.



Figure 4: **One PSD training trajectory.** Raw weights, 8 NFE; released SDTT scores 104.5 at the matched final token exposure.

On masked OpenWebText, we extend one PSD run to 70k updates at batch 128 and length 1,024, matching the released seventh-round SDTT checkpoint's 9.18B training tokens. At eight NFE, PSD's generative perplexity under the SDTT scoring convention decreases from 665.7 at 10k to 537.5 at 35k, then reaches 620.1 at 70k. The released SDTT model scores 104.5 at the same inference budget (Figure 4). PSD's reported raw checkpoints remain screen-passing; screen status alone does not expose the quality gap. This comparison complements a two-run native SDTT first-round experiment: at its 512-NFE design point, those runs score 42.2 and 42.5, versus 43.6 for the released first-round checkpoint and 52.1 for the teacher under the same scoring convention. Matching token exposure leaves a score gap in this configured comparison. Its single PSD run, fixed optimizer settings, and released comparator do not isolate the contribution of the round curriculum from other training differences (Appendix E.3).

5 AN EXPOSURE-MATCHED CONTROL FOR INTERVAL ALLOCATION

Training intervals provide a focused case study in the comparison design. Their distribution determines which parts of a flow map receive supervision. We use a fixed mixture over four quarter intervals, two half intervals, and the full interval in model-conditioning time τ. The mixture assigns more mass to later components while retaining multiple interval lengths. Within a component $[a, b]$, it samples a length $d \sim U(0, b-a)$ and a start uniformly from $[a, b-d]$. The diagonal anchor and full-interval boundary branches remain fixed. With their respective probabilities $1/2$ and $1/64$, the training objective is

$$\mathcal{L}_q(\theta) = \tfrac{1}{2}\,\mathbb{E}\ell_{\text{diag}}(\theta) + \tfrac{1}{64}\,\mathbb{E}\ell_{\text{boundary}}(\theta) + \tfrac{31}{64}\,\mathbb{E}_{(\tau_s,\tau_t)\sim q}\ell_{\text{off}}(\theta;\tau_s,\tau_t). \tag{3}$$

Here q is the mixture density and each expectation also includes the recipe's data and noise sampling. No importance correction is applied: changing q reweights the training objective. Appendix F specifies the component weights and sampling density.

Matching exposure separates coarse coverage from finer allocation. We compare the fixed mixture with a joint exposure-matched control that uses the same components and within-component sampling. Its weights approximately preserve the interval-length distribution and the probability that an interval lies entirely in the late half, to the precision reported in Appendix F. The control redistributes probability between components of the same width, allowing us to examine the remaining differences in temporal allocation.

Finding 10. Exposure-matched interval allocation gives similar measured scores.

Across two runs per arm, the fixed mixture reduces mean NLL by 0.002529 under Qwen and 0.015774 under GPT-2, with mean MAUVE higher by 0.000896 (Table 8). The run-level MAUVE ordering reverses. These small differences support similar measured scores, without establishing statistical equivalence. Matching interval-length exposure and late-half coverage provides a control for finer temporal allocation; it does not match the full joint time density.

Table 8: **Exposure-matched sampling.** PSD, 12k raw, 1 NFE; two runs per arm, 2,048 outputs per run. NLL is scorer mean negative log-likelihood.

Metric	Fixed mixture	Joint control
Qwen NLL ↓	5.196921	5.199450
GPT-2 NLL ↓	5.375370	5.391144
MAUVE ↑	0.025083	0.024187

Interval accuracy and categorical agreement are distinct. For fixed student and reference maps on a common grid, let ε_i denote the local residual and T_j a finite propagation factor between their trajectories. Starting from the same state, Proposition 1 bounds terminal error by

$E_K \leq \sum_i \varepsilon_i \prod_{j>i} T_j$. This separates local approximation from downstream amplification. More inference steps change both quantities, so this bound gives no monotonic improvement guarantee. It does not specify how an interval's residual responds to additional training, so the bound alone cannot select an optimal sampling distribution. The exposure-matched comparison evaluates the remaining allocation difference empirically while preserving coarse coverage.

The final categorical readout introduces a separate condition. If a reference probability vector p has top-two margin $m = p_{(1)} - p_{(2)}$, then $\|p - \hat{p}\|_\infty < m/2$ suffices for the student's maximizing token to agree (Proposition 2). Without margin control, arbitrarily small continuous errors can change every decoded token. Conversely, disagreement under shared noise can coexist with identical marginal output distributions. These distinctions motivate measuring map accuracy, decoded behavior, and distributional quality separately; neither bound establishes a MAUVE or external-perplexity ordering. The proofs and counterexamples in Appendix G formalize these limits.

6 Related work

Flow maps and language distillation. Flow Map Matching formalizes maps between times (Boffi et al., 2025); progressive distillation, consistency models, and shortcut models learn related compressed generation procedures (Salimans & Ho, 2022; Song et al., 2023; Frans et al., 2025). Flow Map Language Models derive and compare semigroup, Lagrangian, and Eulerian objectives for text (Lee et al., 2026), while self-conditioned flow maps develop fixed-point flows (Yoo et al., 2026a). Our study examines how comparisons between configured recipes change with the metric, training-cost convention, and inference budget.

Discrete-language generation and distillation. SEDD and masked diffusion establish discrete denoising approaches, with LLaDA extending masked modeling to larger scales (Lou et al., 2024; Sahoo et al., 2024; Nie et al., 2025). SDTT compresses masked trajectories; Di4C models token dependencies; and DiDi uses divergence-based distillation (Deschenaux & Gulcehre, 2025; Hayakawa et al., 2025; Zheng et al., 2026). Trajectory supervision, stochastic mask states, and posterior-bridge consistency provide further routes to few-step generation (Monsefi et al., 2026a; Yoo et al., 2026b; Amin et al., 2026). Autoregressive distillation and speculative decoding offer other acceleration routes (Kim & Rush, 2016; Leviathan et al., 2023); comparisons across these paradigms require measured latency because NFE does not equalize their computation.

Evaluation and categorical readout. Repetition and sampling precision can affect generated text (Holtzman et al., 2020; Welleck et al., 2020; Zheng et al., 2025). MAUVE measures discrepancies between text distributions, while Generative Frontiers motivates evaluating perplexity with distributional entropy (Pillutla et al., 2021; Pynadath et al., 2026). The blockwise entropy screen diagnoses specified token-usage patterns; it does not estimate full sequence entropy. The supporting margin condition specifies when continuous perturbations preserve argmax, without assigning a cause to the observed recipe differences.

Resource accounting and controlled benchmarks. EDM separates training and sampling choices, while importance sampling and Min-SNR address training-time allocation (Karras et al., 2022; Nichol & Dhariwal, 2021; Hang et al., 2023). Our interval control matches coarse exposure when testing finer allocation. DeepOBS, DomainBed, and AlgoPerf make experimental and resource conventions central to comparisons (Schneider et al., 2019; Gulrajani & Lopez-Paz, 2021; Kasimbeg et al., 2025). Our comparisons likewise specify budget, checkpoint treatment, and inference setting. Appendix J develops these connections.

7 Conclusion

DistillScope shows that flow-map distillation comparisons depend on the question being asked. PSD's lower perplexity does not give it the best MAUVE in the replicated comparison; FP's advantage under one scorer at equal updates reverses when training time is matched; and the same LSD checkpoints change degeneration status with inference budget. Map-readout, initialization, ablation, and transfer results define the scope of these observations. The benchmark provides a common framework for reporting these conditional comparisons, enabling recipe choices tied to an explicit quality criterion and resource budget.

AI USE STATEMENT

Generative AI tools assisted in the revision of manuscript text, checks for grammatical and spelling errors, and refinement of language expression. A separate AI-based output audit assigned descriptive labels to selected generated outputs, as reported in Appendix C.4; these labels do not determine the main metric comparisons. The authors are responsible for the final manuscript, code, and other research artifacts, including AI-assisted content.

ETHICS STATEMENT

This work studies language-model distillation using existing benchmark corpora and pretrained models; no new data were collected from human participants. Distilled models can retain biases and reproduce harmful or sensitive content from their training sources, and lower generation costs can increase the scale of misuse. Perplexity, MAUVE, and degeneration screens do not establish factuality, fairness, privacy, or deployment safety. Any downstream use requires evaluation appropriate to its intended setting. Releases of code and derived artifacts remain subject to the access and distribution terms of the underlying datasets and models.

REPRODUCIBILITY STATEMENT

Training and evaluation code, recipe configurations, and analysis scripts will accompany the submission. We plan a public release of these resources. The paper specifies the trained-model cohorts and comparison conditions; Appendices B–E give detailed protocols and results. Appendix F specifies the interval mixtures and separate per-run metric banks, and Appendix G gives the supporting statements and proofs.

REFERENCES

Hasan Amin, Yuan Gao, Yaser Souri, Subhojit Som, Ming Yin, Rajiv Khanna, and Xia Song. Consistent diffusion language models. *arXiv preprint arXiv:2605.00161*, 2026. URL https://arxiv.org/abs/2605.00161v2.

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

Iskander Azangulov, Kianoosh Ashouritaklimi, Leo Zhang, Simon Vary, and Patrick Rebeschini. Masked language flow models. https://kiaashour.github.io/blog/masked-language-flow-models/, 2026. SFT-1028M checkpoint, step 50,000; assembled with the SMDM-1028M base.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research (TMLR)*, 2025.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In *Proceedings of INTERSPEECH*, pp. 2635–2639, 2014.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. arXiv preprint.

Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast LLMs via self-distillation through time. In *International Conference on Learning Representations (ICLR)*, 2025.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations (ICLR)*, 2025. Oral.

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

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2007.01434.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via Min-SNR weighting strategy, 2023. URL https://arxiv.org/abs/2303.09556.

Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Distillation of discrete diffusion through dimensional correlations. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.00364.

Priya Kasimbeg, Frank Schneider, Runa Eschenhagen, Juhan Bae, Chandramouli Shama Sastry, Mark Saroufim, Boyuan Feng, Less Wright, Edward Z. Yang, Zachary Nado, Sourabh Medapati, Philipp Hennig, Mike Rabbat, and George E. Dahl. Accelerating neural network training: An analysis of the AlgoPerf competition. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/6bdde0373d53d4a501249547084bed43-Abstract-Conference.html.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations (ICLR)*, 2024.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

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.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. TinyGSM: achieving >80% on GSM8k with small language models, 2023. arXiv preprint.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. Oral.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations (ICLR)*, 2025. Oral.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.00927.

