Parcae: Scaling Laws For Stable Looped Language Models

Paper discussion: stability, training budgets, and inference saturation

Reading presentation

Paper: Hayden Prairie, Zachary Novack,
Taylor Berg-Kirkpatrick, and Daniel Y. Fu

arXiv:2604.12946v1 (14 April 2026)

Reading: 16 September 2026

More compute per parameter can help—up to a plateau

- Parcae stabilizes shared-depth language models with constrained state transmission and better depth sampling.
- At 1.3B: Core improves by **2.99 points** at matched data and approximately matched parameters.
- Training: allocate compute to **both recurrence and data**.
- Inference: added recurrence approaches a **training-dependent loss floor**.

Scope

Quality comparisons and compute-matched comparisons answer different questions.

Secs. 4–5; Table 5, PDF p. 8; Eq. (4), p. 11.

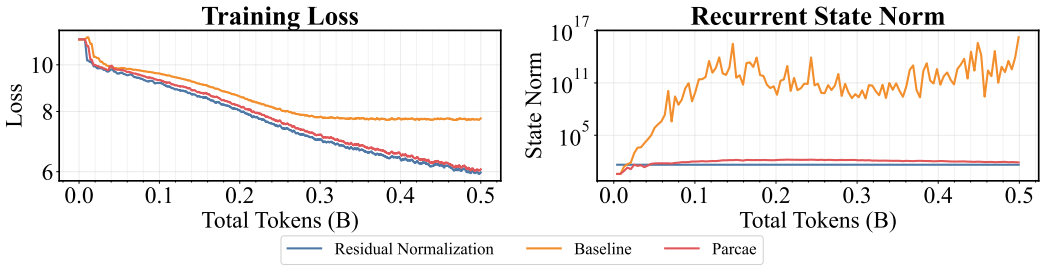
Loop the middle block to reuse stored weights

$$s \xrightarrow{\mathcal{P}} e, \qquad h_0 \xrightarrow{\mathcal{R}(\cdot,e)\times T} h_T \xrightarrow{\mathcal{C}} p$$

- Prelude encodes tokens; Recurrent updates hidden states; Coda predicts tokens.
- The recurrent block uses the **same parameters** at every iteration.
- Training samples depth; inference chooses a depth T.
- Repeated hidden-state computation increases effective depth and compute.

Sec. 2.1, PDF p. 3. This schematic is an instructional summary.

Uncontrolled recurrence can explode the residual state



- Left: training loss. Right: recurrent state norm on a logarithmic axis.
- Residual normalization helps, but stability remains sensitive to learning rate.

Fig. 2, PDF p. 4; Table 2, p. 5. Original author figure.

A linear surrogate isolates the state transmission

$$h_{t+1} = \overline{A} h_t + \overline{B} e + \overline{\mathcal{R}}(h_t, e)$$

- $\overline{A}$: state transmission; $\overline{B}$: input injection.
- $\overline{\mathcal{R}}$: nonlinear residual contribution of Transformer blocks.
- Dropping $\overline{\mathcal{R}}$ gives an LTI surrogate:

$$h_T = \overline{A}^T h_0 + \sum_{k=0}^{T-1} \overline{A}^k \overline{B} e.$$

Here $\overline{A}^T$ is a matrix power. For the linear system, $\rho(\overline{A}) < 1$ implies asymptotic stability.

Sec. 3, Eq. (2), PDF p. 5; App. C, p. 22. Expansion is a teaching derivation.

Negative diagonal dynamics constrain linear memory

$$A = \mathrm{Diag}(-\exp(a)), \qquad \overline{A} = \exp(\Delta A), \qquad \overline{B} = \Delta B$$

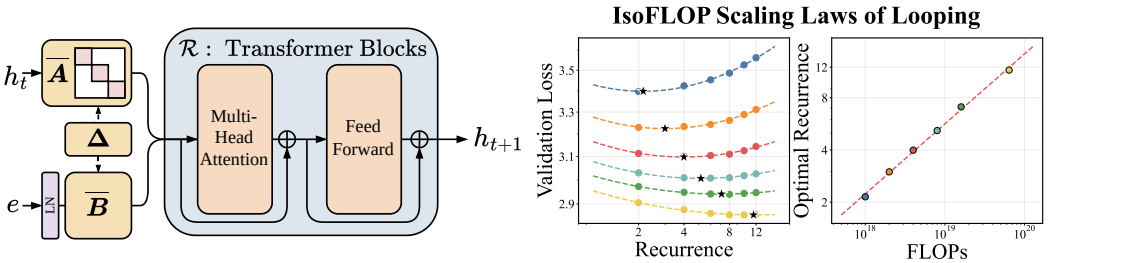
- For $\Delta_i > 0$, each diagonal entry of $\overline{A}$ lies in $(0, 1)$.
- ZOH for $\overline{A}$; Euler for $\overline{B}$.
- Normalize the prelude output: $e = \mathrm{LN}(\mathcal{P}(s))$.
- Predict with $p = \mathcal{C}(Ch_T)$; C is a readout projection.

