SCALING LAWS FOR LOOPED MIXTURE OF EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Looped transformers and Mixture-of-Experts (MoE) offer complementary routes to efficient scaling: recurrence increases computational depth at fixed parameters, while MoE sparsity expands total capacity at fixed active compute. Yet existing scaling laws model recurrence or sparsity in isolation. In this work, we introduce *Loop Scaling Laws*, the first scaling law to jointly model recurrence and sparsity alongside model size and data. At its core is a bounded, sparsity-conditional recurrence mapping that characterizes the effective-parameter gain from looping and how sparsity raises this gain. The laws predict the held-out loss of looped models more accurately than prior alternatives, and recover the standard dense and MoE scaling laws as special cases. Beyond prediction, the fitted laws provide a principled foundation for designing looped MoE models under compute and memory constraints. Downstream evaluations further demonstrate the complementary benefits of the two axes: sparsity delivers $\sim 3\times$ active-parameter efficiency, recurrence yields $\sim 2\times$ total-parameter efficiency on reasoning, and joint scaling further advances the performance frontier. As a practical extension, we show these gains hold at trillion-token scale: at matched training compute, a looped MoE with law-derived recurrence matches a $\sim 2\times$ larger non-looped MoE on the reasoning benchmarks, while enabling test-time scaling through recurrence.

1 INTRODUCTION

Scaling language models has conventionally relied on larger models and more data (Kaplan et al., 2020; Hoffmann et al., 2022). Beyond model and data, two additional axes enable efficient scaling. Mixture-of-Experts (MoE) introduces sparsity, expanding total capacity at fixed active compute, and is now widely adopted by frontier models from cloud (DeepSeek-AI, 2024; Yang et al., 2025) to edge (Chen et al., 2026; Apple, 2026). Recently, looped transformers (Dehghani et al., 2018) introduce recurrence, reusing shared weights across passes to increase computational depth at fixed stored parameters and delivering substantial gains in reasoning (Saunshi et al., 2025; Geiping et al., 2025; Zhu et al., 2025; Jeddi et al., 2026). Looped MoE models naturally combine these complementary axes.

Yet the joint scaling behavior of recurrence and sparsity remains underexplored. Scaling laws for looped transformers capture recurrence, modeling its capacity gain with linear or power-law forms that grow without bound (Prairie et al., 2026; Schwethelm et al., 2026). MoE laws capture sparsity, varied through expert count, but omit recurrence (Clark et al., 2022; Ludziejewski et al., 2025; Abnar et al., 2025). Recent looped MoE studies examine sparsity while fixing recurrence at two passes in their primary scaling analyses (Lee et al., 2026; Wang et al., 2026), leaving the interaction between the two axes unmodeled. Consequently, a unified scaling law that jointly characterizes recurrence and sparsity in looped MoE models has yet to be established.

Establishing such a law requires understanding how recurrence and sparsity shape the effective capacity gain from looping. This raises two central questions: ***(Q1)*** *How much effective capacity does each recurrent pass add, and does this gain grow indefinitely?* ***(Q2)*** *Does greater MoE sparsity increase and sustain this gain across recurrent passes?* Two observations in Figure 1(b) point toward the answers. First, increasing recurrence shows diminishing returns: loss drops sharply over the first few passes, and then approaches a plateau within the observed range. Second, greater sparsity, varied by expert count E, improves and sustains this gain: with more experts, looping achieves lower loss over more passes. Figure 1(a) explains this interaction intuitively: in a dense looped transformer, every recurrent pass reuses the same weights; whereas in a looped MoE, the router can send a token to different experts across passes, allowing each pass to reach a new set of parameters.

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.

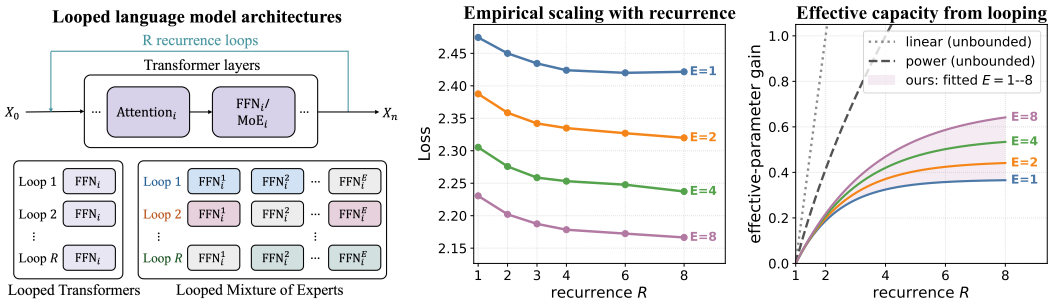


Figure 1: (a) Dense looped transformers reuse the same FFNs, whereas looped MoE models can route tokens to different experts across recurrent passes. (b) At fixed active parameters and training tokens, increasing recurrence yields diminishing returns, with the dense model (E=1) saturating earlier than MoE models (E>1). (c) Our Loop Scaling Laws model a bounded effective-parameter gain conditioned on recurrence R and sparsity (varied by expert count E), unlike prior linear and power-law mappings that assume unbounded gains.

To answer the above questions, we introduce *Loop Scaling Laws*, the first unified law over model size, data, recurrence, and sparsity. The law treats each recurrent pass as adding effective parameters with diminishing returns: each pass adds less than the last, with its total capacity gain approaching a finite asymptote rather than growing without bound. This asymptote also depends on sparsity: more experts raise and sustain the gain over more recurrent passes, so recurrence and sparsity interact within a single form, as illustrated in Figure 1(c). The law subsumes the standard dense, dense-looped, and non-looped MoE scaling laws as special cases. Empirically, it predicts held-out loss more accurately than prior alternatives with unbounded mappings, extrapolating well to unseen recurrence and expert counts.

Beyond prediction, the fitted law provides a principled recipe for looped MoE model design: it selects the compute- and memory-optimal recurrence and expert count under given budgets for resource-constrained deployment. Our empirical results across 14 downstream benchmarks further confirm the complementary gains of sparsity and recurrence. By scaling sparsity, an MoE model can surpass larger dense models with $1/3$ the active parameters, while scaling recurrence enables a looped model to match non-looped models with $1/2$ the total parameters. Jointly scaling both further advances the frontier beyond either axis alone. Together, these results establish joint scaling of recurrence and sparsity as a new parameter-efficient scaling paradigm. We extend this paradigm to practical, trillion-token training: at matched compute, a looped MoE model (0.3B active/1.3B total) with law-derived recurrence matches the reasoning performance of a larger non-looped MoE (0.6B active/2.9B total), trading additional inference compute for approximately $2\times$ parameter efficiency. We also show that the looped MoE unlocks on-demand test-time scaling by varying recurrence at inference.

2 Related Work

Looped transformers increase computational depth by repeatedly applying shared layers without proportional growth in stored parameters. Looped architectures have evolved from single-layer recurrence in Universal Transformers (Dehghani et al., 2018) to full-model (Zhu et al., 2025) and middle-block reuse for programmable, recursive, and latent-reasoning models (Giannou et al., 2023; Saunshi et al., 2025; Zeitoun et al., 2026; McLeish et al., 2025; Geiping et al., 2025). Training objectives range from final-pass next-token prediction loss (Saunshi et al., 2025; Geiping et al., 2025; McLeish et al., 2025) to intermediate-pass supervision (Bae et al., 2024), self-distillation (Goyal et al., 2026), and shortcut consistency (Jeddi et al., 2026). Prior work shows that recurrent depth provides parameter-efficient training- and test-time scaling, especially on reasoning tasks (Saunshi et al., 2025; Geiping et al., 2025; Zhu et al., 2025; Jeddi et al., 2026). Motivated by these findings, we model recurrence as an explicit scaling axis to characterize how performance scales with looping.

Mixture of experts (MoE) models increase total capacity without proportionally increasing per-token computation by routing each token to a sparse subset of expert networks (Shazeer et al., 2017; Fedus et al., 2022). MoE has become a well-established scaling paradigm across deployment regimes, from open-weight and proprietary frontier models deployed in the cloud, e.g., Mixtral (Jiang et al., 2024), DeepSeek-V3 (Dai et al., 2024; DeepSeek-AI, 2024), Qwen3MoE (Yang

Table 1: Comparison of scaling laws for training LLMs. ✓ indicates variables explicitly modeled by each law. N/F denotes model size N / training FLOPs F. S/E denotes MoE sparsity S (varied by expert expansion E).

		Scaling axes					Optimality	
LLM Category	**Scaling law**	N/F	D	R	S/E	**Recurrence mapping**	Compute	Memory
Dense	Chinchilla (2022)	✓	✓	×	×	–	✓	×
MoE	Unified routed (2022)	✓	×	×	✓	–	×	×
	Joint MoE (2025)	✓	✓	×	✓	–	✓	✓
Looped	Parcae (2026)	✓	✓	✓	×	linear; unbounded†	✓	×
	Iso-Depth (2026)	✓	✓	✓	×	power law; unbounded	✓	×
Looped MoE	Sparse Layers (2026)	✓	×	×	×	fixed R=2	✓	×
	SMELT (2026)	✓	✓	×	✓	fixed R=2	✓	×
	Loop scaling laws (ours)	✓	✓	✓	✓	**bounded; sparsity-dependent**	✓	✓

†Parcae fits a saturating law only over *test-time* depth; its trained-recurrence term uses the linear unrolled count.

et al., 2025), and Gemini (Gemini Team, 2024), to resource-constrained models deployed on edge devices, e.g., MobileMoE (Chen et al., 2026) and AFM 3 Core Advanced (Apple, 2026). Recent works have also explored MoE with looping (Lee et al., 2026; Wang et al., 2026), yet fix recurrence at two passes in their scaling analyses, thus leaving the joint scaling of recurrence and sparsity unmodeled. We address this gap by modeling how recurrence scales and interacts with sparsity.

