INSIDE THE SCALING OF RIEMANNIAN FLOW LANGUAGE MODELS: GEOMETRY, GENERATION, AND CONDITIONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Riemannian flow language models have not been studied at practical scale. We train SwaFlow, a suite of hyperspherical flow language models with up to 3B parameters on 2T tokens. As training progresses, token embeddings become less isotropic, shrinking the range of noise levels from which local geometry alone recovers a token. Yet the models recover tokens from increasingly higher noise levels using context, and the gap between these regimes grows with scale. Consistent with this trend, held-out loss and unconditional generation improve with model size and training budget, but conditioning on clean context does not: multiple-choice accuracy stays at chance at every size and budget. We trace this failure to the training distribution rather than the geometry: standard training assigns the same noise level to all positions and never exposes the model to mixed clean–corrupted contexts. Prior work under a different geometry has shown that clean-prefix corruption improves conditional generation; for Riemannian flow language models, scaling alone does not remove this dependence on partially clean training examples. Introducing them only during the final 100B training tokens lifts conditioning off chance at every size, with gains that grow with model size and training budget and carry over to task finetuning on GSM8K. We will release all intermediate checkpoints, optimizer states, and exact data order upon acceptance, enabling these scaling trajectories to be reproduced, branched, and re-evaluated.

1 INTRODUCTION

Discrete diffusion language models generate by iteratively restoring masked or resampled tokens (Sahoo et al., 2024; Lou et al., 2024; Shi et al., 2024; Sahoo et al., 2025) and have reached 7–8B parameters (Nie et al., 2025; Ye et al., 2025). Riemannian flow language models instead corrupt text continuously on a manifold, either the statistical manifold of categorical distributions (Davis et al., 2024; Cheng et al., 2024; Jo & Hwang, 2025) or a hypersphere of learned token embeddings (Deschenaux & Gulcehre, 2026; Chemseddine et al., 2026). We study the hyperspherical kind, which moves each clean embedding along the sphere toward random noise by a continuous ratio α, with $\alpha = 1$ clean. Unlike a mask, α grades how much evidence each position retains about its own token, and the geometry of the learned embeddings can be measured.

Whether a family of language models is worth pursuing depends on how it scales, which is well characterised for autoregressive and masked diffusion models (Kaplan et al., 2020; Hoffmann et al., 2022; Nie et al., 2024). Riemannian flow language models have stayed near 0.1B parameters (Table 3), so how they behave at scale is unknown. We train **SwaFlow**, hyperspherical flow language models of 150M, 500M, 1B and 3B parameters on 2T tokens of Dolma 3 Mix (Team Olmo, 2025), with autoregressive twins at the three smaller sizes that differ only in the objective.

Because α is continuous, held-out loss is a curve $L(\alpha)$, and a scaling study can ask not only how far the loss falls but where along α. We read the curve through two points (§2). The **solvable frontier** α_{sol} is the noise level at which the loss falls below 0.1 nats; the lower it lies, the heavier the noise the model can undo. The **geometric threshold** α_{geo} is computed in closed form from the embedding table: above it, context-free nearest-neighbour decoding recovers the token. When

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.

$\alpha_{\text{sol}} < \alpha_{\text{geo}}$, the interval between them is the **information band**, where the model solves noise levels that per-position geometry cannot, and so must be using context.

Training and scale move the two points in opposite directions. The frontier descends, while α_{geo} more than doubles as the embeddings lose isotropy (Mu & Viswanath, 2018; Ethayarajh, 2019); within the first fifth of training, every training noise level lies below α_{geo}. The band widens with model size N and training tokens D: at a matched $D = 2$T, $\alpha_{\text{geo}}/\alpha_{\text{sol}}$ grows from 2.4 at 150M to 3.5 at 3B (§3). Scale thus improves recovery from context as embeddings grow less separable.

Yet scale does not teach these models to use context that is clean. Held-out cross entropy and unconditional generation improve with N and D (§3.1), but no measure that holds part of the sequence clean does (§4): multiple-choice accuracy stays at chance at every size and budget, far below that of the autoregressive twins, and given a clean prefix the model predicts the continuation worse than the corpus unigram distribution does. The released checkpoints of two earlier hyperspherical flow models are also at chance (Table 2). The cause is the corruption schedule: drawing one α for all positions never produces a sequence that is clean in one part and noised in another, whereas next-token prediction trains on nothing else.

Closing this gap removes the deficit. In **clean-prefix mid-training**, the last 100B pretraining tokens keep a random prefix of half the sequences clean, with all else unchanged and without finetuning. At all four sizes, multiple-choice accuracy leaves chance and a clean prefix now helps predict what follows, for about 1% of held-out cross entropy. The gain grows with N and, at 1B, with the pretraining budget at which mid-training is applied, while a matched uniform-α control stays at chance (§5). Clean prefixes are known from Euclidean diffusion and flow models (Dieleman et al., 2022; Davis et al., 2026) and from task training of a hyperspherical flow (Deschenaux & Gulcehre, 2026). To our knowledge, we are the first to show what their absence costs a Riemannian flow language model, to bring them into its general-corpus pretraining, and to measure their effect beyond task accuracy: a clean prompt, which hurt the prediction of a response, now helps it, and the first finetuning gradient shrinks to within a factor of two of the autoregressive twins' (§5.3).

The mid-trained models also transfer to task finetuning. Finetuned on TinyGSM (Liu et al., 2023), every pretrained base outperforms a randomly initialised model on GSM8K (Cobbe et al., 2021), and accuracy improves with both N and D (§6). Our final model is the 3B model mid-trained from 3T to 4T tokens, together with a weight average that improves it further (§6.2).

In summary, our contributions are: (i) SwaFlow, to be released with every intermediate training state, its optimizer state and the exact data order (Appendix A); (ii) an α-resolved reading of held-out loss, with a geometric threshold freed from the isotropy assumption of Deschenaux & Gulcehre (2026); (iii) the finding that scaling widens the information band yet leaves conditioning on clean context at chance, traced to the corruption schedule; and (iv) clean-prefix mid-training, which removes this deficit at every size, with gains that grow with scale and transfer to task finetuning.

2 SwaFlow

We build SwaFlow, a fully traceable suite of Riemannian flow language models for scaling studies (Table 1): four sizes from 150M to 3B parameters (N), trained on $D = 2$T tokens beside autoregressive twins at three sizes, with full training states every 20B tokens and the exact data order, all to be released upon acceptance (Appendix A).

Table 1: Model shapes, shared with the AR twins. N includes the untied embeddings, so size names are nominal.

SwaFlow	Dim	Layers	Heads	FFN	N	N (AR)	$\alpha_{\max}$
150M	768	12	12	2048	0.175B	0.161B	0.150
500M	1280	24	10	3584	0.686B	0.614B	0.125
1B	2048	24	16	5632	1.588B	1.435B	0.105
3B	3072	28	24	8192	3.741B	—	0.100

SwaFlow is a hyperspherical flow language model (Deschenaux & Gulcehre, 2026): tokens are learned unit embeddings on S^{d-1}, and each clean embedding e_w is corrupted along the great circle toward its own noise z, uniform on the sphere,

$$x_\alpha = \frac{\sin\big((1-\alpha)\Theta\big)\, z + \sin(\alpha\Theta)\, e_w}{\sin\Theta}, \qquad \Theta = \arccos\langle z, e_w\rangle, \tag{1}$$

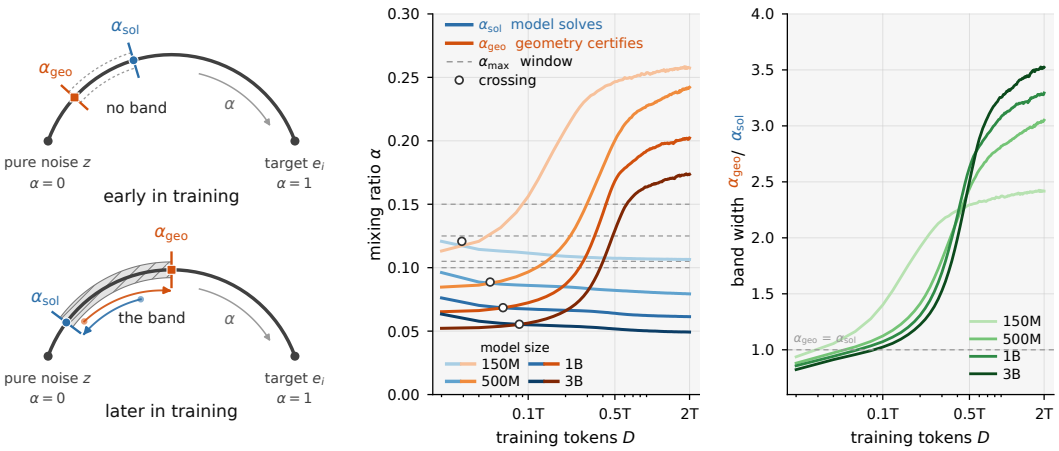


Figure 1: **The information band.** *(Left)* Schematic of the two boundaries before and after they cross. *(Center)* Measured α_{sol} and α_{geo} against training tokens D. *(Right)* Band width $\alpha_{\mathrm{geo}}/\alpha_{\mathrm{sol}}$.

with $\alpha = 1$ clean and $\alpha = 0$ pure noise. One α per sequence is drawn uniformly from a static window $[0, \alpha_{\max}]$, set per model size where an untruncated pilot's loss falls below 0.1 nats (Table 1; Appendix B.3), and the loss is cross entropy against the clean tokens, unweighted in α.

Two points on the held-out cross entropy $L(\alpha)$, and the interval between them, organise the paper.

Solvable frontier α_{sol}. Where $L(\alpha)$ falls below 0.1 nats. Lower is better: the model then recovers tokens from less per-position evidence.