Implementation detail to resolve

The paper does not specify how learned Δ is kept positive.

Sec. 4.1, Eq. (3), PDF p. 6. Positive step sizes are required by this argument.

Parcae combines a linear constraint and nonlinear blocks



Constrain state transmission, normalize input injection, retain learned nonlinear updates.

Fig. 1, PDF p. 2. Original author figure, trimmed as in the paper.

The stability argument has a precise boundary

- The LTI surrogate omits nonlinear updates; it is not the full network Jacobian.
- Spectral radius and spectral norm differ in general; they coincide for this diagonal design.
- Constraining $\overline{A}$ alone does not prove global nonlinear convergence.

Teaching counterexample

$$h_{t+1} = 0.5h_t + \overline{\mathcal{R}}(h_t), \qquad \overline{\mathcal{R}}(h_t) = h_t.$$

The full update multiplies the state by 1.5.

Interpretation of Secs. 2.2–4 and App. C; counterexample is this reading's analysis.

Sample depth per sequence to better cover the objective

$$\min_{\theta} \ \mathbb{E}_{(x,y),\, T\sim\Lambda}[\ell(f_\theta(x;\, T), y)]$$

- Sample T_i independently within a micro-batch.
- Align endings: delay sequence i by $T_{\max} - T_i$ steps.
- Backpropagate through the final $\min(T_i, \mu_{\text{bwd}})$ updates.

Example: depths 2, 4, 6

Wait for 4, 2, 0 steps respectively; all sequences finish together. Waiting must preserve state.

Sec. 4.2, PDF p. 6; App. E, Algorithm 2, p. 23. Example is instructional.

Sample full forward depth before truncating gradients

$$T \sim \Lambda(\mu_{\mathrm{rec}}), \quad n = \max(T - \mu_{\mathrm{bwd}}, 0), \quad k = \min(T, \mu_{\mathrm{bwd}})$$

- Sampling a no-gradient prefix and adding a fixed suffix shifts the depth distribution.
- The correction restores low-depth exposure and improves generalization across T.
- Main runs use $\mu_{\mathrm{bwd}} = \lceil \mu_{\mathrm{rec}}/2 \rceil$.
- Main RDM baselines also receive the sampling-distribution correction.

App. H–I, PDF pp. 27–30. The backward-depth rule is empirical, not proven optimal.

Three evaluation protocols separate three questions

Protocol	Held fixed	Question
End-to-end quality	Parameter scale and tokens	Does looping improve quality per stored parameter?
Training isoFLOP	Parameter scale and training FLOPs	How much recurrence versus data?
Inference scaling	A trained model	How much does increasing T help?

Important: the first protocol does not match training or inference compute.

Sec. 5, PDF pp. 7–11; App. O, pp. 38–40.

The Transformer comparison uses a shared data recipe

- FineWeb-Edu; 32,768-token BPE; context 2,048; batch 256; BF16.
- 140M / 370M / 770M / 1.3B: 11.2B / 29.6B / 61.6B / 104B training tokens.
- Loop the middle third; train at $\mu_{\text{rec}} = 8$, $\mu_{\text{bwd}} = 4$.
- Parcae adds 0.35–0.83% parameters. Learning rates are selected for baselines and reused.
- Core and Core-Extended are aggregate downstream scores; evaluation uses three seeds.

Apps. M, P–R, PDF pp. 38–44. Evaluation seeds are not independent pretraining runs.

Parcae improves PPL over prior residual-norm RDMs

Scale / depth	Validation PPL ↓		6-task Avg. ↑	
	RDM	Parcae	RDM	Parcae
100M / $T = 16$	14.23	**13.59**	45.03	**46.83**
350M / $T = 8$	10.76	**10.09**	49.45	**50.12**

- Huginn data/tokenizer; this is separate from the nanochat setup.
- 350M validation PPL reduction: $\approx 6.23\%$ from table values.
- Individual tasks can regress; average improvement is not universal.

Table 3, PDF p. 7. Abstract says 6.3%; text says 6.2%. Values above follow the table.

