TRANSFORMER DEPTH UP-SCALING: AN INFINITE-DEPTH PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Progressive training iteratively up-scales small checkpoints to reduce the cost of training large language models. We observe that, as base-model depth increases, layer concatenation leads to greater performance degradation immediately after up-scaling, whereas in-place stacking becomes more stable. Inspired by this depth-dependence trend, we consider an infinite-depth formulation of Transformers, treating depth up-scaling as continuation of parameters and study the consistency problems with a controlled ODE. A criterion on this ODE's terminal-state ensures the consistent next-token predictions between the base and the up-scaled models. For base models of depth L, piecewise-constant simple-continuation yields an $\mathcal{O}(L^{-1})$ terminal-state error and thus asymptotically preserves immediate performance. As a concrete continuation scheme, scaled stacking approximates the Euler refinement of simple-continuation, with the same error order uniformly over growth-factors. Moreover, the ODE's adjoint gives $\mathcal{O}(L^{-1})$-consistent directional derivatives, providing a criterion for preserving descent directions. Benchmarking on the Qwen3 family and progressive training on the GPT-2 provide support for these consistency results.

1 INTRODUCTION

Scaling laws suggest that the performance of large language models (LLMs) predictably improves given sufficient data and compute (Kaplan et al., 2020; Hoffmann et al., 2022). Training from scratch, however, incurs increasingly large computational, energy, and engineering costs (Biderman et al., 2023). *Model up-scaling* offers a promising alternative: a larger model is initialized from a trained checkpoint and trained further, rather than starting from random initialization (Samragh et al., 2024; Du et al., 2024; Wang et al., 2024b). Related strategies have been studied under names such as progressive training, staged training, and continual training (Bengio et al., 2006; Hinton et al., 2006; Li et al., 2020; 2022; Shen et al., 2022; Pan et al., 2024; Yano et al., 2025).

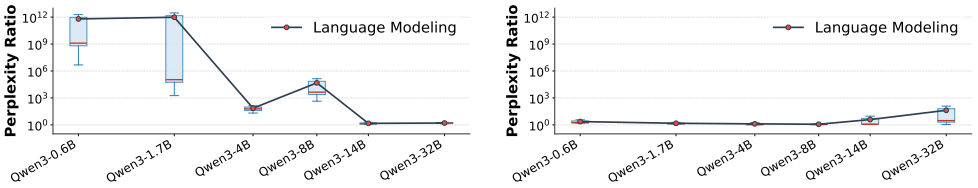
Recent empirical comparisons suggest that *depth up-scaling* is particularly attractive for decoder-only Transformers. Du et al. (2024) report that layer duplication outperforms baselines such as the identity extension used in LLaMA-Pro (Wu et al., 2024) and reduces training FLOPs by nearly 50% compared with training from scratch. In their ablations, full-block *concatenation* ((l_1, l_2, l_1, l_2), see Figure 7) outperforms in-place *stacking* ((l_1, l_1, l_2, l_2), see Figure 8). This concatenation strategy has also been applied to a 560B modern-scale LLM reported by Meituan LongCat Team et al. (2025).

Empirical ablations indicate that exact function preservation during up-scaling (Chen et al., 2016; Shen et al., 2022) is not necessary for progressive training. Nevertheless, preserving performance remains important, since severe initial degradation can cause continual training to collapse (Dong et al., 2020; Du et al., 2024). Young et al. (2024) report that their newly initialized 9B model achieves performance comparable to, but not identical to, that of its base model.

The ablations in Du et al. (2024) favor concatenation over in-place stacking, but consider relatively shallow models up-scaled from 8 to 24 layers. The 560B-parameter model is likewise up-scaled from 14 to 28 layers (Meituan LongCat Team et al., 2025). It is therefore unclear whether this superiority extends to deeper models. We encounter an empirical puzzle when investigating this question. Using the Qwen3 family (Yang et al., 2025), we compare how well in-place stacking and concatenation preserve performance immediately after up-scaling. For the deepest checkpoints, stacking yields perplexity ratios of about $\mathcal{O}(1)$, whereas concatenation leads to much larger increases in perplexity

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). This trend is particularly relevant in the deep-model regime, where training from scratch is most expensive and stable depth up-scaling is most valuable.



(a) Stacking. (b) Concatenation.

Figure 1: Perplexity of each up-scaled model relative to its corresponding base model. As the base-model depth increases, the ratio for stacking approaches the base-model level (1(a)), whereas the ratio for concatenation reaches $\mathcal{O}(10^2)$ on the language-modeling benchmarks at 32B (1(b)).

This trend drives us to study the infinite-depth limits. Pre-norm Transformer blocks take a residual form (He et al., 2016; Vaswani et al., 2017), and a trained model of depth L can be viewed as an L-step forward-Euler discretization. From this perspective, layer parameters are discretized control variables along depth rather than an unordered collection of tensors. Depth up-scaling therefore asks how to continue these controls to a finer grid while maintaining stability and consistency.

Within this framework, we consider two consistency questions. The *zeroth-order* (forward) consistency question asks whether the up-scaled model preserves the base model's next-token prediction. The *first-order* (backward) consistency question asks whether loss differentials computed by backpropagation remain consistent along corresponding parameter directions, and under what conditions descent directions remain valid after up-scaling. The latter concerns the reuse of optimization information during continual training.

Contributions. Our contributions are threefold.

1. We formulate depth up-scaling of pre-norm Transformers as parameter continuation followed by grid refinement in a controlled ordinary differential equation (ODE) framework. From this perspective, we analyze the zeroth-order consistency of next-token predictions and the first-order consistency of descent directions.
2. We establish a terminal-state criterion for zeroth-order consistency and show that simple-continuation (3) satisfies it. We further show that scaled stacking approximates the Euler refinement under simple-continuation, and thus preserves the next-token predictions.
3. We relate the adjoint of simple-continuation to backpropagation and establish first-order consistency of loss differentials under scaled stacking. This yields a criterion for preserving descent directions after depth up-scaling.

2 Background

Residual Transformer Dynamics

Let T be the maximal context length, d be the hidden dimension, and $x_k \in \mathbb{R}^{T\times d}$ be the hidden-state of the k-th block. The standard pre-norm Transformer block contains a multi-head attention (MHA) sublayer and a feed-forward network (FFN) sublayer. We allow arbitrary attention implementations and smooth gated activations, provided that both mappings are smooth, and hence locally Lipschitz, in the hidden state and parameters. The serial MHA-FFN update is

$$x_{k+\frac{1}{2}} = x_k + \boldsymbol{A}(x_k, \theta_k^A),$$
$$x_{k+1} = x_{k+\frac{1}{2}} + \boldsymbol{F}(x_{k+\frac{1}{2}}, \theta_k^F),$$

where $\theta_k := (\theta_k^A, \theta_k^F) \in \Theta$ denotes all parameters in the k-th block. For simplicity, define

$$\boldsymbol{T}(x_k, \theta_k) := \boldsymbol{A}(x_k, \theta_k) + \boldsymbol{F}(x_k + \boldsymbol{A}(x_k, \theta_k), \theta_k),$$

so that the update is $x_{k+1} = x_k + \boldsymbol{T}(x_k, \theta_k)$. Let $\mathcal{V}$ be a finite vocabulary and $z_0 \in \prod_{i=1}^{T} \mathcal{V}$ an input sequence. Training a depth-L model is then equivalent to the constrained optimization problem:

$$\begin{aligned} \min_{\theta^L, V_{\mathrm{em}}, V_{\mathrm{de}}, x^L} \ & \mathbb{E}\left[\ell(x_L^L)\right] \\ \text{s.t. } & x_{k+1}^L = x_k^L + \boldsymbol{T}(x_k^L, \theta_k^L), \quad \forall k = 0, 1, \ldots, L-1, \\ & x_0^L = z_0(\xi) V_{\mathrm{em}}, \quad z_0(\xi) \sim \hat{P}_{\mathrm{data}}, \quad \xi \in \Xi. \end{aligned} \tag{1}$$

Here the tensors are $\theta^L := (\theta_0^L, \ldots, \theta_{L-1}^L)$ and $x^L := (x_1^L, \ldots, x_L^L)$, and the matrices are $V_{\mathrm{em}} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $V_{\mathrm{de}} \in \mathbb{R}^{d \times |\mathcal{V}|}$, respectively. The loss is $\ell(x) := H\big(q(\cdot|z_0), \mathrm{softmax}(\mathrm{Norm}(x)V_{\mathrm{de}})\big)$, where $q(\cdot|z_0)$ is the target next-token distribution and $H(\cdot,\cdot)$ is the cross-entropy. Here and below, the output readout is evaluated at the final non-padding position of the input sequence. It is worth noting that $\boldsymbol{A}$ and $\boldsymbol{F}$ are smooth and locally Lipschitz in the hidden-state and parameters; the training and test sets are finite given a finite vocabulary and a fixed maximal context length.

Let $(\theta^{L,*}, V_{\mathrm{em}}^{L,*}, V_{\mathrm{de}}^{L,*}, x^{L,*})$ be a solution of equation 1, where $\theta^{L,*} = (\theta_k^{L,*})_{k=0}^{L-1}$ and $x^{L,*} = (x_k^{L,*})_{k=1}^{L}$. We assume a boundedness condition on the trained parameters.

Assumption 1. *There exist $M_\theta, M_V > 0$ such that for all $L > 1$, $\max\{\|V_{\mathrm{em}}^{L,*}\|, \|V_{\mathrm{de}}^{L,*}\|\} \le M_V$ and $\|\theta_k^{L,*}\| \le M_\theta$ for all $k = 0, \ldots, L-1$.*

Infinite-Depth View of Trained Transformers

We next introduce the conditions under which an infinite-depth formulation is appropriate for trained Transformers. Such a formulation requires (i) that the hidden-state path neither diverges nor becomes increasingly oscillatory with depth and (ii) that each hidden layer contributes. When depth is interpreted as time on the fixed interval $[0, 1]$, a stable finite-time trajectory requires each residual increment to be of order $1/L$. This scaling follows directly from any uniformly Lipschitz infinite-depth interpolation of the trained hidden-states.

Proposition 1 (Depth-stable path)**.** *Let $\bar{x}^L : [0,1] \to \mathbb{R}^{T \times d}$ be the piecewise-linear interpolation satisfying $\bar{x}^L(k/L) = x_k^{L,*}$. If $\bar{x}^L$ is Lipschitz with a constant independent of L, then*

$$\|x_{k+1}^{L,*} - x_k^{L,*}\| = \mathcal{O}(L^{-1}), \qquad k = 0, \ldots, L-1.$$

Proposition 1 formalizes the residual scale implied by a uniformly stable depth trajectory (Haber & Ruthotto, 2018). Prior work provides empirical evidence for this residual-scaling behavior. For general residual networks of the form $x_{k+1}^{L,*} = x_k^{L,*} + \alpha^{L,*} \boldsymbol{R}(x_k^{L,*}, \theta_k^{L,*})$, Cohen et al. (2021) observe the scalings $|\alpha^{L,*}| \sim \mathcal{O}(L^{-a})$ and $\|\boldsymbol{R}(x_k^{L,*}, \theta_k^{L,*})\| \sim \mathcal{O}(L^{-b})$ with $a + b \approx 1$. For Transformers with the same rescaling coefficients α_k^L, Wang et al. (2024a) propose using prescribed rescaling coefficients $\alpha_k^{L,*} = (2L)^{-1/4}$ to accelerate convergence, while Bachlechner et al. (2021) propose learning the coefficients and observe that they decay approximately as $1/L$ as the depth grows.

Rigorous analysis of the connection between discrete residuals and infinite-depth limits remains an open mathematical problem (Marion et al., 2024; Barboni et al., 2025). We therefore require neither convergence nor the existence of limiting trained parameters. Instead, we impose a local stability condition on the residual mappings, without assuming convergence of the trained parameters.

Assumption 2 (Depth-stable residual dynamics)**.** *Let*

$$\tilde{\boldsymbol{A}}(x, \theta_k^{L,*}) = L \cdot \boldsymbol{A}(x, \theta_k^{L,*}), \qquad \tilde{\boldsymbol{F}}(x, \theta_k^{L,*}) = L \cdot \boldsymbol{F}(x, \theta_k^{L,*})$$

be the rescaled mappings of $\boldsymbol{A}$ and $\boldsymbol{F}$. Assume $\|\tilde{\boldsymbol{A}}(x, \theta_k^{L,*})\| \le M_A$ and $\|\tilde{\boldsymbol{F}}(x, \theta_k^{L,*})\| \le M_F$ for all x, uniformly in L and k. Moreover, $\tilde{\boldsymbol{A}}(\cdot, \theta_k^{L,*})$ and $\tilde{\boldsymbol{F}}(\cdot, \theta_k^{L,*})$ are locally Lipschitz continuous in x, uniformly in L and k.

Assumption 2 states the regularity needed for the composite residual in equation 1. Proposition 1 establishes $\mathcal{O}(1/L)$ scaling along a stable hidden-state path, whereas the assumption imposes boundedness and local Lipschitz regularity off that path, which is sufficient to relate the discrete Transformer to a Carathéodory ODE. Set $h := 1/L$ and define

$$f_L(x, \theta) := \tilde{\boldsymbol{A}}(x, \theta) + \tilde{\boldsymbol{F}}\big(x + h\tilde{\boldsymbol{A}}(x, \theta), \theta\big).$$

Under this assumption, the trained Transformer update can be regarded as a forward-Euler step with step size $h = 1/L$. The associated optimal control formulation is

$$\begin{aligned} \min_{\boldsymbol{\theta}, V_{\mathrm{em}}, V_{\mathrm{de}}, \mathbf{x}} \ & \mathbb{E}\left[\ell(\mathbf{x}(1))\right] \\ \text{s.t. } & \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = f_L(\mathbf{x}, \boldsymbol{\theta}), \\ & \mathbf{x}(0) = z_0(\xi) V_{\mathrm{em}}, \quad z_0(\xi) \sim \hat{P}_{\mathrm{data}}, \quad \xi \in \Xi. \end{aligned} \tag{2}$$

Here $\mathbf{x}(t)$ is an absolutely continuous state trajectory, and $\boldsymbol{\theta}(t)$ is a time-dependent control. This formulation follows the optimal-control perspective of E (2017), with related formulations developed by Li & Hao (2018); Lu et al. (2018); Ruthotto & Haber (2020); Vialard et al. (2020). Generalization analyses in this setting include those of E et al. (2020; 2022). Neural ODEs likewise describe hidden-state evolution through a learned vector (Chen et al., 2018). Crucially, the control need not be smooth or even continuous in time t, but be bounded and Lebesgue measurable (Li et al., 2018):

Property 1. *The admissible control of problem (2) satisfies* $\boldsymbol{\theta} \in \boldsymbol{\Theta} := L^\infty\big([0,1], \Theta\big)$.