Geometric threshold α_{geo}. As $\langle z, e_w \rangle = O(d^{-1/2})$, $x_\alpha \approx \cos\theta\, z + \sin\theta\, e_w$ with $\theta = \alpha\pi/2$: the target scores $\approx \sin\theta$ and any competitor at most $\cos\theta\, m + \sin\theta\, c$, where m and c are the largest cosines from z to any embedding and from e_w to any other. Context-free nearest-neighbour decoding thus succeeds once $\tan\theta > m/(1-c)$, and averaging m over noise draws and c over tokens (c_{mean}) gives a typical-case threshold set by the embedding table alone:

$$\alpha_{\mathrm{geo}} \;=\; \tfrac{2}{\pi} \arctan\big(m \,/\, (1 - c_{\mathrm{mean}})\big). \tag{2}$$

It is sufficient, not necessary: below it nothing is certified, yet a model using context can still succeed. A lower α_{geo} means embeddings separable from per-position evidence under heavier noise; anisotropy, crowding neighbours together, raises it. Deschenaux & Gulcehre (2026) derive it analytically assuming isotropic embeddings, true only near initialisation, and Eq. 2 recovers theirs at $c_{\mathrm{mean}} = 0$ up to small corrections (Appendix C). Reading m and c_{mean} off the trained table frees it from that idealisation, so we can track it as scaled training makes the embeddings anisotropic (§3.2).

Information band. When $\alpha_{\mathrm{sol}} < \alpha_{\mathrm{geo}}$, the model solves an interval the geometry does not certify (Fig. 1, left), and the width $\alpha_{\mathrm{geo}}/\alpha_{\mathrm{sol}}$ is how far its frontier reaches below the certificate. With one edge from the model's loss and the other from its embedding table, the width expresses geometrically how the model changes with training and scale, and can help interpret gains or losses in performance (§3.2).

Setup. The network is a bidirectional DiT (Peebles & Xie, 2023), conditioned on the noise level through adaLN, with a Llama-style block (Grattafiori et al., 2024) (RMSNorm, SwiGLU, RoPE, per-head QK-norm). The corpus is Dolma 3 Mix (Team Olmo, 2025), tokenised with the SmolLM2 tokenizer ($V = 49{,}152$; Allal et al., 2025) into 2048-token sequences; all sizes share one batch and data order, so a step denotes the same tokens at every size. For the analysis rather than final performance, the learning rate stays constant after warmup: checkpoints then differ only in data seen, and any can serve as a branch point (§5). The autoregressive twins share all of this and differ only where the objective requires: causal attention, clean inputs, no noise conditioning, next-token targets. To make distributed training possible, we implement SwaFlow on torchtitan (Liang et al., 2025) and train it with FSDP2 (details in Appendix B).

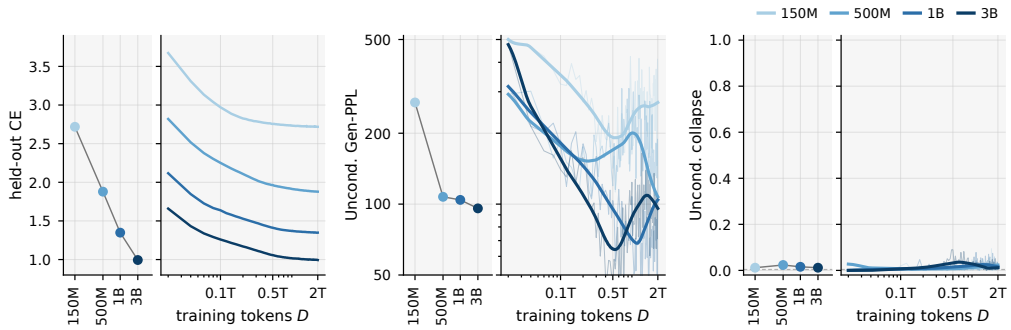


Figure 2: **Unconditional pretraining scales.** Held-out cross entropy, unconditional Gen-PPL and collapse rate, against N at $D = 2\text{T}$ (narrow left axis) and against D (wide right axis).

3 The Information Band

3.1 Aggregate measures scale

Uniform corruption scales as a pretraining objective should (Fig. 2). Held-out cross entropy, read on one α grid shared by all sizes, falls with N and D, and so does the generative perplexity (Gen-PPL) of unconditional samples, scored by an external autoregressive model as is standard (Sahoo et al., 2024; Arriola et al., 2025; Nie et al., 2025). The gain is not bought with degenerate text: the collapse rate, the fraction of samples overrun by repeated 4-grams, stays below 10% at all 400 checkpoints (Appendix E.1). The internal quantities below are thus read off models that train normally.

3.2 Geometry retreats, models advance

The two boundaries of §2 move in opposite directions (Fig. 1, centre; Table 8). Over training the model's frontier α_{sol} descends, more for larger models, while α_{geo} more than doubles. The rise comes from the embeddings, not the noise: m in Eq. 2 even drifts down, while c_{mean} climbs as the embeddings lose isotropy (Appendix D.1). Embedding anisotropy is well documented (Mu & Viswanath, 2018; Gao et al., 2019; Ethayarajh, 2019); on the sphere its cost is the rise of α_{geo}.

Early in training there is no band. At $D = 20\text{B}$, $\alpha_{\text{sol}} > \alpha_{\text{geo}}$ at every size: the geometry certifies noise levels the model cannot yet solve. The two boundaries cross between 29B and 85B tokens, later for larger models (open circles in Fig. 1), and within the first fifth of a 2T budget α_{geo} also passes the window edge $\alpha_{\max}$, leaving the whole training window below it (Appendix D.2).

The band widens with scale. In Fig. 1 (right), larger models and longer training both carry the frontier further below what per-position geometry certifies; Table 8 gives the values. Read with §3.1, the models improve steadily although the geometry certifies none of their training draws: what scale improves is recovery from context, even as the embeddings grow less separable.

3.3 Against next-token prediction

The flow and its autoregressive twin solve different tasks, so we compare difficulty rather than loss. The *equivalence point* α_{eq} is the α at which the flow's held-out cross entropy equals the twin's at the same step; below it, the flow's per-position problem is harder than next-token prediction. It falls with N: larger flow models match autoregressive difficulty from less per-position evidence.

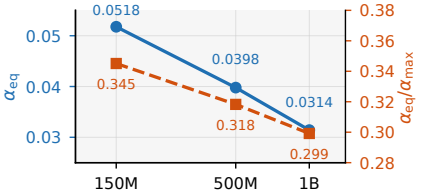


Figure 3: **Equivalence point** α_{eq} (left) and $\alpha_{\text{eq}}/\alpha_{\max}$ (right) at $D = 2\text{T}$.

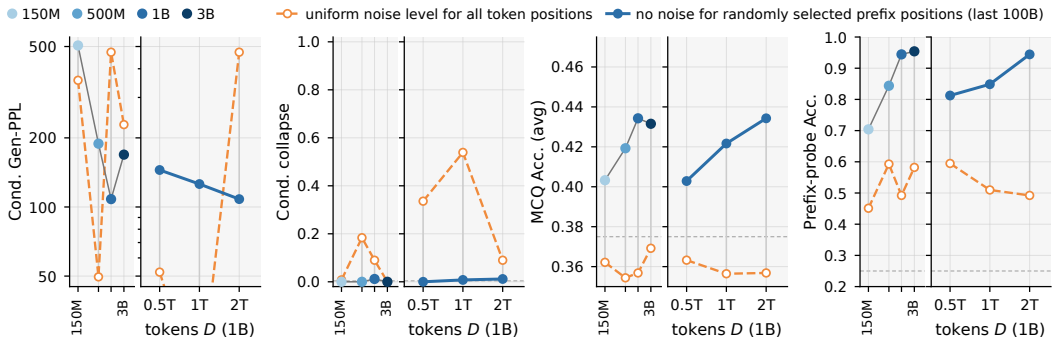


Figure 4: **Conditioned quantities, uniform α (open) vs. clean-prefix mid-training (filled).** Each pair: N at $D = 2$T, then 1B along D. Dashed: chance or real text; off-axis point: 14.9.

4 Unconditional Generation Scales, Conditioning on Clean Context Does Not

The quantities of §3.1 hold no position clean, as in training, and improve with N and D. Those that hold part of the sequence clean and require the model to use it do not (open markers in Fig. 4; values in Table 10). Multiple-choice accuracy over six tasks, scored with the prompt clean (Appendix E.2), stays at chance at all four sizes and every budget measured, whereas the autoregressive twins reach 0.48–0.60 on the same data at the same step. Given a clean 64-token prefix, the cross entropy of the true continuation exceeds the corpus unigram entropy of 7.53 nats at all four sizes: the prefix leaves the model worse off than a context-free unigram predictor. In the **prefix probe**, which asks the model to pick a document's own 32-token continuation from four candidates (Appendix E.3), the flow models reach 0.45–0.59 against 0.80–0.87 for the twins, with no ordering in N and no gain along D. Generation from a clean prefix collapses into repetition on about a third of samples midway through training at the three larger sizes, while unconditional samples from the same checkpoints stay below 10% (Appendix E.1). Because conditional Gen-PPL falls as samples collapse, we report it only beside the collapse rate.

All four measurements present a sequence that is clean in one part and corrupted in the rest. Uniform α never produces such a sequence: one α is drawn per sequence and applied to every position, so a partially clean sequence has probability zero in pretraining. Next-token prediction, in contrast, trains on nothing else, since every prediction is made from a clean prefix. As neither N nor D moves the deficit, we attribute it to this gap in the training distribution rather than to capacity or budget. The attribution can be tested with a single difference: a schedule that leaves part of each sequence clean, with everything else fixed, should remove the deficit (§5).

5 Clean-Prefix Mid-Training Removes the Conditioning Deficit

Section 4 attributes the deficit to the corruption schedule: uniform α never presents a sequence that is clean in one part and corrupted in the rest. We test this attribution with a single difference, holding a prefix of half the sequences clean late in pretraining and keeping everything else fixed. Clean prefixes are a known ingredient, used to make a model promptable or its samples faithful (Dieleman et al., 2022; Davis et al., 2026; Deschenaux & Gulcehre, 2026); here they serve as a controlled test of the diagnosis, and we follow the result along N and D.