Scaling laws characterize predictable power-law relationships among loss, model size, data, and compute, providing a principled foundation for compute-optimal language model training (Kaplan et al., 2020; Hoffmann et al., 2022). MoE scaling laws further incorporate routed expert count (Clark et al., 2022), expert granularity (Krajewski et al., 2024), sparsity (Abnar et al., 2025), enabling optimization under memory constraints (Ludziejewski et al., 2025), efficiency leverage (Tian et al., 2026), and on-device constraints (Chen et al., 2026). More recently, Parcae and Iso-Depth model recurrence for dense looped transformers (Prairie et al., 2026; Schwethelm et al., 2026), while SMELT analyzes looped MoEs with recurrence fixed in their primary scaling analyses (Wang et al., 2026). As summarized in Table 1, prior laws model either recurrence or sparsity, but not their joint effect. Our unified MoE loop scaling law jointly models model size, data, recurrence, and sparsity, while recovering the prior laws for dense and MoE models as equivalent reduced forms.

3 Loop Scaling Laws

3.1 Preliminaries

Scaling law. The standard Chinchilla-style scaling law (Hoffmann et al., 2022; Kaplan et al., 2020) models the relationship between model parameters N, training tokens D, and model loss $\mathcal{L}$ as

$$\mathcal{L}(N, D) = A\,N^{\alpha} + B\,D^{\beta} + c, \tag{1}$$

where A, B, α, β, c are fitted coefficients, $\alpha, \beta < 0$ are the model and data scaling exponents, c is the irreducible loss. Training compute is $F_{\text{train}}{=}6ND$, and inference compute is $F_{\text{inf}}{=}2N$ per token.

Parameter counts and compute. A looped model reuses a block of parameters N_{loop} over R recurrent passes with fixed model parameters N. In a forward pass, its unrolled parameter count is

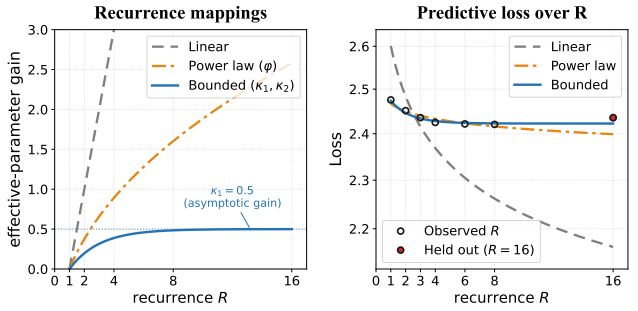
$$N_{\text{unroll}}(R) = N_{\text{act}} + (R-1)\,N_{\text{loop}}. \tag{2}$$

For dense looped models, the parameters are fixed as $N_{\text{act}}{=}N_{\text{total}}{=}N$. For looped MoE models, the active and total parameters N_{act} and N_{total} are fixed. In both cases, $R{=}1$ recovers the corresponding non-looped baseline. Under full backpropagation through all recurrent passes, the training and per-token inference compute are $F_{\text{train}} = 6\,N_{\text{unroll}}(R)\,D$ and $F_{\text{inf}} = 2\,N_{\text{unroll}}(R)$, respectively. While looping recurrence trades compute for quality at fixed parameters, MoE sparsity trades parameters for quality at fixed compute. We explore these complementary scaling axes in the following.

3.2 Loop Scaling Law

The standard scaling law depends only on model N and data D, and thus cannot capture the scaling behaviour of looping recurrence R. Prior scaling laws for looped transformers address this limitation by replacing N with a recurrence-dependent effective parameter count $N_{\text{eff}}(R)$. Parcae's training scaling law (Prairie et al., 2026) uses the fully unrolled parameter count: a linear mapping on R; whereas Iso-Depth (Schwethelm et al., 2026) learns a power-law mapping on R with exponent φ:

$$N_{\text{eff}}^{\text{linear}}(R) = N + (R-1)N_{\text{loop}}, \qquad N_{\text{eff}}^{\text{power}}(R) = N + (R^{\varphi}-1)N_{\text{loop}}. \tag{3}$$



Recurrence mappings for N_{eff}

Mapping	$N_{\text{eff}} - N$	$R \to \infty$
Linear	$(R-1)N_{\text{loop}}$	∞
Power law	$(R^{\varphi}-1)N_{\text{loop}}$	∞
Bounded	$\kappa_1 N_{\text{loop}}\left(1-e^{-(R-1)/\kappa_2}\right)$	$\kappa_1 N_{\text{loop}}$

Held-out prediction RMSE (↓)

Evaluation	Linear	Power law	Bounded
Held-out R	0.2566	0.0313	**0.0092**
Held-out N	0.1106	0.0211	**0.0128**
Held-out D	0.0882	0.0073	**0.0049**

(c)

Figure 2: (a) Illustration of effective-parameter gain under linear, power-law, bounded recurrence mappings. (b) Predictive loss by different recurrence mappings: each mapping is fitted on $R \leq 8$ and applied to predict the held-out-R up to R=16. (c) Formula on recurrence mappings and their effective-parameter gains (top) and their evaluation on held-out recurrence R, model N, and data D with RMSE scores (bottom).

$N_{\text{eff}}^{\text{linear}}(R)$ assigns a constant effective-parameter gain N_{loop} per recurrence; $N_{\text{eff}}^{\text{power}}(R)$ yields diminishing gains with $0<\varphi<1$. Nevertheless, both mappings are unbounded: $\lim_{R\to\infty} N_{\text{eff}}(R) = \infty$, where increasing recurrence substitutes for parameters indefinitely (Figure 2(a)).

A bounded recurrence mapping. However, increasing recurrence yields diminishing returns. Empirically, the model loss drops substantially over the earlier passes, and flattens gradually within the evaluated range (Figure 2(b)). This behavior is consistent with weight sharing: each additional pass increases computational depth without introducing new learned parameters, leading to diminishing marginal gains. We therefore introduce the following bounded monotone mapping:

$$N_{\text{eff}}^{\text{bounded}}(R) = N + \kappa_1 N_{\text{loop}}\left(1 - e^{-(R-1)/\kappa_2}\right), \qquad \kappa_1 > 0,\ \kappa_2 > 0 \tag{4}$$

where κ_1 sets the asymptotic effective-parameter gain from looping: $\kappa_1 N_{\text{loop}}$, while κ_2 controls how quickly this limit is approached as R increases.

Boundary properties. The bounded mapping in Eq. 4 recovers the non-looped baseline at R=1, and approaches a finite effective-parameter bound as $R\to\infty$:

$$N_{\text{eff}}^{\text{bounded}}(1) = N, \qquad \lim_{R\to\infty} N_{\text{eff}}^{\text{bounded}}(R) = N + \kappa_1 N_{\text{loop}}. \tag{5}$$

Thus, looping contributes at most $\kappa_1 N_{\text{loop}}$ additional effective parameters, an asymptotic property of the mapping rather than a guarantee beyond the observed recurrence range.

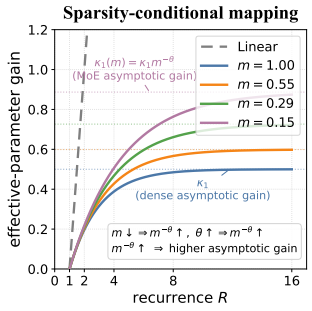
Loop scaling law. For a looped transformer with R recurrence loops, we replace the model-size term N in standard scaling law Eq. 1 with a recurrence-dependent effective parameter count $N_{\text{eff}}(R)$:

$$\mathcal{L}(N, D, R) = A\, N_{\text{eff}}(R)^{\alpha} + B\, D^{\beta} + c, \tag{6}$$

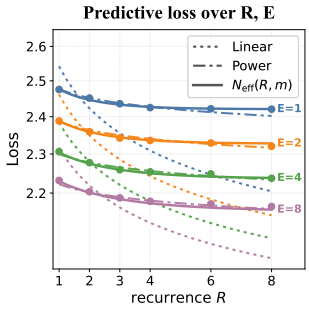
where $N_{\text{eff}}(R)$ introduces recurrence-specific coefficients alongside the base scaling-law coefficients $\{A, \alpha, B, \beta, c\}$: in Eq. 3, the linear mapping uses linear constant and the power-law mapping uses φ, while our bounded mapping in Eq. 4 uses $\{\kappa_1, \kappa_2\}$. Figure 2(a) illustrates the normalized effective-parameter gain $(N_{\text{eff}}(R) - N)/N_{\text{loop}}$ under these mappings. The effective parameter count $N_{\text{eff}}(R)$ is distinct from the unrolled parameter count $N_{\text{unroll}}(R)$: the latter determines compute, whereas the former models the effective parameter capacity attributed to weight-tied recurrence.

Reduced form $\mathcal{L}|_{R=1}$**.** For any recurrence mapping satisfying $N_{\text{eff}}(1)$=N, the loop scaling law in Eq. 6 reduces to the standard non-looped scaling law in Eq. 1 at R=1 (as derived in Proposition A.1).

Comparison of recurrence mappings. We derive the fitted laws with the same parametric fitting and experimental sweep over (R, N, D), as detailed in Appendix C.3. Figure 2(b) compares the predictive loss over recurrence for loop scaling laws under different recurrence mappings, where each law is fitted on the sweep over N, D, $R \leq 8$ and extrapolated to the held-out R=16. The linear mapping substantially overestimates the effective-parameter gain, while the power-law mapping captures diminishing gains but remains overly optimistic beyond the fitted range. In contrast, the bounded mapping captures the loss plateau. The held-out evaluation over unseen N, D, R in Figure 2(c) further shows the law with bounded recurrence mapping achieves the lowest held-out RMSE, indicating it better captures the scaling trends across model size, data, and recurrence.