This property is central to depth up-scaling. A trained Transformer provides the discrete parameters $\{\theta_k^{L,*}\}_{k=0}^{L-1}$. Depth up-scaling therefore requires a continuation rule for these discrete parameters along the grid points. Prior work chooses piecewise-linear interpolation in depth (Cohen et al., 2021; Sander et al., 2022):

$$\boldsymbol{\theta}_{\mathrm{lin}}^{L,*}(t) := \begin{cases} \theta_k^{L,*} + \frac{t-kh}{h}(\theta_{k+1}^{L,*} - \theta_k^{L,*}), & t \in [kh, (k+1)h), \quad k = 0, \dots, L-2, \\ \theta_{L-1}^{L,*}, & t \in [(L-1)h, Lh]. \end{cases}$$

Several infinite-depth analyses impose the convergence of the interpolated parameters. As L tends to infinity, Cohen et al. (2021) prove the convergence of hidden-states $\mathbf{x}_{\mathrm{lin}}^{L,*}(t)$ when the limits of the parameters $\boldsymbol{\theta}_{\mathrm{lin}}^{L,*}(t)$ exist. However, Sander et al. (2022) show that $\boldsymbol{\theta}_{\mathrm{lin}}^{L,*}(t)$ does not converge in general. We therefore work with a notion of continuation that allows bounded measurable controls.

Definition 1 (Weak continuation)**.** *Let* $\mathbf{x}^L(t) = \mathbf{x}(0) + \int_0^t f_L(\mathbf{x}^L(s), \boldsymbol{\theta}^L(s))\,\mathrm{d}s$ *be absolutely continuous on* $[0,1]$*, and let* $\boldsymbol{\theta}^L \in \boldsymbol{\Theta}$ *be a bounded continuation of the discrete parameters* $\{\theta_k^L\}_{k=0}^{L-1}$*. We call* $(\mathbf{x}^L, \boldsymbol{\theta}^L)$ *a weak continuation of the solution of equation 1.*

3 Consistent Depth Up-Scaling for Transformers

3.1 Forward Consistency

We first relate terminal-state consistency to the preservation of the discrete model's output. A Transformer's next-token prediction is determined by its terminal hidden-states. On a finite evaluation domain, the terminal-state therefore controls the convergence of the KL divergence in expectation.

Theorem 1 (Terminal-state criterion)**.** *Suppose Assumption 1 holds. If a weak continuation* $(\mathbf{x}^{L,*}, \boldsymbol{\theta}^{L,*})$ *satisfies* $\lim_{L\to\infty} \|\mathbf{x}^{L,*}(1) - x_L^{L,*}\| = 0$ *for every* $z_0(\xi) \sim p_{\mathrm{data}}$*, then*

$$\lim_{L\to\infty} \mathbb{E}_{z_0 \sim p_{\mathrm{data}}}\left[D_{\mathrm{KL}}\big(p^{L,*}(\cdot|z_0) \,\|\, p_{\mathrm{cont}}^{L,*}(\cdot|z_0)\big)\right] = 0,$$

where $p^{L,*} = \mathrm{softmax}(\mathrm{Norm}(x_L^{L,*})V_{\mathrm{de}}^{L,*})$ *and* $p_{\mathrm{cont}}^{L,*} = \mathrm{softmax}(\mathrm{Norm}(\mathbf{x}^{L,*}(1))V_{\mathrm{de}}^{L,*})$.

The key task is then to identify continuations that are numerically consistent in the terminal-states.

Sander et al. (2022) prove that $\boldsymbol{\theta}_{\mathrm{lin}}^{L,*}(t)$ converges to a Lipschitz function when the residual mapping is linear and the trained parameters vary smoothly across layers. However, these conditions do not hold for Transformers, whose dynamics are nonlinear and whose discretely trained parameters need not vary smoothly, or even continuously, between layers. More fundamentally, the admissible controls need merely be Lebesgue measurable. Smoothness is an additional constraint on the control class and can restrict the approximation and generalization properties studied in E et al. (2022).

More importantly, $\mathbf{x}_{\mathrm{lin}}^{L,*}(t)$ does not generally satisfy the terminal-state criterion under nonlinear dynamics. The one-dimensional nonlinear control problem in Example 1 isolates this failure. For

comparison, define the simple-(weak)continuation associated with $\{\theta_k^{L,*}\}_{k=0}^{L-1}$ as

$$\boldsymbol{\theta}_{\mathrm{simp}}^{L,*}(t) = \theta_k^{L,*}, \qquad t \in [kh, (k+1)h), \quad k = 0, \dots, L-1. \tag{3}$$

The following result shows the terminal-state consistency of simple-continuation.

Theorem 2 (Zeroth-order consistency)**.** *Suppose that the trained parameters $\{\theta_k^{L,*}\}_{k=0}^{L-1}$ satisfy Assumptions 1 and 2. Then*

$$\|\mathbf{x}_{\mathrm{simp}}^{L,*}(1) - x_L^{L,*}\| = \mathcal{O}(L^{-1}).$$

3.2 Depth Up-Scaling via Grid Refinement

Theorem 2 suggests refining the grid under the piecewise-constant simple-continuation $\boldsymbol{\theta}_{\mathrm{simp}}^{L,*}$, rather than constructing a smooth control by interpolation or reordering layers through concatenation. Consider a trained depth-L model. For $M \geq 1$, set $\delta := h/M = 1/(ML)$ and define the index mapping $\pi_M(j) := \lfloor \frac{j}{M} \rfloor$, $j = 0, \dots, ML-1$. Partitioning each interval $[kh, (k+1)h]$ into M subintervals produces an (ML)-step refined grid. Let $\{y_j^{L,M}\}_{j=0}^{ML}$ denote its hidden-states, with $y_0^{L,M} = x_0^{L,*}$. Applying forward Euler to the simple-continuation gives

$$y_{j+1}^{L,M} = y_j^{L,M} + \delta f_L\big(y_j^{L,M}, \theta_{\pi_M(j)}^{L,*}\big), \qquad j = 0, \dots, ML-1. \tag{4}$$

Equation 4 is the ideal Euler refinement associated with simple continuation. Since $\boldsymbol{\theta}_{\mathrm{simp}}^{L,*}$ is piecewise-constant, equation 4 indicates depth upscaling through in-place layer duplication. Following Du et al. (2024), we refer to M as the growth-factor.

To compare this ideal refinement equation 4 with the serial MHA-FFN implementation, expand f_L into its two residual branches. Since the MHA's output is passed to the FFN in a serial Transformer block, the exact refined Euler update is

$$y_{j+1}^{L,M} = y_j^{L,M} + \delta \tilde{\boldsymbol{A}}\big(y_j^{L,M}, \theta_{\pi_M(j)}^{L,*}\big) + \delta \tilde{\boldsymbol{F}}\left(y_j^{L,M} + h\tilde{\boldsymbol{A}}\big(y_j^{L,M}, \theta_{\pi_M(j)}^{L,*}\big), \theta_{\pi_M(j)}^{L,*}\right).$$

This distinction in the FFN evaluation leads to two practical stacking variants.

Direct stacking. Direct stacking is employed as a baseline in Du et al. (2024). Each Transformer block is duplicated without rescaling its residual outputs by M, as illustrated in Figure 8.

$$\widehat{x}_{j+1}^{L,M} := \widehat{x}_j^{L,M} + h\tilde{\boldsymbol{A}}\big(\widehat{x}_j^{L,M}, \theta_{\pi_M(j)}^{L,*}\big) + h\tilde{\boldsymbol{F}}\left(\widehat{x}_j^{L,M} + h\tilde{\boldsymbol{A}}\big(\widehat{x}_j^{L,M}, \theta_{\pi_M(j)}^{L,*}\big), \theta_{\pi_M(j)}^{L,*}\right).$$

Scaled stacking. Scaled stacking instead divides both residual outputs by M, so that each duplicated block uses the refined step size δ. Let $\{x_j^{L,M}\}_{j=0}^{ML}$ be the hidden-states, with $x_0^{L,M} = x_0^{L,*}$.

$$x_{j+1}^{L,M} := x_j^{L,M} + \delta \tilde{\boldsymbol{A}}\big(x_j^{L,M}, \theta_{\pi_M(j)}^{L,*}\big) + \delta \tilde{\boldsymbol{F}}\left(x_j^{L,M} + \delta\tilde{\boldsymbol{A}}\big(x_j^{L,M}, \theta_{\pi_M(j)}^{L,*}\big), \theta_{\pi_M(j)}^{L,*}\right).$$

Both variants approximate the ideal Euler refinement. However, the resulting perturbation can be controlled for scaled stacking, and its accumulation over the ML refined steps remains of order $1/L$ uniformly in M.

Theorem 3. *For $L \geq 2$ and $M \geq 1$, let $\{x_j^{L,M}\}_{j=0}^{ML}$ be the states obtained by applying scaled stacking to a depth-L model with growth-factor M. Under Assumptions 1 and 2, the following bounds hold uniformly over all growth-factors:*

$$\sup_{M \in \mathbb{N}_+} \|x_{ML}^{L,M} - \mathbf{x}_{\mathrm{simp}}^{L,*}(1)\| = \mathcal{O}(L^{-1}), \qquad \sup_{M \in \mathbb{N}_+} \|x_{ML}^{L,M} - x_L^{L,*}\| = \mathcal{O}(L^{-1}).$$

3.3 Backward Consistency

Forward (zeroth-order) consistency does not determine whether continual training follows the optimization dynamics. We thus characterize the backward (first-order) consistency of depth up-scaling.

The simple-continuation admits a consistent continuous adjoint-state, which is approximated by Lagrange multipliers produced during backpropagation on both the coarse and refined grids. This

adjoint consistency controls the loss differentials and directional derivatives. The state–adjoint system follows from Pontryagin's Minimum Principle (PMP) for almost every $t \in [0,1]$:

$$\mathbf{a}^{L,*}_{\text{simp}}(1) = \partial_{\mathbf{x}} \ell\big(\mathbf{x}^{L,*}_{\text{simp}}(1)\big), \quad \forall \xi \in \Xi, \tag{Terminal}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{a}^{L,*}_{\text{simp}}(t) = -\partial_{\mathbf{x}} f_L\big(\mathbf{x}^{L,*}_{\text{simp}}(t), \boldsymbol{\theta}^{L,*}_{\text{simp}}(t)\big)^T \mathbf{a}^{L,*}_{\text{simp}}(t), \quad \forall \xi \in \Xi, \tag{Adjoint}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}^{L,*}_{\text{simp}}(t) = f_L\big(\mathbf{x}^{L,*}_{\text{simp}}(t), \boldsymbol{\theta}^{L,*}_{\text{simp}}(t)\big), \quad \forall \xi \in \Xi. \tag{Dynamic}$$

Definition 2 (Consistent adjoint-state)**.** *We call $\mathbf{a}^{L,*}_{\text{simp}}$ a consistent adjoint-state associated with $(\mathbf{x}^{L,*}_{\text{simp}}, \boldsymbol{\theta}^{L,*}_{\text{simp}})$ if it satisfies equation Terminal and equation Adjoint.*

Proposition 2 (Adjoint well-definedness)**.** *Under Assumptions 1 and 2, the simple-continuation $(\mathbf{x}^{L,*}_{\text{simp}}, \boldsymbol{\theta}^{L,*}_{\text{simp}})$ of the trained depth-L model admits a unique consistent adjoint-state $\mathbf{a}^{L,*}_{\text{simp}}$.*

Assumption 3. *Suppose that $\partial_x \boldsymbol{R}(x, \theta^{L,*}_k)$ and $\partial_\theta \boldsymbol{R}(x, \theta^{L,*}_k)$, for $\boldsymbol{R} \in \{\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{F}}\}$, are bounded and Lipschitz continuous in x independently of L and k, with $x \in K_A$ and K_A defined in Lemma 1.*

Proposition 3. *Under Assumptions 1–3, the Lagrange multipliers $\{\lambda^{L,*}_k\}_{k=0}^L$ implicitly generated by backpropagation at the trained depth-L model, together with the hidden-states $\{x^{L,*}_k\}_{k=0}^L$, form a symplectic discretization of equation Adjoint and equation Dynamic. Moreover,*

$$\max_{0 \le k \le L} \|\lambda^{L,*}_k - \mathbf{a}^{L,*}_{\text{simp}}(kh)\| = \mathcal{O}(L^{-1}), \quad \forall \xi \in \Xi.$$

Proposition 3 indicates that the Lagrange multipliers $\{\lambda^{L,*}_k\}_{k=0}^L$ are consistent with the adjoint-state $\mathbf{a}^{L,*}_{\text{simp}}$. Indeed, the partial derivative of the loss is $\frac{\partial}{\partial \theta^{L,*}_k}\mathbb{E}[\ell(x^{L,*}_L)] = \mathbb{E}[(\lambda^{L,*}_{k+1})^T \frac{\partial x^{L,*}_{k+1}}{\partial \theta^{L,*}_k}]$. Therefore, the consistent multipliers, together with the parameters duplicated through scaled-stacking, yield consistent loss differentials and directional derivatives after depth up-scaling.

Theorem 4 (First-order consistency)**.** *Suppose Assumptions 1–3 hold. Consider the ML-layer scaled-stacking up-scaling of a trained depth-L model. Initialize $\theta^{L,M}_j = \theta^{L,*}_{\pi_M(j)}$ and denote the corresponding Lagrange multipliers by $\{\lambda^{L,M}_j\}_{j=0}^{ML}$. Then*

$$\sup_{M \in \mathbb{N}_+} \max_{0 \le j \le ML} \|\lambda^{L,M}_j - \mathbf{a}^{L,*}_{\text{simp}}(j\delta)\| = \mathcal{O}(L^{-1}), \qquad \forall \xi \in \Xi,$$

$$\sup_{M \in \mathbb{N}_+} \max_{0 \le j < ML} \left\| \frac{1}{\delta} \frac{\partial}{\partial \theta^{L,M}_j} \mathbb{E}[\ell(x^{L,M}_{ML})] - \frac{1}{h} \frac{\partial}{\partial \theta^{L}_{\pi_M(j)}} \mathbb{E}[\ell(x^{L,*}_L)] \right\| = \mathcal{O}(L^{-1}).$$

For base-model directions v^L_k, define their refinement by $v^{L,M}_j := v^L_{\pi_M(j)}$. Then

$$\sup_{M \in \mathbb{N}_+} \left| \sum_{j=0}^{ML-1} \left\langle \frac{\partial}{\partial \theta^{L,M}_j} \mathbb{E}[\ell(x^{L,M}_{ML})], v^{L,M}_j \right\rangle - \sum_{k=0}^{L-1} \left\langle \frac{\partial}{\partial \theta^{L}_k} \mathbb{E}[\ell(x^{L,*}_L)], v^L_k \right\rangle \right| = \mathcal{O}(L^{-1}) \sum_{k=0}^{L-1} h \|v^L_k\|.$$

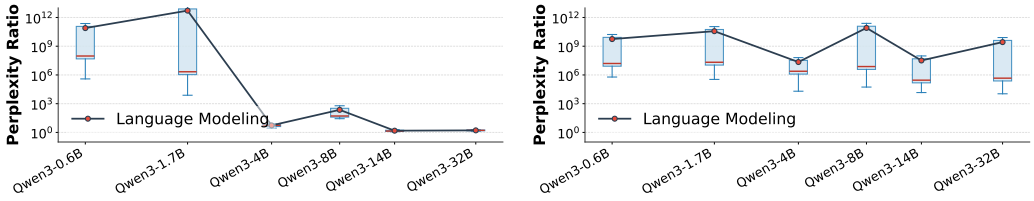
The refined directional derivative thus remains negative along the inherited negative direction of the base model given large depth. For AdamW, this motivates us to scale the copied first-order moments by $1/M$ and the second-order moments by $1/M^2$. Then, the descent direction is guaranteed, eliminating the need for a warm-up at the next stage.