5.1 Setup

Each sequence draws a split point k. With probability $1/2$, $k = 0$ and the sequence is corrupted as under uniform α; otherwise $k \sim \mathcal{U}\{0, \ldots, 1023\}$, the first k positions stay clean and unscored, and the rest are corrupted at the sequence's own α. We call training under this schedule **clean-prefix mid-training**. Its settings were fixed in advance and never swept, since the aim is to test the diagnosis rather than to tune a recipe (Appendix F.1).

We apply it as a branch late in training, so that each model's own run is the control. Each model leaves its uniform run at 1.9T and trains to 2T under the new schedule, while the **uniform arm** continues that run over the same 100B tokens, in the same order and at the same learning rate. The two arms thus differ only in which positions are held clean (Appendix F.1), and neither sees instruction data. For the budget axis, the 1B model is branched in the same way to land at 0.5 and 1.0T.

5.2 Conditioning appears without finetuning

Mid-training removes each failure of §4 at all four sizes (Fig. 4). Multiple-choice accuracy leaves chance, reaching 0.40–0.43 against 0.35–0.37 for the uniform arm, and prefix-probe accuracy rises from 0.45–0.59 to 0.70–0.95, each by at least 6σ at every size. A clean prefix now helps predict what follows: the cross entropy of the true continuation drops below the unigram entropy, to 6.9–7.2 nats (Table 10). Generation from a clean prefix collapses on at most 1.2% of draws at every size and budget, where the uniform arm reaches 54% (1B at 1T). The price is 0.7–1.2% of held-out cross entropy, an order of magnitude below the 7.7–11.3% that one epoch of instruction finetuning costs the same uniform checkpoints, and unconditional generation is unchanged (Appendix F.2).

Along D. At 1B the gain grows with the budget at which mid-training is applied (Fig. 4, right panel of each pair). Branches landing at 0.5, 1.0 and 2.0T reach 0.403, 0.422 and 0.434 on multiple choice and 0.81, 0.85 and 0.94 on the probe, while the uniform arm stays at chance on multiple choice and loses probe accuracy. As every branch spends the same 100B tokens, the gain reflects what the preceding uniform pretraining provides, which grows with D even as the uniform arm's own scores stay at chance or decline. The three points share one size and do not separate N from D.

Along N. The gain grows with size: from 150M to 3B, multiple-choice accuracy rises from 0.403 to 0.432 and prefix-probe accuracy from 0.70 to 0.95, with boolq supplying most of the multiple-choice gain (Appendix F.2). On the probe the mid-trained flow trails its autoregressive twin at 150M, ties at 500M and leads at 1B, while on multiple choice the twins stay ahead at every size.

5.3 What mid-training changes

The change shows directly in the loss. We score the responses of a held-out instruction set, Dolci-Instruct-SFT (Team Olmo, 2025), with the prompt either clean or corrupted. Under uniform α the clean prompt hurts: response cross entropy is 1.6–2.6 times higher than with the prompt corrupted as in training. After mid-training the clean prompt helps at every size, while the cross entropy under a corrupted prompt rises by only 0.01–0.03 nats (Table 10): the use of clean context is added, not traded against the task the model was trained on. At 1B the inversion is already in place at the first branch point, 0.5T (Appendix F.3), so what grows with D is the downstream ability it supports rather than the inversion itself.

How much is left for finetuning shows in its first gradient on this instruction set (Fig. 5). All bases, flow and autoregressive, are finetuned with one recipe, so the differences lie in the base alone. From a uniform-α base the norm is 9.6–38.9, an order of magnitude above that of the autoregressive twins (1.0–1.1). Mid-training lowers it 5–25-fold at every size, to 1.0–1.9, within a factor of two of the twins: a mid-trained flow model enters finetuning about where an autoregressive one does. Consistently, finetuning on top of mid-training adds less than mid-training itself, +0.005 to +0.015 in multiple-choice accuracy against +0.04 to +0.08 (Appendix F.2); the post-training of §6 therefore starts from mid-trained bases.

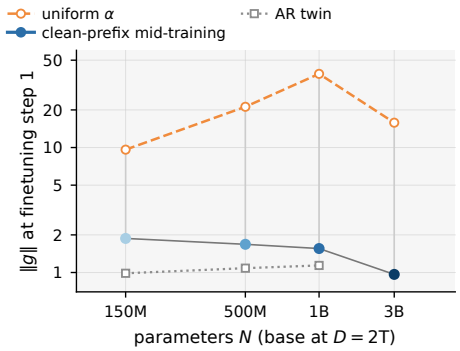


Figure 5: **Gradient norm at the first finetuning step** (bases at $D = 2$T).

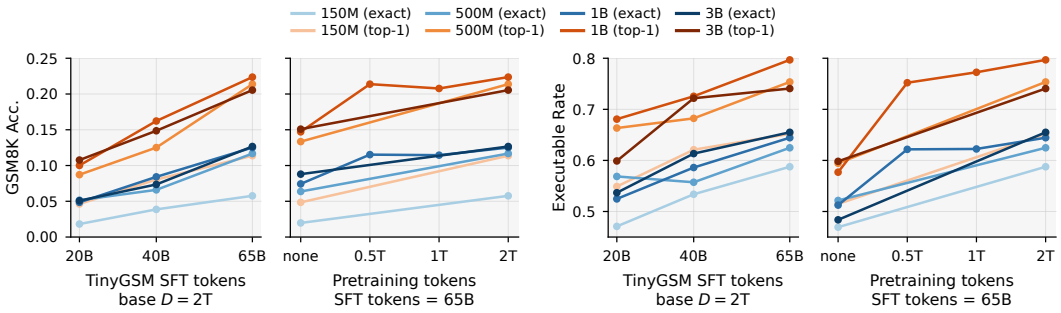


Figure 6: **Task-specific finetuning on TinyGSM.** GSM8K accuracy (left pair) and executable rate (right pair). In each pair, left: along finetuning tokens from 2T bases; right: along the base's pretraining tokens at 65B ("none": random initialisation; only 1B has intermediate bases). Blue: exact sampler; orange: top-1; darker: larger. Values: Table 11.

6 Transfer to Finetuning and the Final Model

6.1 Pretraining transfers to task finetuning

With conditioning in place (§5), we take the mid-trained bases as given and ask whether what a base gained from pretraining carries over to task-specific finetuning. Each base is finetuned on TinyGSM (Liu et al., 2023) for up to 65B tokens, beside a randomly initialised model of the same size ("none"), and scored on GSM8K (Cobbe et al., 2021) 0-shot by executing the generated program, with two samplers: *exact*, the reference implementation's default, and *top-1* (Appendix G.2).

Pretraining transfers (Fig. 6). At every size a pretrained base finishes ahead of a random start, by 3.2–6.2σ, and accuracy improves along both N and D: with the exact sampler it rises from 5.8% at 150M to 12.7% at 3B, and at 1B from 7.4% without pretraining to 12.5% from the 2T base, with the executable rate rising alongside (Appendix G.3). Finetuning itself is far from saturated: accuracy rises at every step from 20 to 40 to 65B in every condition, so these are levels at a finite budget, not ceilings.

6.2 The final model and where it stands

Our final model is the 3B model branched from its uniform run at 3T and mid-trained to 4T tokens with the recipe of §5. Beside its raw weights we will release an exponential moving average reconstructed from the saved states (Appendix F.4). Weight averaging of this kind was proposed to make pretraining horizon-free, without a learning-rate decay phase (Meterez et al., 2026), and no run in this paper decays its learning rate; the average lowers held-out cross entropy from 1.006 to 0.968 and raises multiple-choice accuracy from 0.418 to 0.434 (+2.3σ), a lead that survives instruction finetuning (+1.6σ). It thus carries over to a flow language model.

Table 2 places the base models on multiple choice, task by task. Of the Riemannian flow language models we can score, only the mid-trained ones are off chance: the released checkpoints of two hyperspherical flow models (Deschenaux & Gulcehre, 2026; Chemseddine et al., 2026) sit at chance (0.378 and 0.380), as do our uniform-α models, so size alone does not lift a model off chance; mid-training does. Next-token prediction remains well ahead: the autoregressive twins reach 0.481–0.604 against at most 0.434, and lead on five of the six tasks.

7 Related Work

Riemannian flows over discrete data. One line places categorical data on the statistical manifold of categorical distributions and flows along its geodesics: under the Fisher–Rao metric, which maps the manifold to the positive orthant of a sphere with one dimension per category (Davis et al., 2024; Cheng et al., 2024), or under a family of α-geometries (Cheng et al., 2025); RDLM builds a continuous diffusion on the same orthant (Jo & Hwang, 2025). Their language experiments are

Table 2: Base models on multiple choice, task by task (first row: chance). SwaFlow rows are the base models this paper studies, clean-prefix mid-trained unless marked *Uniform*; the 4T rows are the final model, raw and averaged (EMA). S-FLM (Deschenaux & Gulcehre, 2026) and SFCD (Chemseddine et al., 2026) are their authors' released checkpoints, scored under the same rule (Appendix E.2). Standard errors are .004–.022 per task and .005 for the average. †Derived; §non-embedding.