(a)



(b)

MoE recurrence mappings for N_{eff}

Mapping	$N_{\text{eff}} - N_{\text{act}}$
Linear	$(R-1)N_{\text{loop}}$
Power law	$(R^{\varphi}-1)N_{\text{loop}}$
Bounded $N_{\text{eff}}(R)$	$\kappa_1 N_{\text{loop}}\left(1-e^{-(R-1)/\kappa_2}\right)$
Bounded $N_{\text{eff}}(R,m)$	$\kappa_1(m)N_{\text{loop}}\left(1-e^{-(R-1)/\kappa_2(m)}\right)$

Held-out prediction RMSE (↓)

Evaluation	Linear	Power	$N_{\text{eff}}(R)$	$N_{\text{eff}}(R,m)$
Held-out R	0.2847	0.0480	0.0190	**0.0100**
Held-out E	0.0896	0.0113	0.0075	**0.0047**
Held-out N	0.0656	0.0077	0.0065	**0.0050**
Held-out D	0.0824	0.0074	0.0058	**0.0043**

(c)

Figure 3: (a) Illustration of effective-parameter gain with sparsity-conditional recurrence mapping. (b) Predictive loss under different MoE recurrence mappings fitted on runs with $R{\leq}8$; markers denote observations used for fitting. (c) Formula on MoE recurrence mappings and their effective-parameter gains (top), and their evaluation on held-out recurrence $R{=}16$, expert count E, model size N, and data D with RMSE (bottom).

3.3 MoE Loop Scaling Law

The loop scaling law in Section 3.2 assumes dense looped models. In looped MoE models, sparse routing may select different experts across recurrent passes, allowing tokens to traverse different parameter paths despite weight sharing (Figure 1(a)). Motivated by this expert-path diversity (see analysis in Appendix B), we introduce MoE sparsity as a condition for recurrence mapping, allowing sparsity to modulate the effective-parameter gain from looping.

A sparsity-conditional recurrence mapping. Let $m = N_{\text{act}}/N_{\text{total}} \in (0, 1]$ denote the MoE active-parameter ratio, where smaller m indicates greater sparsity and $m{=}1$ is the dense case. For top-k routing over E_{route} experts, $m \approx k/E_{\text{route}}$. We extend the dense bounded mapping in Eq. 4 to looped MoE models by replacing N with N_{act} and parameterizing its coefficients as functions of m:

$$N_{\text{eff}}(R, m) = N_{\text{act}} + \kappa_1(m)N_{\text{loop}}\left(1 - e^{-(R-1)/\kappa_2(m)}\right), \quad \kappa_j(m) = \kappa_j m^{-\theta}\ (j = 1, 2). \tag{7}$$

The fitted sparsity-scaling exponent $\theta \geq 0$ quantifies how strongly sparsity amplifies the parameter gain from recurrence, with $\theta = 0$ recovering a sparsity-independent recurrence mapping. As sparsity increases (m decreases), the factor $m^{-\theta}$ lifts the recurrence curve vertically through $\kappa_1(m)$ and stretches it horizontally through $\kappa_2(m)$, thus raising the asymptotic effective-parameter gain that is approached over more recurrent passes, as illustrated in Figure 3(a).

Boundary properties. The sparsity-conditional mapping in Eq. 7 recovers the non-looped baseline at $R{=}1$, and approaches a finite effective-parameter bound as $R{\to}\infty$. It also recovers the dense bounded mapping in Eq. 4 at $m{=}1$, and approaches a linear-in-R limit as $m{\to}0$:

$$\begin{aligned} &N_{\text{eff}}(1, m) = N_{\text{act}}, && \lim_{R\to\infty} N_{\text{eff}}(R, m) = N_{\text{act}} + \kappa_1 m^{-\theta} N_{\text{loop}}, \\ &N_{\text{eff}}(R, 1) = N_{\text{eff}}^{\text{bounded}}(R), && \lim_{m\to 0} N_{\text{eff}}(R, m) = N_{\text{act}} + \frac{\kappa_1}{\kappa_2}(R-1)N_{\text{loop}}, \quad \theta > 0. \end{aligned} \tag{8}$$

Thus, at any fixed $m{>}0$, looping contributes at most $\kappa_1 m^{-\theta} N_{\text{loop}}$ additional effective parameters; in the extreme-sparsity limit $m \to 0$ (expert count $E \to \infty$), the gain grows linearly with R. These asymptotic limits are properties rather than guarantees beyond the observed recurrence range.

MoE loop scaling law. For a looped MoE model with R recurrence loops and active-parameter ratio m, we replace the active-parameter term N_{act} in MoE scaling law (Clark et al., 2022; Ludziejewski et al., 2025) with the sparsity-conditional effective parameter count $N_{\text{eff}}(R, m)$ in Eq. 7:

$$\mathcal{L}(N_{\text{act}}, D, R, E, m) = A\,\hat{E}^{\,\delta}\,N_{\text{eff}}(R, m)^{\,\alpha+\gamma\ln\hat{E}} + B\,\hat{E}^{\,\omega}\,D^{\,\beta+\zeta\ln\hat{E}} + c, \tag{9}$$

where $N_{\text{eff}}(R, m)$ introduces the fitted coefficients $\{\kappa_1, \kappa_2, \theta\}$ alongside the base MoE scaling-law coefficients $\{A, \alpha, B, \beta, c, \delta, \gamma, \omega, \zeta, E_{\text{start}}, E_{\text{max}}\}$. Following Clark et al. (2022), $\hat{E}$ is a monotonic transformation: $\hat{E}^{-1} = \left(E - 1 + \left(E_{\text{start}}^{-1} - E_{\text{max}}^{-1}\right)^{-1}\right)^{-1} + E_{\text{max}}^{-1}$, where E denotes the number of experts under top-1 routing (Clark et al., 2022; Ludziejewski et al., 2025). For top-k routing with E_{route} routed experts, the expert count generalizes to $E = E_{\text{route}}/k$.

Reduced forms. The MoE loop scaling law in Eq. 9 subsumes prior scaling laws as special cases: (1) $\mathcal{L}|_{E=1}$, at $E = 1$, it recovers the dense loop scaling law in Eq. 6 after reparameterization; (2)

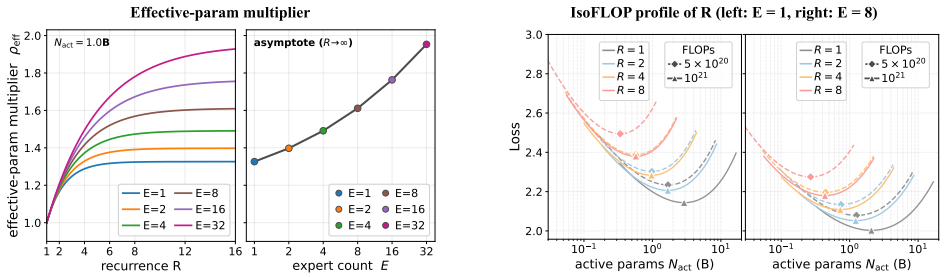


Figure 4: (a,b) Effective-parameter multiplier over recurrence at N_{act}=1.0B, and its asymptote over expert count. (c,d) Recurrence IsoFLOP profiles for dense (E=1) and MoE (E=8) models; markers denote minima.

$\mathcal{L}|_{R=1}$, at $R = 1$, it recovers the standard MoE scaling law; and (3) $\mathcal{L}|_{R=1,E=1}$, at $E = 1, R = 1$, it recovers the standard dense scaling law in Eq. 1. Full derivations are given in Proposition A.2.

Comparison of MoE recurrence mappings. We derive the fitted MoE loop scaling laws using the same parametric fitting and experimental sweep over $(R, N_{\text{act}}, D, E)$, as detailed in Appendix C.3. Figure 3(b) illustrates the predictive loss over recurrence R and MoE sparsity varied by expert count E, where each law is fitted on the sweep over varying N_{act}, D, E, and $R \leq 8$. The same fitted laws are evaluated on held-out recurrence R=16 in Figure 3(c). As observed in the dense comparison, the linear mapping overestimates recurrence gains and the power-law mapping remains optimistic; importantly, neither captures how saturation varies with sparsity. In contrast, our sparsity-conditional bounded mapping tracks the loss plateau across expert counts. Figure 3(c) further shows the law with sparsity-conditional bounded mapping in Eq. 7 achieves the lowest held-out RMSE across all held-out axes, as compared to the laws with linear, power-law in Eq. 3 and sparsity-independent mappings Eq. 4. This confirms our formulated MoE loop scaling law in Eq.9 has a better predictive fit and captures how sparsity modulates the asymptotic gain from recurrence.

4 Experiments and Findings

4.1 Scaling Experiments and Analysis

Scaling experiments and fitted law. We jointly sweep over four scaling axes: model parameters $N_{\text{act}} \in \{0.3, 0.6, 1.0\}$B, training tokens $D \in \{100, 200, \ldots, 500\}$B, expert count $E \in \{1, 2, 4, 8, 16\}$, and recurrence $R \in \{1, 2, 3, 4, 6, 8\}$. We use standard middle-cycle looping (Geiping et al., 2025) to loop over the middle block, leaving the first two prelude layers and last two coda layers unshared. As our formulation is agnostic to specific looping strategies, we leave other strategies to future work. The full training and law fitting details are provided in Appendix C.

How do recurrence and sparsity improve effective capacity from looping? To quantify the effective capacity gain from the fitted MoE loop scaling law, we define the effective-parameter multiplier:

$$\rho_{\text{eff}}(R, m) = \frac{N_{\text{eff}}(R, m)}{N_{\text{act}}} = 1 + \kappa_1(m) \frac{N_{\text{loop}}}{N_{\text{act}}} \left(1 - e^{-(R-1)/\kappa_2(m)}\right). \tag{10}$$

Its asymptote is $\rho_{\text{eff}}(\infty, m) = 1 + \kappa_1(m) N_{\text{loop}}/N_{\text{act}}$, which characterizes the maximum effective-parameter multiplier relative to N_{act}. Figure 4(a) shows that, at fixed N_{act}, the effective-parameter multiplier increases with larger recurrence but approaches a finite asymptote, while greater sparsity raises this asymptote and delays saturation over recurrent passes; Figure 4(b) shows this asymptote itself rises with expert count. Alternatively, this gain can be measured relative to the recurrent-block parameters N_{loop} as the effective-parameter gain $g_{\text{eff}} = (N_{\text{eff}} - N_{\text{act}})/N_{\text{loop}}$ (see Figures 1(c) and 3(a)), which yields the same scaling trends over recurrence and sparsity, as summarized below.

Finding: Scaling recurrence adds effective-parameter gain up to a finite, sparsity-dependent asymptote, while scaling sparsity raises and sustains this gain over more recurrent passes.