4 Experiments

We evaluate the observable consequences of the two consistency notions. Immediate post-growth behavior probes zeroth-order preservation, while continual training examines both zeroth- and first-order consistency. For reproducibility, the immediate post-growth behavior is tested on open-source models, while the continual training is conducted using the nanoGPT project.

POST-GROWTH PERFORMANCE

We evaluate each growth rule before any additional training, thereby isolating the quality of the up-scaled initialization. Our primary testbed is the dense Qwen3 family ranging from 0.6B to 32B parameters (Yang et al., 2025), evaluated in BF16 with `lm-evaluation-harness` (Gao et al., 2023) on 14 accuracy, exact-match, and perplexity metrics. Scores are normalized by the corresponding base-model scores: 100% indicates preservation for accuracy and exact match, while 1 indicates preservation for perplexity. Given that Qwen3-0.6B and Qwen3-1.7B both have 28 blocks, and Qwen3-4B and Qwen3-8B both have 36 blocks, trends indexed by model-scale need not be monotone. Architectures, tasks, and metric definitions are listed in Appendix B.3.



(a) Simple continuation. (b) Linear interpolation.

Figure 2: Immediate perplexity ratios after doubling the depth of dense Qwen3 models ($M = 2$); the vertical axis is logarithmic, and each box summarizes the three language-modeling benchmarks. Simple-continuation approaches the base-model level at 14B and 32B, whereas linear interpolation remains severely degraded across the evaluated checkpoints.

Simple-continuation versus linear interpolation. We first test the prediction that simple-continuation preserves immediate model quality in the deep regime, whereas linear parameter interpolation need not do so. To implement the ideal Euler update on the refined grid for a serial Transformer while retaining the KV cache, we use the update rule:

$$\hat{y}_{j+1}^{L,M} = \hat{y}_j^{L,M} + \delta\tilde{\boldsymbol{A}}\big(\hat{y}_j^{L,M}, \theta^{L,*}_{\pi_M(j)}\big) + \delta\tilde{\boldsymbol{F}}\left(\hat{y}_j^{L,M} + M\delta\tilde{\boldsymbol{A}}\big(\hat{y}_j^{L,M}, \theta^{L,*}_{\pi_M(j)}\big), \theta^{L,*}_{\pi_M(j)}\right).$$

Up to BF16 arithmetic, this construction implements the refined Euler update with step size $1/(ML)$. We use a growth-factor of $M = 2$ in this experiment.

As shown in Figure 2, simple-continuation is highly inaccurate for the shallower checkpoints but approaches the base-model perplexity at 14B and 32B. Linear interpolation remains unstable throughout the evaluated range. The remaining results in Figure 10, together with the practical implementations in Figure 11, show the same qualitative separation. These results support the predicted deep-regime stabilization and show that interpolation between adjacent trained layers is not a reliable initialization.

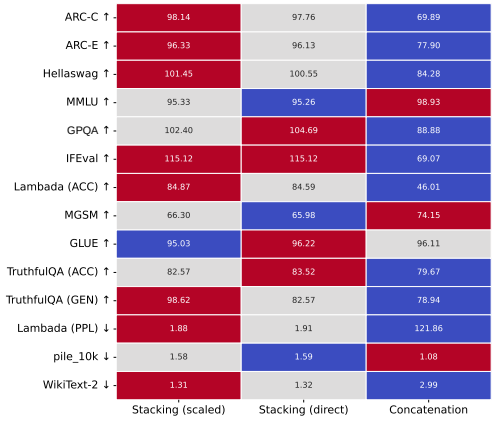


Figure 3: Immediate Qwen3-32B performance relative to the base model. Scaled stacking leads direct stacking in 10 of 14 metrics, and both stacking variants outperform concatenation on most metrics.

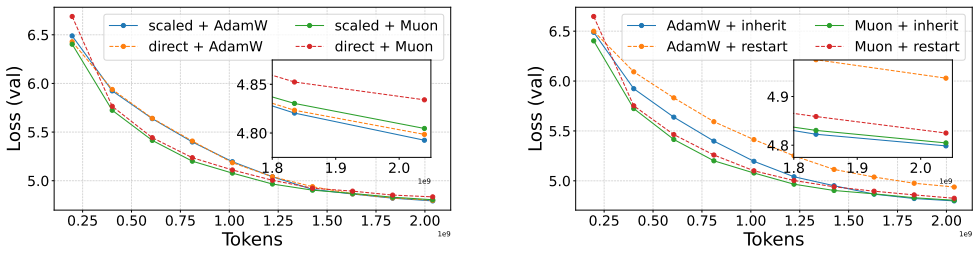
Stacking versus concatenation. We next compare direct stacking, scaled stacking, and concatenation. At 32B (Figure 3), the two stacking variants retain the highest accuracy and remain stable. Concatenation degrades generative metrics and increases Lambada perplexity by a factor of 121.86. Scaled stacking attains better values on 10 of 14 metrics than direct stacking. Across the dense Qwen3 family, Figure 12 shows a scale-ladder trend: concatenation can be stable for shallow checkpoints but degrade for deeper ones, whereas stacking recovers sharply in the deep regime.

Additional comparisons across growth-factors and on an MoE model are reported in Appendix B.5.

CONTINUAL TRAINING

We study continual training with GPT-2 models[1] (Radford et al., 2019) on OpenWebText (Gokaslan & Cohen, 2019). Given that the speedup of concatenation-based progressive training over training from scratch is well-demonstrated (Du et al., 2024), we restrict our comparison to stacking against concatenation. Starting with compact yet deep models, we demonstrate that the predicted consistency is independent of model scale. Each comparison expands the same 12.6M, 64-layer checkpoint to 25.2M and 128 layers. We then scale our experiments to a 0.9B, 512-layer model, maintaining the depth-to-width proportions in the small-scale setup.

Figure 4 examines the two predictions derived from consistency. Scaled stacking outperforms direct stacking, and optimizer-state inheritance benefits both AdamW (Loshchilov & Hutter, 2019) and Muon (Jordan et al., 2024). The improvement is especially marked for AdamW, which overtakes Muon with optimizer-state inheritance. Refer to Figures 15 and 16 for detailed comparisons.



(a) Residual scaling.

(b) Optimizer-state inheritance.

Figure 4: Validation loss after the 12.6M-to-25.2M depth up-scaling. Left: scaled stacking lowers loss compared to direct stacking under both Muon and AdamW. Right: inheriting the descent-direction lowers loss relative to restart for both optimizers, with a much larger gain under AdamW.

As illustrated in Figure 5, concatenation yields a larger initial loss spike and slower recovery following up-scaling. Conversely, scaled stacking induces milder perturbation, and achieves lower loss and higher accuracy after continual training across both optimizers.

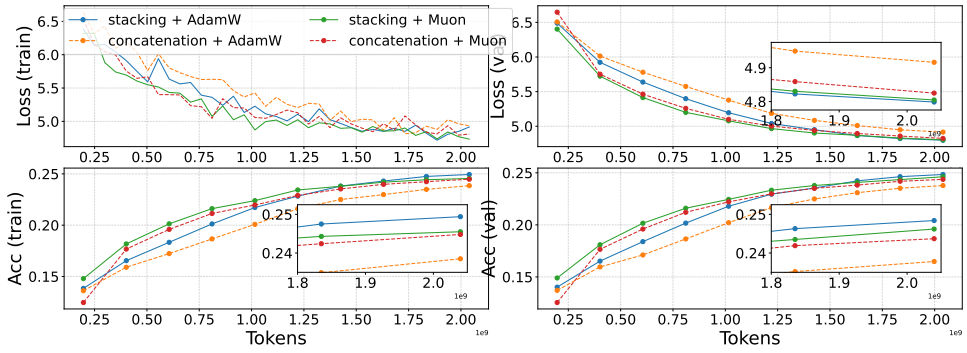


Figure 5: Continual training for the 12.6M, 64-layer to 25.2M, 128-layer expansion. Scaled stacking achieves lower loss and higher accuracy than concatenation under both AdamW and Muon.

In our large-scale experiments, we upscale a 453.2M, 256-layer checkpoint to 906.4M and 512 layers, followed by next-stage training on 10B tokens. Based on the observation from smaller-scale experiments that optimizer-state inheritance provides a particularly pronounced speedup for

[1] https://github.com/karpathy/nanogpt

AdamW, these experiments are conducted using AdamW. As shown in Figure 6, concatenation causes a much larger loss spike and slow recovery, whereas scaled stacking consistently outperforms concatenation, confirming that its advantage holds at the 0.9B scale.

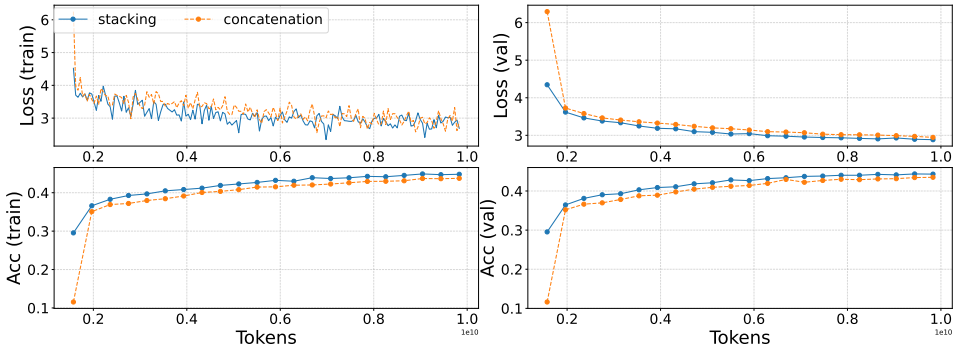


Figure 6: Continual training for the 453.2M, 256-layer to 906.4M, 512-layer expansion. Under AdamW, scaled stacking achieves lower loss and higher accuracy than concatenation.

5 Conclusions

We study Transformer depth up-scaling through a controlled-ODE formulation. Under this perspective, piecewise-constant simple-continuation is consistent in preserving both next-token predictions and descent directions. This motivates duplicating layers in place, scaling their residual outputs, and inheriting optimizer state. Specifically, for depth-L base models, scaled stacking approximates the corresponding Euler refinement with an $\mathcal{O}(L^{-1})$ error order, uniformly over growth-factors. Moreover, the ODE adjoint under simple-continuation is consistent with the Lagrange multipliers, providing a criterion for preserving descent directions and helping stabilize continual training after growth. Experiments further support these zeroth- and first-order consistency results.

6 Discussions

Zeroth-order consistency is relevant to looped Transformers, which reuse layers across recurrent inference steps (Dehghani et al., 2019; Geiping et al., 2025). Recent studies propose using layer-loops rather than block/model-loops (Gao et al., 2026), which coincides with in-place stacking in terms of zeroth-order consistency. Our analysis also advances the understanding of this regime with an asymptotic guarantee for deeper models: the residual-scaled layer-loop has $\mathcal{O}(L^{-1})$ terminal-state and directional-derivative errors for depth-L models. Experiments on the 40- and 64-layer Qwen3 models extend current empirical results (Chen et al., 2026), providing further evidence for the validity of layer-loops.

Prior studies of ResNet limits focus on the convergence of hidden-state or parameter trajectories (Sander et al., 2022). Instead, our objective is to identify the convergence of the terminal state, which suffices for preserving model outputs. Therefore, the convergence of the trajectories across depths does not matter. However, these studies still leave open how or whether general residual networks converge to ODE or SDE limits (Cohen et al., 2021; Barboni et al., 2025).

The coarse-to-fine structure connects with multigrid methods for ODE-constrained optimization (Borzì & Schulz, 2009; Chang et al., 2018; Zou et al., 2024). Specifically, a trained shallow model acts as a coarse-grid solution whose controls are prolonged to a finer discretization before next-stage training. Our construction performs a single coarse-to-fine transfer, while a classical multigrid cycle would additionally alternate restriction, prolongation, and coarse-grid correction. Extending the approach with such cycles is a natural direction. However, the nonlinear Transformer residual dynamics make the resulting optimization complexity difficult to characterize. Deriving rigorous complexity bounds and establishing provable computational gains remain open problems.

AI USE STATEMENT

In this work, generative AI tools were used solely for assistance in literature review, initial text drafting, code development and debugging, and verifying mathematical proofs. Generative AI was not employed for conceptualizing research ideas, deriving mathematical proofs, conducting theoretical analysis, or designing and analyzing experiments. All AI-assisted outputs were rigorously reviewed and validated by the authors. Specifically, LLM-generated literature references were manually verified against original sources to prevent hallucinations and plagiarism, and generated code was thoroughly tested for functional correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1352–1361. PMLR, 2021. URL https://proceedings.mlr.press/v161/bachlechner21a.html.

Raphaël Barboni, Gabriel Peyré, and François-Xavier Vialard. Understanding the training of infinitely deep and wide ResNets with conditional optimal transport. *Communications on Pure and Applied Mathematics*, 78(11):2149–2205, 2025. doi: 10.1002/cpa.70004.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, volume 19, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/5da713a690c067105aeb2fae32403405-Abstract.html.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pp. 95–136. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.bigscience-1.9.

Alfio Borzì and Volker Schulz. Multigrid methods for PDE optimization. *SIAM Review*, 51(2): 361–395, 2009. doi: 10.1137/060671590.

Bo Chang, Lili Meng, Eldad Haber, Frederick Tung, and David Begert. Multi-level residual networks from dynamical systems view. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyJS-OgR-.

Lizhang Chen, Jonathan Li, Chen Liang, Ni Lao, and Qiang Liu. Training-free looped transformers. *arXiv preprint arXiv:2605.23872*, 2026. URL https://arxiv.org/abs/2605.23872.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

Tianqi Chen, Ian J. Goodfellow, and Jonathon Shlens. Net2Net: Accelerating learning via knowledge transfer. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.05641.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with Pathways. *Journal of Machine Learning Research*, 24(240): 1–113, 2023. URL https://jmlr.org/papers/v24/22-1144.html.