Xiaocheng Lu, Hualei Zhang, Shuhan Guo, Jie Zhang, Xiaoyi Pang, Jian Liu, Haoxi Li, Bohai Gu, Haoxuan Che, Jingcai Guo, and Song Guo. OPTD: On-policy transition distillation with consistency-guided adaptive compression for few-step diffusion language models, 2026. arXiv preprint.

Amin Karimi Monsefi, Nikhil Bhendawade, Manuel Rafael Ciosici, Dominic Culver, Yizhe Zhang, and Irina Belousova. FS-DFM: Fast and accurate long text generation with few-step diffusion language models. In *International Conference on Learning Representations (ICLR)*, 2026a.

Amin Karimi Monsefi, Dominic Culver, Nikhil Bhendawade, Manuel R. Ciosici, Yizhe Zhang, and Irina Belousova. Trajectory as the teacher: Few-step discrete flow matching via energy-navigated distillation, 2026b. arXiv preprint.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

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

Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. Oral.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. Generative frontiers: Why evaluation matters for diffusion language models. *arXiv preprint arXiv:2604.02718*, 2026. URL https://arxiv.org/abs/2604.02718v1.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.

Frank Schneider, Lukas Balles, and Philipp Hennig. DeepOBS: A deep learning optimizer benchmark suite. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1903.05499.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.02502.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models, 2016. URL https://arxiv.org/abs/1511.01844v3.

Zhongyao Wang. Why do few-step text latents fail when image latents work? non-commitment at sharp categorical readouts. *arXiv preprint arXiv:2606.30705*, 2026.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations (ICLR)*, 2020.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024. arXiv preprint.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models, 2025. arXiv preprint.

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. *arXiv preprint arXiv:2607.00714*, 2026a.

Jaehoon Yoo, Wonjung Kim, Chanhyuk Lee, and Seunghoon Hong. Infinite mask diffusion for few-step distillation. *arXiv preprint arXiv:2605.10518*, 2026b. URL https://arxiv.org/abs/2605.10518v1.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model, 2024. arXiv preprint.

Haoyang Zheng, Xinyang Liu, Cindy Xiangrui Kong, Nan Jiang, Zheyuan Hu, Weijian Luo, Wei Deng, and Guang Lin. Ultra-fast language generation via discrete diffusion divergence instruct. In *International Conference on Learning Representations (ICLR)*, 2026.

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

APPENDIX CONTENTS

The supplementary material summarizes limitations, specifies the training and evaluation protocols, provides detailed comparisons and transfer results, and develops the proofs and diagnostic studies supporting the main analysis, with an extended discussion of related work.

A Limitations . . . **15**

B Training procedures and benchmark configurations . . . **15**
B.1 Environments and comparison units . . . 15
B.2 Continuous transport and time coordinates . . . 15
B.3 The five training objectives . . . 16
B.4 Optimization and checkpoint conventions . . . 17

C Measurement and uncertainty . . . **17**
C.1 Degeneration statistics and calibration . . . 17
C.2 Quality metrics and map-readout measurements . . . 18
C.3 Run-level reporting and sensitivity . . . 18
C.4 Precision and descriptive output audits . . . 18

D Detailed comparisons and ablations . . . **19**
D.1 Per-run perplexity and MAUVE . . . 19
D.2 Training exposure and measured training cost . . . 19
D.3 Map-readout results on six corpora . . . 20
D.4 Cloze, conditional, and length-transfer measurements . . . 20
D.5 Architecture and initialization controls . . . 21
D.6 Objective ablations and sampling warps . . . 21
D.7 Continuous-flow OpenWebText transfer . . . 22

E Masked generation and conditional-task extensions . . . **22**
E.1 Absorbing-state transport and the masked models . . . 22
E.2 Native SDTT and the matched short-budget comparison . . . 22
E.3 Equal-exposure long-budget comparison . . . 23
E.4 Masked-framework degeneration profiles . . . 23
E.5 The 1.4B adapter study . . . 24
E.6 GSM8K with LLaDA-8B adapters . . . 24
E.7 Executable correctness in the TinyGSM setting . . . 24

F Interval-sampling protocol . . . **25**
F.1 Component distribution and objective . . . 25
F.2 Analytic exposure matching . . . 26
F.3 Training and evaluation . . . 26

G Proofs and implications of the supporting analysis . . . **26**
G.1 Finite propagation . . . 26
G.2 Categorical stability and sequence distributions . . . 27
G.3 Why the distinctions matter . . . 28

H Additional diagnostic studies . . . **28**
H.1 Fixed-point iteration and generation . . . 28
H.2 Finite propagation in controlled systems . . . 28
H.3 Teacher smoothing and trajectory profiles . . . 29
H.4 Generative behavior in simpler environments . . . 30

I Computation and reproducibility . . . **30**
I.1 Hardware and measured training cost . . . 30
I.2 Precision, accumulation, and numerical checks . . . 30
I.3 Release scope . . . 31
I.4 Use of large language models . . . 31

J Extended related work . . . **31**

A LIMITATIONS

The most detailed comparisons concern five configured recipes on short-form LM1B, with two or three training runs per condition; the long masked-PSD comparison uses one training trajectory. PH and FP are local baselines, and changes in data, initialization, and training procedure limit causal attribution across extensions. The readout–generation contrast includes length extrapolation, and conditional-task results do not establish preserved reasoning performance. These results support comparisons at the reported settings rather than a universal recipe ranking. The entropy and repetition screens are calibrated diagnostics, not human-validated quality measures; perplexity and MAUVE do not establish factuality or usefulness. NFE counts network evaluations rather than deployment latency. Similar sampling-control scores do not establish equivalence or a generally superior sampler. The supporting bounds give sufficient conditions for propagation and readout stability, not a causal explanation of the empirical differences.

B TRAINING PROCEDURES AND BENCHMARK CONFIGURATIONS

B.1 ENVIRONMENTS AND COMPARISON UNITS

The core benchmark trains five configured recipes on LM1B with length-128 BERT-uncased token blocks. The extensions change the data distribution, sequence length, architecture, initialization, or transport, as specified in Table 9. A comparison unit is a trained checkpoint evaluated with a specified sampler and inference budget. Several checkpoints from one training trajectory are dependent observations, and multiple metrics on one output bank are distinct measurements of the same model.

Table 9: **Training settings.** Within a row, changes in checkpoint or inference budget are reported separately. Cross-row comparisons are scoped to the complete settings shown. OWT denotes OpenWebText.

Setting	Block length	Global batch	Training updates
LM1B core	128	128	12k, 25k, 50k, 100k
LM1B capacity and initialization	128	128	12k
Continuous-flow OWT	1,024	128	10k
Masked OWT, six recipes	1,024	64	10k
Masked OWT, extended PSD	1,024	128	10k to 70k
LLaDA-8B/GSM8K adapters	512	32 / 64	2k / 6k

The LM1B reference student has width 768, 12 transformer blocks, head dimension 64, conditioning width 128, dropout 0.1, and MLP ratio four. The vocabulary contains 30,522 tokens. At the reference architecture, teacher weights initialize the student and the second time-conditioning pathway is zero-initialized. For the new widths in the capacity study, the architecture's native initialization replaces the weight transfer. These alternatives are compared directly in Appendix D.5.

B.2 CONTINUOUS TRANSPORT AND TIME COORDINATES

Write x_1 for one-hot data and x_0 for Gaussian noise. The continuous training interpolant is $x_s = (1-s)x_0 + sx_1$. Write $t = m_e(\tau)$ for the monotone conditioning-to-transport map in environment e, so $\tau = m_e^{-1}(t)$. Interpolation coefficients use transport times; network inputs use conditioning times. Where differentiable, the derivative conversion is $\mathrm{d}\tau/\mathrm{d}t = (m_e^{-1})'(t)$. An inference grid specified in conditioning coordinates must be mapped as $t_k = m_e(\tau_k)$ before forming jump coefficients; a uniform conditioning grid is generally not uniform in transport time. For $0 \le s \le t \le 1$ and $s < 1$, the jump with readout D is

$$X_{s\to t}(x_s; D) = \frac{1-t}{1-s}x_s + \frac{t-s}{1-s}D. \tag{4}$$

At $t = 1$ this is the readout itself. The rational expression is not defined at $(s,t) = (1,1)$; no endpoint-identity claim relies on that singular pair. Core generation composes the learned jumps on the inference grid and applies tokenwise argmax at the terminal step. The initial noise supplies randomness; the continuous-flow sampler is deterministic conditional on it.

B.3 THE FIVE TRAINING OBJECTIVES

Let $D_\nu^{st} = D_\nu(x_s, \tau_s, \tau_t)$ be the student's vocabulary probability vector at a token position, D_θ the teacher denoiser, and sg stop-gradient. The shared diagonal loss is

$$\ell_{\text{diag}} = -\sum_v \text{sg}[D_\theta(x_s, \tau_s)]_v \log D^{ss}_{\nu,v}. \tag{5}$$

Per-position losses are averaged over positions and training rows. Half of the default batch is diagonal; $1/64$ is the full-interval boundary branch and $31/64$ is interior. The default interior sampler draws a uniform interval length in conditioning time and then a uniform start conditional on that length. PH restricts this support within each round.

Semigroup self-distillation (PSD). For midpoint $\tau_u = (\tau_s + \tau_t)/2$, with transport time u, evaluate the online student twice without target gradients:

$$\begin{aligned} D_{su} &= D_\nu(x_s, \tau_s, \tau_u), & D_{ut} &= D_\nu(X_{s\to u}(x_s; D_{su}), \tau_u, \tau_t), \\ \lambda_{sut} &= \frac{(1-t)(u-s)}{(1-u)(t-s)}, & \mathcal{T}_{\text{PSD}} &= \text{sg}[\lambda_{sut} D_{su} + (1-\lambda_{sut}) D_{ut}]. \end{aligned} \tag{6}$$

The off-diagonal loss is $-\sum_v \mathcal{T}_v \log D^{st}_{\nu,v}$. The mixture identity follows by expanding the two affine jumps in Equation 4. The target uses the online student, and does not use EMA weights. The PSD target retains its probability readout without entmax projection.