Model	N	D	Corpus	BoolQ	OBQA	PIQA	HSwag	ARC-E	WinoG	Avg.
Chance				*.500*	*.250*	*.500*	*.250*	*.250*	*.500*	*.375*
Riemannian flow language models										
S-FLM	170M†	0.52T†	OWT	.482	.252	.492	.263	.269	.507	.378 ± .005
SFCD	93M§	0.52T†	OWT	.391	.226	.538	.264	.343	.518	.380 ± .005
SwaFlow (Uniform)										
• 150M	175M	2T	Dolma 3	.377	.214	.513	.251	.332	.486	.362 ± .005
• 500M	686M	2T	Dolma 3	.379	.196	.504	.253	.314	.481	.354 ± .005
• 1B	1.59B	2T	Dolma 3	.386	.220	.526	.258	.266	.485	.357 ± .005
• 3B	3.74B	2T	Dolma 3	.379	.252	.520	.247	.300	.517	.369 ± .005
SwaFlow										
• 150M	175M	2T	Dolma 3	.621	.218	.521	.263	.303	.493	.403 ± .005
• 500M	686M	2T	Dolma 3	.611	.256	.541	.282	.356	.470	.419 ± .005
• 1B	1.59B	2T	Dolma 3	.611	.232	.573	.286	.385	.518	.434 ± .005
• 3B	3.74B	2T	Dolma 3	.617	.226	.552	.294	.409	.492	.432 ± .005
• 3B	3.74B	4T	Dolma 3	.621	.226	.528	.286	.344	.501	.418 ± .005
• 3B, EMA	3.74B	4T	Dolma 3	.622	.222	.526	.308	.426	.501	.434 ± .005
Autoregressive language models										
AR-150M	161M	2T	Dolma 3	.572	.284	.640	.350	.507	.533	.481 ± .005
AR-500M	614M	2T	Dolma 3	.559	.354	.713	.499	.615	.565	.551 ± .005
AR-1B	1.44B	2T	Dolma 3	.659	.378	.740	.583	.675	.586	.604 ± .005

on character-level Text8 or LM1B, at up to 110M parameters. A second line places learned token embeddings on $\mathbb{S}^{d-1}$: S-FLM (Deschenaux & Gulcehre, 2026) interpolates along geodesics, and Chemseddine et al. (2026) study geodesic and von Mises–Fisher paths, both with 12-layer, 768-dimensional networks. Only this line has an embedding table, and so only it has the c_{mean} and α_{geo} of §3; on the statistical manifold the categories are fixed vertices. We keep S-FLM's objective and truncated schedule and change the scale (Table 3), so the α axis this line defines is what §3–§5 measure. We claim no advantage for the manifold: in an LM1B ablation, Lee et al. (2026) find their Euclidean interpolant ahead of Riemannian diffusion.

Euclidean continuous diffusion and flows. Continuous language models outside this family corrupt in Euclidean space: learned embeddings under Gaussian noise, left free or normalised to a fixed length (Li et al., 2022; Dieleman et al., 2022; Gulrajani & Hashimoto, 2023; Chen et al., 2026; Yang et al., 2026), one-hot vectors (Lee et al., 2026; Davis et al., 2026), and encoder or autoencoder latents (Hu et al., 2026; Guo et al., 2026). They have been trained at 1.3B (Gulrajani & Hashimoto, 2023), 1.7B (Davis et al., 2026) and, as latent flow matching generated block by block, 2.3B parameters (Guo et al., 2026). Davis et al. (2026) train their categorical flow map at full budget at that one size and track its generation quality every 20k training steps; our analysis instead decomposes held-out loss along α across four sizes on one data order, follows conditioning measures along D, and reads states we will release. Two Euclidean observations bear on ours. Chen et al. (2026) find the loss nearly zero for $t \in [0.2, 1]$, with $t = 1$ clean, so that a schedule uniform in t spends most of training where nothing is learned — the observation behind a truncated window — and answer it with a learned Gumbel-shaped noise schedule rather than a truncation. And the sweep of Davis et al. (2026) shows that a non-zero clean-prefix probability (Dieleman et al., 2022) is needed for sample fidelity; §5 supplies the other half, identifying the failure it repairs and isolating the repair against a matched control. Their multiple-choice accuracy, above chance on some tasks (ARC-E 47.1 against 24.7) and below it on others, is scored by a likelihood bound at another size on a model trained with clean prefixes, so it does not conflict with §4.

Discrete diffusion. Masked diffusion has scaled further, with fitted laws (Nie et al., 2024) and 7–8B models (Nie et al., 2025; Ye et al., 2025), under a different objective. A mask over discrete states carries no embedding geometry, so the quantities §3 is built from do not exist there.

Table 3: Artifacts of Riemannian flow language models, against what a scaling analysis needs; Table 4 adds Euclidean flow and diffusion models. ● present, ◐ in final checkpoints only, – absent or not found in the paper, its repository or Hugging Face. †Derived; §non-embedding.

	Corpus	Max tokens	Data order	Smallest model	Largest model	Sizes	Sizes with matched AR	Intermediate checkpoints	Final checkpoints	Optimizer states
Riemannian: learned token embeddings on a hypersphere										
S-FLM (Deschenaux & Gulcehre, 2026)	OWT, TinyGSM	0.52T†	–	170M†	170M†	1	1	–	●	◐
SFCD (Chemseddine et al., 2026)	LM1B, OWT, TinyGSM	0.52T†	–	93M§	93M§	1	0	–	●	◐
SwaFlow (ours)	Dolma 3	**4T**	●	175M	**3.7B**	**4**	**3**	● (per 20B)	●	●
Riemannian: the statistical manifold of categorical distributions										
RDLM (Jo & Hwang, 2025)	Text8, LM1B	0.13T†	–	92M	110M	2	0	–	●	◐
Fisher-Flow (Davis et al., 2024)	LM1B	0.33T	–	110M	110M	1	0	–	–	–
SFM (Cheng et al., 2024)	Text8	0.39–1.57T†	–	92M	92M	1	0	–	–	–

Scaling laws. Scaling studies fit one held-out loss against N and D (Kaplan et al., 2020; Hoffmann et al., 2022), and continuous diffusion over text has its own: Gulrajani & Hashimoto (2023) fit compute-optimal laws and place Plaid at $64\times$ the compute of an autoregressive model, and RePlaid (Yang et al., 2026), a Euclidean diffusion model trained on a variational bound rather than a flow, fits IsoFLOP laws under one protocol with autoregressive and masked-diffusion baselines, reporting the first such law for continuous diffusion that rivals discrete diffusion and a $20\times$ gap to autoregression with self-conditioning; its loss per noise level is read at one size. §3 instead decomposes held-out loss along α across N and D, where it can fall at every α while the frontier moves toward heavier noise. Autoregressive training fixes the corruption level and has no such axis.

Post-training and conditioning. Conditioning on a clean instruction is usually bought afterwards by instruction tuning (Wei et al., 2022; Ouyang et al., 2022), and late pretraining usually changes the data (Gururangan et al., 2020; Ibrahim et al., 2024; Hu et al., 2024; OLMo Team, 2025), masking patterns aside (Bavarian et al., 2022; Tay et al., 2023). §5 keeps the data and changes the objective, since here the objective is what withholds conditioning.

8 Discussion and Conclusion

Read along α rather than as one number, the held-out loss of a hyperspherical flow language model separates three things that scale does. Embedding anisotropy raises the noise level at which per-position evidence is certified while the model's frontier sinks, so the training window soon falls wholly below the certified region and no later gradient comes from inside it (§3). Generation from noise improves with size and budget, whereas conditioning on clean context stays at chance (§4). Conditioning is withheld by the schedule, not by scale: clean prefixes, applied late with all else fixed, remove the deficit at every size for about 1% of held-out cross entropy, with gains that grow with N and D and carry over to task finetuning (§5, §6).

Artifacts. Beyond these findings, SwaFlow is a resource for studying how continuous language models scale. Where released models of this family have been final checkpoints near 0.1B parameters (Table 3), it pairs four hyperspherical flow models from 150M to 3B parameters with autoregressive twins at the three smaller sizes, all trained on one public corpus in one recorded order, with full training states every 20B tokens, the mid-training branches, the post-trained models and the evaluation harness to be released. Any state can be resumed, so a new intervention can branch from any point of training as in §5; any step expands into the documents it consumed, so behaviour can be traced to data; and the twins isolate the objective from everything else (Appendix A).

Limitations. α_{geo} is a typical-case threshold and carries no failure probability. The autoregressive control stops at 1B, the diagnosis of §4 is tested only on hyperspherical flows, and the mid-training configuration was fixed in advance and never swept.

Future work. The diagnosis of §4 rests on drawing one noise level per sequence, not on the sphere, so the same deficit and the same remedy should be tested in other continuous language models. The frontier α_{sol} can be read off any model with a noise-level axis, whereas α_{geo} needs a learned embedding table; for embeddings normalised to a fixed length under a variance-preserving Gaussian path (Chen et al., 2026; Yang et al., 2026), an analogue looks derivable.

AI USE STATEMENT

We used an LLM-based assistant to polish the writing of this paper. The research question, the method, and the derivations were developed by the authors, who take full responsibility for the content of this paper.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

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

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient training of language models to fill in the middle. *arXiv preprint arXiv:2207.14255*, 2022.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Jannis Chemseddine, Gregor Kornhardt, and Gabriele Steidl. Spherical flows for sampling categorical data. *arXiv preprint arXiv:2605.05629*, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

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

Chaoran Cheng, Jiahan Li, Jian Peng, and Ge Liu. Categorical flow matching on statistical manifolds. In *Advances in Neural Information Processing Systems*, 2024.

Chaoran Cheng, Jiahan Li, Jiajun Fan, and Ge Liu. α-Flow: A unified framework for continuous-state discrete flow matching models. *arXiv preprint arXiv:2504.10283*, 2025.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Oscar Davis, Samuel Kessler, Mircea Petrache, İsmail İlkan Ceylan, Michael Bronstein, and Avishek Joey Bose. Fisher flow matching for generative modeling over discrete data. In *Advances in Neural Information Processing Systems*, 2024.

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

Justin Deschenaux and Caglar Gulcehre. Language modeling with hyperspherical flows. *arXiv preprint arXiv:2605.11125*, 2026.

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

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations (ICLR)*, 2019.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, et al. A framework for few-shot language model evaluation, 2024. Version 0.4, Zenodo.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *International Conference on Machine Learning (ICML)*, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Hongcan Guo, Qinyu Zhao, Yian Zhao, Shen Nie, Rui Zhu, Qiushan Guo, Feng Wang, Tao Yang, Hengshuang Zhao, Guoqiang Wei, and Yan Zeng. Continuous latent diffusion language model. *arXiv preprint arXiv:2605.06548*, 2026.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051, 2021.

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

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, et al. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*, 2024.

Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling. In *Advances in Neural Information Processing Systems*, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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.

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

Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, et al. TorchTitan: One-stop PyTorch native solution for production ready LLM pre-training. In *International Conference on Learning Representations (ICLR)*, 2025.