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. URL https://arxiv.org/abs/1803.05457.

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. URL https://arxiv.org/abs/2110.14168.

Alain-Sam Cohen, Rama Cont, Alain Rossier, and Renyuan Xu. Scaling properties of deep residual networks. In *Proceedings of the International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2039–2048. PMLR, 2021. URL https://proceedings.mlr.press/v139/cohen21b.html.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Chengyu Dong, Liyuan Liu, Zichao Li, and Jingbo Shang. Towards adaptive residual network training: A Neural-ODE perspective. In *Proceedings of the International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2616–2626. PMLR, 2020. URL https://proceedings.mlr.press/v119/dong20c.html.

Wenyu Du, Tongxu Luo, Zihan Qiu, Zeyu Huang, Yikang Shen, Reynold Cheng, Yike Guo, and Jie Fu. Stacking your transformers: A closer look at model growth for efficient LLM pre-training. In *Advances in Neural Information Processing Systems*, volume 37, pp. 10491–10540, 2024. doi: 10.52202/079017-0336.

Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017. doi: 10.1007/s40304-017-0103-z.

Weinan E, Chao Ma, and Lei Wu. Machine learning from a continuous viewpoint, i. *Science China Mathematics*, 63(11):2233–2266, 2020. doi: 10.1007/s11425-020-1773-8.

Weinan E, Chao Ma, and Lei Wu. The Barron space and the flow-induced function spaces for neural network models. *Constructive Approximation*, 55(1):369–406, 2022. doi: 10.1007/s00365-021-09549-y.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020. URL https://arxiv.org/abs/2101.00027.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, et al. A framework for few-shot language model evaluation, December 2023. Version v0.4.0.

Zitian Gao, Yilong Chen, Yihao Xiao, Xinyu Yang, Ran Tao, Joey Zhou, and Bryan Dai. Loop the Loopies! *arXiv preprint arXiv:2607.16051*, 2026. URL https://arxiv.org/abs/2607.16051.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *Advances in Neural Information Processing Systems*, volume 38, pp. 41340–41391, 2025. doi: 10.52202/085713-1380.

Aaron Gokaslan and Vanya Cohen. OpenWebText corpus, 2019. URL https://skylion007.github.io/OpenWebTextCorpus/.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2018. doi: 10.1088/1361-6420/aa9a90.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning AI with shared human values. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=dNy_RKzJacY.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006. doi: 10.1162/neco.2006.18.7.1527.

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. URL https://arxiv.org/abs/2203.15556.

Lingzi Jin, Xiao Wang, and Xiaojun Chen. Nonconvex nonsmooth multicomposite optimization and its applications to recurrent neural networks. *SIAM Journal on Optimization*, 35(4):2343–2371, 2025. doi: 10.1137/25M1728168.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

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. URL https://arxiv.org/abs/2001.08361.

Sanghoon Kim, Dahyun Kim, Chanjun Park, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, et al. SOLAR 10.7B: Scaling large language models with simple yet effective depth up-scaling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pp. 23–35. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-industry.3.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. Shallow-to-deep training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 995–1005. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.72.

Changlin Li, Bohan Zhuang, Guangrun Wang, Xiaodan Liang, Xiaojun Chang, and Yi Yang. Automated progressive learning for efficient training of vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12486–12496, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Li_Automated_Progressive_Learning_for_Efficient_Training_of_Vision_Transformers_CVPR_2022_paper.html.

Qianxiao Li and Shuji Hao. An optimal control approach to deep learning and applications to discrete-weight neural networks. In *Proceedings of the International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2985–2994. PMLR, 2018. URL https://proceedings.mlr.press/v80/li18b.html.

Qianxiao Li, Long Chen, Cheng Tai, and Weinan E. Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*, 18(165):1–29, 2018. URL https://jmlr.org/papers/v18/17-653.html.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.229.

Wei Liu, Xin Liu, and Xiaojun Chen. An inexact augmented Lagrangian algorithm for training leaky ReLU neural network with group sparsity. *Journal of Machine Learning Research*, 24(212):1–43, 2023. URL https://jmlr.org/papers/v24/22-0491.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *Proceedings of the International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3276–3285. PMLR, 2018. URL https://proceedings.mlr.press/v80/lu18d.html.

Pierre Marion, Yu-Han Wu, Michael Sander, and Gérard Biau. Implicit regularization of deep residual networks towards neural ODEs. In *International Conference on Learning Representations*, volume 2024, pp. 40795–40834, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/b29a95e7d9f1e1a6dfc567b556733744-Abstract-Conference.html.

Meituan LongCat Team, Bayan, Bei Li, Bingye Lei, Bo Wang, Bolin Rong, Chao Wang, Chao Zhang, Chen Gao, Chen Zhang, et al. LongCat-Flash technical report. *arXiv preprint arXiv:2509.01322*, 2025. URL https://arxiv.org/abs/2509.01322.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Yu Pan, Ye Yuan, Yichun Yin, Jiaxin Shi, Zenglin Xu, Ming Zhang, Lifeng Shang, Xin Jiang, and Qun Liu. Preparing lessons for progressive training on language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18860–18868, 2024. doi: 10.1609/aaai.v38i17.29851.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1144.

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

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Lars Ruthotto and Eldad Haber. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 62(3):352–364, 2020. doi: 10.1007/s10851-019-00903-1.

Mohammad Samragh, Seyed Iman Mirzadeh, Keivan Alizadeh-Vahid, Fartash Faghri, Minsik Cho, Moin Nabi, Devang Naik, and Mehrdad Farajtabar. Scaling smart: Accelerating large language model pre-training with small model initialization. In *Proceedings of the 4th NeurIPS Efficient Natural Language and Speech Processing Workshop*, volume 262 of *Proceedings of Machine Learning Research*, pp. 1–13. PMLR, 2024. URL https://proceedings.mlr.press/v262/samragh24a.html.

Michael Sander, Pierre Ablin, and Gabriel Peyré. Do residual neural networks discretize neural ordinary differential equations? In *Advances in Neural Information Processing Systems*, volume 35, pp. 36520–36532, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/ecc38927fe5148c66bee64ee8fed1e76-Abstract-Conference.html.

Sheng Shen, Pete Walsh, Kurt Keutzer, Jesse Dodge, Matthew Peters, and Iz Beltagy. Staged training for transformer language models. In *Proceedings of the International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19893–19908. PMLR, 2022. URL https://proceedings.mlr.press/v162/shen22f.html.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. Language models are multilingual chain-of-thought reasoners. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=fR3wGCk-IXp.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

François-Xavier Vialard, Roland Kwitt, Susan Wei, and Marc Niethammer. A shooting formulation of deep learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11828–11838, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/89562dccfeb1d0394b9ae7e09544dc70-Abstract.html.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355. Association for Computational Linguistics, 2018. doi: 10.18653/v1/W18-5446.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 billion parameter autoregressive language model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. DeepNet: Scaling Transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6761–6774, 2024a. doi: 10.1109/TPAMI.2024.3386927.

Yite Wang, Jiahao Su, Hanlin Lu, Cong Xie, Tianyi Liu, Jianbo Yuan, Haibin Lin, Ruoyu Sun, and Hongxia Yang. LEMON: Lossless model expansion. In *International Conference on Learning Representations*, volume 2024, pp. 3338–3367, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/0e705ac30e573d1526f81a0fd071a151-Abstract-Conference.html.

Yue Wang, Chao Zhang, and Xiaojun Chen. An augmented Lagrangian method for training recurrent neural networks. *SIAM Journal on Scientific Computing*, 47(1):C22–C51, 2025. doi: 10.1137/23M1627614.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ying Shan, and Ping Luo. LLaMA Pro: Progressive LLaMA with block expansion. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6518–6537. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.352.

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. URL https://arxiv.org/abs/2505.09388.

Kazuki Yano, Sho Takase, Sosuke Kobayashi, Shun Kiyono, and Jun Suzuki. Efficient construction of model family through progressive training using model expansion. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=fuBrcTH8NM.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. Yi: Open foundation models by 01.AI. *arXiv preprint arXiv:2403.04652*, 2024. URL https://arxiv.org/abs/2403.04652.

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*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

Longwei Zou, Han Zhang, and Yangdong Deng. A multi-level framework for accelerating training transformer models. In *International Conference on Learning Representations*, volume 2024, pp. 12205–12228, 2024.

Transformer Depth Up-Scaling: An Infinite-Depth Perspective —Supplementary Materials—

CONTENTS

1 Introduction **1**

2 Background **2**

3 Consistent Depth Up-Scaling for Transformers **4**

3.1 Forward Consistency 4

3.2 Depth Up-Scaling via Grid Refinement 5

3.3 Backward Consistency 5

4 Experiments **6**

5 Conclusions **9**

6 Discussions **9**

A Theoretical Analysis **17**

A.1 Forward Regularity and Well-Definedness 17

A.2 Forward Consistency 18

A.3 Backward Regularity and Well-Definedness 21

A.4 Backward Consistency 22

B Experiments **27**

B.1 Depth Up-Scaling Diagrams 27

B.2 Model Architectures 27

B.3 Experimental Setup 28

B.4 An Empirical Diagnostic for Residual Scaling 29

B.5 Additional Experiments 31

B.5.1 Linear-Interpolation Continuations 31

B.5.2 Stacking and Concatenation across Dense Models 33

B.5.3 Growth-Factor Sensitivity 33

B.5.4 Mixture-of-Experts Model 34

B.5.5 Continual Training Experiments 34

A Theoretical Analysis

A.1 Forward Regularity and Well-Definedness

Lemma 1. *Under Assumptions 1 and 2, all trained hidden-states $\{x_k^{L,*}\}_{k=0}^{L}$ lie in a fixed compact set K_x for every $L \geq 2$. There is also a fixed compact set K_A containing $x + s\tilde{\boldsymbol{A}}(x, \theta_k^{L,*})$ for every $L \geq 2$, $k = 0, \ldots, L-1$, $x \in K_x$, and $s \in [0, 1/L]$. Moreover, on K_x, $f_L(\cdot, \theta_k^{L,*})$ is uniformly bounded and globally Lipschitz in the state variable, with constants independent of L and k.*

Proof. Set $M_f := M_A + M_F$. Assumption 2 gives, for every x, L, and k,

$$\|f_L(x, \theta_k^{L,*})\| \leq \|\tilde{\boldsymbol{A}}(x, \theta_k^{L,*})\| + \left\|\tilde{\boldsymbol{F}}\left(x + \frac{1}{L}\tilde{\boldsymbol{A}}(x, \theta_k^{L,*}), \theta_k^{L,*}\right)\right\| \leq M_f.$$

By Assumption 1 and the finiteness of the data domain, there is $M_0 > 0$, independent of L and the input z_0, such that $\|x_0\| = \|z_0(\xi)V_{\mathrm{em}}^{L,*}\| \leq M_0$. Since the trained update is $x_{k+1}^{L,*} = x_k^{L,*} + L^{-1}f_L(x_k^{L,*}, \theta_k^{L,*})$,

$$\|x_k^{L,*}\| \leq M_0 + \frac{k}{L}M_f \leq M_0 + M_f =: R.$$

Hence, all trained states lie in the compact ball $K_x := \overline{B}_R(0)$.

For $x \in K_x$ and $s \in [0, 1/L]$, every relevant MHA-shifted state satisfies

$$\left\|x + s\tilde{\boldsymbol{A}}(x, \theta_k^{L,*})\right\| \leq R + M_A.$$

Define the compact set $K_A := \overline{B}_{R+M_A}(0)$, on which we have uniform Lipschitz continuity:

$$\begin{aligned}
\|\tilde{\boldsymbol{A}}(x, \theta_k^{L,*}) - \tilde{\boldsymbol{A}}(y, \theta_k^{L,*})\| &\leq L_A\|x - y\|, && \forall x, y \in K_x, \\
\|\tilde{\boldsymbol{F}}(u, \theta_k^{L,*}) - \tilde{\boldsymbol{F}}(v, \theta_k^{L,*})\| &\leq L_F\|u - v\|, && \forall u, v \in K_A.
\end{aligned}$$

For $x, y \in K_x$, define $\tilde{x} = x + L^{-1}\tilde{\boldsymbol{A}}(x, \theta_k^{L,*})$ and $\tilde{y} = y + L^{-1}\tilde{\boldsymbol{A}}(y, \theta_k^{L,*})$. Then

$$\begin{aligned}
\|f_L(x, \theta_k^{L,*}) - f_L(y, \theta_k^{L,*})\| &\leq L_A\|x - y\| + L_F\|\tilde{x} - \tilde{y}\| \\
&\leq \left[L_A + L_F\left(1 + \frac{L_A}{L}\right)\right]\|x - y\| \\
&\leq \left[L_A + L_F(1 + L_A)\right]\|x - y\|.
\end{aligned}$$

Thus f_L is globally Lipschitz on K_x with the depth-uniform constant $L_f := L_A + L_F(1 + L_A)$, and its uniform bound on K_x is M_f. □

Lemma 2. *The ODE $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = f_L(\mathbf{x}, \boldsymbol{\theta}_{\mathrm{simp}}^{L,*}(t))$ with $\|\mathbf{x}(0)\| \leq M_0$ admits a unique Carathéodory solution $\mathbf{x}_{\mathrm{simp}}^{L,*}(t)$ on $[0, 1]$. The solution $\mathbf{x}_{\mathrm{simp}}^{L,*}(t)$ remains in the compact set $K_x = \overline{B}_{M_0+M_f}(0)$ defined in Lemma 1 for all $t \in [0, 1]$. Moreover, along this trajectory, $f_L(\mathbf{x}_{\mathrm{simp}}^{L,*}(t), \boldsymbol{\theta}_{\mathrm{simp}}^{L,*}(t))$ is uniformly bounded by M_f and Lipschitz in the state variable with constant L_f.*

Proof. By construction, the simple-continuation control $\boldsymbol{\theta}_{\mathrm{simp}}^{L,*}(t)$ takes values in the finite set $\{\theta_k^{L,*} : 0 \leq k < L\}$ and hence the same depth-uniform bounds apply. Since $f_L(\mathbf{x}_{\mathrm{simp}}^{L,*}(t), \boldsymbol{\theta}_{\mathrm{simp}}^{L,*}(t))$ is measurable in time and uniformly locally Lipschitz in the state, the standard Carathéodory theorem gives a unique absolutely continuous solution. Hence, for almost every $t \in [0, 1]$ and every state x, $\|f_L(x, \boldsymbol{\theta}_{\mathrm{simp}}^{L,*}(t))\| \leq M_f$.