How do recurrence and sparsity push the IsoFLOP frontier? We examine the scaling behavior over recurrence and sparsity on an IsoFLOP basis based on the fitted law. Figures 4(c,d) show predicted loss against N_{act} under fixed training compute: 5×10^{20} and 10^{21} FLOPs. When increasing sparsity from E=1 (dense) to E=8 (MoE), i.e., comparing Figure 4(c) and (d), lower loss is attained under

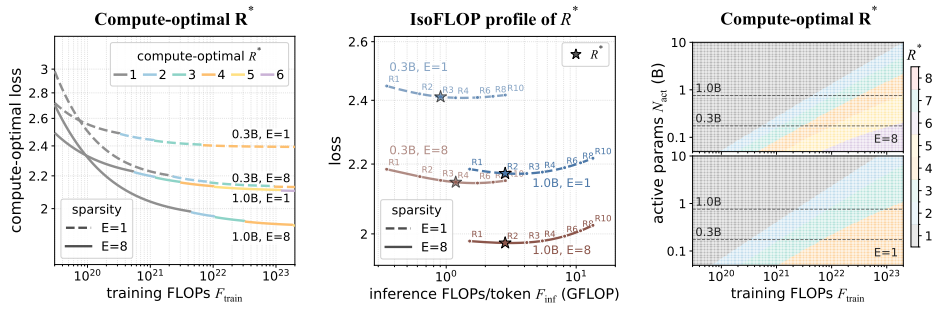


Figure 5: Predicted compute-optimal recurrence $R^\star$ of dense (E=1) and MoE (E=8) models. (a) Recurrence with compute-optimal loss across training-compute budgets. (b) IsoFLOP profiles over recurrence at $\bar{F}_{\text{train}}$=5×10^{21} FLOPs; stars mark $R^\star$. (c) $R^\star$ across compute and active model size for E=8 (top) and E=1 (bottom).

the same compute budget, indicating that higher sparsity pushes the IsoFLOP performance frontier. When increasing recurrence at fixed sparsity, higher recurrence can outperform lower recurrence at the same active parameter count N_{act} with additional compute; for example, R=2 at 10^{21} FLOPs achieves lower loss than R=1 at 5×10^{20} FLOPs for both E=1 and E=8, as summarized below.

Finding: Higher sparsity improves the IsoFLOP frontier at matched compute, while recurrence pushes this frontier further at the same parameter size when given more compute.

4.2 Compute- and Memory-Optimal Recurrence and Sparsity

Given the fitted law, which recurrence and sparsity should be selected under resource constraints? Our MoE loop scaling law provides a principled foundation for optimizing architectures under practical resource constraints of training compute, and deployment memory. At fixed weight memory, recurrence R trades additional compute for better performance; whereas at fixed compute, sparsity trades additional weight memory for improved performance. These complementary tradeoffs motivate our following analyses that apply the fitted law to predict the compute-optimal recurrence at fixed sparsity, memory-optimal sparsity at fixed recurrence, and their joint optimum

Compute-optimal recurrence at fixed sparsity. At fixed model sparsity (fixed expert count E), increasing recurrence R leaves the model weights N_{total} and weight memory $\mathcal{M}_{\text{weight}}$ unchanged, but increases the training compute through unroll parameters $N_{\text{unroll}}(R)$. Therefore, recurrence can be compute-optimal if it achieves the minimal loss among recurrences trained under the same compute, while providing a meaningful loss reduction over the preceding recurrence:

$$R^\star = \arg\min_R \mathcal{L}(N_{\text{act}}, D, R, E, m), \qquad \text{s.t.} \quad 6N_{\text{unroll}}(R)D = \bar{F}_{\text{train}}, \quad \Delta\mathcal{L}(R) \geq \epsilon. \tag{11}$$

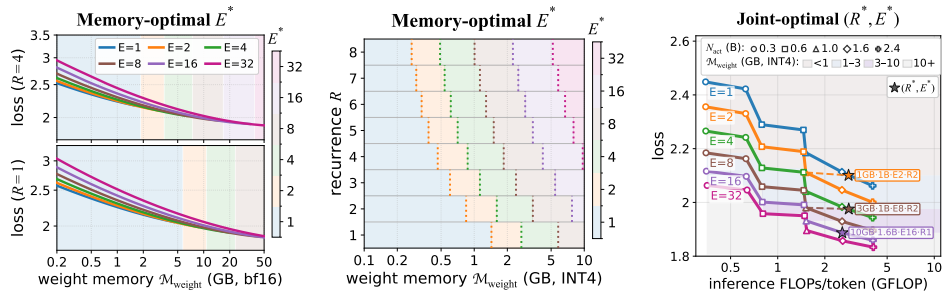
$\bar{F}_{\text{train}}$ is the given training-compute budget. For $R \geq 2$, $\Delta\mathcal{L}(R) = \mathcal{L}_{R-1} - \mathcal{L}_R$ measures the predicted loss reduction from the additional recurrent pass $R{-}1{\to}R$. The tolerance ϵ specifies the minimum meaningful loss reduction. In practice, we set ϵ to the fitted-law RMSE, below which predicted improvements cannot be reliably distinguished from fitting error. If no pass satisfies $\Delta\mathcal{L}(R) \geq \epsilon$, $R^\star$=1. Figure 5 shows three trends on compute-optimal $R^\star$: (a) $R^\star$ increases with more training-compute budget and is higher for smaller active models; (b) along each IsoFLOP profile, the selected $R^\star$ lies at or near the loss minimum, with subsequent recurrent passes providing negligible gains; and (c) across the F_{train}–N_{act} design space, greater sparsity shifts the compute-optimal regime toward higher recurrence. These trends can be summarized below.

Finding: Larger training compute budgets and greater sparsity favor higher recurrence.

Memory-optimal sparsity at fixed recurrence. At fixed recurrence R, increasing sparsity through expert count E expands the total parameters N_{total} and thus weight memory, while leaving the active compute unchanged. Therefore, sparsity can be memory-optimal by selecting the expert count E that achieves the minimal loss when trained under the same compute:

$$E^\star = \arg\min_E \mathcal{L}(N_{\text{act}}, D, R, E, m), \quad \text{s.t. } F_{\text{train}} = \bar{F}_{\text{train}}, \ \mathcal{M}_{\text{weight}} \leq M_{\text{budget}}, \ \Delta\mathcal{L}(E) \geq \epsilon. \tag{12}$$

M_{budget} is the weight-memory budget and $\mathcal{M}_{\text{weight}} = b_w N_{\text{total}}/8$, where b_w is the weight precision in bits. Total deployment memory may include KV cache and other runtime state, but their optimiza-



(a) (b) (c)

Figure 6: Predicted memory-optimal expert count $E^\star$ and joint optimum $(N^\star_{\text{act}}, E^\star, R^\star)$ at $\bar{F}_{\text{train}}{=}5{\times}10^{21}$ FLOPs. (a) $E^\star$ across bf16 weight-memory budgets at $R{=}4$ (top) and $R{=}1$ (bottom). (b) $E^\star$ across INT4 weight memory and recurrence. (c) Predicted joint optimum $(N^\star_{\text{act}}, E^\star, R^\star)$ across INT4 weight memory.

tion is outside our scope; we therefore consider weight memory only. For expert counts $\{E_i\}$ and $i{\geq}2$, $\Delta\mathcal{L}(E_i) = \mathcal{L}_{E_{i-1}} - \mathcal{L}_{E_i}$ measures the loss reduction from expert expansion $E_{i-1}{\to}E_i$. The same tolerance ϵ as in Eq. 11 requires each expansion yields a meaningful loss reduction. If no expansion satisfies $\Delta\mathcal{L}(E_i) \geq \epsilon$, $E^\star{=}E_1{=}1$. Figure 6(a) and (b) show two trends on memory-optimal $E^\star$: (a) $E^\star$ increases with more memory budget, while higher recurrence favors larger $E^\star$ under the same memory; and (b) across the $\mathcal{M}_{\text{weight}}{-}R$ design space, larger memory budgets and higher recurrence shift the optimal regime toward greater sparsity. These trends are summarized below.

Finding: Larger weight memory budgets and higher recurrence favor greater sparsity.

Joint optimum on recurrence and sparsity. Given a training-compute budget and a memory budget, the recurrence R and sparsity (varied by expert count E) can be jointly compute- and memory-optimal if they achieve the minimal loss under the same resource constraints:

$$(N^\star_{\text{act}}, E^\star, R^\star) = \arg\min_{N_{\text{act}}, E, R} \mathcal{L}(N_{\text{act}}, D, R, E, m) \tag{13}$$
$$\text{s.t. compute: } 6N_{\text{unroll}}(R)D = \bar{F}_{\text{train}},\ \Delta\mathcal{L}(R) \geq \epsilon;\ \text{memory: } \mathcal{M}_{\text{weight}} \leq M_{\text{budget}},\ \Delta\mathcal{L}(E) \geq \epsilon.$$

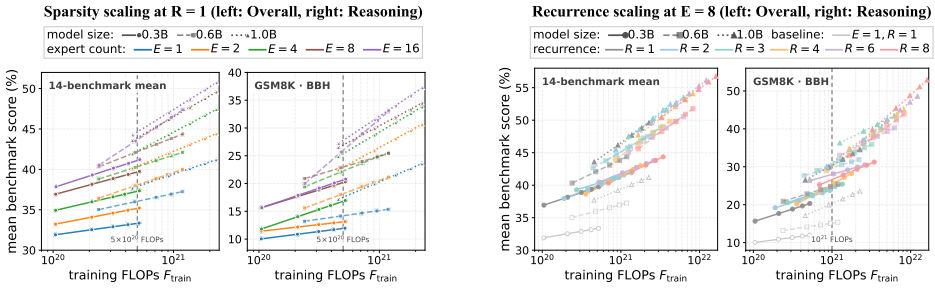
Given a training compute budget $\bar{F}_{\text{train}}$ and weight memory budget M_{budget}, we use the fitted scaling law to identify the predicted optimal looped MoE configuration from a model ladder spanning active model size, expert count, and recurrence (Appendix C.2), which guides model design under resource constraints. For each candidate configuration (N_{act}, E, R), we set $D = \bar{F}_{\text{train}}/[6N_{\text{unroll}}(R)]$, retain candidates satisfying the compute and memory constraints, and select the optimum with lowest predicted loss. Figure 6(c) shows that under fixed training compute, tighter memory favors higher recurrence and smaller expert counts, e.g., selecting a 1.0B active model with $E{=}8$ and $R{=}2$ at 3 GB; whereas larger memory favors greater sparsity and less recurrence, e.g., selecting a 1.6B active model with $E{=}16$ and $R{=}1$ at 10 GB, as summarized below.