Lagrangian self-distillation (LSD). A target-time JVP gives $\partial_{\tau_t} \log D_\nu^{st}$ at fixed x_s and τ_s. Converting log-probability derivatives to probability derivatives requires both the coordinate change and componentwise multiplication:

$$\partial_t D_\nu^{st} = D_\nu^{st} \odot \partial_{\tau_t} \log D_\nu^{st} \frac{\mathrm{d}\tau_t}{\mathrm{d}t}. \tag{7}$$

The configured target is

$$\mathcal{T}_{\text{LSD}} = \text{entmax}_{2.0}\left(\text{sg}\left[D_\nu(X_{s\to t}(x_s; D_\nu^{st}), \tau_t, \tau_t) - \frac{(t-s)(1-t)}{1-s}\partial_t D_\nu^{st}\right]\right). \tag{8}$$

The off-diagonal loss is cross-entropy against this target. Both the transported self-evaluation and the derivative use the online student; the diagonal branch is the shared teacher anchor. The entmax parameter is 2.0 (Peters et al., 2019). The derivative-zero ablation removes the second term inside the brackets while retaining the other target operations.

Eulerian self-distillation (ESD). The teacher anchors the source prediction $D_s = D_\theta(x_s, \tau_s)$. The total source derivative is computed by a JVP with state tangent $(D_s - x_s)/(1-s)$ and the source-time chain-rule tangent. For interior times with $t < 1$,

$$\mathcal{T}_{\text{ESD}} = \text{sg}\left[D_s + \frac{(1-s)(t-s)}{1-t}\frac{\mathrm{d}}{\mathrm{d}s} D_\nu^{st}\right], \qquad \ell^{\text{off}}_{\text{ESD}} = \sum_v (D^{st}_{\nu,v} - \mathcal{T}_{\text{ESD},v})^2. \tag{9}$$

This recipe uses squared probability error off the diagonal. Its implementation follows the ESD branch of the reference code, including its transport-time conditioning on diagonal rows. Thus the comparison is between configured recipes, including their numerical and target conventions, rather than changing only a symbolic loss term.

Progressive halving (PH). The local PH implementation has four equal-budget rounds. In round $r \in \{1, 2, 3, 4\}$ the maximum conditioning-time jump is 2^{r-4}; the boundary row is $(0, 2^{r-4})$. Its target has the PSD composition form, with both evaluations taken from a frozen copy of the raw student at the round boundary. The first target copy is the initialized student. The off-diagonal loss is cross-entropy. Under fresh initialization this first target is therefore not a teacher-quality denoiser. The diagonal branch still supplies teacher supervision on half the training rows. These fresh-initialization runs characterize this local target-refresh procedure; they do not evaluate progressive distillation from a trained first-round target.

Fixed point (FP). The local FP baseline initializes $D^{(0)} = \mathrm{sg}[D_\nu^{st}]$ and performs four detached iterations with damping $\alpha = 0.3$:

$$D^{(j+1)} = (1-\alpha)D^{(j)} + \alpha D_\nu(X_{s\to t}(x_s; D^{(j)}), \tau_t, \tau_t), \qquad \mathcal{T}_{\mathrm{FP}} = \mathrm{sg}[D^{(4)}]. \tag{10}$$

The target is fitted with cross-entropy. The finite iteration count and damping define this baseline; the measured iteration behavior does not certify a global contraction or convergence to a unique fixed point.

PSD, LSD, ESD, PH, and FP all detach target construction. EMA is used for evaluation in the continuous-flow core, not for these targets. PH is a local progressive-compression implementation, and FP a local self-consistency baseline; neither is presented as a native reproduction of every method in its lineage (Salimans & Ho, 2022; Deschenaux & Gulcehre, 2025; Yoo et al., 2026a).

B.4 Optimization and checkpoint conventions

Core training uses AdamW with $\beta = (0.9, 0.95)$, a 1,000-update warmup, and constant learning rate 3×10^{-4} for PSD, LSD, ESD, and FP, and 10^{-3} for PH. Global batch 128 is implemented by microbatch 64 and two accumulation steps. Raw weights are used at 12k and 25k; EMA weights with decay 0.9999 at 50k and 100k. Within-checkpoint NFE comparisons keep the weight choice fixed. Differences across training budgets do not isolate additional updates from the change in weight treatment. Equal-update comparisons match training exposure; equal-time comparisons use measured per-update cost on fixed hardware. The masked framework and its released baselines have separate optimizer and checkpoint conventions, specified in Appendix E.1.

C Measurement and uncertainty

C.1 Degeneration statistics and calibration

For each generated token block y, let $\hat{p}_y(v)$ be its empirical token frequency, $H(y) = -\sum_v \hat{p}_y(v) \log \hat{p}_y(v)$, and $\mathrm{rep4}(y) = 1 - U_4(y)/(L-3)$ for $L \geq 4$, where U_4 counts distinct four-grams. We average these statistics over blocks. Pooled entropy uses token frequencies over the entire output bank. The within-block entropy ceiling is $\log L$, which makes calibration specific to sequence length and tokenizer. The screen is a conjunction of bounds on these aggregate statistics, not a classifier of individual sentences.

Table 10: **Environment-specific degeneration screens.** Entropies are in nats. The 8B setting additionally requires median answer-span length at least eight tokens and trims EOS/EOT before computing window statistics.

Evaluation setting	Mean block entropy	Pooled entropy	Mean rep-4
LM1B, 128 tokens	$[3.553, 4.594]$	≤ 9	≤ 0.008
OWT, 1,024 tokens	$[4.6687, 6.2608]$	≤ 9.1997	≤ 0.1479
1.4B OWT study	$[3.60, 6.19]$	≤ 8.85	≤ 0.249
8B answer spans, window 128	$[2.8694, 4.9762]$	≤ 8.5509	≤ 0.1842

The LM1B reference consists of 6,124 held-out blocks, with mean entropy 4.336 and standard deviation 0.085. Its lower threshold is the smaller of two reference-student means, 4.145 and 4.053, minus 0.5. The upper threshold is the midpoint between the human-reference mean and $\log 128$. The rep-4 threshold is the reference 95th percentile; the reference mean is 0.0013. OWT uses the same general calibration principle with reference entropy 5.4267, student-anchor entropy 5.1687, and teacher-anchor entropy 5.5902. Its repetition reference is higher, with mean 0.0453 and 95th percentile 0.1479.

Passing the screen denotes absence of the specified aggregate pathologies. It does not establish semantic quality. In particular, high-diversity incoherent text can pass an entropy band. The masked-framework one-step rows illustrate this limitation: some screen-passing rows have generative perplexity between roughly 2,700 and 7,000.

C.2 Quality metrics and MAP-readout measurements

The core generation scorers are Qwen2.5-1.5B-Instruct and GPT-2-large (Yang et al., 2024; Radford et al., 2019). They score the generated text separately. Generative perplexity quantifies predictability under the external scorer, not the distilled model's own likelihood. The principal recipe effect averages log-perplexity over training runs before exponentiation. Other tables explicitly report arithmetic run means or individual run values.

Core MAUVE uses 5,000 generated samples, GPT-2-large features, PCA retaining 90% of variance, and fixed clustering randomness (Pillutla et al., 2021). The interval-sampling comparison instead uses its separate 2,048-output bank per run. Metric values computed from different bank sizes or protocols are not combined. MAUVE is retained when its ordering differs from perplexity; no single metric is designated as semantic ground truth.

The map-readout suite corrupts reference data to conditioning times $\tau \in \{0.25, 0.5\}$, predicts clean tokens with a $(\tau \to 1)$ call, and averages token NLL over both noise levels. It evaluates denoising at supplied states. It is neither an autoregressive likelihood nor a discrete-diffusion ELBO, and is not numerically compared with published likelihood perplexities. Cloze and conditional probes likewise specify their corruption and readout. Reference tokens in these denoising probes must not be interpreted as a context-only generation task.

C.3 Run-level reporting and sensitivity

Training runs are the replication units. Output blocks characterize a fitted checkpoint, and several checkpoints or metrics from one run do not create additional training replications. We show individual values or observed ranges where available. A range across two or three runs is a description of that sample, not a confidence interval.

For the principal three-versus-three comparison, the difference of mean log-perplexities is -0.15997744 for Qwen and -0.18625813 for GPT-2. The corresponding geometric-mean reductions are 14.78% and 16.99%. Enumerating the $\binom{6}{3} = 20$ label assignments yields two-sided $p = 0.10$ for each absolute-difference statistic under exchangeability. This sensitivity calculation does not establish a randomized assignment design. Its limited resolution is retained alongside the effect size.

Repeated evaluation of fixed checkpoints gave relative scorer variation of 0.96% for Qwen and 0.25% for GPT-2 in the 1,024-sample LM1B configuration. Between-run variation was larger and changed with environment: pooled relative spreads were 13.3% and 13.7% in the continuous OWT comparison, and 24.7% and 26.7% in the masked comparison. These are descriptive variability estimates from small collections of cells. Uncertainty is assessed within the environment and comparison being reported.

C.4 Precision and descriptive output audits

Core sampling uses float64 state and interpolation arithmetic with fp32 network forwards. An independent 64-sequence evaluation compared fp32 and fp64 states under the same noise. Token agreement was 1.0000, 0.98621, and 0.98364 for student NFE 1, 4, and 8; teacher agreement was 0.90991 and 0.96582 at NFE 8 and 128. Changing the noise bank instead gave agreement 0.02832, 0.02673, 0.02710, 0.02209, and 0.02246 in the same order (Table 11A). These are numerical sensitivity measurements, not estimates of a quality change.

The separate AI-only audit assigned overlapping labels to 760 outputs: 303 pronounced repetition, 312 high-diversity nonsense, 693 ordinary incoherence, 36 acceptable text, and 39 too short or empty. Uncertain labels numbered 1, 7, 7, and 1 for the first four categories. The study uses one GPT-5 Codex judge; it does not measure human agreement or screen sensitivity and specificity. The counts describe the audited collection and do not estimate the quality of any particular best-performing recipe.