Since the Carathéodory solution $\mathbf{x}_{\mathrm{simp}}^{L,*}(t)$ is absolutely continuous, it satisfies

$$\|\mathbf{x}_{\mathrm{simp}}^{L,*}(t)\| = \left\|\mathbf{x}_{\mathrm{simp}}^{L,*}(0) + \int_0^t f_L(\mathbf{x}_{\mathrm{simp}}^{L,*}(s), \boldsymbol{\theta}_{\mathrm{simp}}^{L,*}(s))\,\mathrm{d}s\right\| \leq M_0 + \int_0^t M_f\,\mathrm{d}s = M_0 + M_f t.$$

Therefore, for every $t \in [0,1]$,

$$\|\mathbf{x}_{\text{simp}}^{L,*}(t)\| \le M_0 + M_f = R.$$

Thus, $\mathbf{x}_{\text{simp}}^{L,*}(t) \in K_x$ for all $t \in [0,1]$. Because the trajectory remains in K_x and the control takes values in $\{\theta_k^{L,*} : 0 \le k < L\}$, Lemma 1 supplies the same uniform Lipschitz constant L_f along the entire trajectory. □

A.2 Forward Consistency

Proof of Theorem 1

Theorem 1. *Suppose Assumption 1 holds. If a weak continuation* $(\mathbf{x}^{L,*}, \boldsymbol{\theta}^{L,*})$ *satisfies* $\lim_{L\to\infty} \|\mathbf{x}^{L,*}(1) - x_L^{L,*}\| = 0$ *for every* $z_0(\xi) \sim p_{\text{data}}$*, then*

$$\lim_{L\to\infty} \mathbb{E}_{z_0\sim p_{\text{data}}} \Big[D_{\text{KL}}\big(p^{L,*}(\cdot|z_0) \,\|\, p_{\text{cont}}^{L,*}(\cdot|z_0)\big)\Big] = 0,$$

where $p^{L,*} = \text{softmax}(\text{Norm}(x_L^{L,*})V_{\text{de}}^{L,*})$ *and* $p_{\text{cont}}^{L,*} = \text{softmax}(\text{Norm}(\mathbf{x}^{L,*}(1))V_{\text{de}}^{L,*})$.

Proof. Recall the construction of the loss function in equation 1. For a fixed z_0, define the discrete and continued logits by

$$u_L := \text{Norm}(x_L^{L,*})V_{\text{de}}^{L,*}, \qquad v_L := \text{Norm}(\mathbf{x}^{L,*}(1))V_{\text{de}}^{L,*}.$$

Given that the normalization term is Lipschitz continuous, Assumption 1 gives

$$\|v_L - u_L\| \le L_{\text{Norm}} M_V \|\mathbf{x}^{L,*}(1) - x_L^{L,*}\| \xrightarrow{L\to\infty} 0.$$

Let $G_L(z) := D_{\text{KL}}(p^{L,*}(\cdot|z_0) \,\|\, \text{softmax}(z))$. Since $\nabla G_L(z) = \text{softmax}(z) - p^{L,*}(\cdot|z_0)$, its norm is bounded by 2, independent of L and z_0. As $G_L(u_L) = 0$,

$$0 \le D_{\text{KL}}\big(p^{L,*}(\cdot|z_0) \,\|\, p_{\text{cont}}^{L,*}(\cdot|z_0)\big) \le 2\|v_L - u_L\| \longrightarrow 0.$$

The evaluation domain is finite, so the convergence also holds after taking the expectation. □

CounterExample

Example 1 (Counterexample for linear interpolation)**.** *Consider the optimal control problem*

$$\min_{\boldsymbol{\theta}} \ \ell(\mathbf{x}(1)) := (\mathbf{x}(1) - e^2)^2$$
$$\text{s.t. } \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = f_L(\mathbf{x}, \boldsymbol{\theta}) \equiv \boldsymbol{\theta}^2\mathbf{x}, \qquad \mathbf{x}(0) = 1.$$

Let $h = 1/L$*, so the forward-Euler discretization is*

$$x_{k+1}^L = x_k^L + h(\theta_k^L)^2 x_k^L = x_k^L\big(1 + h(\theta_k^L)^2\big).$$

For every even $L \ge 4$*, set*

$$\theta_k^{L,*} = \begin{cases} 2, & \textit{even } k < L-2, \\ 0, & \textit{odd } k < L-2. \end{cases}$$

The first $L-2$ *steps give* $x_{L-2}^{L,*} = (1+4h)^{(L-2)/2}$*. Then choose* $\theta_{L-2}^{L,*} = \theta_{L-1}^{L,*} = a$ *such that*

$$x_{L-2}^{L,*}(1 + ha^2)^2 = e^2.$$

Such an $a \in [0,2]$ *exists for all sufficiently large even* L*, and in fact* $a \to 2$ *as* $L \to \infty$*. Hence the discrete trajectory attains the optimal terminal state* $x_L^{L,*} = e^2$.

Next, linearly interpolate the controls between adjacent layers. Away from the final two intervals, each pair of controls gives the integral

$$\int_0^h \left(2 - \frac{2s}{h}\right)^2 \mathrm{d}s + \int_0^h \left(\frac{2s}{h}\right)^2 \mathrm{d}s = \frac{8h}{3}.$$

Summing up the integrals yields

$$\int_0^1 (\boldsymbol{\theta}_{\text{lin}}^{L,*}(t))^2\, dt = (L-3)\frac{4h}{3} + \frac{7}{3}ha^2 \longrightarrow \frac{4}{3}.$$

The trajectory driven by the linear interpolation consequently satisfies

$$\mathbf{x}_{\text{lin}}^{L,*}(1) = \exp\left(\int_0^1 (\boldsymbol{\theta}_{\text{lin}}^{L,*}(t))^2\, \mathrm{d}t\right) \longrightarrow e^{4/3} \neq e^2.$$

Thus the discrete controls attain the optimal terminal state, whereas their linear interpolation fails the terminal-state criterion.

For comparison, let $\boldsymbol{\theta}^*(t) = 2 \cdot \mathbf{1}_C(t)$, where $C \subseteq [0,1]$ is a Cantor set of Lebesgue measure $1/2$. The resulting hidden-state is $x^*(t) = \exp(4m(C \cap [0,t]))$ with terminal value $x^*(1) = e^2$. This $\boldsymbol{\theta}^*(t)$ is Lebesgue measurable but is even not Riemann integrable, illustrating that the optimal controls need not possess the continuity imposed by linear-interpolation. Moreover, $\boldsymbol{\theta}^*(t)$ is not the limit of any bounded measurable continuation of $\{\theta_k^{L,*}\}_{k=0}^{L-1}$.

PROOF OF THEOREM 2

Theorem 2. *Suppose that the trained parameters $\{\theta_k^{L,*}\}_{k=0}^{L-1}$ satisfy Assumptions 1 and 2. Then*

$$\|\mathbf{x}_{\text{simp}}^{L,*}(1) - x_L^{L,*}\| = \mathcal{O}(L^{-1}).$$

Proof. Consider the ODE driven by the simple-continuation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}_{\text{simp}}^{L,*} = f_L\big(\mathbf{x}_{\text{simp}}^{L,*}, \boldsymbol{\theta}_{\text{simp}}^{L,*}(t)\big).$$

The trained states $\{x_k^{L,*}\}_{k=0}^{L}$ are exactly the forward-Euler iterates with step size $h = 1/L$ for this piecewise-constant control. By Lemma 1, these discrete states lie in K_x. Because the simple control is measurable and takes values in the trained parameter set, Lemma 2 gives a unique Carathéodory solution $\mathbf{x}_{\text{simp}}^{L,*}(t)$ that remains in the same compact set K_x for all $t \in [0,1]$. On this domain, $f_L(\cdot, \boldsymbol{\theta}_{\text{simp}}^{L,*}(t))$ is uniformly bounded by M_f and globally Lipschitz continuous in x with constant L_f.

Define the grid-point error by $e_k := \|\mathbf{x}_{\text{simp}}^{L,*}(kh) - x_k^{L,*}\|$. On each interval $[kh, (k+1)h]$, the exact trajectory satisfies $\mathbf{x}_{\text{simp}}^{L,*}((k+1)h) = \mathbf{x}_{\text{simp}}^{L,*}(kh) + \int_0^h f_L(\mathbf{x}_{\text{simp}}^{L,*}(kh+s), \theta_k^{L,*})\, \mathrm{d}s$. Subtracting the Euler update $x_{k+1}^{L,*} = x_k^{L,*} + hf_L(x_k^{L,*}, \theta_k^{L,*})$ and taking norms gives the recursion

$$\begin{aligned}
e_{k+1} &= \left\|\mathbf{x}_{\text{simp}}^{L,*}(kh) - x_k^{L,*} + \int_0^h \left[f_L(\mathbf{x}_{\text{simp}}^{L,*}(kh+s), \theta_k^{L,*}) - f_L(x_k^{L,*}, \theta_k^{L,*})\right] \mathrm{d}s\right\| \\
&\leq e_k + \int_0^h L_f \|\mathbf{x}_{\text{simp}}^{L,*}(kh+s) - x_k^{L,*}\|\, \mathrm{d}s \\
&\leq e_k + \int_0^h L_f \Big(\|\mathbf{x}_{\text{simp}}^{L,*}(kh+s) - \mathbf{x}_{\text{simp}}^{L,*}(kh)\| + \|\mathbf{x}_{\text{simp}}^{L,*}(kh) - x_k^{L,*}\|\Big)\, \mathrm{d}s \\
&\leq e_k + hL_f e_k + L_f \int_0^h \left(\int_0^s \|f_L(\mathbf{x}_{\text{simp}}^{L,*}(kh+\tau), \theta_k^{L,*})\|\, \mathrm{d}\tau\right) \mathrm{d}s \\
&\leq (1 + hL_f)e_k + L_f M_f \int_0^h s\, \mathrm{d}s \\
&= (1 + hL_f)e_k + \frac{L_f M_f}{2}\, h^2.
\end{aligned}$$

Since $e_0 = 0$, iterating this recursion yields

$$e_L \leq \frac{L_f M_f}{2} h^2 \sum_{j=0}^{L-1} (1 + hL_f)^j \leq \frac{L_f M_f}{2} h^2 L e^{L_f} = \frac{L_f M_f}{2} e^{L_f} \frac{1}{L}.$$

Consequently, $\lim_{L\to\infty} \|\mathbf{x}_{\text{simp}}^{L,*}(1) - x_L^{L,*}\| = 0$. □

PROOF OF THEOREM 3

Theorem 3. *For $L \geq 2$ and $M \geq 1$, let $\{x_j^{L,M}\}_{j=0}^{ML}$ be the states obtained by applying scaled stacking to a depth-L model with growth-factor M. Under Assumptions 1 and 2, the following bounds hold uniformly over all growth-factors:*

$$\sup_{M\in\mathbb{N}_+} \|x_{ML}^{L,M} - \mathbf{x}_{\mathrm{simp}}^{L,*}(1)\| = \mathcal{O}(L^{-1}), \qquad \sup_{M\in\mathbb{N}_+} \|x_{ML}^{L,M} - x_L^{L,*}\| = \mathcal{O}(L^{-1}).$$

Proof. Fix $L \geq 2$ and $M \geq 1$, and set $h := 1/L$ and $\delta := 1/(ML)$. Define

$$g_{L,M}(x,\theta) := \tilde{\boldsymbol{A}}(x,\theta) + \tilde{\boldsymbol{F}}\left(x + \delta\tilde{\boldsymbol{A}}(x,\theta),\theta\right).$$

Thus a scaled-stacking substep with parent-layer indexed by k is $x \mapsto x + \delta g_{L,M}(x,\theta_k^{L,*})$.

We first identify a common compact set. Every refined-Euler or scaled-stacking substep has increment norm at most $\delta(M_A + M_F) = \delta M_f$. Starting from $\|x_0\| \leq M_0$, after at most ML substeps, we obtain

$$\|x\| \leq M_0 + ML\delta M_f = M_0 + M_f = R.$$

Hence all states lie in $K_x = \overline{B}_R(0)$, while for every parent layer k both intermediate arguments $x + h\tilde{\boldsymbol{A}}(x,\theta_k^{L,*})$ and $x + \delta\tilde{\boldsymbol{A}}(x,\theta_k^{L,*})$ lie in $K_A = \overline{B}_{R+M_A}(0)$. The constants L_A, L_F, L_f from Lemma 1 therefore apply uniformly in L and M. In particular, $g_{L,M}$ is Lipschitz on K_x with constant at most L_f.

Scaled stacking versus exact Euler refinement. Let $\{y_j^{L,M}\}_{j=0}^{ML}$ be the exact Euler refinement of the simple-continuation and $\{x_j^{L,M}\}_{j=0}^{ML}$ the scaled-stacking states, with $y_0^{L,M} = x_0^{L,M} = x_0^{L,*}$. For $j = 0,\dots,ML-1$, set $k := \pi_M(j)$; then

$$y_{j+1}^{L,M} = y_j^{L,M} + \delta f_L(y_j^{L,M},\theta_k^{L,*}), \qquad x_{j+1}^{L,M} = x_j^{L,M} + \delta g_{L,M}(x_j^{L,M},\theta_k^{L,*}).$$

For every $x \in K_x$,

$$\begin{aligned}\|g_{L,M}(x,\theta_k^{L,*}) - f_L(x,\theta_k^{L,*})\| &\leq L_F(h-\delta)\|\tilde{\boldsymbol{A}}(x,\theta_k^{L,*})\| \\ &\leq L_F M_A h.\end{aligned}$$

Set $d_j := \|x_j^{L,M} - y_j^{L,M}\|$. Then

$$d_{j+1} \leq (1+\delta L_f)d_j + \delta L_F M_A h, \qquad d_0 = 0,$$

and hence by Gronwall's inequality,

$$d_{ML} \leq ML\delta L_F M_A h(1+\delta L_f)^{ML} \leq L_F M_A e^{L_f} h.$$

This bound is independent of the growth-factor M.