Each stabilization addresses a different failure mode

140M cumulative ablation	Validation loss ↓		
	$T = 1$	$T = 4$	$T = 8$
RDM	Divergent training		
Constrained $\overline{A}$	8.99	3.15	2.97
+ Per-sequence sampling	3.38	3.01	2.98
+ Prelude norm	**3.28**	**2.97**	**2.95**

Constrained transmission enables training; depth sampling especially helps shallow inference; prelude normalization further improves loss.

Table 4, PDF p. 7. Not every downstream metric improves monotonically.

Matched parameters and data yield better Core scores

Scale	Core ↑		Validation PPL ↓	
	Transformer	Parcae	Transformer	Parcae
140M	13.00	14.04	21.48	19.06
370M	17.46	20.00	15.79	14.49
770M	22.42	25.07	13.08	12.49
1.3B	25.45	**28.44**	11.95	**11.42**

At 1.3B, Core: $25.45 \pm 0.08 \rightarrow 28.44 \pm 0.28$;
Core-Extended: $15.90 \pm 0.23 \rightarrow 17.08 \pm 0.09$.

Compute is not matched; Parcae is evaluated at $T = 8$.

Table 5, PDF p. 8. Original error bars retained; the paper does not specify SD versus SE.

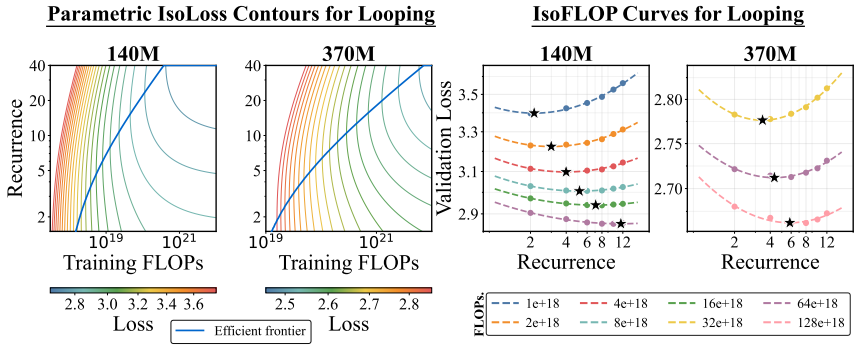
87.5% means closing a gap, not saving parameters

$$\frac{\mathrm{Core}_{770M\ \textit{Parcae}} - \mathrm{Core}_{770M\ \textit{Transformer}}}{\mathrm{Core}_{1.3B\ \textit{Transformer}} - \mathrm{Core}_{770M\ \textit{Transformer}}} = \frac{25.07 - 22.42}{25.45 - 22.42} \approx 87.5\%.$$

- This measures the fraction of the quality gap to the next larger baseline that is closed.
- Actual parameter counts are about 777M versus 1,334M (1.72×).
- Cross-scale comparisons also use different token budgets.

Sec. 5.1, PDF p. 7; Tables 5, 19, 21, pp. 8, 42, 44.

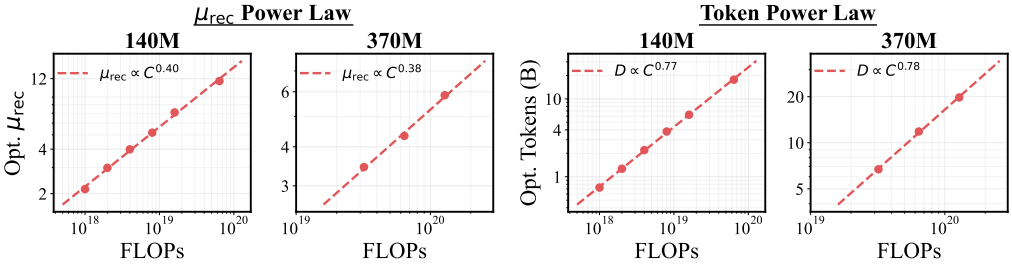
At fixed FLOPs, recurrence has an interior optimum



Right: more recurrence eventually displaces too much training data.

Fig. 4, PDF p. 8. Left horizontal axis is training FLOPs. Original author figure.

Optimal recurrence and tokens both grow with budget



Scale	$\mu^*_{\text{rec}} \propto C^{\gamma_\mu}$	$D^* \propto C^{\gamma_D}$
140M	$\gamma_\mu = 0.40$	$\gamma_D = 0.77$
370M	$\gamma_\mu = 0.38$	$\gamma_D = 0.78$

Fig. 5, PDF p. 9. Finite-range empirical fits, not universal exponents.