Table 11: **Numerical sensitivity and descriptive output labels.** A: token agreement after changing sampler-state precision with fixed noise, or changing the noise bank, using 64 sequences per setting. B: overlapping positive labels from one GPT-5 Codex judge on 760 selected outputs; percentages use 760 as denominator. The bank does not estimate benchmark-wide prevalence.

A Token agreement (%)

Model	NFE	Same noise	New noise
Student	1	100.000	2.832
Student	4	98.621	2.673
Student	8	98.364	2.710
Teacher	8	90.991	2.209
Teacher	128	96.582	2.246

B Descriptive output audit

Label	Count	%
Repetition	303	39.9
Diverse nonsense	312	41.1
Incoherence	693	91.2
Acceptable	36	4.7
Short / empty	39	5.1

D Detailed comparisons and ablations

D.1 Per-run perplexity and MAUVE

Table 12 supplies the individual values underlying the cross-metric comparison. This cohort is reported separately from the principal geometric-mean comparison. Perplexity uses 1,024 generated blocks per cell; MAUVE uses 5,000. All checkpoints are 100k-update EMA weights.

Table 12: **Individual run values for the cross-metric comparison.** The numbered columns identify runs within each recipe; they do not imply a randomized paired design across recipes. Lower perplexity and higher MAUVE are preferred.

Metric	Recipe	Run 1	Run 2	Run 3
Qwen PPL, 8 NFE	PSD	171.9	159.4	160.4
	LSD	193.0	190.0	192.8
GPT-2 PPL, 8 NFE	PSD	213.9	195.8	202.6
	LSD	245.5	244.1	247.1
MAUVE, 8 NFE	PSD	0.518	0.482	0.479
	LSD	0.559	0.602	0.578
Qwen PPL, 4 NFE	PSD	195.4	179.9	180.5
	LSD	245.5	257.5	256.7
GPT-2 PPL, 4 NFE	PSD	236.2	214.6	219.5
	LSD	306.6	319.0	313.8

PSD has lower perplexity at both displayed inference budgets, while LSD has higher MAUVE at eight NFE. The three-run ranges are disjoint on each displayed comparison. These remain small samples of trained models, and their several metrics are not independent confirmations.

D.2 Training exposure and measured training cost

At 100k updates and eight NFE, the two-run EMA comparison gives GPT-2 perplexity 204.9 for PSD and 185.4 for FP, and Qwen perplexity 165.6 and 172.2. Using PSD as denominator, FP is 9.5% lower under GPT-2 and 4.0% higher under Qwen. Matching measured training time extends PSD to 135,576 updates while FP remains at 100k. At that allocation, FP's reported perplexity is 7.1% higher under GPT-2 and 7.7% higher under Qwen. Figure 2 uses the same denominator for both comparisons. The normalized entries are PPL ratios, not absolute perplexities; the equal-time denominator is the extended PSD checkpoint. The one-NFE measurements below use a different inference budget and cannot supply the absolute eight-NFE scores.

A separate two-run PSD extension to 300k updates retains screen-passing outputs across the 96 evaluated checkpoint, NFE, and weight-kind cells. Its best reported one-NFE checkpoint is at approximately 135.6k updates, with Qwen perplexity 225.9 and GPT-2 perplexity 266.9. Longer training does not improve every reported quality measure. This trajectory result does not establish a universally optimal stopping point.

D.3 MAP-READOUT RESULTS ON SIX CORPORA

The suite evaluates PTB, WikiText, AG News, LM1B, ArXiv abstracts, and PubMed abstracts. The entries in Table 13 are mean denoising NLL over two corruption levels and two training runs. PSD is evaluated at 135,576 updates and the other LM1B students at 100k, all with EMA weights. The OWT students use raw 10k checkpoints. The teacher rows are reference evaluations, not additional student runs. This suite is distinct from the equal-100k capability comparison in Table 4; its recipe columns must not be paired with the latter generation measurements as if they represented identical budgets.

Table 13: **Map-readout denoising NLL in nats per token.** Lower is better. Bold denotes the lowest student mean within each row, without a significance claim. Blank FP entries indicate that FP is outside the three-recipe continuous OWT comparison. These NLLs are not model likelihoods.

Corpus	PSD	LSD	PH	FP	Teacher
LM1B-trained models					
PTB	**3.842**	3.977	4.351	5.910	3.657
WikiText	**4.502**	4.653	5.373	6.858	4.076
AG News	**4.157**	4.332	4.771	6.267	3.869
LM1B	**3.606**	3.711	4.111	5.297	3.408
ArXiv	**5.370**	5.607	6.589	8.379	4.885
PubMed	**4.318**	4.499	5.374	6.917	3.853
OWT-trained models					
PTB	**4.093**	4.221	4.539		3.809
WikiText	**3.197**	3.301	3.831		2.874
AG News	**4.471**	4.654	4.973		4.142
LM1B	**3.381**	3.533	3.686		3.192
ArXiv	**4.309**	4.403	5.080		4.072
PubMed	**3.787**	3.910	4.448		3.493

The mean LM1B-family ordering is PSD, LSD, PH, FP on all six corpora; the OWT-family ordering is PSD, LSD, PH. These results describe the models' denoising readouts at the tested states, independently of the orderings of their free-running generated text.

D.4 CLOZE, CONDITIONAL, AND LENGTH-TRANSFER MEASUREMENTS

Table 14 retains the separate 100k, two-run capability comparison. The denoising probes read clean-token predictions from corrupted reference states. The conditional probes use globally interpolated context states. Generative log-perplexity instead evaluates new length-256 samples at eight NFE, twice the 128-token training length. This is a length-extrapolation measurement. The retrieval-copy probe is a reference-conditioned readout measurement, not an executable reasoning task. The underlying units and denominators differ across these tasks, so their magnitudes are not averaged together.

Table 14: **Denoising and generated-text measurements.** EM is exact match; NLL and log-PPL use natural logarithms. *The teacher's length-256 outputs are degenerate, with mean entropy 1.85 and rep-4 0.50; these low log-perplexities do not indicate superior text quality.

Measure	PSD	LSD	PH	FP	Teacher
Cloze EM, $\tau = 0.5 \uparrow$	0.6701	0.6690	0.6444	0.6555	0.6777
Cloze NLL, $\tau = 0.5 \downarrow$	3.183	3.243	3.710	4.035	3.088
Cloze NLL, $\tau = 0.25 \downarrow$	4.008	4.157	4.508	6.488	3.715
Conditional next-token EM $\uparrow$	0.6035	0.6104	0.5815	0.5791	0.6240
Conditional suffix NLL $\downarrow$	4.284	4.461	4.800	6.692	4.002
Retrieval-copy EM $\uparrow$	0.5820	0.5810	0.5570	0.5479	0.5970
256-token GPT-2 log-PPL $\downarrow$	5.662	5.829	5.430	5.234	2.64*
256-token Qwen log-PPL $\downarrow$	5.506	5.671	5.271	5.163	2.62*

FP also changes differently across the two measurement axes with budget. Its mean map-readout NLL rises from 12k raw checkpoints, three runs, to 100k EMA checkpoints, two runs: PTB 5.307 to 5.910; WikiText 6.237 to 6.858; AG News 5.725 to 6.267; LM1B 4.830 to 5.297; ArXiv 7.629 to 8.379; and PubMed 6.216 to 6.917. Its Qwen generative perplexity at eight NFE decreases from 184.0 to 172.2. Because both checkpoint treatment and run composition differ between budgets, this is a descriptive contrast rather than an isolated causal effect of more updates.

D.5 Architecture and initialization controls

The capacity study uses the three architectures in Table 15. The two new sizes have two runs per recipe at 12k updates and batch 128. Their optimizer settings equal the reference settings without a per-size search. Head dimension remains 64 and conditioning width 128. Native initialization uses zero final layers and zero adaptive-normalization modulations; the target-time embedder's final layer is also zero-initialized.

Table 15: **Measured student parameter counts.** Non-embedding parameters span about 29 times, while total parameters span about 11.8 times. These are different capacity measures.

Total params	Non-embedding	Width	Blocks	Heads	Initialization
28,618,618	9,054,016	320	6	5	Fresh
139,342,394	92,430,080	768	12	12	Teacher
336,383,738	262,123,712	1,216	14	19	Fresh

Table 16: **PH outcomes across architecture and initialization settings.** 12k updates, raw weights. Counts denote passing training runs. Architecture and initialization change together; no reference-size fresh-PH control is available.

Total parameters	Initialization	1 NFE	8 NFE
28.6M	Fresh	0/2	2/2
139.3M	Teacher	3/3	3/3
336.4M	Fresh	0/2	2/2

Table 16 separates the measured PH settings. PH's mean within-block entropies at one NFE are 2.942 and 0.299 at the smaller size, and 1.057 and 0.265 at the larger size, below the LM1B screen floor. Both sizes pass at eight NFE. The separate reference-size fresh-initialization control covers PSD and LSD, one run each. Their screen outcomes agree with the teacher-initialized comparison at all four NFEs: PSD passes at 1, 2, 4, and 8; LSD fails at 1 and 2 and passes at 4 and 8. PSD's eight-NFE Qwen perplexity changes from 125.1 to 407.4 and MAUVE from 0.456 to 0.054. The control does not cover PH, so the PH capacity contrast cannot be assigned specifically to initialization or capacity. In addition, fresh PH begins with a frozen, untrained off-diagonal target, while the shared diagonal teacher anchor remains active (Appendix B.3).

D.6 Objective ablations and sampling warps

The LSD study uses a factorial over derivative-term removal, time-sampling warp, and batch composition, with additional clipping and sampling controls, totaling 15 runs at 12k updates. All 15 fail the one-NFE screen and pass at eight NFE. In particular, none of the four derivative-zero configurations passes at one NFE. The result concerns removing that correction while retaining the online-student target, readout projection, and other recipe operations.

The early and late sampling warps are $x \mapsto x^2$ and $x \mapsto \sqrt{x}$ in conditioning time. The two late-warp runs fail at two NFE and pass at four and eight. Their Qwen perplexities at eight NFE are 212.8 and 259.5. Single-run early-warp and base comparisons give 153.7 and 181.1; these values do not provide a replicated early-versus-base improvement estimate. At two NFE, the three-run base recipe grid has mean entropies 2.62, 4.14, and 2.05, showing variation near the screen boundary.