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. *arXiv preprint arXiv:2312.09241*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

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

Alexandru Meterez, Pranav Ajit Nair, Depen Morwani, Cengiz Pehlevan, and Sham Kakade. Anytime pretraining: Horizon-free learning-rate schedules with weight averaging. *arXiv preprint arXiv:2602.03702*, 2026.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations (ICLR)*, 2018.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. *arXiv preprint arXiv:2410.18514*, 2024.

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. *arXiv preprint arXiv:2502.09992*, 2025.

OLMo Team. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. URL https://arxiv.org/abs/2212.09748.

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.

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

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9P9Y8FOSOk.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

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

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. UL2: Unifying language learning paradigms. In *International Conference on Learning Representations (ICLR)*, 2023.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations (ICLR)*, 2022.

Zhihan Yang, Wei Guo, Shuibai Zhang, Subham Sekhar Sahoo, Yongxin Chen, Arash Vahdat, Morteza Mardani, and John Thickstun. Continuous diffusion scales competitively with discrete diffusion for language. *arXiv preprint arXiv:2605.18530*, 2026.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, et al. PyTorch FSDP: Experiences on scaling fully sharded data parallel. *Proceedings of the VLDB Endowment*, 16(12):3848–3860, 2023.

A Artifacts

Analysing what scaling does to a flow language model needs several sizes trained under one recipe, full training states on a token-indexed grid, a data order that is shared across runs and can be recovered, and an autoregressive control on the same data. SwaFlow supplies all four, and we will release it whole upon acceptance: the four flow models, their seven mid-training branches and the three autoregressive twins as 836 full training states, one every 20B tokens (Table 5), the 37 post-trained models of §6, an order log, the evaluation harness and code, and the table every number in this paper is read from.

Each part is there to be used. Any state can be resumed, not only evaluated: the branches of §5 were resumed from intermediate states of finished runs. Any step expands into the documents it consumed, re-read from the public corpus (Appendix A.3). The twins share the data, its order and the schedule, so they isolate the objective. And the harness recomputes $L(\alpha)$ and both boundaries from weights alone, including for checkpoints we did not train (Appendix A.4).

A.1 Against prior releases

Table 4 reads prior continuous language models against that list. The Riemannian family keeps the corrupted state on a Riemannian manifold and moves it along geodesics (SFCD also along von Mises–Fisher paths); α-Flow (Cheng et al., 2025), at about 92M on Text8 only, is omitted. Euclidean

flows put a one-hot vector, an embedding or a latent under Gaussian noise, and Euclidean diffusion models are trained on a variational bound. Discrete diffusion is not listed. These works ask other questions, for which a final checkpoint is the natural artifact.

Token counts marked † are steps × batch × length: S-FLM's OWT batch is taken from its released script, Text8 counts characters, and SFM's budget reads two ways, so both are printed. S-FLM prints no parameter count; 170M is ours for its stated shape. Sizes count language models, RDLM's two on different corpora; *sweep* is an IsoFLOP study over several sizes whose autoregressive models are trained under the same protocol, in a number not stated; ELF's autoregressive baselines are for conditional tasks only. ◐ marks final checkpoints that carry optimizer state, which we confirmed in one released file per model. SwaFlow's maximum is the 3B clean-prefix branch of §6.2; the ladder and its twins share 2T.

Table 4: Artifacts of continuous language models, against what a scaling analysis needs; SwaFlow's will be released upon acceptance. ● present, ◐ in final checkpoints only, – absent or not found in the paper, its repository or Hugging Face, n/s not stated, ? not checked. †Derived; §non-embedding.

	Corpus	Max tokens	Data order	Smallest model	Largest model	Sizes	Sizes with matched AR	Intermediate checkpoints	Final checkpoints	Optimizer states
Riemannian: learned token embeddings on a hypersphere										
S-FLM (Deschenaux & Gulcehre, 2026)	OWT, TinyGSM	0.52T†	–	170M†	170M†	1	1	–	●	◐
SFCD (Chemseddine et al., 2026)	LM1B, OWT, TinyGSM	0.52T†	–	93M§	93M§	1	0	–	●	◐
SwaFlow (ours)	Dolma 3	**4T**	●	175M	**3.7B**	**4**	**3**	● (20B)	●	●
Riemannian: the statistical manifold of categorical distributions										
RDLM (Jo & Hwang, 2025)	Text8, LM1B	0.13T†	–	92M	110M	2	0	–	●	◐
Fisher-Flow (Davis et al., 2024)	LM1B	0.33T	–	110M	110M	1	0	–	–	–
SFM (Cheng et al., 2024)	Text8	0.39–1.57T†	–	92M	92M	1	0	–	–	–
Euclidean flows										
FLM/FMLM (Lee et al., 2026)	LM1B, OWT	0.52T	–	179M	870M	3	0	–	●	◐
Scaling CFM (Davis et al., 2026)	Nemotron-PT	2.1T	–	400M	1.7B	2	1	–	–	–
LangFlow (Chen et al., 2026)	LM1B, OWT	0.52T†	–	130M	130M	1	1	–	●	–
ELF (Hu et al., 2026)	OWT	0.045T	–	105M	652M	3	0	–	●	–
Cola DLM (Guo et al., 2026)	n/s	n/s	–	2.3B	2.3B	1	1	–	●	–
Euclidean diffusion										
Plaid (Gulrajani & Hashimoto, 2023)	OWT2	0.31T	–	n/s	1.3B	sweep	sweep	–	●	?
RePlaid (Yang et al., 2026)	SlimPajama, LM1B, OWT	n/s	–	n/s	n/s	sweep	sweep	–	–	–

A.2 Inventory

Table 5: The training states to be released, one every 20B tokens. Each holds weights, optimizer state, learning-rate schedule and dataloader position, and has a row in the aggregate table. Clean-prefix branches are given by the tokens they span.

Model	Uniform α	Clean-prefix branches	Full states
Flow 150M, 500M	0–2T	1.9–2T	$2 \times (101 + 5)$
Flow 1B	0–2T	0.4–0.5T, 0.9–1T, 1.9–2T	$101 + 3 \times 5$
Flow 3B	0–3T	1.9–2T, 3–4T	$150 + 5 + 50$
Autoregressive 150M, 500M, 1B	0–2T	—	3×101
Total			836

Pretraining. Table 5 breaks down the 836 states. No state was deleted during training, so each grid is complete, and each run trained from scratch also keeps weights-only snapshots at steps $1, 2, 4, \ldots, 512$. The full states occupy about 15.4 TB, 12.7 TB of them the flow models and their branches and 2.7 TB the autoregressive twins.

Branches. The seven clean-prefix branches of Table 5 are the four sizes, each resumed from its own 1.9T state and trained to 2T (§5); the 1B model branched to land at 0.5T and 1.0T (§5.2); and the 3B model continued from 3T to 4T, the final model of §6.2.

Random-number state. The generator state behind the α and noise draws is kept only for the last 20 full-state points of each run. A branch from an earlier point resumes optimizer, schedule and dataloader exactly but draws a different, identically distributed stream; among the branches above this applies to the 3B branch at 1.9T and to the two early 1B branches.

Post-trained models. The 23 general-purpose models are one epoch of Dolci on the bases of Appendix G.1, the three autoregressive twins among them; the 14 task-specific models are the TinyGSM runs of Appendix G.2.

Aggregate table. Every number in this paper is read from one table of 909 rows and 113 columns, one row per evaluated state, including the weight averages of §6.2 and the released checkpoints of other models that our harness scores (Table 2), which are marked as such.

A.3 The order log

Appendix B.2 defines the order; the log makes it usable after the fact, and holds no text of its own. It has three parts. The build plan fixes the seed, the mixture weights, the shard layout and the parameters of the keyed permutation; each run records the checksum of the plan it read and its global batch; and a command-line tool combines the two to expand any step or checkpoint into (source, input file, revision, row).

Two conventions. Rank r of W data-parallel ranks reads slot $iW + r$ for its i-th sample, so the ranks jointly cover the slot space exactly once. The dataloader's state is then a single integer, step t consumes exactly the slots $[tB, (t+1)B)$ for global batch B, and a checkpoint at step N has seen $[0, NB)$. And every shard is tokenised from exactly one input file, so a document's origin is a (file, row) pair.

Verification. Three checks accompany the log: a *round-trip* check compares, for one step, the tokens the dataloader delivered, the blocks the plan resolves those slots to, and the rows the log names, re-read from the source files and re-tokenised; a *rebuild* check regenerates the plan and the held-out split byte for byte; and a *held-out* check confirms that no held-out block appears in any slot. All three pass on a 1B-token build produced by the same pipeline, and they ship with the log, so they can be rerun on the full build.

A.4 The harness and its three α grids

The harness reads full training states after the fact, separately from the validator that runs during training, and computes every per-α quantity in this paper. It evaluates on three α grids, and the grid decides which comparisons are valid. The *training grid*, eight points spanning each size's own window, is what each run logged as it trained, and is compared only within a run. The *common grid*, eight points on $[0, 0.10]$ and identical to the 3B training grid, is the one on which cross entropy is compared across sizes. A *window grid* of 32 points over each size's window locates α_{sol}, interpolating in log cross entropy: at 150M α_{sol} is 0.107 even at $D = 2$T, outside the common grid, and being a value of α rather than a loss it may be read on a different grid at each size. The noise of each evaluation is seeded by the point's index within its grid, so cross entropies are never compared across grids.

α_{geo} needs only the embedding table and noise draws, and α_{eq} needs in addition the held-out cross entropy of an autoregressive model on the same data. The weights of the S-FLM reference implementation load into our implementation key for key, which is what lets the decomposition run on a checkpoint we did not train.

A.5 Licensing

Weights will be released under Apache-2.0. The training code is built on torchtitan (Liang et al., 2025), under BSD-3, with parts derived from the S-FLM reference implementation (Deschenaux & Gulcehre, 2026) under Apache-2.0. The corpus, Dolma 3 Mix (Team Olmo, 2025), is under ODC-By 1.0, which asks attribution to the Allen Institute for AI and puts no condition on models trained from it. Because ODC-By also permits redistribution, the order log, the held-out splits and the tokenised shards can be published beside the weights, so a reader can check what a checkpoint saw rather than take our word for it.