Finding: At fixed compute, tight memory favors recurrence, and more memory favors sparsity.

4.3 Downstream Scaling with Recurrence and Sparsity

Empirical scaling on downstream tasks. We extend our scaling analysis of looped MoE models to downstream tasks, covering 14 benchmarks, and report *Overall* as the mean across all tasks and *Reasoning* as the mean of GSM8K and BBH. Full evaluation protocols are provided in Appendix D.1.

How do recurrence and sparsity translate into downstream performance? Figure 7 shows the two scaling axes provide distinct but complementary gains. In panels (a,b), increasing sparsity at fixed $R{=}1$ consistently improves both Overall and Reasoning across all three model sizes. These sparsity gains can offset a $\sim$3$\times$ increase in active model size: at $5{\times}10^{20}$ FLOPs, the 0.3B MoE models with $E{\geq}8$ outperform the larger 1.0B dense model on both Overall and Reasoning. In panels (c,d), increasing recurrence at fixed $E{=}8$ produces more reasoning-oriented gains. These recurrence gains can offset a $\sim$2$\times$ increase in total parameters on Reasoning: at 10^{21} FLOPs, the 0.3B model with $R{\geq}4$ matches or exceeds the 0.6B model with $R{=}1$, while the 0.6B model with $R{\geq}3$ matches the 1.0B model with $R{=}1$. Moreover, models that jointly increase sparsity and recurrence ($E{=}8, R{>}1$) consistently outperform the dense baseline ($E{=}1, R{=}1$) at matched training compute, showing that their gains accumulate when scaled together. Thus, sparsity and recurrence provide complementary



(a) (b) (c) (d)

Figure 7: Downstream scaling of sparsity and recurrence. (a,b) Scaling E at R=1; (c,d) Scaling R at E=8, with the dense baseline E=1, R=1. (a,c) report *Overall* performance, and (b,d) report *Reasoning* performance; dashed lines mark compute-matched comparisons.

Table 2: Downstream results of A0.6B-2.9B MoE and A0.3B-1.3B LoopMoE across test-time recurrence R, trained at matched compute FLOPs $\approx 1.5\times10^{22}$. Relative inference compute is normalized to the non-looped baseline. Δ reports gains over LoopMoE at R=1. Evaluation protocols are provided in Appendix D.1.

Model	R	$\frac{F_{\text{inf}}^{0.3B}(R)}{F_{\text{inf}}^{0.6B}(1)}$	Reasoning		Science			Commonsense Reasoning				Reading		Knowledge			Overall (Δ)
			BBH[3]	GSM8K[8]	ARC-C[25]	ARC-E	OBQA	HS	PIQA	SIQA	Wino	BoolQ	DROP[3]	MMLU[5]	NQ[5]	TQA[5]	
A0.6B-2.9B MoE	1	1.0×	29.8	42.9	50.0	66.9	40.8	68.1	77.2	51.6	64.4	70.1	39.7	46.9	13.8	39.8	50.2
A0.3B-1.3B LoopMoE	1	0.4×	18.6	10.5	36.1	57.3	33.8	52.7	70.7	43.8	54.6	58.1	23.0	29.2	5.7	18.2	36.6
	2	0.8×	26.7	30.0	42.2	60.3	38.2	59.9	72.5	46.4	58.5	62.8	33.7	37.6	8.3	24.1	43.0 (+6.4)
	3	1.1×	30.1	41.0	44.9	60.9	39.6	62.4	73.2	50.2	61.3	65.9	38.4	41.6	8.9	25.6	46.0 (+9.4)
	4	1.5×	31.5	41.5	46.5	63.9	39.0	62.9	73.9	51.1	60.6	64.4	39.3	42.2	9.8	27.1	46.7 (+10.1)
	5	1.8×	31.8	41.2	47.5	64.3	40.4	63.1	74.0	51.1	61.0	64.3	39.0	42.2	9.9	27.0	46.9 (+10.3)

routes to parameter-efficient scaling, with sparsity achieving $\sim3\times$ active-parameter efficiency on overall performance and recurrence achieving $\sim2\times$ total-parameter efficiency on reasoning.

How does a looped MoE compare to a $2\times$ larger non-looped MoE at near-matched compute? We explore looped MoE at trillion-token training, and apply our recurrence selection criterion (Eq. 11) to derive the recurrence for an A0.3B-1.3B LoopMoE (0.3B active/1.3B total, E=8), and compare it with a larger A0.6B-2.9B MoE (0.6B active/2.9B total, E=8, R=1) as the non-looped reference (derivation and training details are in Appendix D.2). Table 2 compares A0.3B-1.3B LoopMoE and A0.6B-2.9B MoE at matched compute. Despite using $\sim2\times$ fewer active and total parameters, A0.3B-1.3B LoopMoE at $R \in \{4, 5\}$ matches A0.6B-2.9B MoE on reasoning tasks by using more inference compute, while trailing on *Overall*. Beyond parameter efficiency, LoopMoE supports test-time scaling: varying recurrence from R=1 to R=5 raises inference compute from low to high, boosting *Overall* performance on demand by a substantial $+10.3$ points ($36.6 \rightarrow 46.9$; Table 2).

5 Conclusion

We introduced *Loop Scaling Laws*, a unified predictive law that jointly models recurrence and sparsity alongside model size and data, closing a pressing gap between MoE and looped model scaling theories. Its bounded, sparsity-conditional mapping captures the diminishing recurrence gains observed within the evaluated range and models how sparsity raises the fitted capacity. The fitted law guides looped MoE model design by selecting recurrence and sparsity under compute and memory constraints, providing practical recipes for resource-constrained settings.

More broadly, our work reframes the scaling design space, positioning recurrence and sparsity as complementary axes for the next-generation parameter-efficient scaling paradigm. As effective scaling relies not only on how many model parameters are added, but also on how effectively additional computation can use them, these two axes should be co-designed according to resource constraints. By combining both axes properly, looped MoE models can advance the performance–efficiency frontier while enabling on-demand test-time scaling through recurrence.

To unlock the greater potential of jointly scaling recurrence and sparsity, we highlight several future directions. First, advanced looping strategies may be explored to enrich recurrent latent representations and boost the effective capacity gain. Second, efficient scaling of looped MoE models could be generalized across broader scales, spanning from small models on the edge to large models in the cloud. Third, the memory and latency of looped inference could be further improved by optimizing recurrent states or adding early-exit gating, thus scaling recurrence only when necessary.

AI USE STATEMENT

AI tools were used to polish part of the manuscript. They also assisted with part of the experiments, e.g., detecting and restarting failed runs on our infrastructure and executing evaluations according to the author-defined workflows. The research ideas, methodology, and experimental design were developed by the authors. AI-assisted outputs were reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

We provide comprehensive details for reproducing our findings and results. Appendix A provides derivations of the reduced forms of loop scaling laws. Appendix C documents details of scaling experiments and parametric fitting. Appendix D gives details of evaluation setups and experiments on downstream benchmarks.

REFERENCES

Samira Abnar, Harshay Shah, Dan Busbridge, Alaaeldin Mohamed Elnouby Ali, Josh Susskind, and Vimal Thilak. Parameters vs FLOPs: Scaling laws for optimal sparsity for mixture-of-experts language models. *arXiv preprint arXiv:2501.12370*, 2025.

Apple. Introducing the third generation of Apple's foundation models. Apple Machine Learning Research, June 2026. URL `https://machinelearning.apple.com/research/introducing-third-generation-of-apple-foundation-models`. Published June 8, 2026.

Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. Relaxed recursive transformers: Effective parameter sharing with layer-wise LoRA. *arXiv preprint arXiv:2410.20672*, 2024.

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.

Yanbei Chen, Hanxian Huang, Ernie Chang, Jacob Szwejbka, Digant Desai, Zechun Liu, Vikas Chandra, and Raghuraman Krishnamoorthi. MobileMoE: Scaling on-device mixture of experts. *arXiv preprint arXiv:2605.27358*, 2026.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *International conference on machine learning*, 2022.

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 NAACL-HLT*, 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.

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.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal Transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of NAACL-HLT*, 2019.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 2022.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 2024.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025.

Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. *arXiv preprint arXiv:2301.13196*, 2023.

Sahil Goyal, Swayam Agrawal, Gautham Govind Anil, Prateek Jain, Sujoy Paul, and Aditya Kusupati. ELT: Elastic looped transformers for visual generation. *arXiv preprint arXiv:2604.09168*, 2026.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 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. *arXiv preprint arXiv:2203.15556*, 2022.

Ahmadreza Jeddi, Marco Ciccone, and Babak Taati. LoopFormer: Elastic-depth looped transformers for latent reasoning via shortcut modulation. *arXiv preprint arXiv:2602.11451*, 2026.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

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.

Jakub Krajewski, Jan Ludziejewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, et al. Scaling laws for fine-grained mixture of experts. *arXiv preprint arXiv:2402.07871*, 2024.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 2019.

Ryan Lee, Jacob Biloki, Edward J. Hu, and Jonathan May. Sparse layers are critical to scaling looped language models. *arXiv preprint arXiv:2605.09165*, 2026.

Jan Ludziejewski, Maciej Pióro, Jakub Krajewski, Maciej Stefaniak, Michał Krutul, Jan Małaśnicki, Marek Cygan, Piotr Sankowski, Kamil Adamczewski, Piotr Miłoś, et al. Joint MoE scaling laws: Mixture of experts can be memory efficient. *arXiv preprint arXiv:2502.05172*, 2025.