Exact Euler refinement versus the simple-continuation. Set $e_j := \|\mathbf{x}_{\mathrm{simp}}^{L,*}(j\delta) - y_j^{L,M}\|$. For $j = 0,\dots,ML-1$, set $k := \pi_M(j)$; then $[j\delta,(j+1)\delta] \subseteq [kh,(k+1)h]$, and the control equals $\theta_k^{L,*}$ almost everywhere on this refined interval. Lemma 2 gives $\|\mathbf{x}_{\mathrm{simp}}^{L,*}(j\delta+s) - \mathbf{x}_{\mathrm{simp}}^{L,*}(j\delta)\| \leq M_f s$ for $0 \leq s \leq \delta$. Therefore, the local truncation error satisfies

$$\begin{aligned}&\left\|\mathbf{x}_{\mathrm{simp}}^{L,*}((j+1)\delta) - \mathbf{x}_{\mathrm{simp}}^{L,*}(j\delta) - \delta f_L(\mathbf{x}_{\mathrm{simp}}^{L,*}(j\delta),\theta_k^{L,*})\right\| \\ &\qquad \leq \int_0^\delta L_f\|\mathbf{x}_{\mathrm{simp}}^{L,*}(j\delta+s) - \mathbf{x}_{\mathrm{simp}}^{L,*}(j\delta)\|\,\mathrm{d}s \leq \frac{L_f M_f}{2}\delta^2.\end{aligned}$$

Thus

$$e_{j+1} \leq (1+\delta L_f)e_j + \frac{L_f M_f}{2}\delta^2, \qquad e_0 = 0,$$

and

$$e_{ML} \leq \frac{L_f M_f}{2} ML\delta^2(1+\delta L_f)^{ML} \leq \frac{L_f M_f}{2} e^{L_f}\frac{1}{ML}.$$

Combining the estimates d_j and e_j gives

$$\|x_{ML}^{L,M} - \mathbf{x}_{\mathrm{simp}}^{L,*}(1)\| \le \frac{e^{L_f}}{L}\left(L_F M_A + \frac{L_f M_f}{2M}\right) \le \frac{C}{L},$$

where $C := e^{L_f}(L_F M_A + L_f M_f/2)$ is independent of L and M. From the proof of Theorem 2,

$$\|\mathbf{x}_{\mathrm{simp}}^{L,*}(1) - x_L^{L,*}\| \le \frac{C_{\mathrm{simp}}}{L}, \qquad C_{\mathrm{simp}} := \frac{L_f M_f}{2} e^{L_f}.$$

Hence the triangle inequality gives

$$\sup_{M\in\mathbb{N}_+} \|x_{ML}^{L,M} - x_L^{L,*}\| \le \frac{C'}{L}, \qquad C' := C + C_{\mathrm{simp}},$$

with C' independent of L and M.

□

A.3 Backward Regularity and Well-Definedness

Lemma 3. *Suppose Assumptions 1–3 hold. Then there exist $C_f, C_a, C_g > 0$, independent of L, k, and M, such that for every $x, y \in K_x$ and $k = 0, \dots, L-1$,*

$$\begin{aligned} &\|f_L(x, \theta_k^{L,*})\| + \|\partial_x f_L(x, \theta_k^{L,*})\| + \|\partial_\theta f_L(x, \theta_k^{L,*})\| \le C_f, \\ &\|\partial_x f_L(x, \theta_k^{L,*}) - \partial_x f_L(y, \theta_k^{L,*})\| + \|\partial_\theta f_L(x, \theta_k^{L,*}) - \partial_\theta f_L(y, \theta_k^{L,*})\| \le C_f \|x - y\|. \end{aligned} \tag{5}$$

Moreover, for $M \ge 1$, let $\delta = h/M - 1/(ML)$ and define

$$g_{L,M}(x, \theta) := \tilde{\boldsymbol{A}}(x, \theta) + \tilde{\boldsymbol{F}}\big(x + \delta \tilde{\boldsymbol{A}}(x, \theta), \theta\big).$$

Uniformly in M, $x \in K_x$, and k, we have

$$\begin{aligned} &\|g_{L,M}(x, \theta_k^{L,*}) - f_L(x, \theta_k^{L,*})\| + \|\partial_x g_{L,M}(x, \theta_k^{L,*}) - \partial_x f_L(x, \theta_k^{L,*})\| \\ &\quad + \|\partial_\theta g_{L,M}(x, \theta_k^{L,*}) - \partial_\theta f_L(x, \theta_k^{L,*})\| \le C_g(h - \delta) \le C_g h. \end{aligned} \tag{6}$$

The consistent adjoint-state of the trained simple-continuation satisfies

$$\|\mathbf{a}_{\mathrm{simp}}^{L,*}\|_\infty + \left\| \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{a}_{\mathrm{simp}}^{L,*} \right\|_\infty \le C_a. \tag{7}$$

Proof. Let B_1, L_1 denote the uniform bounds and Lipschitz constants in Assumption 3. For the loss ℓ defined in Section 2, softmax cross-entropy has bounded first and second derivatives with respect to the logits. Together with the smoothness of the fixed normalization map Norm, compactness of K_x, and the bound on V_{de} in Assumption 1, the chain rule yields constants $C_\ell, L_\ell > 0$, independent of L and z_0, such that

$$\|\partial_x \ell(x)\| \le C_\ell, \qquad \|\partial_x \ell(x) - \partial_x \ell(y)\| \le L_\ell \|x - y\|, \quad x, y \in K_x.$$

Fix L and k. For $s \in [0, h]$, define

$$\Phi_{L,s}(x, \theta) := \tilde{\boldsymbol{A}}(x, \theta) + \tilde{\boldsymbol{F}}\big(x + s\tilde{\boldsymbol{A}}(x, \theta), \theta\big).$$

Thus $f_L = \Phi_{L,h}$ and $g_{L,M} = \Phi_{L,\delta}$. For $x \in K_x$, Assumption 2 gives

$$\|x + s\tilde{\boldsymbol{A}}(x, \theta_k^{L,*})\| \le R + M_A,$$

so every argument below lies in K_A. The chain rule gives

$$\begin{aligned} \partial_x \Phi_{L,s}(x, \theta) &= \partial_x \tilde{\boldsymbol{A}}(x, \theta) + \partial_x \tilde{\boldsymbol{F}}\big(x + s\tilde{\boldsymbol{A}}(x, \theta), \theta\big)\big(I + s\partial_x \tilde{\boldsymbol{A}}(x, \theta)\big), \\ \partial_\theta \Phi_{L,s}(x, \theta) &= \partial_\theta \tilde{\boldsymbol{A}}(x, \theta) + s\partial_\theta \tilde{\boldsymbol{F}}\big(x + s\tilde{\boldsymbol{A}}(x, \theta), \theta\big)\partial_\theta \tilde{\boldsymbol{A}}(x, \theta) + \partial_\theta \tilde{\boldsymbol{F}}\big(x + s\tilde{\boldsymbol{A}}(x, \theta), \theta\big). \end{aligned}$$

Moreover, at $\theta = \theta_k^{L,*}$,

$$\left\| x + s\tilde{\boldsymbol{A}}(x, \theta_k^{L,*}) - y - s\tilde{\boldsymbol{A}}(y, \theta_k^{L,*}) \right\| \le (1 + B_1)\|x - y\|.$$

Assumptions 2 and 3, applied to these identities, yield equation 5 with a constant C_f depending only on M_A, M_F, B_1, and L_1.

For the field-consistency estimate, observe that

$$\left\| x + \delta \tilde{\boldsymbol{A}}(x, \theta_k^{L,*}) - x - h\tilde{\boldsymbol{A}}(x, \theta_k^{L,*}) \right\| \leq M_A(h - \delta).$$

Assumption 3 therefore gives

$$\|g_{L,M}(x, \theta_k^{L,*}) - f_L(x, \theta_k^{L,*})\| \leq B_1 M_A(h - \delta).$$

Given the structures of f_L and $g_{L,M}$, subtracting the preceding chain-rule identities with $s = \delta$ and $s = h$ yields

$$\|\partial_x g_{L,M}(x, \theta_k^{L,*}) - \partial_x f_L(x, \theta_k^{L,*})\| \leq \left[L_1 M_A(1 + B_1) + B_1^2\right](h - \delta),$$
$$\|\partial_\theta g_{L,M}(x, \theta_k^{L,*}) - \partial_\theta f_L(x, \theta_k^{L,*})\| \leq \left[B_1^2 + L_1 M_A(B_1 + 1)\right](h - \delta).$$

Summing these three bounds proves equation 6 for a depth- and growth-factor-independent C_g.

Finally, the adjoint equation has the integral form

$$\mathbf{a}_{\text{simp}}^{L,*}(t) = \partial_{\mathbf{x}} \ell\big(\mathbf{x}_{\text{simp}}^{L,*}(1)\big) + \int_t^1 \partial_{\mathbf{x}} f_L\big(\mathbf{x}_{\text{simp}}^{L,*}(s), \boldsymbol{\theta}_{\text{simp}}^{L,*}(s)\big)^T \mathbf{a}_{\text{simp}}^{L,*}(s)\, \mathrm{d}s.$$

Its coefficient is bounded and measurable, so this linear terminal-value problem has a unique absolutely continuous solution. Backward Gronwall and equation Adjoint give

$$\|\mathbf{a}_{\text{simp}}^{L,*}\|_\infty \leq C_\ell e^{C_f}, \qquad \left\| \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{a}_{\text{simp}}^{L,*} \right\|_\infty \leq C_f C_\ell e^{C_f}.$$

Hence equation 7 holds with $C_a := (1 + C_f) C_\ell e^{C_f}$. □

PROOF OF PROPOSITION 2

Proposition 2. *Under Assumptions 1 and 2, the simple-continuation* $(\mathbf{x}_{\text{simp}}^{L,*}, \boldsymbol{\theta}_{\text{simp}}^{L,*})$ *of the trained depth-L model admits a unique consistent adjoint-state* $\mathbf{a}_{\text{simp}}^{L,*}$.

Proof. Define

$$\mathcal{A}(t) := \partial_{\mathbf{x}} f_L\big(\mathbf{x}_{\text{simp}}^{L,*}(t), \boldsymbol{\theta}_{\text{simp}}^{L,*}(t)\big).$$

Lemma 2 ensures that the state remains in K_x, and the control takes values in the finite trained parameter set. The smoothness of f_L therefore makes $\mathcal{A}$ bounded and measurable for each L, while the smoothness of ℓ makes its terminal derivative well defined. Hence the linear terminal-value problem

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{a}_{\text{simp}}^{L,*}(t) = -\mathcal{A}(t)^T \mathbf{a}_{\text{simp}}^{L,*}(t), \qquad \mathbf{a}_{\text{simp}}^{L,*}(1) = \partial_{\mathbf{x}} \ell\big(\mathbf{x}_{\text{simp}}^{L,*}(1)\big),$$

has a unique absolutely continuous solution by Carathéodory's existence theorem. □

A.4 BACKWARD CONSISTENCY

We first show that $\mathbf{a}_{\text{simp}}^{L,*}(\tau)$ is the sensitivity of the terminal loss relative to the state at time τ.

Lemma 4. *Let* $(\mathbf{x}_{\text{simp}}^{L,*}, \boldsymbol{\theta}_{\text{simp}}^{L,*})$ *be the simple-continuation of the trained depth-L solution, and let* $\mathbf{a}_{\text{simp}}^{L,*}$ *be its consistent adjoint-state. Fix* $\tau \in [0, 1]$ *and a state perturbation* η *at time* τ. *If* ζ *solves*

$$\frac{\mathrm{d}}{\mathrm{d}t} \zeta(t) = \partial_{\mathbf{x}} f_L\big(\mathbf{x}_{\text{simp}}^{L,*}(t), \boldsymbol{\theta}_{\text{simp}}^{L,*}(t)\big)\zeta(t), \qquad \zeta(\tau) = \eta, \tag{8}$$

then

$$\partial_{\mathbf{x}_{\text{simp}}^{L,*}(\tau)} \left[\ell(\mathbf{x}_{\text{simp}}^{L,*}(1))\right][\eta] = \langle \mathbf{a}_{\text{simp}}^{L,*}(\tau), \eta \rangle, \qquad \forall \xi \in \Xi.$$

Proof. Proposition 2 shows that the equation 8 is well-defined. Write

$$\mathcal{A}(t) := \partial_{\mathbf{x}} f_L\big(\mathbf{x}^{L,*}_{\mathrm{simp}}(t), \boldsymbol{\theta}^{L,*}_{\mathrm{simp}}(t)\big).$$

For almost every $t \in [\tau, 1]$, the adjoint and linearized state equations give

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\langle \mathbf{a}^{L,*}_{\mathrm{simp}}(t), \zeta(t)\rangle &= \Big\langle -\mathcal{A}(t)^T \mathbf{a}^{L,*}_{\mathrm{simp}}(t), \zeta(t)\Big\rangle + \Big\langle \mathbf{a}^{L,*}_{\mathrm{simp}}(t), \mathcal{A}(t)\zeta(t)\Big\rangle \\ &= 0.\end{aligned}$$

Hence $\langle \mathbf{a}^{L,*}_{\mathrm{simp}}(t), \zeta(t)\rangle$ is constant on $[\tau, 1]$. Using equation Terminal and $\zeta(\tau) = \eta$ yields

$$\partial_{\mathbf{x}^{L,*}_{\mathrm{simp}}(\tau)}\big[\ell(\mathbf{x}^{L,*}_{\mathrm{simp}}(1))\big][\eta] = \Big\langle \partial_{\mathbf{x}}\ell\big(\mathbf{x}^{L,*}_{\mathrm{simp}}(1)\big), \zeta(1)\Big\rangle = \langle \mathbf{a}^{L,*}_{\mathrm{simp}}(\tau), \eta\rangle.$$

□

Lemma 5. *Set $h = 1/L$ and write the discrete Transformer dynamics equivalently as*

$$x^L_{k+1} = x^L_k + h f_L(x^L_k, \theta^L_k), \qquad k = 0, \dots, L-1.$$

Define the Lagrangian

$$\boldsymbol{L}(x, \theta, \lambda) := \mathbb{E}\Bigg[\ell(x^L_L) + \sum_{k=0}^{L-1} (\lambda^L_{k+1})^T \left(x^L_k + h f_L(x^L_k, \theta^L_k) - x^L_{k+1}\right)\Bigg].$$

The Lagrange multipliers of these constraints satisfy the following backward recursion for all $\xi \in \Xi$:

$$\begin{aligned}&\lambda^L_L = \partial_{x^L_L}\ell(x^L_L),\\ &\lambda^L_k = \lambda^L_{k+1} + h\partial_x f_L(x^L_k, \theta^L_k)^T \lambda^L_{k+1}, \qquad k = L-1, \dots, 0.\end{aligned}$$

For every $\xi \in \Xi$, they coincide with the discrete loss sensitivities,

$$\lambda^L_k = \partial_{x^L_k}\ell(x^L_L), \qquad k = 1, \dots, L.$$

Proof. Stationarity of the Lagrangian with respect to the states gives the terminal identity and the backward multiplier recursion. The well-definedness of the corresponding KKT system has been proved by Liu et al. (2023); Wang et al. (2025); Jin et al. (2025). If $\lambda^L_{k+1} = \partial_{x^L_{k+1}}\ell(x^L_L)$, the chain rule and the forward update imply

$$\begin{aligned}\partial_{x^L_k}\ell(x^L_L) &= \big(I + h\partial_x f_L(x^L_k, \theta^L_k)\big)^T \lambda^L_{k+1}\\ &= \lambda^L_{k+1} + h\partial_x f_L(x^L_k, \theta^L_k)^T \lambda^L_{k+1} = \lambda^L_k.\end{aligned}$$

Backward induction proves the sensitivity formula. Differentiating the same update with respect to θ^L_k gives

$$\partial_{\theta^L_k}\ell(x^L_L) = h\partial_\theta f_L(x^L_k, \theta^L_k)^T \lambda^L_{k+1}.$$

Thus

$$\frac{1}{h}\frac{\partial}{\partial \theta^L_k}\mathbb{E}[\ell(x^L_L)] = \mathbb{E}\left[\partial_\theta f_L(x^L_k, \theta^L_k)^T \lambda^L_{k+1}\right].$$

□

PROOF OF PROPOSITION 3

Proposition 3. *Under Assumptions 1–3, the Lagrange multipliers $\{\lambda^{L,*}_k\}_{k=0}^L$ implicitly generated by backpropagation at the trained depth-L model, together with the hidden-states $\{x^{L,*}_k\}_{k=0}^L$, form a symplectic discretization of equation Adjoint and equation Dynamic. Moreover,*

$$\max_{0 \le k \le L} \|\lambda^{L,*}_k - \mathbf{a}^{L,*}_{\mathrm{simp}}(kh)\| = \mathcal{O}(L^{-1}), \quad \forall \xi \in \Xi.$$

Proof. For a fixed trained layer parameter $\theta_k^{L,*}$, introduce the generating function

$$S_k(x,\lambda) := \lambda^T\big(x + hf_L(x,\theta_k^{L,*})\big).$$

Its recursions hold for all $\xi \in \Xi$:

$$x_{k+1}^{L,*} = \partial_\lambda S_k(x_k^{L,*},\lambda_{k+1}^L) = x_k^{L,*} + hf_L(x_k^{L,*},\theta_k^{L,*}),$$
$$\lambda_k^L = \partial_x S_k(x_k^{L,*},\lambda_{k+1}^L) = \lambda_{k+1}^L + h\partial_x f_L(x_k^{L,*},\theta_k^{L,*})^T\lambda_{k+1}^L.$$

Lemma 5 identifies the second relation with ordinary backpropagation. The pair of relations is precisely the symplectic-Euler scheme for the state–adjoint system.