Training loss is fitted using effective unfolded capacity

$$\widehat{\mathcal{L}}_{train}(\mu, D) = E + X\mathbf{N}(\mu)^{-x} + YD^{-y}$$

- $\mathbf{N}(\mu)$ counts effective unrolled parameters, not stored weights.
- Huber objective on log loss, optimized with L-BFGS.
- Held-out loss prediction error: 1.3% (140M), 0.8% (370M).

$$C_{\mathrm{FLOPs}} \approx (2\widehat{N}_1 + 6\widehat{N}_2)D + \text{attention cost}$$

$\widehat{N}_1$: no backward pass; $\widehat{N}_2$: forward and backward.

Sec. 5.2, PDF p. 9; App. D, p. 22; App. K, p. 35.

Compute-matched downstream wins depend on budget

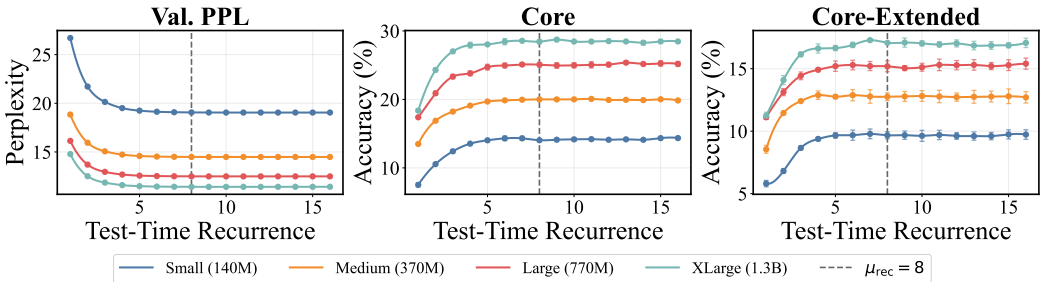
Scale	FLOPs (10^{18})	μ^*	Loop Core	Fixed Core
140M	2	2	9.0	**10.5**
140M	8	6	10.5	**11.8**
140M	16	8	**14.6**	13.0
140M	64	10	**16.2**	15.0
370M	32	4	15.2	**16.8**
370M	128	6	**20.1**	18.1

Fixed baseline here: Parcae with $\mu = 1$.

Lower validation loss does not guarantee better Core at every budget.

Table 6, PDF p. 9; Table 16 with error bars, p. 40. Selected original rows.

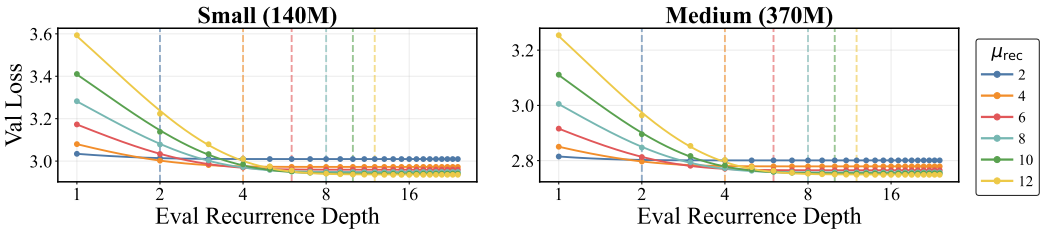
Inference gains plateau near the training depth



Four model scales; all trained with $\mu_{\text{rec}} = 8$.

Fig. 7, PDF p. 10. PPL decreases; aggregate task scores increase, then saturate.

Saturating exponentials fit inference curves well



$$\mathcal{L}(T) = \mathcal{L}_\infty + Ze^{-zT}$$

$\mathcal{L}_\infty$: training-dependent floor; Z: gap scale; z: decay rate.

The exponential beats the tested power-law alternatives, including in depth extrapolation.

Fig. 8, PDF p. 10; Table 12, p. 36. Caption says power laws; actual fit is exponential.

A unified law separates the floor from inference progress

$$\widehat{\mathcal{L}}_{unified}(T \mid \mu, D) = \underbrace{E + X\mathbf{N}(\mu)^{-x} + YD^{-y}}_{\text{training-dependent floor}} + \underbrace{Ze^{-zT/\mu}}_{\text{inference gap}}$$

- T/μ conditions the decay on training depth.
- Full held-out prediction error: **0.85–1.31%**.
- With an empirically measured floor: **0.10–0.17%**.
- The measured-floor variant is an oracle diagnostic, not a fully predictive fit.

Eq. (4), PDF p. 11; App. L, pp. 36–38.