B Architecture, data and training

This appendix gives the setup behind §2.

B.1 ARCHITECTURE

Table 1 gives the model shapes and Table 7 the hyperparameters, shared by the flow models and the autoregressive twins except where noted.

B.2 DATA

Table 6 gives the corpus as the build plan sees it: Dolma 3 Mix (ODC-By 1.0) pinned to one revision, at its published source weights with no mixture engineering, tokenised into 2048-token sequences packed across document boundaries. The dataset card counts tokens with the OLMo tokenizer; every token count in this paper is our own, under SmolLM2.

Table 6: Dolma 3 Mix as tokenised for this work, from its build plan. Weights are the corpus' own and are not altered. Epochs are the fraction of each source consumed within the 2T budget of the ladder; no source repeats within the 4T of the longest run.

Source	Weight	Tokens	Epochs at 2T
`common_crawl`	76.07%	4.7930T	0.318
`olmocr_science_pdfs`	13.57%	0.8034T	0.338
`stack_edu`	6.89%	0.5210T	0.265
`finemath_3plus`	2.56%	0.1794T	0.286
`rpj_proofpile_arxiv`	0.86%	0.0546T	0.316
`dolma1_7_wiki_en`	0.04%	0.0027T	0.301
Total		6.3540T	

One deviation from the corpus' reference recipe. The web split marks byte ranges for removal by suffix-array deduplication without stating how the removal is to be applied, and we do not apply them; they touch 14.3% of documents and 1.63% of body bytes. Our corpus is therefore not byte-identical to that of OLMo models of the same lineage, which leaves the comparisons of this paper, all internal, unaffected.

Order. Order is not a stored shuffle but a keyed permutation evaluated on demand: slot i maps to a block of the corpus through a fixed bijection, so a run's trajectory through the data is a function of the plan's seed and the global batch alone. Two consequences matter here. A run's step N and another run's step N read the same slots whenever the global batch matches, which is what makes the autoregressive ladder of §2 a controlled comparison rather than an approximate one; and any step expands, after the fact, into the documents it consumed, which is what the order log encodes. Held-out data is separated by the same construction, 0.1% of every source, and therefore provably never enters a training slot.

B.3 TRAINING

Table 7: Training setup, identical across the four flow sizes and the autoregressive ladder except where noted. Values marked † are conventional or inherited defaults that this work did not measure.

Sequence Length	$2,048$ tokens
Global Batch	$4,096 = 8{,}388{,}608$ tokens
Optimizer	AdamW (Loshchilov & Hutter, 2019) with $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, wd $0.1^{\dagger}$
Learning Rate	$6{\times}10^{-4}$ (150M), $3{\times}10^{-4}$ (500M, 1B, 3B) with max-norm clip 2.5
Scheduler	2K-step linear warmup†, then constant; no run in this paper decays
t sampling	antithetic within rank; per-rank independent seeds; base seed 20260810
Parallelism	FSDP2 (Zhao et al., 2023; Liang et al., 2025) with up to 32 nodes $\times$ 8 H200
AC	selective-op activation checkpointing (Chen et al., 2016) with no compilation
Checkpoints	every 20B tokens with optimizer states

Learning rate. A sweep at a global batch of 8.39M tokens, meant to exclude wrong orders of magnitude rather than to find an optimum, left 6×10^{-4} at 150M and 3×10^{-4} at every larger size; not being monotone in d, the four values are measurements, not a curve.

Window. The window $[0, \alpha_{\max}]$ of each size (Table 1) was fixed from an untruncated pilot of 891M tokens, as the α at which its loss crosses 0.1 nats, rounded up.

Schedule. Every run warms up linearly over 2,000 steps and then holds its learning rate constant, with no decay phase. The choice serves the analysis rather than final performance: a shorter run is then a prefix of a longer one, checkpoints at a matched D are comparable across sizes, and any state can serve as a branch point.

C THE GEOMETRIC THRESHOLD α_{geo}

Setup. Let e_i be the unit embedding of the target token and z a uniform draw on S^{d-1}. The interpolant of Eq. 1 is $x_\alpha = \big[\sin((1-\alpha)\Theta)\, z + \sin(\alpha\Theta)\, e_i\big] / \sin\Theta$ with $\Theta = \arccos\langle z, e_i\rangle$. In d dimensions $\langle z, e_i\rangle = O(d^{-1/2})$, so $\Theta = \pi/2 + O(d^{-1/2})$ and, writing $\theta = \alpha\pi/2$,

$$x_\alpha \;=\; \cos\theta\; z \;+\; \sin\theta\; e_i \;+\; O(d^{-1/2}). \tag{3}$$

As α rises the target's component grows and the noise component shrinks; everything below is stated in this limit.

Scores. The context-free decoder is nearest neighbour in embedding space, $\hat{\imath} = \arg\max_k \langle x_\alpha, e_k\rangle$, which is the strongest inference available from per-position evidence alone. From Eq. 3,

$$\langle x_\alpha, e_i\rangle = \cos\theta\,\langle z, e_i\rangle + \sin\theta = \sin\theta + O(d^{-1/2}), \qquad \langle x_\alpha, e_j\rangle = \underbrace{\cos\theta\,\langle z, e_j\rangle}_{\text{noise}} + \underbrace{\sin\theta\,\langle e_i, e_j\rangle}_{\text{leakage}}. \tag{4}$$

A competitor is helped from two independent directions: noise that happens to point at e_j, and the target's own signal leaking into e_j because the two embeddings are similar. The second term is the one an isotropy assumption removes.

Bounding the best competitor. Put $m = \max_j \langle z, e_j\rangle$ and $c = \max_{j\neq i}\langle e_i, e_j\rangle$. Because the maximum of a sum is at most the sum of the maxima,

$$\max_{j\neq i}\,\langle x_\alpha, e_j\rangle \;\;\le\;\; B \;:=\; \cos\theta\, m + \sin\theta\, c. \tag{5}$$

B is in general *not attained*: m and c are achieved by different tokens, so B describes an adversary that combines the noise luck of one token with the embedding proximity of another. No such token need exist, which is precisely why the resulting threshold is conservative.

The threshold. Decoding succeeds when the target outscores every competitor. Imposing the strictly stronger requirement $\sin\theta > B$ and chaining with Eq. 5,

$$\sin\theta \;>\; B \;\ge\; \max_{j\neq i}\langle x_\alpha, e_j\rangle, \tag{6}$$

so success follows. Solving the requirement for θ, and using $\cos\theta > 0$ on $(0, \pi/2)$,

$$\begin{aligned} \sin\theta > \cos\theta\, m + \sin\theta\, c \;\iff\; \sin\theta\,(1-c) > \cos\theta\, m \;\iff\; \tan\theta > \frac{m}{1-c} \\ \iff\; \alpha > \frac{2}{\pi}\arctan\frac{m}{1-c}, \end{aligned} \tag{7}$$

which is Eq. 2 with $c = c_{\text{mean}}$.

What α_{geo} is, and what it is not. The implication runs one way. Above α_{geo}, context-free decoding provably recovers the token; below α_{geo} nothing is certified, and since B over-estimates the best competitor, decoding frequently still succeeds there. α_{geo} therefore *certifies triviality above itself and is silent below itself*; it is not a statement that per-position evidence is insufficient at lower α, and the gap $\alpha_{\text{geo}}/\alpha_{\text{sol}}$ should be read as the span over which the model operates without that certificate.

Two further points fix its statistical status. First, m is a one-sided maximum taken over the whole vocabulary rather than over $j \neq i$, since the probe draws z with no target attached; including e_i raises m with probability $\approx 1/|V| = 2\times 10^{-5}$ and only in the conservative direction. Second,

we evaluate Eq. 2 at m_{mean} and c_{mean}, the means over 1024 noise draws and 1024 sampled tokens of the respective maxima (fixed seed, CPU generator, so the same draws are used at every size and checkpoint). The distribution of the maximum is concentrated — mean and median agree to within 1% — so α_{geo} is a *typical-case* threshold rather than one holding with a stated probability. Substituting the recorded 95th percentile of m gives a $\delta = 0.05$ analogue, raising α_{geo} by 15–19% at $D = 2$T without changing any trend.

Relation to the analytic form. Deschenaux & Gulcehre (2026) assume embeddings i.i.d. uniform on the sphere and bound m by a union bound over the vocabulary, obtaining $\alpha_{\text{geo}}(\delta) = \frac{2}{\pi} \arcsin\left(t_\delta\right)$ with $t_\delta = \sqrt{2\log(2(|V|-1)/\delta)/d}$. At $c_{\text{mean}} = 0$, Eq. 2 recovers this form up to two corrections that do not move during training: it keeps the factor $\cos\theta$ on the noise term, which turns $\arcsin$ into $\arctan$ and changes the threshold by at most 1.1%, and it measures m rather than bounding it. What moves is the factor $1/(1 - c_{\text{mean}})$ that isotropy fixes at 1: it grows from 1.08–1.19 at $D = 20$B to 3.0–4.0 at $D = 2$T and carries the whole rise of α_{geo} (Appendix D.1).

D Behind the band

Every $L(\alpha)$ in this paper is held-out cross entropy at a fixed α, on the held-out split of Appendix B.2, with each noise draw seeded by its source, block and α alone. The draws therefore do not depend on batch size or degree of sharding, and every size and checkpoint is scored on the same data and the same noise.

Table 8 gives the two boundaries, c_{mean} and the band's width early in training and at a matched $D = 2$T. The width is a ratio because both boundaries fall with scale, so a gap stated in α would not compare across sizes. It also grows with D: for the 3B model alone it is 1.22 at 0.2T, 3.52 at 2T and 3.61 at 3T. Table 8's ordering in N therefore holds only because D is matched there.

Table 8: The band early in training and at a matched $D = 2$T. Cross entropy is on the common α grid; the two boundaries swap order between the blocks.