We next establish the consistency estimate. Set $t_k := kh$ and define $e_k^x := \|x_k^{L,*} - \mathbf{x}_{\mathrm{simp}}^{L,*}(t_k)\|$. On $[kh,(k+1)h]$, the control is constant. By equation 5,

$$\Big\|\mathbf{x}_{\mathrm{simp}}^{L,*}(t_{k+1}) - \mathbf{x}_{\mathrm{simp}}^{L,*}(t_k) - hf_L(\mathbf{x}_{\mathrm{simp}}^{L,*}(t_k),\theta_k^{L,*})\Big\| \le \frac{C_f^2}{2}h^2.$$

Therefore

$$e_{k+1}^x \le (1+hC_f)e_k^x + \frac{C_f^2}{2}h^2, \qquad e_0^x = 0.$$

Discrete Gronwall yields

$$\max_{0\le k\le L} e_k^x \le C_{\mathrm{state}}h, \qquad C_{\mathrm{state}} := \frac{C_f^2}{2}e^{C_f}. \tag{9}$$

On the common compact state set, let L_ℓ be a Lipschitz constant of $\partial_{\mathbf{x}}\ell$, and set $e_k^a := \|\lambda_k^L - \mathbf{a}_{\mathrm{simp}}^{L,*}(t_k)\|$. The terminal discrepancy satisfies

$$e_L^a \le L_\ell C_{\mathrm{state}}h.$$

Integrating equation Adjoint backward over $[kh,(k+1)h]$ and comparing it with the multiplier recursion gives

$$e_k^a \le (1+hC_f)e_{k+1}^a + C_{\mathrm{adj}}h^2,$$

where, using equation 5, equation 7, and equation 9, one may take

$$C_{\mathrm{adj}} := C_fC_a(C_{\mathrm{state}} + C_f + 1).$$

Here $C_fC_a(C_{\mathrm{state}} + C_f)h^2$ controls the change in $\partial_{\mathbf{x}}f_L$ caused by the state discretization and the within-step state variation, while $C_fC_ah^2$ controls the within-step adjoint variation. Backward discrete Gronwall now gives

$$\max_{0\le k\le L} e_k^a \le e^{C_f}\big(L_\ell C_{\mathrm{state}} + C_{\mathrm{adj}}\big)h =: C_\lambda h.$$

The constants C_{state}, C_{adj}, and C_λ are independent of L, k, and ξ, proving the proposition. □

PROOF OF THEOREM 4

Theorem 4. *Suppose Assumptions 1–3 hold. Consider the ML-layer scaled-stacking up-scaling of a trained depth-L model. Initialize $\theta_j^{L,M} = \theta_{\pi_M(j)}^{L,*}$ and denote the corresponding Lagrange multipliers by $\{\lambda_j^{L,M}\}_{j=0}^{ML}$. Then*

$$\sup_{M\in\mathbb{N}_+}\max_{0\le j\le ML}\|\lambda_j^{L,M} - \mathbf{a}_{\mathrm{simp}}^{L,*}(j\delta)\| = \mathcal{O}(L^{-1}), \qquad \forall\xi\in\Xi,$$

$$\sup_{M\in\mathbb{N}_+}\max_{0\le j<ML}\left\|\frac{1}{\delta}\frac{\partial}{\partial\theta_j^{L,M}}\mathbb{E}[\ell(x_{ML}^{L,M})] - \frac{1}{h}\frac{\partial}{\partial\theta_{\pi_M(j)}^{L}}\mathbb{E}[\ell(x_L^{L,*})]\right\| = \mathcal{O}(L^{-1}).$$

For base-model directions v_k^L, define their refinement by $v_j^{L,M} := v_{\pi_M(j)}^L$. Then

$$\sup_{M\in\mathbb{N}_+}\left|\sum_{j=0}^{ML-1}\left\langle\frac{\partial}{\partial\theta_j^{L,M}}\mathbb{E}[\ell(x_{ML}^{L,M})], v_j^{L,M}\right\rangle - \sum_{k=0}^{L-1}\left\langle\frac{\partial}{\partial\theta_k^{L}}\mathbb{E}[\ell(x_L^{L,*})], v_k^{L}\right\rangle\right| = \mathcal{O}(L^{-1})\sum_{k=0}^{L-1}h\|v_k^L\|.$$

Proof. Fix $L \geq 2$ and $M \geq 1$, and set $h := 1/L$, $\delta := h/M$, and $t_j := j\delta$. For $j = 0, \ldots, ML-1$, write $k := \pi_M(j)$. At the copied initialization,

$$x_{j+1}^{L,M} = x_j^{L,M} + \delta g_{L,M}(x_j^{L,M}, \theta_k^{L,*}), \qquad \theta_j^{L,M} = \theta_k^{L,*}. \tag{10}$$

Assumptions 1 and 2 keep the continuous, coarse-grid, ideal-refinement, and scaled-stacking states in the common compact set K_x; see Lemmas 1 and 2 and the proof of Theorem 3. Hence Lemma 3 applies throughout, with constants independent of L, M, j, and ξ.

Fine-grid state consistency. Let $y_j^{L,M}$ be the ideal Euler refinement driven by f_L, with $y_0^{L,M} = x_0^{L,M} = \mathbf{x}_{\text{simp}}^{L,*}(0)$. Using equation 6 gives $\|\partial_{\mathbf{x}} g_{L,M}\| \leq C_f + C_g$. Hence $d_j := \|x_j^{L,M} - y_j^{L,M}\|$ satisfies

$$d_{j+1} \leq \big(1 + \delta(C_f + C_g)\big) d_j + \delta C_g (h - \delta), \qquad d_0 = 0.$$

Consequently, the Gronwall's inequality gives

$$\max_{0 \leq j \leq ML} d_j \leq C_{\text{bias}}(h - \delta), \qquad C_{\text{bias}} := C_g e^{C_f + C_g}.$$

The ordinary Euler error $r_j := \|y_j^{L,M} - \mathbf{x}_{\text{simp}}^{L,*}(t_j)\|$ obeys

$$r_{j+1} \leq (1 + \delta C_f) r_j + \frac{C_f^2}{2}\delta^2, \qquad r_0 = 0,$$

and therefore

$$\max_{0 \leq j \leq ML} r_j \leq C_{\text{Euler}}\delta, \qquad C_{\text{Euler}} := \frac{C_f^2}{2} e^{C_f}.$$

Since $(h - \delta) + \delta = h$, these estimates give

$$\max_{0 \leq j \leq ML} \|x_j^{L,M} - \mathbf{x}_{\text{simp}}^{L,*}(t_j)\| \leq C_{\text{fine}} h, \qquad C_{\text{fine}} := \max\{C_{\text{bias}}, C_{\text{Euler}}\}. \tag{11}$$

Thus the two-stage bias is controlled by $h - \delta$, whereas the Euler discretization error is controlled by δ; their sum is uniformly $\mathcal{O}(h)$.

Fine-grid adjoint consistency. Applying the same chain-rule calculation as in Lemma 5 to equation 10 gives

$$\lambda_{ML}^{L,M} = \partial_{\mathbf{x}} \ell(x_{ML}^{L,M}), \qquad \lambda_j^{L,M} = \lambda_{j+1}^{L,M} + \delta \partial_{\mathbf{x}} g_{L,M}(x_j^{L,M}, \theta_k^{L,*})^T \lambda_{j+1}^{L,M}. \tag{12}$$

Set $e_j^a := \|\lambda_j^{L,M} - \mathbf{a}_{\text{simp}}^{L,*}(t_j)\|$. The bounds in equation 5, equation 6, and equation 11 give

$$\left\| \partial_x g_{L,M}(x_j^{L,M}, \theta_k^{L,*}) - \partial_{\mathbf{x}} f_L(\mathbf{x}_{\text{simp}}^{L,*}(t_j), \theta_k^{L,*}) \right\| \leq C_{\text{jac}} h, \qquad C_{\text{jac}} := C_g + C_f C_{\text{fine}}.$$

Let L_ℓ again denote a Lipschitz constant of $\partial_x \ell$ on the compact state set. The terminal error satisfies $e_{ML}^a \leq L_\ell C_{\text{fine}} h$. Comparing equation 12 with the integral form of equation Adjoint on $[t_j, t_{j+1}]$ gives

$$e_j^a \leq \big(1 + \delta(C_f + C_g)\big) e_{j+1}^a + \delta C_{\text{jac}} C_a h + C_{\text{quad}} \delta^2,$$

where $C_{\text{quad}} := C_a(C_f^2/2 + C_f)$ controls the within-subinterval variation of the continuous state and adjoint. Backward discrete Gronwall, together with $ML\delta = 1$ and $\delta \leq h$, yields

$$\max_{0 \leq j \leq ML} e_j^a \leq C_{\lambda,\text{fine}} h, \qquad C_{\lambda,\text{fine}} := e^{C_f + C_g}\big(L_\ell C_{\text{fine}} + C_{\text{jac}} C_a + C_{\text{quad}}\big). \tag{13}$$

Normalized-gradient consistency. Applying the same chain-rule calculation to the coarse and fine updates gives the exact identities

$$\frac{1}{\delta} \frac{\partial}{\partial \theta_j^{L,M}} \mathbb{E}[\ell(x_{ML}^{L,M})] = \mathbb{E}\left[\partial_\theta g_{L,M}(x_j^{L,M}, \theta_k^{L,*})^T \lambda_{j+1}^{L,M} \right],$$

$$\frac{1}{h} \frac{\partial}{\partial \theta_k^L} \mathbb{E}[\ell(x_L^{L,*})] = \mathbb{E}\left[\partial_\theta f_L(x_k^{L,*}, \theta_k^{L,*})^T \lambda_{k+1}^L \right].$$

The first-order bounds and the coarse and fine state estimates imply

$$\|\partial_\theta g_{L,M}(x_j^{L,M}, \theta_k^{L,*}) - \partial_\theta f_L(x_k^{L,*}, \theta_k^{L,*})\| \le C_{par} h,$$

where $C_{par} := C_g + C_f(C_{\mathrm{fine}} + C_f + C_{\mathrm{state}})$. Indeed, insert the continuous states at t_j and kh and use $0 \le t_j - kh < h$. Similarly, equation 13, Proposition 3, and equation 7 give

$$\|\lambda_{j+1}^{L,M} - \lambda_{k+1}^{L}\| \le (C_{\lambda,\mathrm{fine}} + C_a + C_\lambda)h.$$

Since $\|\partial_\theta f_L\| \le C_f$ and $\|\lambda_{j+1}^{L,M}\| \le C_a + C_{\lambda,\mathrm{fine}}$, the two exact identities yield

$$\left\| \frac{1}{\delta} \frac{\partial}{\partial \theta_j^{L,M}} \mathbb{E}[\ell(x_{ML}^{L,M})] - \frac{1}{h} \frac{\partial}{\partial \theta_k^{L}} \mathbb{E}[\ell(x_L^{L,*})] \right\| \le C_{\mathrm{grad}} h, \tag{14}$$

where $C_{\mathrm{grad}} := C_{par}(C_a + C_{\lambda,\mathrm{fine}}) + C_f(C_{\lambda,\mathrm{fine}} + C_a + C_\lambda)$ is independent of M.

Loss differentials and descent. Pairing equation 14 with $v_j^{L,M} = v_{\pi_M(j)}^{L}$ and using $M\delta = h$ yields

$$\left| \sum_{j=0}^{ML-1} \left\langle \frac{\partial}{\partial \theta_j^{L,M}} \mathbb{E}[\ell(x_{ML}^{L,M})], v_j^{L,M} \right\rangle - \sum_{k=0}^{L-1} \left\langle \frac{\partial}{\partial \theta_k^{L}} \mathbb{E}[\ell(x_L^{L,*})], v_k^{L} \right\rangle \right| \le C_{\mathrm{grad}} h \sum_{k=0}^{L-1} h \|v_k^L\|.$$

The right-hand side is independent of M, proving the stated differential estimate. In particular, a negative coarse directional derivative remains negative after refinement whenever its magnitude exceeds this error. □

B EXPERIMENTS

B.1 DEPTH UP-SCALING DIAGRAMS

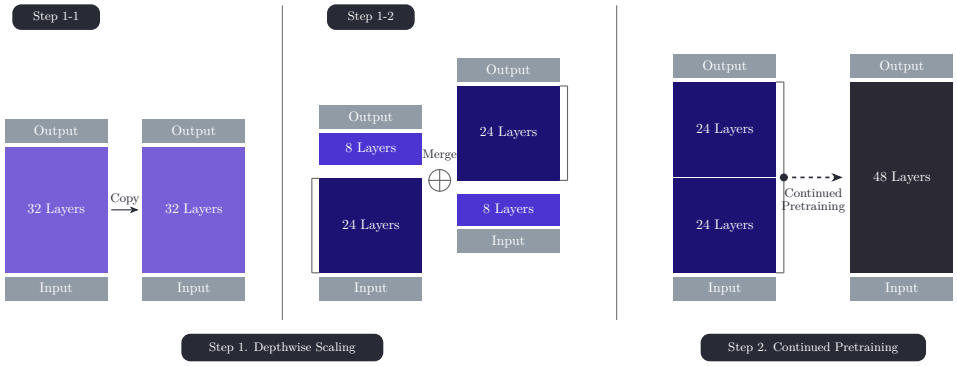


Figure 7: Layer concatenation. Kim et al. (2024) drop prefix and suffix blocks from the base model, while Du et al. (2024) select full blocks of the base model (i.e., all 32 layers in the diagram). The selected blocks are then duplicated and concatenated to form the deeper initialization.