The main payoff is parameter efficiency with boundaries

- **Established:** improved stability in tested runs; higher quality per parameter at matched tokens.
- **Compute allocation:** recurrence and data should grow together in the tested high-budget regimes.
- **Inference:** depth helps until a training-dependent plateau.
- **Open:** full nonlinear stability, larger-scale laws, and joint optimization of parameters/data/depth.
- **Deployment:** latency, throughput, KV cache, and activation memory need separate measurement.

Sec. 6, PDF p. 11; evidence boundaries from this reading. Main talk: approximately 25 minutes.

Reference and reading provenance

Hayden Prairie, Zachary Novack, Taylor Berg-Kirkpatrick, Daniel Y. Fu.

Parcae: Scaling Laws For Stable Looped Language Models.
arXiv:2604.12946v1, 14 April 2026.

`https://arxiv.org/abs/2604.12946v1`

- Read the version-matched PDF and author TeX, including Apps. A–R.
- Figures are original author assets. Selected tables preserve original values.
- Teaching examples and critical interpretations are marked explicitly.
- This presentation does not reproduce model training.

Backup: per-sequence sampling especially helps low depth

Scale	Sampling	PPL $T = 1$	PPL $T = 8$	PPL $T = 16$
100M	Per-batch	300.32	16.65	13.81
	Per-sequence	70.47	14.08	13.59
350M	Per-batch	167.61	10.40	10.24
	Per-sequence	17.92	10.09	10.11

- The paper reports 1.8% extra pretraining wall-clock time for per-sequence sampling.
- This is sampling overhead, not the cost of looping versus a Transformer.

App. G, Table 8, PDF pp. 25–26.

Backup: late instability can originate in the input

- An early 1.3B configuration develops late loss spikes and large state norms.
- The authors inspect state-transition, injection, readout, and actual activation expansion.
- The large jump occurs at the first injection; prelude activations are implicated.
- Prelude normalization enables the reported stable 1.3B run.
- Improved gradient flow is an author interpretation, not directly demonstrated.

App. J, Figs. 17–24, PDF pp. 31–34.

Backup: depth extrapolation favors a saturating exponential

Fit on $T \leq \mu$; evaluate on held-out $T > \mu$.

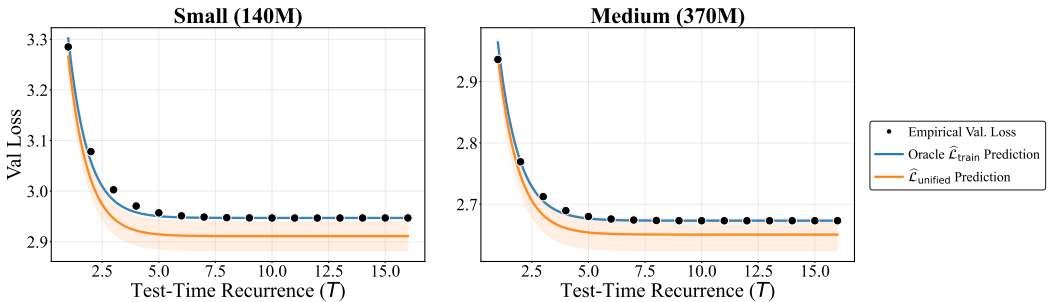
	$\mathcal{L}_\infty + Ze^{-zT}$	$\mathcal{L}_\infty + Z(1+T)^{-z}$	$\mathcal{L}_\infty + ZT^{-z}$	ZT^{-z}
140M	**3.18**	21.41	43.99	397.90
370M	**2.29**	18.51	38.68	369.83

Average Huber loss, $\times 10^{-7}$; lower is better.

The result compares these candidates on the observed depth range. It does not establish a unique universal law.

App. L, Table 12, PDF p. 36.

Backup: the floor explains most of the prediction gap



Orange: full prediction. Blue: measured floor. Black dots: empirical loss.

App. L, Fig. 26, PDF p. 38. Both model sizes are held out from the isoFLOP fit.

Backup: reproduction requires resolving reporting issues

- RDM learning rates differ by $10\times$ between two paragraphs of App. Q.1.
- App. I gives conflicting fixed backward depths in a figure caption and the text/table.
- App. O reports a data-point count inconsistent with its listed model/depth grid.
- App. O says eight fit parameters; the final fixed-γ equation shows seven.
- Use primary table values for PPL improvements; some narrative percentages disagree.

PDF pp. 7–8, 29, 40, 42–43. These discrepancies are documented in the Chinese notes.