A separate pilot numerical comparison crosses exact versus finite-difference JVPs with bf16 versus fp32 for LSD and ESD. All eight configurations have entropy below the pilot criterion of two nats. This criterion is distinct from the final three-clause screen. The pilot does not establish that precision

or derivatives are irrelevant in every regime; it shows that these numerical alternatives did not remove the observed low-entropy outcome there.

D.7 Continuous-flow OpenWebText transfer

PSD, PH, and LSD train on length-1,024 OWT blocks for 10k updates at global batch 128, two runs each. At one NFE, the separate run entropies are PSD 3.862 and 4.057; PH 4.146 and 4.088; LSD 4.204 and 4.703. The OWT lower bound is 4.6687. All six raw checkpoints pass the full OWT screen at two NFE. This setting changes corpus, tokenizer, block length, and training exposure together, so it does not isolate any one factor behind the difference from LM1B.

The OWT LAMBADA probe uses on-manifold context and a uniform-simplex soft mask on the target span, one call, and exact match across all target wordpieces. Out of 5,153 items, the teacher solves 16; PSD solves 14 and 7, LSD 11 and 11, and PH 4 and 2 in their two runs. The GPT-2-small context-only reference solves 1,800. These low flow-map completion rates are specific to the tested probe and models; they do not establish an impossibility of conditioning flow-map language models.

E Masked generation and conditional-task extensions

E.1 Absorbing-state transport and the masked models

The masked OWT comparison uses the released MDLM-110M teacher (Sahoo et al., 2024; Deschenaux & Gulcehre, 2025): width 768, 12 blocks and heads, conditioning width 128, sequence length 1,024, and vocabulary size 50,258 including a mask token M. The teacher's backbone is time-unconditioned. The flow-map students add two time embedders with zero final layers, preserving the teacher prediction at initialization. The native SDTT student remains time-unconditioned, with its step size specified by the distillation round.

For reveal level g, a data token y is retained with probability g and masked otherwise:

$$\Pr(x_g \mid y) = g\delta_y + (1-g)\delta_{\mathrm{M}}, \qquad g = 1-(1-\epsilon)t_{\mathrm{code}}, \quad \epsilon = 10^{-5}. \tag{11}$$

Here $t_{\mathrm{code}} \in [0,1]$ is the masked model's corruption-time coordinate: zero denotes clean data, and one denotes mask probability $1-\epsilon$. Generation decreases t_{code} and increases reveal level g; the symbols s, u, t below denote reveal levels, not this corruption coordinate. The affine conversion gives $\mathrm{d}g/\mathrm{d}t_{\mathrm{code}} = -(1-\epsilon)$. The transition from s to u reveals a still-masked position with probability $(u-s)/(1-s)$, drawing its token from the predicted categorical distribution. Already revealed positions are retained. The PSD and PH target mixture uses $\lambda = (u-s)/(t-s)$, determined by the relative reveal masses. LSD uses the target correction $(t-s)\partial_t D_{s,t}$; ESD uses $(t-s)\mathrm{d}_s D_{s,t}$ with the derivative through a soft one-hot embedding and the absorbing transport's state tangent. These implement the masked recipe variants; the continuous-transport interpolation coefficient is not reused.

The SUBS readout suppresses the mask logit and preserves observed tokens on unmasked positions. FP uses damping 0.3 and four iterations, sharing one realization of reveal and categorical noise across the iterations. Evaluation uses the MDLM/SDTT ancestral sampler with float64 probability arithmetic. Remaining masks are denoised in the terminal pass, according to the shared implementation's evaluation convention.

E.2 Native SDTT and the matched short-budget comparison

The native SDTT arm uses the published reverse-KL variant and a two-step-to-one-step target in its first round, compressing the 1,024-step schedule to 512 steps. Training uses learning rate 6×10^{-5}, AdamW with $\beta = (0.9, 0.999)$ and $\epsilon = 10^{-8}$, zero weight decay, 2,500 warmup updates, gradient clipping at one, and EMA decay 0.9999. The five flow-map recipes retain their configured optimizer settings rather than receiving a masked-environment tuning sweep. These comparisons evaluate complete procedures under the stated budgets, including their optimization and checkpoint conventions; they do not isolate objective functions.

The short-budget comparison trains six recipes, including SDTT, for 10k updates at global batch 64, with two runs each. The data convention is BOS, 1,022 OWT tokens, and EOS. Flow-map arms

use raw checkpoints; native SDTT additionally uses its operative EMA checkpoints. The released first-round checkpoint was trained at batch 128 and is a contextual reference rather than a matched-exposure arm.

The SDTT generative-perplexity convention scores raw token IDs with GPT-2-large and exponentiates the mean per-sample mean NLL. At 512 NFE, the two locally trained SDTT runs score 42.2 and 42.5, the released first-round checkpoint 43.6, and the teacher 52.1. These results demonstrate functioning first-round distillation under the stated local setting; they are not a reproduction of every experiment in the original SDTT paper.

At eight NFE, the screen-passing short-budget arms have broad score ranges: approximately 370–757 under the SDTT perplexity convention, 407–822 under Qwen, and MAUVE 0.005–0.066. The released seventh-round model gives 104.5, 117.7, and 0.247 respectively. This short-budget comparison does not match the released model's training exposure.

The SDTT-style MAUVE configuration uses the first 100 tokens, GPT-2-large features, five clustering seeds, and scaling factor five. Our banks are unconditional; this differs from the original paper's conditional-prefix evaluation. Its absolute MAUVE values are therefore compared only within the local evaluation protocol, separately from the core MAUVE configuration.

E.3 Equal-exposure long-budget comparison

The long PSD run matches the released seventh-round model's updates, batch size, and block length:

$$70{,}000 \times 128 \times 1{,}024 = 9{,}175{,}040{,}000 \quad \text{training tokens.}$$

One PSD run reaches the full budget; another reaches the 10k point at batch 128. The long run keeps its PSD optimizer settings fixed. Raw checkpoints at 10k, 20k, 35k, 50k, and 70k updates are evaluated at NFE 1, 2, 4, and 8. The released SDTT model is evaluated as released. The different target construction, round structure, optimizer history, and released-weight treatment remain part of the comparison; equal token exposure does not isolate these factors. The single trajectory does not estimate between-run uncertainty or an intrinsic scaling limit of PSD, and the raw/EMA difference does not identify its cause.

Table 17: **Masked OWT training-budget comparison.** PSD rows belong to one training trajectory. Perplexity uses the SDTT scoring convention. The released comparator has the same final token exposure.

Model	Updates	PPL at 8 NFE	Weights
PSD	10,000	665.7	Raw
PSD	20,000	550.9	Raw
PSD	35,000	537.5	Raw
PSD	50,000	541.1	Raw
PSD	70,000	620.1	Raw
PSD	70,000	958.1	EMA
Released **SDTT**, round 7	70,000	104.5	As released

The 24 raw checkpoint–NFE cells from the five-point long trajectory and the second run's 10k checkpoint all pass the masked-framework screen. This is a count of evaluated cells, not 24 training replications. The long-budget score gap persists among these screen-passing samples.

E.4 Masked-framework degeneration profiles

The masked comparison also contains a profile that differs from core LM1B recovery. In one LSD run, mean within-block entropy decreases from 2.32 at one NFE to 1.41 at eight, while rep-4 rises from 0.27 to 0.57. The other LSD run passes at the evaluated NFEs. FP fails at one through four NFE and passes at eight in both runs. Thus additional inference can accompany either recovery or greater degeneration under the tested masked recipes.

The low-entropy LSD run has generative perplexity 9.0 at eight NFE under the SDTT scoring convention, below the teacher's 512-step score of 52.1. Another FP row reaches Qwen perplexity 1.9 with rep-4 0.965. These values illustrate why very low external perplexity must be interpreted alongside the generated output's statistics.

E.5 THE 1.4B ADAPTER STUDY

The OWT extension uses a released flow-model teacher with 1,461,894,656 assembled parameters and a TinyLlama/LLaMA-2-format tokenizer (Azangulov et al., 2026; Zhang et al., 2024). The student has a frozen base and a two-time adapter, with 437.7M trainable parameters. It uses the teacher family's ancestral token-commitment sampler. Four recipes run for 2k updates at batch 32, one run each. FP and LSD are also trained for 12k updates, two runs each; raw weights and 1,024-output banks are used for these comparisons.

At 2k updates and eight NFE, Qwen perplexities are FP 105.4, PH 667.8, LSD 672.9, and PSD 929.5. These point estimates do not persist as a reliable ordering at the longer budget. Table 18 shows the 12k entropy and repetition values. Both FP runs fail at one NFE; at larger NFE one passes and one fails. Both LSD runs fail at one NFE and pass at 2, 4, and 8. The screen uses the environment-specific thresholds in Table 10.

Table 18: **1.4B OWT outputs at 12k updates.** Mean within-block entropy in nats and mean repeated-four-gram fraction. Values rounded to zero are displayed at their reported precision.

Recipe / NFE	Run 1 entropy	Run 1 rep-4	Run 2 entropy	Run 2 rep-4
FP / 1	0.16	0.99	0.00	1.00
FP / 2	1.19	0.79	4.58	0.009
FP / 4	1.29	0.77	5.12	0.000
FP / 8	1.48	0.70	5.10	0.000
LSD / 1	1.10	0.63	0.97	0.67
LSD / 2	4.90	0.005	4.35	0.006
LSD / 4	4.89	0.002	4.95	0.000
LSD / 8	4.82	0.002	4.94	0.001

The teacher's 128-step self-samples also repeat, with mean entropy 2.14, rep-4 0.71, and Qwen perplexity 4.6. The similarly repetitive FP run at eight NFE has entropy 1.48, rep-4 0.70, and Qwen perplexity 5.1. Their similar output statistics do not isolate a dynamical mechanism of transfer. The experiment establishes that the favorable short-budget FP score did not reliably persist at 12k in this setting.