	$D = 20$B tokens				$D = 2$T tokens			
	150M	500M	1B	3B	150M	500M	1B	3B
held-out CE	3.673	2.820	2.117	1.660	2.718	1.879	1.348	0.994
α_{sol}	0.121	0.096	0.076	0.063	0.107	0.079	0.061	0.049
α_{geo}	0.113	0.085	0.065	0.052	0.257	0.242	0.202	0.174
c_{mean}	0.160	0.120	0.095	0.078	0.669	0.729	0.741	0.753
$\alpha_{\text{geo}}/\alpha_{\text{sol}}$	0.937	0.881	0.857	0.823	2.417	3.051	3.294	3.523

D.1 How the geometry scales in d

The noise geometry scales in d as its idealisation predicts; the embedding geometry does not. The fits, over the four widths $d = 768, 1280, 2048, 3072$:

- $m \propto d^{-0.515}$ at $D = 2$T and $d^{-0.497}$ at $D = 20$B, $R^2 > 0.999$ in both cases, against the $d^{-1/2}$ expected of a maximum over $|V|$ near-Gaussian projections. This is the one quantity the idealisation describes, and it describes it at both budgets.
- c_{mean} does not follow the idealisation: at a fixed $D = 2$T it *rises* with size, from 0.669 at 150M to 0.753 at 3B, as well as with D (Table 8).

The isotropic threshold falls. Setting $c_{\text{mean}} = 0$ in Eq. 2 leaves a threshold driven by m alone. It *falls* between $D = 20$B and $D = 2$T, to 0.91–0.94 of its 20B value at the four sizes, and m itself drifts down by 6–9%. The rise of α_{geo} reported in §3.2 is therefore carried entirely by the anisotropy factor $1/(1 - c_{\text{mean}})$, which grows from 1.08–1.19 at 20B to 3.0–4.0 at 2T, and is slightly opposed by the noise geometry.

D.2 THE TWO CROSSINGS

Table 9 gives both crossings of Fig. 1 and the deficit they start from. At $D = 20$B the model's frontier lies above the threshold at every size, and does so by more at larger sizes: relative to the threshold, a larger model starts further behind its own geometry.

Table 9: The two crossings, by log-D interpolation of the measured series. The first is α_{sol} against α_{geo}, the second α_{geo} against the fixed window edge $\alpha_{\max}$. Both come later with scale, and the first also comes at a lower α.

	150M	500M	1B	3B
$\alpha_{\mathrm{sol}}/\alpha_{\mathrm{geo}}$ at $D = 20$B	1.07	1.14	1.17	1.22
first crossing, D	29.2B	49.3B	62.6B	84.8B
first crossing, α	0.117	0.088	0.068	0.055
second crossing ($\alpha_{\mathrm{geo}} = \alpha_{\max}$), D	90B	218B	278B	382B
as a fraction of a 2T budget	4.5%	10.9%	13.9%	19.1%

The figure marks each first crossing at the nearest measured checkpoint, which differs from the interpolated α only in the third decimal.

D.3 THE EQUIVALENCE POINT

At $D = 2$T, α_{eq} is 0.0518, 0.0398 and 0.0314 for 150M, 500M and 1B, and 0.345, 0.318 and 0.299 of the respective training windows: it falls with size in both forms, so the narrowing window does not account for the trend. Its dependence on D is an order of magnitude weaker than its dependence on N: $\mathrm{d}\ln\alpha_{\mathrm{eq}}/\mathrm{d}\ln D$ is -0.005, -0.027 and -0.030 (Fig. 7, right).

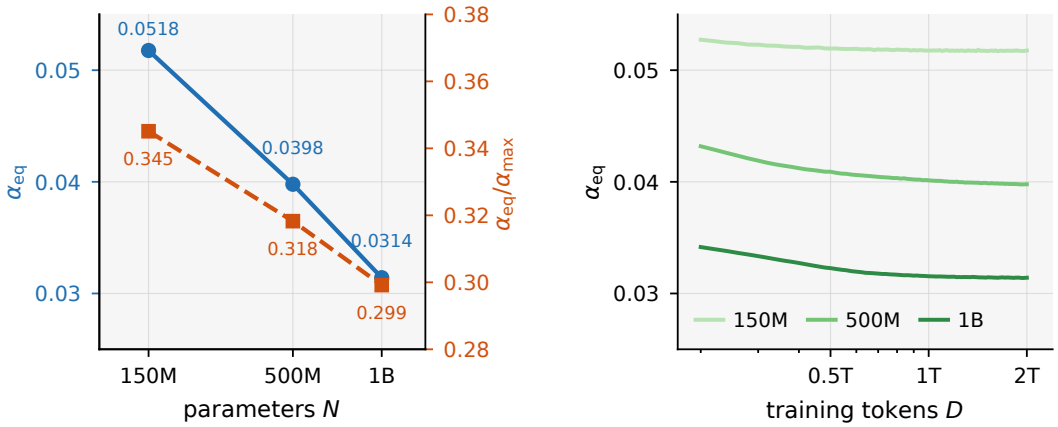


Figure 7: The equivalence point α_{eq}. Left: against N at $D = 2$T, raw and normalised by the window (as Fig. 3). Right: against D from 200B.

Placed on the band of §3.2, α_{eq} sits at 0.49–0.51 of the model's own frontier α_{sol} and at 0.15–0.20 of α_{geo}, at all three sizes. Since α is drawn uniformly on the window, the normalised value $\alpha_{\mathrm{eq}}/\alpha_{\max}$ above is also the probability that a training draw falls below the equivalence point, so 30–34% of all draws pose a strictly harder per-position problem than next-token prediction on the same data.

Two limits apply. The autoregressive ladder stops at 1B, so the trend in N rests on three points. And we do not compare rates of improvement at α_{eq}, where the two cross entropies are equal by construction, so the measurement orders difficulty rather than identifying what closes the gap.

E GENERATION, MULTIPLE CHOICE AND THE PREFIX PROBE

This appendix gives how the generative quantities of §3.1 and §4 are measured, and the scoring rules and individual values behind the discriminative ones. The values at $D = 2\text{T}$ are in Tables 2 (multiple choice, per task) and 10.

E.1 GENERATION

Generative quantities are measured at every 20B checkpoint to 2T, 100 per size: 256 samples of 512 tokens at 64 function evaluations on an α grid inside the model's training window and temperature 1, from pure noise for the unconditional arm and after a clean 64-token held-out prefix for the conditional one, both scored from position 64 onward, i.e. on 448 tokens. Gen-PPL uses SmolLM2-1.7B (Allal et al., 2025). A sample's rep-4 is the share of its tokens covered by some 4-gram that occurs in it more than once, 1.0 under unbounded repetition; a sample collapses when its rep-4 exceeds 0.9, and the collapse rate is the collapsing fraction of the 256. The real-text references score held-out text at the same positions: Gen-PPL 9.64, rep-4 0.157, and a collapse rate of 0.39%. The thick lines of the Gen-PPL and collapse panels of Fig. 2 are robust LOWESS fits in $\log D$ (Cleveland, 1979) with span 0.45 and two bisquare iterations, fitted to log Gen-PPL or to the rate itself, plus a correction that pins each curve to its first and last checkpoints and decays to zero within 0.3 decades of either end; everything inside is the LOWESS fit.

Conditional Gen-PPL inverts. Over the 400 checkpoints measured to 2T, log conditional Gen-PPL and the collapse rate correlate at $r = -0.73$: the one checkpoint that scores below the real-text floor, 500M at 0.82T with 5.6, is the one collapsing most, on 64.8% of its samples. We therefore read conditional Gen-PPL only beside the collapse rate (Table 10).

The conditional collapse rate. Every model above 150M passes through a hump midway through training: on the smoothed series the rate peaks at 36%, 40% and 32% for 500M, 1B and 3B, at 0.69, 0.83 and 0.61T, and single checkpoints reach 62–65%; on the 1B budget axis of Fig. 4 it is 33.6% at 0.5T and 53.9% at 1.0T. 150M stays below 6% on the smoothed series and reaches 13% at its worst checkpoint. At 2T the single checkpoints of Table 10 sit at 0.8%, 18.4%, 9.0% and 0.0%, and the medians over 1.5–2T are 3.1%, 20.3%, 18.0% and 5.9%. Unconditional generation from the same checkpoints never exceeds 9.4%.

Why no distribution-distance score. We do not report scores such as MAUVE (Pillutla et al., 2021). At the sample budgets that four sizes across a dozen checkpoints allow, the measurement is dominated by sample size: at 8,192 tokens the unigram total variation between a draw of held-out text and the corpus is 0.54, larger than the 0.43–0.44 of our generations.

E.2 MULTIPLE CHOICE

The score. Six tasks (Clark et al., 2019; Mihaylov et al., 2018; Bisk et al., 2020; Zellers et al., 2019; Clark et al., 2018; Sakaguchi et al., 2021) are formatted after Gao et al. (2024); the chance level of their average is 0.375. For a prompt x and an option c of $|c|$ tokens, let $\mathrm{CE}(c \mid x)$ be the summed cross entropy over the option positions of a sequence in which the prompt is held clean and the option positions are at $\alpha = 0$ — pure noise, so the option is read from the prompt alone. Let $\tilde{x}$ be the prompt with every position replaced by one neutral filler token, a newline. The option scores

$$s(c) \;=\; \frac{1}{|c|}\Big[\mathrm{CE}(c \mid x) \;-\; \mathrm{CE}(c \mid \tilde{x})\Big], \tag{8}$$

and the option with the smallest score wins. The bracket is a PMI correction (Holtzman et al., 2021): length, option positions and the noise draw are identical between its two terms, so the difference cancels the per-token term measuring how readily a token is recovered from noise, which otherwise dominates a flow model's option scores.

Both normalisations are needed. Without the correction the score ranks options by how recoverable their tokens are; without the division by $|c|$ it ranks them by brevity. Every flow MCQ accuracy in this paper uses Eq. 8.