Sean McLeish, Ang Li, John Kirchenbauer, Dayal Singh Kalra, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Jonas Geiping, Tom Goldstein, and Micah Goldblum. Teaching pretrained language models to think deeper with retrofitted recurrence. *arXiv preprint arXiv:2511.07384*, 2025.

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*, 2018.

Hayden Prairie, Zachary Novack, Taylor Berg-Kirkpatrick, and Daniel Y. Fu. Parcae: Scaling laws for stable looped language models. *arXiv preprint arXiv:2604.12946*, 2026.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of EMNLP-IJCNLP*, 2019.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. *arXiv preprint arXiv:2502.17416*, 2025.

Kristian Schwethelm, Daniel Rueckert, and Georgios Kaissis. How much is one recurrence worth? Iso-Depth scaling laws for looped language models. *arXiv preprint arXiv:2604.21106*, 2026.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.

Changxin Tian, Kunlong Chen, Jia Liu, Ziqi Liu, Zhiqiang Zhang, and Jun Zhou. Towards greater leverage: Scaling laws for efficient MoE language models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7r2lkhDGUj.

Shaowen Wang, Ge Zhang, Kairong Luo, Yuhao Wu, Shaofan Liu, Jiaheng Liu, Wenhao Huang, Shen Yan, and Jian Li. SMELT: Scaling laws for compute-matched MoE looped transformers. *arXiv preprint arXiv:2609.01343*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Abbas Zeitoun, Lucas Torroba-Hennigen, and Yoon Kim. Hyperloop Transformers. *arXiv preprint arXiv:2604.21254*, 2026.

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*, 2019.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, Lu Li, Jiajun Shi, Kaijing Ma, Shanda Li, Taylor Kergan, Andrew Smith, Xingwei Qu, Mude Hui, Bohong Wu, Qiyang Min, Hongzhi Huang, Xun Zhou, Wei Ye, Jiaheng Liu, Jian Yang, Yunfeng Shi, Chenghua Lin, Enduo Zhao, Tianle Cai, Ge Zhang, Wenhao Huang, Yoshua Bengio, and Jason Eshraghian. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

APPENDIX

A REDUCED FORMS OF LOOP SCALING LAWS

A.1 DENSE LOOP SCALING LAW

Proposition A.1 (Dense loop scaling law equivalence)**.** *At R=1, the dense loop scaling law recovers the standard non-looped scaling law.*

Proof.
Reduced form $\mathcal{L}|_{R=1}$**.** With recurrence fixed at R=1, the recurrence mapping $N_{\text{eff}}(R)$ (Eq. 4) recovers the non-looped baseline: $N_{\text{eff}}(1) = N$. Specifically,

$$N_{\text{eff}}(R)|_{R=1} = N + \kappa_1 N_{\text{loop}} \left(1 - e^{-(1-1)/\kappa_2}\right) = N.$$

The loop scaling laws reduce to

$$\begin{aligned} \mathcal{L}(N, D, R)|_{R=1} &= A\, N_{\text{eff}}(1)^{\alpha} + B\, D^{\beta} + c \\ &= A\, N^{\alpha} + B\, D^{\beta} + c \\ &= \mathcal{L}(N, D). \end{aligned} \tag{14}$$

This reduced form is equivalent to the standard non-looped scaling law in Eq. 1 (Hoffmann et al., 2022; Kaplan et al., 2020). □

A.2 MOE LOOP SCALING LAW

Proposition A.2 (MoE loop scaling law equivalence)**.** *The MoE loop scaling law recovers the dense loop scaling law at E=1, the non-looped MoE scaling law at R=1, and the standard dense scaling law at E=1 and R=1, with coefficient reparameterization where required.*

Proof.
Reduced form $\mathcal{L}|_{E=1}$**.** With expert count at E=1, we recover the dense case, for which m=1 and N_{act}=N. Since $\kappa_j(m)|_{m=1} = \kappa_j m^{-\theta}\big|_{m=1} = \kappa_j 1^{-\theta} = \kappa_j$ for $j = 1, 2$, the recurrence mapping in Eq. 7 therefore reduces to the dense looped model formula:

$$\begin{aligned} N_{\text{eff}}(R, m)|_{E=1} &= \left[N_{\text{act}} + \kappa_1(m) N_{\text{loop}} \left(1 - e^{-(R-1)/\kappa_2(m)}\right)\right]\Big|_{m=1, N_{\text{act}}=N} \\ &= N + \kappa_1 N_{\text{loop}} \left(1 - e^{-(R-1)/\kappa_2}\right) = N_{\text{eff}}^{\text{bounded}}(R). \end{aligned}$$

Let $\hat{E}_1 \equiv \hat{E}|_{E=1}$. We reparameterize the coefficients as $\widetilde{A} = A\hat{E}_1^{\delta}$, $\widetilde{\alpha} = \alpha + \gamma \ln \hat{E}_1$, $\widetilde{B} = B\hat{E}_1^{\omega}$, and $\widetilde{\beta} = \beta + \zeta \ln \hat{E}_1$. The MoE loop scaling law in Eq. 9 then reduces to

$$\begin{aligned} \mathcal{L}(N_{\text{act}}, D, R, E, m)|_{E=1} &= A\, \hat{E}_1^{\delta} N_{\text{eff}}(R, 1)^{\alpha + \gamma \ln \hat{E}_1} + B\, \hat{E}_1^{\omega} D^{\beta + \zeta \ln \hat{E}_1} + c \\ &= \widetilde{A}\, N_{\text{eff}}^{\text{bounded}}(R)^{\widetilde{\alpha}} + \widetilde{B}\, D^{\widetilde{\beta}} + c \\ &= \mathcal{L}(N, D, R). \end{aligned} \tag{15}$$

This reduced form is equivalent to the loop scaling law in Eq. 6 for dense looped transformers.

Reduced form $\mathcal{L}|_{R=1}$**.** With recurrence fixed at R=1, the recurrence mapping in Eq. 7 reduces to

$$\begin{aligned} N_{\text{eff}}(R, m)|_{R=1} &= \left[N_{\text{act}} + \kappa_1(m) N_{\text{loop}} \left(1 - e^{-(R-1)/\kappa_2(m)}\right)\right]\Big|_{R=1} \\ &= N_{\text{act}} + \kappa_1(m) N_{\text{loop}} \left(1 - e^{0}\right) = N_{\text{act}}. \end{aligned}$$

The MoE loop scaling law in Eq. 9 then reduces to

$$\begin{aligned} \mathcal{L}(N_{\text{act}}, D, R, E, m)|_{R=1} &= A\, \hat{E}^{\,\delta} N_{\text{eff}}(1, m)^{\,\alpha + \gamma \ln \hat{E}} + B\, \hat{E}^{\,\omega} D^{\,\beta + \zeta \ln \hat{E}} + c \\ &= A\, \hat{E}^{\,\delta} N_{\text{act}}^{\alpha + \gamma \ln \hat{E}} + B\, \hat{E}^{\,\omega} D^{\,\beta + \zeta \ln \hat{E}} + c \\ &= \mathcal{L}(N_{\text{act}}, D, E). \end{aligned} \tag{16}$$

This reduced form is equivalent to the MoE scaling law for non-looped MoE models (Ludziejewski et al., 2025) .

Reduced form $\mathcal{L}|_{R=1,E=1}$**.** Setting both $R{=}1$ and $E{=}1$ gives $m{=}1$, $N_{\text{act}}{=}N$, and $\kappa_j(m)\big|_{m=1} = \kappa_j m^{-\theta}\big|_{m=1} = \kappa_j 1^{-\theta} = \kappa_j$ for $j = 1, 2$. The recurrence mapping in Eq. 7 thus reduces to the dense non-looped baseline:

$$\begin{aligned} N_{\text{eff}}(R,m)|_{R=1,E=1} &= \left[N_{\text{act}} + \kappa_1(m) N_{\text{loop}} \left(1 - e^{-(R-1)/\kappa_2(m)}\right)\right]\Big|_{R=1,m=1,N_{\text{act}}=N} \\ &= N + \kappa_1 N_{\text{loop}} \left(1 - e^{0}\right) = N. \end{aligned}$$

Using the same reparameterized coefficients $\widetilde{A} = A\hat{E}_1^{\delta}$, $\widetilde{\alpha} = \alpha + \gamma \ln \hat{E}_1$, $\widetilde{B} = B\hat{E}_1^{\omega}$, and $\widetilde{\beta} = \beta + \zeta \ln \hat{E}_1$ as in Eq. 15, the MoE loop scaling law in Eq. 9 then reduces to

$$\begin{aligned} \mathcal{L}(N_{\text{act}}, D, R, E, m)|_{R=1,E=1} &= A\,\hat{E}_1^{\delta} N_{\text{eff}}(1,1)^{\alpha+\gamma \ln \hat{E}_1} + B\,\hat{E}_1^{\omega} D^{\beta+\zeta \ln \hat{E}_1} + c \\ &= \widetilde{A}\, N^{\widetilde{\alpha}} + \widetilde{B}\, D^{\widetilde{\beta}} + c \\ &= \mathcal{L}(N, D). \end{aligned} \tag{17}$$

This reduced form is equivalent to the standard non-looped scaling law in Eq. 1 (Hoffmann et al., 2022; Kaplan et al., 2020). □

B Expert-Path Diversity Across Recurrence

Expert-path diversity. The sparsity-conditional mapping is motivated by MoE sparse routing: a token may access different experts across recurrent passes. We measure this directly from the router decisions of trained looped MoE models. For an input token t, looped MoE layer ℓ, and pass r, let $\mathcal{E}_{t,\ell,r}$ be the top-k experts selected from E_{route} routed experts. We define the expert-path diversity as

$$\Psi(R) = \frac{1}{k}\, \mathbb{E}_{t,\ell}\left[\left|\bigcup_{r=1}^{R} \mathcal{E}_{t,\ell,r}\right|\right], \tag{18}$$

where the union deduplicates experts reused across recurrent passes, the expectation averages over tokens and the $n_{\ell,\text{loop}}$ looped layers, and normalizing by k expresses the expert-path diversity per active expert. At $R{=}1$ (non-looped) or $E_{\text{route}}{=}k{=}1$ (dense), $\Psi{=}1$ by definition, as a single pass or a single expert does not have path diversity over recurrent passes.