E.6 GSM8K WITH LLADA-8B ADAPTERS

The teacher is LLaDA-8B-Instruct (Nie et al., 2025); students use rank-32 LoRA on attention and MLP projections plus two time embedders, totaling 119.6M trainable parameters. Training uses 7,473 GSM8K training examples, chat-templated with boxed final answers and padded to length 512 (Cobbe et al., 2021). Short runs use 2k updates at effective batch 32. PSD and an SDTT-style objective are also evaluated after 6k updates at batch 64, two runs each. The latter increases both the update count and the number of examples per update.

Sampling uses temperature zero, no guidance, low-confidence remasking, EOS/EOT confidence forcing, and a full 512-token canvas. Answers are scored with a boxed-answer extractor followed by a last-number fallback. The 6k student evaluations use all 1,319 test items; the teacher reference uses a fixed 200-item subset. The teacher scores 0%, 6.5%, 7.5%, and 65.5% at NFE 1, 4, 8, and 128 on that subset. These teacher percentages contextualize the teacher on its own subset; they are not paired full-test comparisons with the student scores. We do not compute teacher–student accuracy gaps across the unequal subsets.

E.7 EXECUTABLE CORRECTNESS IN THE TINYGSM SETTING

The TinyGSM extension evaluates distillation from a code-generating teacher for 10k updates, with two runs each of PSD, LSD, PH, and FP. TinyGSM denotes the synthetic Python-solution dataset; the original paper's 81.5% result combines a generator with a verifier selecting among multiple candidates (Liu et al., 2023). That composite result is not a directly comparable teacher baseline. Generated programs are executed on all 1,319 GSM8K test cases. At 128 NFE the teacher solves four items, or 0.30%; at NFE 1, 2, 4, and 8 it solves none. At eight NFE, PSD solves one and zero items in its two runs, LSD one and one, and PH and FP zero in both. All eight student runs

Table 19: **LLaDA-8B adapter students on GSM8K.** Raw 6k checkpoints, batch 64, full 1,319-item test set. All displayed runs fail the answer-window entropy screen. This experiment evaluates the stated adapter configuration and budgets, not the general feasibility of large-model few-step distillation.

Recipe / NFE	Run 1 accuracy	Run 2 accuracy	Run 1 entropy	Run 2 entropy
PSD / 1	0.76%	0.08%	1.46	1.15
PSD / 4	1.06%	1.29%	2.13	2.05
PSD / 8	1.44%	3.94%	2.30	2.09
SDTT-style / 1	0.61%	0.45%	1.59	1.59
SDTT-style / 4	3.03%	2.65%	1.69	1.77
SDTT-style / 8	4.25%	3.18%	1.78	1.86

pass the corresponding generation screen at eight NFE. The teacher and students operate near the accuracy floor of this execution protocol. These counts provide no useful recipe ordering or evidence of preserved reasoning capability. Screen status and executable solve rate are reported as separate observations for this configured evaluation.

F Interval-sampling protocol

F.1 Component distribution and objective

The interval mixtures operate in model-conditioning time $\tau \in [0, 1]$. The transport-time parameterization of a recipe is applied after sampling its conditioning times. Table 20 specifies both mixtures. The displayed weights are rounded to six decimal places; normalize each column to obtain probabilities π_c.

Table 20: **Fixed component weights for the PSD sampling comparison.** Both samplers use the same seven interval components and the same conditional rule within each component. These weights apply to the interior branch.

Component $[a_c, b_c]$	Fixed mixture	Joint exposure control
$[0, 0.25]$	0.021270	0.041618
$[0.25, 0.5]$	0.059017	0.041618
$[0.5, 0.75]$	0.126007	0.183977
$[0.75, 1]$	0.244896	0.183977
$[0, 0.5]$	0.059017	0.056067
$[0.5, 1]$	0.244896	0.247847
$[0, 1]$	0.244896	0.244896

Conditional on component c of width $w_c = b_c - a_c$, draw $d \sim U(0, w_c)$ and $u \sim U(0, 1)$ independently and set

$$\tau_s = a_c + (w_c - d)u, \qquad \tau_t = \tau_s + d. \tag{12}$$

Thus interval length is uniform conditional on the component. The density is

$$q_c(\tau_s, \tau_t) = \frac{\mathbf{1}\{a_c < \tau_s < \tau_t < b_c\}}{w_c\big(w_c - (\tau_t - \tau_s)\big)}, \qquad q = \sum_c \pi_c q_c. \tag{13}$$

The formula follows from the length density $1/w_c$ and the conditional start density $1/(w_c - d)$. It integrates to one because, for each length d, the admissible starts have total length $w_c - d$. This density is not uniform over triangle area.

Half of the training rows use the diagonal teacher-anchor branch. A fraction $1/32$ of the remaining rows use the exact full-interval boundary pair $(\tau_s, \tau_t) = (0, 1)$, leaving $31/64$ of all rows for the interior mixture. The continuous component $[0, 1]$ is distinct from this boundary point mass. Both arms apply the sampled losses without importance correction, yielding Equation 3.

F.2 Analytic exposure matching

For an interior draw, the interval-length density and expectation are

$$g(d) = \sum_c \frac{\pi_c}{w_c} \mathbf{1}\{0 < d < w_c\}, \qquad \mathbb{E}d = \frac{1}{2} \sum_c \pi_c w_c. \tag{14}$$

After normalization, the total component weights at each of the widths $1/4$, $1/2$, and 1 differ between mixtures by less than 7×10^{-7}. Equation 14 therefore yields nearly identical length distributions, each with mean approximately 0.2548. For the event that an interval lies entirely in the late half, $\tau_s \geq 1/2$, the third and fourth quarter components and the late-half component contribute their full probability. The full-interval component contributes a fraction

$$\Pr(\tau_s \geq 1/2 \mid [0,1]) = \int_0^{1/2} \frac{1/2 - \ell}{1 - \ell} \, \mathrm{d}\ell = \frac{1 - \log 2}{2}. \tag{15}$$

Consequently, late-half exposure is approximately 0.6534 for both mixtures. Matching these summaries does not match their complete joint time density: the control redistributes probability within the early and late halves. The unchanged diagonal and boundary branches preserve this interpretation for the full training distribution.

F.3 Training and evaluation

Both arms use the PSD recipe implemented by `SchedDistiller`, τ conditioning, 12k training updates, and raw checkpoint weights. The microbatch is 64 with two accumulation steps, giving effective batch size 128. The diagonal fraction is 0.5 and the boundary parameter is 32. Each arm has two training runs. Each run generates its own 2,048-output bank at one NFE. Qwen NLL, GPT-2 NLL, and MAUVE are computed separately for each bank and then averaged across the two runs. Banks are not pooled before metric computation. Table 8 reports these averages; the MAUVE ordering between arms reverses across the two runs.

G Proofs and implications of the supporting analysis

G.1 Finite propagation

Fix a time grid and reference maps Φ_k and student maps X_k, for $k = 0, \ldots, K-1$. Starting from the same noise state, let $x_{k+1} = \Phi_k(x_k)$ and $\hat{x}_{k+1} = X_k(\hat{x}_k)$. Adding and subtracting $X_k(x_k)$ gives the exact decomposition

$$\hat{x}_{k+1} - x_{k+1} = \underbrace{X_k(\hat{x}_k) - X_k(x_k)}_{a_k\text{: propagated discrepancy}} + \underbrace{X_k(x_k) - \Phi_k(x_k)}_{b_k\text{: local residual}}. \tag{16}$$

Proposition 1 (Finite propagation). *For a chosen norm, write $E_k = \|\hat{x}_k - x_k\|$ and $\varepsilon_k = \|b_k\|$. If finite nonnegative factors T_k satisfy $\|a_k\| \leq T_k E_k$, then*

$$E_{k+1} \leq \varepsilon_k + T_k E_k, \qquad E_K \leq \sum_{i=0}^{K-1} \varepsilon_i \prod_{j=i+1}^{K-1} T_j, \tag{17}$$

where $E_0 = 0$ and an empty product equals one.

This standard triangle-inequality bound separates the residual introduced on each interval from its subsequent amplification. A valid finite factor can be defined as $\|a_k\|/E_k$ when $E_k > 0$, and zero otherwise. Bounding it with derivatives requires control along the entire segment between the two input states. The realized factor is a diagnostic of a specified pair of trajectories; using it in the bound does not independently predict their error. Moreover, in Euclidean norm, $\|a_k + b_k\|_2^2 = \|a_k\|_2^2 + \|b_k\|_2^2 + 2\langle a_k, b_k \rangle$, so the two contributions can reinforce or cancel. The proposition describes fixed maps and does not imply that increasing NFE or allocating more training to an interval must reduce the terminal error.

Proof of Proposition 1. Equation 16 follows by adding and subtracting $X_k(x_k)$. The triangle inequality and the assumed finite-factor bound give

$$E_{k+1} = \|a_k + b_k\| \le \|a_k\| + \|b_k\| \le T_k E_k + \varepsilon_k.$$

Repeated substitution, using nonnegativity of every T_k, gives

$$E_K \le E_0 \prod_{k=0}^{K-1} T_k + \sum_{i=0}^{K-1} \varepsilon_i \prod_{j=i+1}^{K-1} T_j.$$

The initial term vanishes because the trajectories start from the same state. □

If $E_k = 0$, then the inputs coincide and $a_k = 0$, so setting $T_k = 0$ is valid. Otherwise $T_k = \|a_k\|/E_k$ is a valid realized factor. A sufficient derivative condition is continuous differentiability on a neighborhood of the segment joining x_k and $\hat{x}_k$, with induced operator norm $\|DX_k(z)\|_{\mathrm{op}} \le L_k$ everywhere on that segment. Integration along the segment then gives $\|a_k\| \le L_k E_k$. A derivative evaluated only at an endpoint or a finite collection of probes does not establish this segment condition.

For random inputs, the proposition holds pointwise. Expectations of its right-hand side remain expectations of products of residuals and finite factors; no independence or replacement by products of means is implied. In the Euclidean case the squared decomposition includes $2\langle a_k, b_k\rangle$. This term may have either sign, including after expectation when the vectors are square-integrable.