The autoregressive twins. They are scored by their plain conditional likelihood of the option given the prompt, under the same length normalisation, since neither α nor the PMI correction has a

Table 10: Uniform α against clean-prefix mid-training at a matched $D = 2$T (step 238,400); the arms differ only in where α is applied over the last 100B tokens. SFT-format rows score held-out instruction responses under a clean or corrupted prompt; $\|g\|$ is the gradient norm at the first step of an identical instruction-finetuning run from that checkpoint. Conditional Gen-PPL is read only with the collapse rate above it: lower is not better (Appendix E.1). Subscripts: difference from uniform α.

	uniform α				clean-prefix mid-training			
	150M	500M	1B	3B	150M	500M	1B	3B
MCQ Acc. (chance 0.375)	0.362	0.354	0.357	0.369	$0.403_{+0.041}$	$0.419_{+0.065}$	$0.434_{+0.077}$	$0.432_{+0.063}$
Prefix-probe Acc. (chance 0.25)	0.451	0.593	0.492	0.582	$0.704_{+0.253}$	$0.844_{+0.251}$	$0.944_{+0.452}$	$0.954_{+0.372}$
Clean-prefix CE (unigram 7.53)	8.00	8.37	7.72	8.53	$7.16_{-0.84}$	$7.07_{-1.30}$	$6.88_{-0.84}$	$6.92_{-1.61}$
SFT-format CE, clean prompt	4.010	3.080	3.151	2.109	$2.411_{-1.599}$	$1.652_{-1.428}$	$1.162_{-1.989}$	$0.860_{-1.249}$
SFT-format CE, noised prompt	2.492	1.695	1.205	0.892	$2.519_{+0.027}$	$1.714_{+0.019}$	$1.221_{+0.016}$	$0.905_{+0.013}$
$\|g\|$ at fine-tuning step 1	9.62	21.19	38.87	15.81	$1.87_{-7.75}$	$1.68_{-19.51}$	$1.56_{-37.31}$	$0.97_{-14.84}$
Held-out CE (common grid)	2.718	1.879	1.348	0.994	$2.738_{+0.020}$	$1.896_{+0.017}$	$1.360_{+0.012}$	$1.006_{+0.012}$
Cond. collapse rate (%)	0.8	18.4	9.0	0.0	$0.0_{-0.8}$	$0.0_{-18.4}$	$1.2_{-7.8}$	$0.0_{\pm 0.0}$
Cond. Gen-PPL (real text 9.64)	356	50	472	228	505_{+149}	189_{+139}	108_{-364}	169_{-59}

counterpart there. It is therefore the accuracies and not the scores that are commensurable across the two ladders: at $D = 2$T the twins reach 0.481, 0.551 and 0.604 at 150M, 500M and 1B.

Absolute values. On a quantity pinned at chance a difference counts noise as signal, so we quote absolute values throughout, with standard errors. At $D = 2$T the average is 0.362, 0.354, 0.357 and 0.369 (± 0.005) from 150M to 3B. Along the 150M D axis it lies inside 0.360–0.388 across twenty checkpoints from 0.1 to 2T, a series that straddles chance and whose furthest excursion from it is 3.1σ. Over all four sizes and the 27 checkpoints from 0.5 to 2T at which it was measured, it lies inside 0.354–0.388.

What the average contains. The per-task split (Table 2) shows that the mean hides no task-level ability. Hellaswag, piqa and winogrande never depart from chance by more than 0.03 at any size; openbookqa sits at or below its 0.25; arc-easy is the only task appreciably above chance (0.27–0.33). Boolq — binary, chance 0.50 — sits at 0.377–0.386, systematically *below* chance rather than at it, so the score prefers the wrong option. Because the average is a sum of such task-level offsets rather than a quantity continuous in either axis, we never draw it as a curve.

Scope. The statement is scoped to this arm and this rule — a uniform-α model scored by in-window cross entropy under a clean prompt — and §7 places it beside a concurrent model scored differently.

E.3 The prefix probe

Construction. Each item holds the first 64 tokens of a held-out document clean and offers four 32-token continuations: the document's own and those of three other documents. A model that uses its context should pick the true continuation almost always, so the probe is built to have a large effect size. The flow models score the continuations at $\alpha = 0$ with the same PMI correction as Eq. 8 (the continuations share one length, so the length division is moot), and the autoregressive twins by plain log-likelihood. There are $n = 1024$ items, chance is 0.25, and standard errors are ± 0.011–0.016. The values at $D = 2$T are in Table 10. Along D, the probe was measured at 1B only.

The 1B decay. Under uniform α the 1B model's probe accuracy falls from 0.595 at 0.5T to 0.510 at 1.0T to 0.492 at 2T, a drop of 4.7σ. Over the same span its cross entropy on the gold continuation *improves* ($8.34 \to 7.87 \to 7.72$ nats), so what decays is not absolute likelihood under a clean prefix but the separation of the true continuation from the distractors. We have no mechanism for that decay and do not offer one.

F Clean-prefix mid-training in detail

Table 10 gives the per-size values at $D = 2$T.

F.1 Design

In expectation 12.5% of the positions of a sequence are held clean, and they are neither corrupted nor scored. The mixing probability, the maximum split point and the length of 100B tokens (11,920 steps) were fixed before any branch was run and were not swept. Each branch resumes the uniform run's optimizer, learning-rate schedule and dataloader position, and k is a function of the sequence's slot index alone. The branches at 1.9T of 150M–1B also resume the random-number state, so their two arms see identical α and noise draws; the 3B branch at 1.9T and the two early 1B branches start where that state is not kept (Appendix A.2) and draw a different, identically distributed stream.

F.2 Results at a matched budget

Per task (Table 2), boolq supplies most of the multiple-choice gain at every size, and arc-easy adds to it at 1B and 3B. On the probe the autoregressive twins score 0.797, 0.852 and 0.867 at 150M, 500M and 1B.

One epoch of instruction finetuning costs the same uniform-α checkpoints 7.7–11.3% of held-out cross entropy (Appendix G.1). Under mid-training, unconditional Gen-PPL at 2T stays within its checkpoint-to-checkpoint spread and the unconditional collapse rate below 2%.

Finetuning on top of mid-training adds +0.014, +0.010, +0.005 and +0.015 in multiple-choice accuracy from 150M to 3B.

F.3 The budget axis

The 1B branches leave the uniform run at 0.4, 0.9 and 1.9T. At all three landing points, including 0.5T, SFT-format cross entropy is lower with the prompt clean than corrupted for the mid-trained arm, and more than twice as high for the uniform arm. The multiple-choice gain along D is carried by boolq; the other five tasks move by less than 2σ.

F.4 Weight averaging

The averaged 3B model is the exponential moving average of Meterez et al. (2026), $\bar{\theta}_{t+1} = \tau_t \bar{\theta}_t + (1 - \tau_t)\theta_t$ with $\tau_t = 2^{-f/t}$ at step t and $f = 100$. We reconstruct it after the fact from the states saved every 20B tokens: between consecutive states at steps T_{i-1} and T_i the average decays by $(T_{i-1}/T_i)^{f \ln 2}$, and the weight of that interval falls on the state at T_i.

G Post-training in detail

This appendix gives the recipes behind §6.

G.1 General-purpose post-training

Recipe. One epoch of Dolci-Instruct-SFT is 1,107 steps and 1.16B tokens, with examples packed by concatenating and chunking. The prompt is held clean and unscored and the response is corrupted at the sequence's own α, which makes the recipe the supervised counterpart (Wei et al., 2022; Ouyang et al., 2022) of the mid-training mask of §5.1. The same recipe is applied to the mid-trained checkpoints, to the uniform-α checkpoints and to the autoregressive twins.

Cost. Held-out cross entropy on the common grid rises by 2.4–5.4% from the mid-trained checkpoints and by 7.7–11.3% from the uniform-α ones.

G.2 Task-specific setup

Recipe. Each TinyGSM (Liu et al., 2023) example fills one 512-token block, and the loss is taken on the padding as well as on the answer; the prompt is clean and unscored as in Appendix G.1. The learning rate is 3×10^{-4} in every cell, and checkpoints are read at 20, 40 and 65B finetuning tokens. "None" is a randomly initialised model of the same size given the same finetuning. The ten

conditions are the four sizes from none, the four sizes from their 2T mid-trained checkpoints, and the 1B model from its 0.5T and 1T mid-trained branches (§5.2).

Layout. We do not pack examples here. Under packing, the target of the positions after an answer is the next problem, so the model learns to fabricate one there, and a flow model, which decodes every position of the block in parallel, then spends those positions in competition with the answer. With one example per block the target after the answer is padding, which is deterministic; the reference implementation's TinyGSM scripts use the same layout.

Scoring and sampling. GSM8K ($n = 1319$; Cobbe et al., 2021) is prompted 0-shot, and the generated Python is executed; a generation is correct when the program's returned value matches the gold answer (Gao et al., 2023; Chen et al., 2021). Generation uses 64 function evaluations at temperature 1.0 with a greedy final decode. The *exact* sampler marginalises the per-step velocity over the vocabulary and is the reference implementation's default; the *top-1* sampler aims the velocity at the arg-max token. Beside accuracy we report the executable rate, the fraction of generations whose function returns without an exception or a timeout.

G.3 Results

Accuracy rises from 20 to 40 to 65B finetuning tokens in every condition, with both samplers. With the top-1 sampler, the executable rate of the 1B model rises with its base's pretraining budget at every finetuning budget. Table 11 gives the end points of Fig. 6.

Table 11: GSM8K accuracy after 65B tokens of TinyGSM finetuning (0-shot, scored by execution, $n = 1319$), with the executable rate in parentheses. "None" is a randomly initialised model; every other base is clean-prefix mid-trained. These are the end points of Fig. 6 (right pair).

Sampler	Base	150M	500M	1B	3B
exact	none	0.0197 (0.469)	0.0637 (0.522)	0.0743 (0.513)	0.0879 (0.484)
	2T	0.0576 (0.588)	0.1168 (0.625)	0.1251 (0.644)	0.1266 (0.655)
top-1	none	0.0485 (0.515)	0.1334 (0.594)	0.1471 (0.577)	0.1509 (0.598)
	2T	0.1137 (0.652)	0.2138 (0.754)	0.2237 (0.797)	0.2055 (0.741)