Table B.1: Expert-path diversity Ψ across varying recurrence R and expert count E (mean ± std).

E	$R{=}1$	$R{=}2$	$R{=}4$	$R{=}8$
1	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
2	1.0000 ± 0.0000	1.1212 ± 0.0692	1.2165 ± 0.0465	1.5088 ± 0.0529
4	1.0000 ± 0.0000	1.1917 ± 0.0974	1.2973 ± 0.0703	1.5300 ± 0.0785
8	1.0000 ± 0.0000	1.2099 ± 0.0868	1.4159 ± 0.0901	1.7035 ± 0.0861

Empirical measures of Ψ. We report Ψ for looped models across varying recurrence and expert count, where Ψ is computed on the same validation set of 800 prompts from different tasks, e.g., math, code, and knowledge. Table B.1 shows that at fixed recurrence, expert-path diversity Ψ increases with larger expert count consistently. This indicates that greater sparsity lets each recurrent pass reach a broader set of expert parameters. As Ψ is only an empirical routing statistic, we introduce sparsity as a condition for modeling the modulated effective-parameter gain (Eq. 7).

C Scaling Experiments and Parametric Fitting

C.1 Training Details

Model training setups. We follow the experimental setup similar to recent work on MoE scaling laws (Chen et al., 2026), using the same pre-training data. All models are trained from scratch on the same open-licensed, web-heavy data mixture, supplemented with math, code, knowledge, and science data. We train with a context length of 2,048 and sequence packing, tied input-output embeddings, and RoPE with base frequency 500,000. We use the llama tokenizer with a vocabulary of

202k. We use AdamW with β_1=0.9, β_2=0.95, ϵ=10^{-15}, weight decay 0.1, and gradient clipping at 1.0. Model weights are trained in BF16, while optimizer states, gradients, and MoE router computations are maintained in FP32. All scaling-sweep experiments run on 8 nodes (64 NVIDIA H100 96 GB GPUs) with a global batch size of 3,072 and a sequence length of 2,048, corresponding to approximately 6.29 million tokens per batch.

Looped MoE setups. For MoE models, we adopt sigmoid gating with per-token top-k normalization, auxiliary-loss-free load balancing with bias-update rate λ_{lb}=10^{-3}, router z-loss with coefficient λ_z=10^{-4}, and drop-and-pad token dispatch with capacity factor 1.5. For the scaling sweep experiments used for loop scaling laws in Sections 3.2–4.1, we apply the next-token prediction loss to the output of the final recurrent pass. For the looped MoE in Table 2, we add per-loop supervision (Bae et al., 2024) to allow the same trained checkpoint to be evaluated at different recurrence at test time.

C.2 Model Ladder and Scaling Sweep

Model ladder setups. We construct a model scaling ladder with five active model scales $N_{\text{act}} \in \{0.3, 0.6, 1.0, 1.6, 2.4\}$B. Each rung fixes its own model configuration d_m, n_h, n_ℓ, $n_{\ell,\text{loop}}$, and N_{loop}. Here, d_m is the model dimension; n_h and n_{KV} are the numbers of attention and KV heads; n_ℓ and $n_{\ell,\text{loop}}$ are the numbers of total and recurrent layers; d_h is the head dimension; N_{act}, N_{total} denote the active and total parameter counts and N_{loop} is the parameter count of the recurrent block. All configurations use n_{KV}=4 and d_h=64, with $n_h = d_m/d_h$. Since we adopt the standard middle-block looping (Geiping et al., 2025), the first two and last two layers remain unshared, giving $n_{\ell,\text{loop}} = n_\ell - 4$. Table C.1 summarizes the model ladder setups.

Table C.1: Model scaling ladder over five model scales. Parameter counts are in billions (B). N_{act} and N_{total} include embedding parameters, whereas N_{loop} includes only parameters in the recurrent block.

	Architecture dimensions					$N_{\text{total}}(E)$ (B)				
N_{act} (B)	d_m	n_h	n_ℓ	$n_{\ell,\text{loop}}$	N_{loop} (B)	E=1	E=2	E=4	E=8	E=16
0.3	768	12	20	16	0.1	0.3	0.5	0.8	1.3	2.5
0.6	1024	16	26	22	0.3	0.6	0.9	1.6	2.9	5.5
1.0	1280	20	32	28	0.7	1.0	1.6	2.9	5.4	10.5
1.6	1536	24	38	34	1.2	1.6	2.7	4.8	9.1	17.7
2.4	1792	28	44	40	1.8	2.4	4.1	7.5	14.3	27.8

Scaling sweep runs. Our experimental scaling sweep uses $N_{\text{act}} \in \{0.3, 0.6, 1.0\}$B across training tokens $D \in \{100, 200, 300, 400, 500\}$, expert counts $E \in \{1, 2, 4, 8, 16\}$ which vary sparsity and active-parameter ratios m, and recurrences $R \in \{1, 2, 3, 4, 6, 8\}$. Following Kaplan et al. (2020), the fitted parameter counts exclude embeddings.

Model optimization design space. For the resource-constrained joint optimization of recurrence and sparsity in Figure 6(c) (Section 4.2), we evaluate the fitted law over five model scales on the model ladder: $N_{\text{act}} \in \{0.3, 0.6, 1.0, 1.6, 2.4\}$B, expert counts $E \in \{1, 2, 4, 8, 16, 32\}$, and recurrences $R \in \{1, \ldots, 10\}$. This gives 300 candidate configurations, where $N_{\text{act}} \in \{1.6, 2.4\}$B, expert counts E>16, and recurrences R>8 lie in the extrapolation range of the fitted law.

C.3 Parametric Fitting

Fitting the dense loop scaling laws. For fitting the dense loop scaling law, we instantiate the effective parameter count $N_{\text{eff}}(R)$ in Eq. 6 using one of the following recurrence mappings:

$$N_{\text{eff}}(R) = \begin{cases} N + (R-1)N_{\text{loop}}, & \text{linear,} \\ N + (R^\varphi - 1)N_{\text{loop}}, & \text{power law,} \\ N + \kappa_1 N_{\text{loop}}\left(1 - e^{-(R-1)/\kappa_2}\right), & \text{bounded.} \end{cases} \tag{19}$$

Here, N is the stored model parameter count and N_{loop} is the parameter count of the recurrent block, as defined in Section 3.1. The linear mapping coincides with the unrolled parameter count $N_{\text{unroll}}(R)$, whereas the power-law and bounded mappings model the effective-parameter gain from looping. The linear mapping introduces no recurrence-specific fitted coefficient, while the power-law and bounded mappings introduce φ and $\{\kappa_1, \kappa_2\}$, respectively.

At R=1, the dense loop scaling law recovers the standard non-looped scaling law (Proposition A.1, Eq. 14). We therefore derive the loop scaling law using a staged fitting procedure: first establish the non-looped scaling law with base coefficients $\{A, \alpha, B, \beta, c\}$ using the R=1 subset of the scaling sweep across model size N and data D, and then extend it to the loop scaling law to jointly estimate all coefficients using the scaling sweep across R, N, and D. For a fair comparison in Figure 2, we apply the same fitting procedure to the loop scaling law under the linear and power-law mappings (Eq. 3) and the bounded mapping (Eq. 4), and evaluate each formulation on the same held-out R, N, and D, where the held-out slices are R=16, N=1.0B, and $D \in (400, 500]$B. Figure 2(c) reports the RMSE on each held-out slice after fitting on the remaining runs.

Fitting the MoE loop scaling laws. For fitting the MoE loop scaling law, we instantiate the effective parameter count $N_{\text{eff}}(R, m)$ in Eq. 9 using one of the following recurrence mappings:

$$N_{\text{eff}}(R, m) = \begin{cases} N_{\text{act}} + (R-1)N_{\text{loop}}, & \text{linear}, \\ N_{\text{act}} + (R^{\varphi}-1)N_{\text{loop}}, & \text{power law}, \\ N_{\text{act}} + \kappa_1 N_{\text{loop}} \left(1 - e^{-(R-1)/\kappa_2}\right), & \text{bounded}, \\ N_{\text{act}} + \kappa_1(m) N_{\text{loop}} \left(1 - e^{-(R-1)/\kappa_2(m)}\right), & \text{sparsity-conditional}. \end{cases} \tag{20}$$

For the sparsity-conditional mapping, $\kappa_j(m) = \kappa_j m^{-\theta}$ for $j \in \{1, 2\}$. Here, N_{act} is the active parameter count, N_{loop} is the active parameter count of the recurrent block, and $m = N_{\text{act}}/N_{\text{total}}$ is computed from the active and total parameter counts as defined in Section 3.1. The linear mapping coincides with $N_{\text{unroll}}(R)$, while the linear, power-law, and bounded recurrence mappings are independent of m; their outer MoE scaling laws retain the dependence on expert count E. The linear mapping introduces no recurrence-specific fitted coefficient, while the power-law, bounded, and sparsity-conditional mappings introduce φ, $\{\kappa_1, \kappa_2\}$, and $\{\kappa_1, \kappa_2, \theta\}$, respectively.

At R=1, the MoE loop scaling law recovers the standard non-looped MoE scaling law (Proposition A.2, Eq. 16). We derive the MoE loop scaling law using a staged fitting procedure: first establish the non-looped MoE scaling law with base coefficients $\{A, \alpha, B, \beta, c, \delta, \gamma, \omega, \zeta, E_{\text{start}}, E_{\text{max}}\}$ using the R=1 subset of the scaling sweep across active model size N_{act}, training tokens D, and expert count E; and then we extend it to the MoE loop scaling law to jointly estimate all coefficients using the full sweep across the four scaling axes: R, N_{act}, D, and E. For a fair comparison in Figure 3, we apply the same fitting procedure to the MoE loop scaling law under the linear, sparsity-independent power-law, bounded, and sparsity-conditional recurrence mappings, and evaluate each formulation on the same held-out R, E, N_{act}, and D, where the held-out slices are R=16, E=16, N_{act}=1.0B, and $D \in (400, 500]$B. In Figure 3(c), we report the RMSE scores on the held-out runs of recurrence R, expert count E, model N_{act}, and data D, after fitting each law on the remaining runs.