G.2 Categorical stability and sequence distributions

At the terminal step, the benchmark applies argmax to each probability vector. For a reference vector p and student vector q, define the reference margin $m = p_{(1)} - p_{(2)}$ and the coordinate error $D = \|p - q\|_\infty$, where $p_{(1)}$ and $p_{(2)}$ are the largest and second-largest coordinates of p, irrespective of their vocabulary indices.

Proposition 2 (Margin-controlled readout)**.** *If $D < m/2$, reference and student argmax tokens agree. For random inputs or token positions and any $\delta > 0$, with a common fixed rule for ties,*

$$\Pr(\arg\max p \ne \arg\max q) \le \Pr(m \le 2\delta) + \Pr(D > \delta) \le \Pr(m \le 2\delta) + \frac{\mathbb{E}D^2}{\delta^2}. \tag{18}$$

Proof of Proposition 2. When $m > 0$, let i be the unique maximizing coordinate of p. For any $j \ne i$,

$$q_i - q_j \ge p_i - p_j - |q_i - p_i| - |q_j - p_j| \ge m - 2D.$$

Thus $D < m/2$ makes i the unique maximizing coordinate of q as well. On the event $\{m > 2\delta, D \le \delta\}$ this strict inequality holds. Disagreement is therefore contained in $\{m \le 2\delta\} \cup \{D > \delta\}$, including reference ties in the first event. The union bound proves the first probability inequality. Applying Markov's inequality to D^2 proves the second. □

For a sequence of length L, let m_r, D_r denote the corresponding quantities at position r. Under the same initial-noise coupling, let Y and $\hat{Y}$ be the reference and student token sequences, with laws P and Q. For arbitrary thresholds $\delta_r > 0$,

$$\mathrm{TV}(P, Q) \le \Pr(Y \ne \hat{Y}) \le \sum_{r=1}^{L} \left[\Pr(m_r \le 2\delta_r) + \frac{\mathbb{E}D_r^2}{\delta_r^2} \right]. \tag{19}$$

The second inequality is a union bound over positions followed by Proposition 2; it needs no independence between tokens. The first follows because for any sequence event A, $|\Pr(Y \in A) - \Pr(\hat{Y} \in A)| \le \Pr(Y \ne \hat{Y})$, followed by a supremum over A. With the coordinate maximum norm on the full state, $D_r \le E_K$ for every r, so Proposition 1 supplies an upper envelope for the error terms. The bound can be loose for long sequences or small margins.

G.3 WHY THE DISTINCTIONS MATTER

Small continuous error can still change every decoded token. For $0 < \epsilon < 1/2$, consider $p = (1/2 + \epsilon, 1/2 - \epsilon)$ and $q = (1/2 - \epsilon, 1/2 + \epsilon)$. Their coordinate error is $2\epsilon \to 0$, but their argmax tokens always differ. The reference margin also tends to zero. A guarantee uniform over such examples therefore needs a margin condition or another restriction on the readout distribution.

Paired disagreement need not imply different output laws. Let Z be uniform on $\{-1, +1\}$ and define reference and student outputs $\Phi(Z) = Z$ and $X(Z) = -Z$. Under this coupling their squared error is always four and their outputs always disagree, yet both marginal laws are uniform on the same two values. Equation 19 is an upper bound, not an identity between a particular coupling's disagreement and distributional error. Finally, closeness to the reference law alone does not establish its closeness to the data distribution or determine the quality measured by a particular language-model scorer.

H ADDITIONAL DIAGNOSTIC STUDIES

H.1 FIXED-POINT ITERATION AND GENERATION

An empirical contraction diagnostic for the damped FP iteration does not determine whether its generated text passes the degeneration screen. Figure 5 compares the reported diagnostic at the one-NFE interval with its range over screen-passing inference settings. Four of the five displayed one-NFE failures have values below one. The fifth has value 1.145. All 37 reported checks on screen-passing cells have values below one, between 0.766 and 0.901. These are empirical iteration measurements on evaluated states; they do not bound the operator uniformly over its state space.

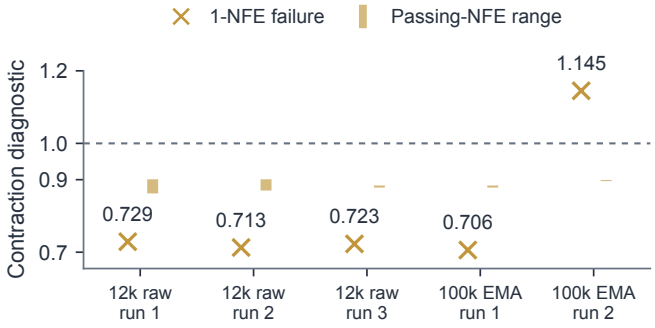

Figure 5: **Iteration stability does not imply generation quality.** Crosses show one-NFE failures. Segments give observed diagnostic ranges across passing settings at NFE 2, 4, and 8. The line at one is a numerical reference, not a certified global contraction bound.

On the training-distribution diagnostic, the contraction value remains approximately 0.78 between 12k and 100k updates while the initial Picard update norm decreases from 0.172 to 0.047. A separate sampled Jacobian proxy varies across the five displayed runs as 0.083, 5.779, 0.102, 0.156, and 0.133. These quantities summarize different aspects of the iteration. Neither an observed update ratio nor a derivative probe establishes a guarantee about the eventual decoded sequence. Convergence toward a fixed prediction can be compatible with repetitive text, just as additional inference can change the generation regime without changing the trained weights.

H.2 FINITE PROPAGATION IN CONTROLLED SYSTEMS

The controlled systems comprise a two-atom relaxed categorical teacher and a two-dimensional, three-component Gaussian-mixture teacher with analytic velocity and Jacobian. The former uses $z_t = (1-t)\epsilon + te_K$ for a sampled category K. Reference trajectories use fourth-order Runge–Kutta integration, with 128 substeps per segment and 256 over the full interval. Reported integration discrepancies are 7×10^{-9} for the two-atom system and 2×10^{-5} for the mixture system. These trajectories are numerical references, not exact flow maps.

The injected-residual experiment uses perturbations with scale 0.02 times the state RMS on banks of 1,024 states. Its 888 evaluation cells combine two teachers, three runs, two residual profiles, two direction choices, and 37 composition schedules. The residual profiles are constant across intervals or proportional to interval duration. Injected students have the form $X_k = \Phi_k + \varepsilon_k g_k$ with $\|g_k\|_2 = 1$, using coherent or random directions. The schedule count comprises four chain families

at $N \in \{1, 2, 3, 4, 6, 8, 12, 16\}$ and one semigroup family at $N \in \{1, 2, 4, 8, 16\}$, giving $4 \times 8 + 5 = 37$ settings. The error is the sample mean of Euclidean endpoint discrepancies. Cells share teachers, states, and perturbations and are therefore not 888 independent replications. With realized finite factors computed from the paired trajectories, the propagated envelope is an application of Proposition 1; the experiment measures its slack rather than testing a new law.

Table 21: **Propagation diagnostics in controlled systems.** Ratios compare the specified envelope or approximation with realized error. A ratio below one is underestimation. Realized finite factors and sampled derivatives have different mathematical status.

Construction	Reported ratio	Interpretation
Realized finite factors, coherent residual directions	Median 1.002; 90th percentile 1.015; maximum 1.03	Little directional cancellation in this construction.
Realized finite factors, random residual directions	Median 1.45; maximum 18.2	Cancellation can make the triangle-inequality envelope loose.
Sampled local-Jacobian replacement	Minimum 0.024	A local proxy can underestimate finite propagation.
Fitted small students, plug-in approximation	Range 0.83 to 1.9	The measured plug-in quantity is not a certified upper bound.

For the fitted-student experiment, 30 training runs per teacher yield median rank correlations of approximately 0.70 and 0.90 between the plug-in prediction and realized composition error. The ratio range in Table 21 includes underestimation. These results motivate measuring finite propagation directly when quantitative guarantees are required.

The language-model composition studies evaluate 18 distinct checkpoint pairs, with a nine-pair subset also examined in a separate analysis. Sampled probe-maximum envelopes exceed the reported realized errors but are 10.7 to 16.0 times as large on the 18-pair grid. A directional proxy underestimates error on all 18 pairs. Observed upper coverage on these checkpoints does not establish the segment-wide derivative condition in Appendix G. The signed cross term in the exact squared-error identity is retained in the analysis; its magnitude and sign cannot be replaced by a universal categorical constant.

H.3 Teacher smoothing and trajectory profiles

The teacher comparison trains a label-smoothed model with target $(1 - 0.1)e_y + 0.1\,\mathbf{1}/V$, using the reference 12-block, width-768 architecture. Training uses batch 512, AdamW with learning rate 3×10^{-4}, 2,500 warmup updates, $\beta = (0.9, 0.999)$, and zero weight decay. The compared teacher checkpoints are the unsmoothed model at 45k updates with EMA and the smoothed model at 50k. Their 128-NFE Qwen perplexities are 244.6 and 259.4; GPT-2 perplexities are 313.5 and 332.0.

PSD and LSD students are trained for 12k updates, with two runs per recipe and teacher. Table 22 reports the eight-NFE recipe gaps. The Qwen gap changes little in its point estimate, but the uncertainty of that change is large. Since smoothing and teacher checkpoint budget differ together, this comparison does not isolate a single teacher property.

Table 22: **LSD minus PSD generative-perplexity gaps at eight NFE.** Positive values favor PSD. Values after $\pm$ are the reported standard errors from two-run comparisons, not confidence intervals. Changes use the underlying unrounded values.

Scorer	Unsmoothed teacher	Smoothed teacher	Change in gap
Qwen	70.0 ± 58.9	69.1 ± 3.3	-0.8 ± 59.0
GPT-2	71.4 ± 37.9	78.7 ± 10.4	7.3 ± 39.3

The accompanying trajectory diagnostic follows a 128-step Euler trajectory and probes the denoising posterior every four steps. Let $\bar{\dot{D}}(t)$ denote mean posterior speed, estimated by an ℓ_1 finite difference divided by the time increment, and $\bar{g}(t)$ the mean signed statistic $g(t) = 2q_t(k_0) - 1$, where k_0 is

the initially maximizing token. For $w(s,t) = (t-s)(1-t)/(1-s)$, define

$$B = \mathbb{E}_{s,t}[w(s,t)\overline{\dot{D}}(t)\bar{g}(t)], \qquad C = \mathbb{E}_{s,t}[w(s,t)\overline{\dot{D}}(t)], \qquad m_{\text{eff}} = B/C \quad (C > 0). \tag{20}$$

The interval expectation is approximated with 200,000 samples. The construction multiplies separately averaged profiles; it is not the expectation of the paired tokenwise product. Also, $2q_t(k_0) - 1$ is not the top-two margin used in Proposition 2.

For the teacher, m_{eff} is approximately -0.785%. The corresponding 100k EMA student means are approximately -0.655% for PSD, -0.437% for LSD, -0.542% for PH, -0.637% for FP, and -0.950% for ESD. These similar signs coexist with different generation regimes. They describe the measured profiles and do not supply a classifier for degeneration or a causal explanation of the PSD–LSD comparison.

H.4 Generative behavior in simpler environments

Across 16 grammar-based environments built from Dyck and probabilistic context-free grammar variants, PH passes all 32 evaluated one- and eight-NFE cells, while ESD fails 30 of 32. LSD passes at eight NFE in all 16 environments. In the 19 PSD–LSD environment–NFE comparisons where both pass, PSD has the better reported quality score. The environments and inference settings share modeling choices, so these counts describe coverage across the suite rather than independent draws from all language tasks.

Separate continuous-readout checkerboard and Gaussian-mixture experiments produce low-diversity LSD outputs in 14 of 16 cells at both one and eight NFE, while PH passes throughout. These controls show that the tested failure pattern can also occur without an argmax text readout. The categorical-stability proposition gives a sufficient readout condition, and does not assert that every distillation failure originates at readout.

I Computation and reproducibility

I.1 Hardware and measured training cost

Experiments use single-GPU jobs on RTX A6000 48 GB, RTX 6000 Ada, and A100 80 GB hardware. The measured LM1B cost benchmark uses length 128 and global batch 128 on an A6000. Table 23 gives its per-1,000-update costs. These local measurements characterize training in the stated configuration; they are not deployment latency or cross-hardware throughput measurements.

Table 23: **Measured training costs in the LM1B configuration.** These rounded reference measurements are separate from the per-step timing used for the PSD/FP time-matched experiment.

Recipe	PSD	LSD	ESD	PH	FP
A6000-hours / 1k updates	0.23	0.39	0.26	0.23	0.30

The long masked PSD experiment records 184.5 Ada GPU-hours. The LLaDA-8B adapter study records 196.33 GPU-hours across its jobs. Hardware and workloads differ, so these study-level measurements are reported individually. The main paper's inference budget is the number of network evaluations. Converting NFE to end-to-end latency would require measurements with a specified deployment implementation and hardware.

I.2 Precision, accumulation, and numerical checks

Small-model training uses fp32 with TF32 matrix multiplication enabled, and recipe forwards are pinned to fp32. The core generation benchmark uses float64 state arithmetic and fp32 network forwards. The masked models compute SUBS readouts in fp32. LLaDA-8B uses a bf16 backbone with fp32 LoRA adapters and time embedders, with deterministic math-SDPA attention on gradient paths and chunked target computation.

The core LM1B batch is 64 sequences with two accumulation steps. Continuous OWT uses eight sequences with 16 accumulation steps. The long masked run uses four with 32, and the 8B com-

parisons use four with eight or 16, yielding the batches in Table 9. Accumulation changes effective batch size without introducing additional training replications.

Implementation checks cover jump coefficients, target construction, gradient stopping, and the conversion between conditioning and transport times. Metric checks cover entropy, repeated four-grams, perplexity aggregation, and distinct-token statistics. Analytic density and controlled-system checks supplement these tests. Such checks assess numerical implementation, while the reported language-generation comparisons provide the empirical evidence.

I.3 Release scope

Training and evaluation code, configured recipes, and figure-generation scripts will accompany the submission. We plan to open-source these resources publicly. Reproducing a comparison requires its teacher and tokenizer, training configuration, checkpoint treatment, sampler, output-bank size, and metric settings. The interval-sampling comparison also requires the weights and within-component rule in Appendix F. Results computed per run should be aggregated at that level, including the separate MAUVE banks. Model and dataset access remain subject to their respective upstream distributions.

I.4 Use of large language models

The AI use statement following the conclusion describes assistance with literature review, manuscript drafting and revision, and analysis and figure code. The AI-only output audit is a separate measurement procedure, with its judge, sample count, and scope specified in Appendix C.4.

J Extended related work

Diffusion language modeling. Discrete diffusion permits structured token corruption and reversal, as developed in D3PM (Austin et al., 2021). SEDD learns ratios of discrete probabilities, while MDLM simplifies masked diffusion training (Lou et al., 2024; Sahoo et al., 2024). Analog Bits instead models discrete data through continuous representations and self-conditioning (Chen et al., 2023). Diffusion Duality connects masked diffusion and autoregressive modeling (Sahoo et al., 2025), and LLaDA and Dream demonstrate diffusion language modeling at larger scales (Nie et al., 2025; Ye et al., 2025). These approaches establish the broader generation landscape. DistillScope concentrates on distilling continuous flow maps with a categorical terminal readout, where training intervals and the number of composed evaluations can be varied within a common implementation.

Learning maps for few-step generation. Progressive distillation compresses a teacher's sampling procedure over successive rounds (Salimans & Ho, 2022). Consistency Models learn mappings toward a shared trajectory endpoint, with subsequent work improving continuous-time training and extending consistency to trajectory mappings (Song et al., 2023; Lu & Song, 2025; Kim et al., 2024). Flow Map Matching formalizes maps between times (Boffi et al., 2025); Shortcut Models, MeanFlow, and Align Your Flow develop related ways to learn finite-time generative transitions (Frans et al., 2025; Geng et al., 2025; Sabour et al., 2025). Flow Map Language Models bring this formulation to language and already compare semigroup, Lagrangian, and Eulerian objectives (Lee et al., 2026). Self-conditioned Flow Map Language Models develop fixed-point flows (Yoo et al., 2026a). Our study takes these objectives as its starting point and examines the stability of their comparisons under changes in metric, training-cost accounting, and inference budget. The degeneration profiles characterize our stated budget and checkpoint settings, separately from the operating points reported by Lee et al. (2026). The PH and FP arms are local implementations motivated by these lineages, with the scope specified in Table 1.

Distillation in discrete text spaces. Self-Distillation Through Time compresses masked diffusion sampling over multiple distillation rounds (Deschenaux & Gulcehre, 2025). Di4C addresses dimensional correlations, and DiDi uses divergence-based distillation for fast generation (Hayakawa et al., 2025; Zheng et al., 2026). FS-DFM combines trajectory supervision with a few-step discrete-flow formulation (Monsefi et al., 2026a); TS-DFM shapes the teacher trajectories, while OPTD trains on student-induced states with adaptive compression (Monsefi et al., 2026b; Lu et al., 2026). Infinite

Mask Diffusion and Consistent Diffusion Language Models further develop few-step distillation and consistency in discrete settings (Yoo et al., 2026b; Amin et al., 2026). Differences in state spaces, teacher trajectories, and decoding rules make these methods complementary comparators. A shared flow-map recipe study answers a narrower question than comparing all such systems by their reported best scores.

Autoregressive acceleration. Sequence-level knowledge distillation trains a student on teacher-generated sequences and can reduce decoding cost, including reliance on beam search (Kim & Rush, 2016). Speculative decoding uses a draft model and target-model verification to accelerate autoregressive sampling while preserving the target distribution (Leviathan et al., 2023). These address different computational units from whole-sequence flow-map evaluations. Equal NFE alone does not equalize their token work, model sizes, caching, or proposal and verification cost. A cross-paradigm deployment comparison therefore requires a common workload and measured hardware-specific latency at specified quality. Our comparisons concern distillation recipes within the stated generative frameworks and do not establish superiority over autoregressive acceleration.

Sampling and training allocation. DDIM and DPM-Solver accelerate generation by changing inference trajectories or their numerical integration (Song et al., 2021; Lu et al., 2022). EDM makes training, preconditioning, and sampling choices explicit within a common design space (Karras et al., 2022). Training-time allocation is a separate axis: Improved DDPM uses importance sampling with correction to reduce the variance of a variational objective, whereas Min-SNR reweights timestep losses (Nichol & Dhariwal, 2021; Hang et al., 2023). Our fixed interval mixture reweights losses over pairs of times. Its exposure-matched control accounts for interval length and late-time coverage, providing a baseline for claims about finer allocation.

Degeneration, readout, and evaluation. Neural text degeneration and unlikelihood training establish repetition as a problem that likelihood alone does not resolve (Holtzman et al., 2020; Welleck et al., 2020). Sparse sequence models study alternatives to dense output probabilities (Peters et al., 2019), while precision-sensitive categorical sampling can change diffusion generation behavior (Zheng et al., 2025). Work on sharp categorical readouts examines the difficulty of few-step text generation (Wang, 2026). Our margin condition states when a continuous perturbation preserves argmax; it does not identify the cause of an empirical recipe difference. MAUVE evaluates discrepancies between text distributions (Pillutla et al., 2021), and both general generative-model analysis and recent diffusion-language evaluation studies explain why one metric cannot capture every aspect of sample quality (Theis et al., 2016; Pynadath et al., 2026). Generative Frontiers motivates evaluating generative perplexity together with distributional entropy. Our blockwise token-entropy screen serves a narrower diagnostic purpose and is not an estimator of full sequence entropy. The additional evidence here is the replicated PSD–LSD ordering reversal between perplexity and MAUVE, together with separate denoising and generation orderings under the stated benchmark conditions.

Controlled empirical benchmarks. DeepOBS packages common workloads for optimizer comparisons (Schneider et al., 2019). DomainBed demonstrates the importance of model selection and consistent experimental protocols in domain generalization (Gulrajani & Lopez-Paz, 2021). The AlgoPerf competition analyzes training algorithms through time-to-result and explicit tuning protocols (Kasimbeg et al., 2025). Following this experimental tradition, DISTILLSCOPE makes the comparison conditions part of the result: training updates, measured training cost, inference evaluations, checkpoint treatment, and the evaluation metric each determine the question being answered.