Numerical optimization and final fitted coefficients. We fit the final MoE loop scaling law using more than 1,000 loss observations from the scaling sweep across four axes N_{act}, D, E, R (Appendix C.2). We initialize the scaling-law fit using `scipy.optimize.curve_fit` with nonlinear least squares and an MSE objective, and then refine the fit using `scipy.optimize.minimize` with L-BFGS-B optimization and a log-Huber objective. Table C.2 reports the final fitted coefficients of the MoE Loop Scaling Law in Eq. 9, using all the scaling sweep runs over recurrence R, expert count E, model N_{act}, and data D.

Table C.2: Fitted coefficients of MoE Loop Scaling Law in Eq. 9. RMSE: root-mean-square error of the fit.

Base scaling law coefficients					MoE coefficients						Recurrence coefficients			Fit
A	α	B	β	c	δ	γ	ω	ζ	E_{start}	E_{max}	κ_1	κ_2	θ	RMSE
0.8085	−0.1951	3.4291	−0.7548	1.3555	−0.1935	−0.0137	−0.2056	0.0881	1.3174	57.1201	0.3682	1.4037	0.3286	0.0037

Sparsity exponent ablation. We ablate a less constrained variant of Eq. 7 with separate sparsity exponents, $\kappa_1(m) = \kappa_1 m^{-\theta_1}$ and $\kappa_2(m) = \kappa_2 m^{-\theta_2}$. The RMSE remains 0.0037 on the final fitted law. As adding extra exponent yields no meaningful gain, we adopt the shared-θ formulation.

C.4 Bootstrap Results of the MoE Loop Scaling Law

To quantify uncertainty in our fitted MoE loop scaling law in Table C.2, we use a bootstrapping protocol similar to prior works (Hoffmann et al., 2022; Ludziejewski et al., 2025). We repeatedly sample 90% of the observations without replacement 100 times, and refit the law on each sampled

subset. For the fitted coefficients, we report the full-data estimates and 90% bootstrap percentile confidence intervals (CIs), defined by the 5th and 95th percentiles over the 100 refit runs.

Table C.3: Bootstrap results on the MoE loop scaling law. (a) Coefficients fitted on full-data or the 90% sampled subsets with intervals spanning the 5th–95th percentiles over 100 refit runs. (b) Derived joint optimum under the compute and memory budgets (with INT4 weights) used in Figure 6(c) on the 100 refit runs.

(a) Coefficient uncertainty

Coefficient	Estimate	90% CI
κ_1	0.3682	[0.3658, 0.3723]
κ_2	1.4037	[1.3896, 1.4243]
θ	0.3286	[0.3183, 0.3350]

(b) Joint-optimum uncertainty

Compute budget	Memory budget	$N^\star_{\text{act}}$ 90% CI	$E^\star$ 90% CI	$R^\star$ 90% CI
5×10^{21}	1 GB	[1.0, 1.0]B	[2, 2]	[2, 2]
5×10^{21}	3 GB	[1.0, 1.0]B	[8, 8]	[2, 2]
5×10^{21}	10 GB	[1.6, 1.6]B	[16, 16]	[1, 1]

Table C.3(a) presents the bootstrap results for coefficients of the sparsity-conditional recurrence mapping. The narrow intervals of 100 refits indicate these coefficient estimates remain stable under bootstrapping. Table C.3(b) shows the 90% bootstrap CIs of the joint optimum for resource-constrained architecture selection in Figure 6(c) (Section 4.2). The collapsed intervals indicate the selected configurations remain the same across 100 refits, and confirm that the fitted law yields robust estimates for architecture optimization under compute and memory budgets.

D Downstream Scaling and Evaluation

D.1 Downstream Evaluation Protocols

Empirical scaling setup on downstream tasks. For the downstream scaling analysis in Section 4.3, we compare models with active model sizes of 0.3B, 0.6B, 1.0B, while varying expert count $E \in \{1, 2, 4, 8, 16\}$ and recurrence $R \in \{1, 2, 3, 4, 6, 8\}$. We evaluate downstream performance as a function of training compute on 14 benchmarks spanning commonsense, knowledge, science, reading comprehension, and reasoning, enabling comparisons across configurations at matched training FLOPs. We report *Overall* as the mean over 14 benchmarks. Motivated by prior work showing that recurrent depth can preferentially improve latent reasoning (Saunshi et al., 2025; Geiping et al., 2025; Zhu et al., 2025), we also report *Reasoning* as the mean of GSM8K and BBH to evaluate reasoning performance. The benchmark suite and evaluation protocols are provided in the following.

Table D.1: Evaluation setups of 14 foundational benchmarks. Task names and metrics follow LM-EVAL.

Category	Benchmark	Task name	n-shot	Metric
Reasoning	BBH	`leaderboard_bbh`	3	`acc_norm`
	GSM8K	`gsm8k_cot`	8	`exact_match,flexible-extract`
Science	ARC-C	`arc_challenge`	25	`acc_norm`
	ARC-E	`arc_easy`	0	`acc_norm`
	OBQA	`openbookqa`	0	`acc_norm`
Commonsense Reasoning	HellaSwag	`hellaswag`	0	`acc_norm`
	PIQA	`piqa`	0	`acc_norm`
	SIQA	`social_iqa`	0	`acc`
	WinoGrande	`winogrande`	0	`acc`
Reading	BoolQ	`boolq`	0	`acc`
	DROP	`drop`	3	`f1`
Knowledge	MMLU	`mmlu`	5	`acc`
	NQ	`nq_open`	5	`exact_match`
	TQA	`triviaqa`	5	`exact_match`

Evaluation protocols. We evaluate the models in Section 4.3 on 14 widely used benchmarks spanning five core competencies: (1) reasoning (BBH (Suzgun et al., 2023), GSM8K (Cobbe et al., 2021)), (2) science (ARC-C/E (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018)), (3) commonsense (HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), WinoGrande (Sakaguchi et al., 2021)), (4) reading (BoolQ (Clark et al., 2019), DROP (Dua et al., 2019)), and (5) knowledge (MMLU (Hendrycks et al., 2020), Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017)). We use the Language Model Evaluation Harness (LM-EVAL) (Gao et al., 2024) with the vLLM backend, bfloat16 model precision, and greedy decoding for generation-based tasks. Detailed benchmark-specific few-shot settings and metrics are given in Table D.1. For compactness in Table 2, we use the following benchmark abbreviations: BBH: BIG-Bench Hard,

ARC-C/E: ARC-Challenge/Easy, OBQA: OpenBookQA, HS: HellaSwag, Wino: WinoGrande, NQ: Natural Questions, TQA: TriviaQA.

D.2 Compute-Matched Trillion-Token Experiment

Scaling-law-guided model design. In Section 4.3, we compare a looped MoE model with a non-looped MoE reference under matched training compute. Given fixed compute FLOPs, we derive the compute-optimal recurrence for the looped MoE model by evaluating each candidate recurrence on an IsoFLOP basis, as summarized in Algorithm D.1.

Algorithm D.1: IsoFLOP recurrence selection

Input: compute budget $\bar{F}_{\text{train}}$ or reference model $(N_{\text{ref}}, D_{\text{ref}})$; target looped MoE architecture $(N_{\text{act}}, N_{\text{loop}}, E, m)$; recurrence set $\mathcal{R}$; tolerance ϵ.

1. If a reference model is provided, set $\bar{F}_{\text{train}} = 6N_{\text{ref}}D_{\text{ref}}$.
2. For each $R \in \mathcal{R}$ of the target looped MoE, compute its unrolled parameter count and compute-matched training-token budget:
$$N_{\text{unroll}}(R) = N_{\text{act}} + (R-1)N_{\text{loop}}, \qquad D^R_{\text{target}} = \bar{F}_{\text{train}}/[6N_{\text{unroll}}(R)].$$
3. Compute the looping effective parameter count and the predicted loss using Eqs. 7 and 9:
$$N_{\text{eff}}(R,m) = N_{\text{act}} + \kappa_1(m)N_{\text{loop}}\left(1 - e^{-(R-1)/\kappa_2(m)}\right),$$
$$\mathcal{L}_R = \mathcal{L}(N_{\text{act}}, D^R_{\text{target}}, R, E, m).$$
4. Select the compute-optimal recurrence according to Eq. 11:
$$R^\star = \arg\min_R \mathcal{L}(N_{\text{act}}, D^R_{\text{target}}, R, E, m),$$
$$\text{s.t.} \quad 6N_{\text{unroll}}(R)D^R_{\text{target}} = \bar{F}_{\text{train}}, \quad \Delta\mathcal{L}(R) \geq \epsilon.$$

Output: Compute-optimal recurrence $R^\star$.

We set ϵ to the fitted-law RMSE, which measures the predictive discrepancy in the model loss. A marginal improvement below this resolution is treated as indistinguishable from fitting error and thus insufficient to justify the cost of an additional recurrent pass.

Matched-compute training and test-time evaluation. For Table 2, the procedure above selects $R^\star$=5 for the A0.3B-1.3B LoopMoE (E=8), which we compare with an A0.6B-2.9B non-looped MoE (E=8, R=1). The models are trained for ∼3T and ∼6T tokens, respectively, which correspond to approximately 1.5×10^{22} FLOPs. We calculate the compute FLOPs as $F_{\text{train}} = 6N_{\text{unroll}}(R)D$, where D is the training tokens and $N_{\text{unroll}}(R)$ is the unrolled parameter count at recurrence R. A0.3B-1.3B LoopMoE is trained as described in Appendix C.1. As LoopMoE gives valid output per recurrent pass, we evaluate the model by varying recurrence $R \in \{1, 2, \ldots, 5\}$ at test time, to switch inference effort from low to high on demand. To compare the inference compute of A0.3B-1.3B LoopMoE and A0.6B-2.9B non-looped MoE, we report their relative inference compute ratio as $F^{0.3\text{B}}_{\text{inf}}(R)/F^{0.6\text{B}}_{\text{inf}}(1)$, where $F_{\text{inf}}(R) = 2N_{\text{unroll}}(R)$.