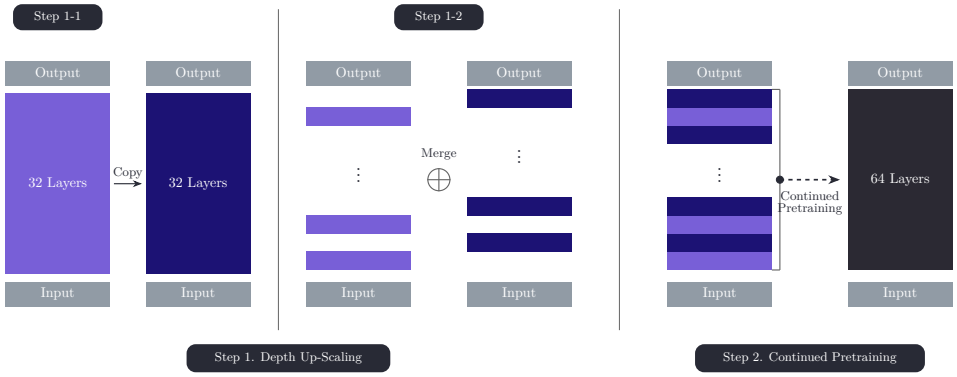


Figure 8: Layer stacking (Du et al., 2024). Each block is duplicated in place, preserving layer order while increasing the number of depth steps.

B.2 MODEL ARCHITECTURES

Table 1: Architectures of the evaluated models.

Model	Layers	Heads (Q / KV)	Architecture
Qwen3-0.6B	28	(GQA) 16 / 8	Dense
Qwen3-1.7B	28	(GQA) 16 / 8	Dense
Qwen3-4B	36	(GQA) 32 / 8	Dense
Qwen3-8B	36	(GQA) 32 / 8	Dense
Qwen3-14B	40	(GQA) 40 / 8	Dense
Qwen3-32B	64	(GQA) 64 / 8	Dense
Qwen3-30B-A3B	48	(GQA) 32 / 4	MoE

B.3 Experimental Setup

Implementation of scaled stacking. During continual training, we implement scaled stacking by applying a $1/M$ gate to each MHA and FFN residual mapping. After continual training or for inference and immediate evaluation, these gates can be folded into the MHA output matrices W_i^O and the FFN down-projection matrix W^{down}:

$$\boldsymbol{A}_M(x) := \sum_{i=1}^{m} \text{Attn}(\text{Norm}(x)) \left(\frac{1}{M} W_i^O\right), \quad \boldsymbol{F}_M(x) := \sigma(\text{Norm}(x)) \left(\frac{1}{M} W^{\text{down}}\right). \tag{15}$$

This implementation preserves the standard serial block interface and KV-cache structure.

Post-Growth Performance

Models and benchmarks. We evaluate dense Qwen3 checkpoints from 0.6B to 32B parameters and the Qwen3-30B-A3B mixture-of-experts model; their architectures are listed in Table 1. Evaluation uses `lm-evaluation-harness` (Gao et al., 2023) on ARC (Clark et al., 2018), GLUE (Wang et al., 2018), GPQA (Rein et al., 2024), HellaSwag (Zellers et al., 2019), IFEval (Zhou et al., 2023), Lambada (Paperno et al., 2016), MGSM (Cobbe et al., 2021; Shi et al., 2023), MMLU (Hendrycks et al., 2021b;a), Pile 10K (Gao et al., 2020), TruthfulQA (Lin et al., 2022), and WikiText-2 (Merity et al., 2017). We group Knowledge & Commonsense (K&C) and Language Modeling (LM) as basic abilities, and Generative Answering (GA) and Language Alignment (LA) as advanced abilities.

Table 2: Benchmark groups used for immediate post-growth evaluation.

Group	Abilities	Benchmarks
Basic	K&C (ACC ↑) LM (PPL ↓)	ARC-C, ARC-E, HellaSwag, MMLU Lambada, Pile 10K, WikiText-2
Advanced	GA (EM ↑) LA (ACC ↑)	GPQA, IFEval, Lambada, MGSM, TruthfulQA GLUE, IFEval, TruthfulQA

Benchmark metrics. The tasks share three metrics: accuracy (acc_norm), flexible-extraction exact match ($exact_match\text{-}flexible_extract$), and word-level perplexity ($word_perplexity$). We normalize each score by that of the corresponding base model. If $\mathcal{M}_{\text{new}}$ is up-scaled from $\mathcal{M}_{\text{old}}$, the normalized score is

$$\mathcal{S}(\mathcal{M}_{\text{new}}) := \frac{\text{Score}(\mathcal{M}_{\text{new}})}{\text{Score}(\mathcal{M}_{\text{old}})}.$$

For multiple-choice tasks, let b denote random-guess accuracy; for example, $b = 0.25$ for four choices. To prevent chance-level predictions from appearing to preserve nontrivial capability, we subtract this baseline before normalization:

$$\mathcal{S}(\mathcal{M}_{\text{new}}) := \frac{\max\{\text{Score}(\mathcal{M}_{\text{new}}) - b, 0\}}{\text{Score}(\mathcal{M}_{\text{old}}) - b}.$$

For accuracy and exact match, a normalized score of 1 (reported as 100% in figures) denotes base-model performance; for perplexity, a ratio of 1 denotes the same reference level and lower is better.

Evaluation protocol. We use 5-shot evaluation for ARC-C, ARC-E, GLUE, GPQA, HellaSwag, MGSM, and MMLU, and 0-shot evaluation for the remaining tasks. For GPQA, we evaluate the generative-answering and chain-of-thought settings on the Diamond, Extended, and Main subsets. Lambada and MGSM contain multilingual subtasks; we report their averages. All base models are downloaded without modification from the Qwen collection[2] on Hugging Face, and all benchmarks are evaluated in the default BF16 precision.

[2] `https://huggingface.co/collections/Qwen/qwen3`

CONTINUAL TRAINING

We train decoder-only GPT-2 models on OpenWebText with a context length of 1024 and evaluate post-growth training under AdamW and Muon. For each target configuration, all compared growth rules start from the same base checkpoint and use the same post-growth token budget. We report training and validation loss, and token accuracy against the number of tokens processed after expansion. Table 3 summarizes the configurations recorded for the base and progressive stages. The 50.4M, 256-layer model is grown from the 12.6M checkpoint with $M = 4$, whereas the 50.4M, 64-layer model is trained as the base checkpoint for the 100.8M target.

Table 3: Recorded configurations for base-model and post-growth training.

Size	Context	Heads	Base	Layers	Batch	Schedule	Warmup	Tokens
12.6M	1024	8	-	64	1M	cosine	100	0.2B
50.4M	1024	8	-	64	1M	cosine	100	0.8B
453.2M	1024	8	-	256	1M	cosine	400	1.6B
25.2M	1024	8	12.6M	128	1M	cosine	0	2B
50.4M	1024	8	12.6M	256	1M	cosine	0	3B
100.8M	1024	8	50.4M	128	1M	cosine	0	3B
0.9B	1024	8	453.2M	512	1M	cosine	0	10B

For all models, dropout and bias are disabled. For each training run, we use a learning rate ranging from $6e-5$ to $6e-4$. The warmup schedule is linear. For AdamW, we set $\beta_1 = 0.9$ and $\beta_2 = 0.95$; for Muon, the number of Newton-Schulz steps is 5, the momentum is $\beta = 0.95$, and Nesterov is enabled. For both optimizers, we use weight decay $\lambda = 0.1$ and clip the gradient norm to 1.0. The growth-timing is selected randomly without manual tuning.

B.4 AN EMPIRICAL DIAGNOSTIC FOR RESIDUAL SCALING

Assumption 2 controls the residual mappings across model depth. Chang et al. (2018) demonstrate that this assumption holds for ResNet models on CIFAR-10 and CIFAR-100, where the average L^2-norm of the residual modules decreases with model depth. We here derive and test another end-to-end consequence of this condition: with the parameters fixed, replacing serial MHA-FFN blocks by parallel blocks yields an $\mathcal{O}(L^{-1})$ terminal-state discrepancy.

Serial and parallel dynamics. Recall the notation of Section 3.2: for $L \geq 2$ and $M \in \mathbb{N}_+$, a depth-L model has step size $h = 1/L$, an M-fold refinement has step size $\delta = h/M$, and $\pi_M(j) = \lfloor j/M \rfloor$ maps fine-grid step j to its parent layer. For $s \in [0, h]$, define the field

$$\Phi_{L,s}(x, \theta) := \tilde{\boldsymbol{A}}(x, \theta) + \tilde{\boldsymbol{F}}\big(x + s\tilde{\boldsymbol{A}}(x, \theta), \theta\big).$$

The serial field of the trained model is $f_L = \Phi_{L,h}$, while the scaled-stacking field in the proof of Theorem 3 is $g_{L,M} = \Phi_{L,\delta}$. A parallel block evaluates MHA and FFN at the same hidden-states and gives

$$f_L^{\mathrm{para}}(x, \theta) := \Phi_{L,0}(x, \theta) = \tilde{\boldsymbol{A}}(x, \theta) + \tilde{\boldsymbol{F}}(x, \theta).$$

Thus, f_L, $g_{L,M}$, and f_L^{para} differ only in the MHA-induced displacement at which the FFN is evaluated: h, δ, and 0, respectively.

To compare the scaled serial and parallel computations on the same refined grid, let $\{z_j^{L,M}\}_{j=0}^{ML}$ denote the parallel states obtained from the copied parameters:

$$z_0^{L,M} = x_0^{L,*}, \qquad z_{j+1}^{L,M} = z_j^{L,M} + \delta f_L^{\mathrm{para}}\big(z_j^{L,M}, \theta_{\pi_M(j)}^{L,*}\big), \quad j = 0, \ldots, ML-1. \tag{16}$$

Recall that scaled-stacking uses the same initial state, fine grid, and copied parameters, but follows the update

$$x_{j+1}^{L,M} = x_j^{L,M} + \delta g_{L,M}\big(x_j^{L,M}, \theta_{\pi_M(j)}^{L,*}\big).$$

Both trajectories remain in the compact set K_x from Lemma 1, because each of their ML increments is bounded by $\delta(M_A + M_F)$. The FFN arguments lie in the fixed compact and convex set K_A. The

uniform local Lipschitz condition in Assumption 2, together with a finite-cover argument along line segments in K_A, therefore yields a constant L_F, independent of L, M, and k, such that

$$\|\tilde{\boldsymbol{F}}(u, \theta_k^{L,*}) - \tilde{\boldsymbol{F}}(v, \theta_k^{L,*})\| \le L_F \|u - v\|, \qquad u, v \in K_A.$$

This is the depth-uniform constant supplied by Lemma 1. Consequently, for every $x \in K_x$,

$$\left\| g_{L,M}(x, \theta_k^{L,*}) - f_L^{\mathrm{para}}(x, \theta_k^{L,*}) \right\| \le L_F M_A \delta. \tag{17}$$

The field discrepancy is therefore $\mathcal{O}(\delta)$, and the corresponding same-input state update differs by at most $L_F M_A \delta^2$.

The estimate also controls the accumulated network discrepancy. Indeed, f_L^{para} is Lipschitz on K_x with constant $L_A + L_F \le L_f$. For $d_j := \|x_j^{L,M} - z_j^{L,M}\|$, equation 16 and equation 17 give

$$d_{j+1} \le (1 + \delta L_f) d_j + L_F M_A \delta^2, \qquad d_0 = 0.$$

A discrete Gronwall estimate then yields

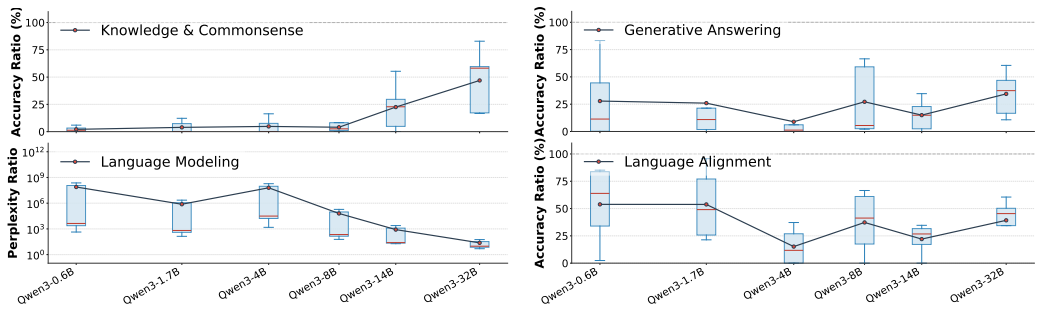
$$d_{ML} \le L_F M_A \delta^2 \sum_{r=0}^{ML-1} (1 + \delta L_f)^r \le L_F M_A e^{L_f} \delta = \frac{L_F M_A e^{L_f}}{ML}. \tag{18}$$

When $M = 1$, we have $\delta = h$, $\pi_1(k) = k$, and $x_k^{L,1} = x_k^{L,*}$. In this case, $\{z_k^{L,1}\}_{k=0}^{L}$ is precisely the trajectory obtained by replacing every serial block in the trained model by its parallel counterpart while retaining $\{\theta_k^{L,*}\}_{k=0}^{L-1}$, and equation 18 becomes

$$\|x_L^{L,*} - z_L^{L,1}\| = \mathcal{O}(L^{-1}).$$

The serial-to-parallel conversion therefore provides an end-to-end diagnostic of the predicted depth-dependent perturbation.

Serial-to-parallel conversion. Qwen3 uses serial blocks by default. For each checkpoint, we replace every block by the update in equation 16 with $M = 1$, keeping all trained parameters fixed and applying no additional training. We then evaluate the converted model immediately.



(a) Basic Abilities (b) Advanced Abilities

Figure 9: Serial-to-parallel conversion of Qwen3 checkpoints, evaluated without any additional training. For basic-abilities, the performance gaps for both language modeling and knowledge & commonsense narrow markedly toward the deepest checkpoint. The advanced abilities gradually improve from 4B onward.

In Figure 9, the serial-to-parallel performance gap narrows markedly toward the deepest checkpoint for both language modeling and knowledge & commonsense, while the advanced-ability metrics improve progressively from 4B onward. Because the conversion keeps all trained parameters fixed and changes only the within-block FFN evaluation point, this recovery with depth is consistent with the diminishing terminal-state discrepancy predicted by equation 18. These results provide empirical support for the depth-stable residual dynamics in Assumption 2.

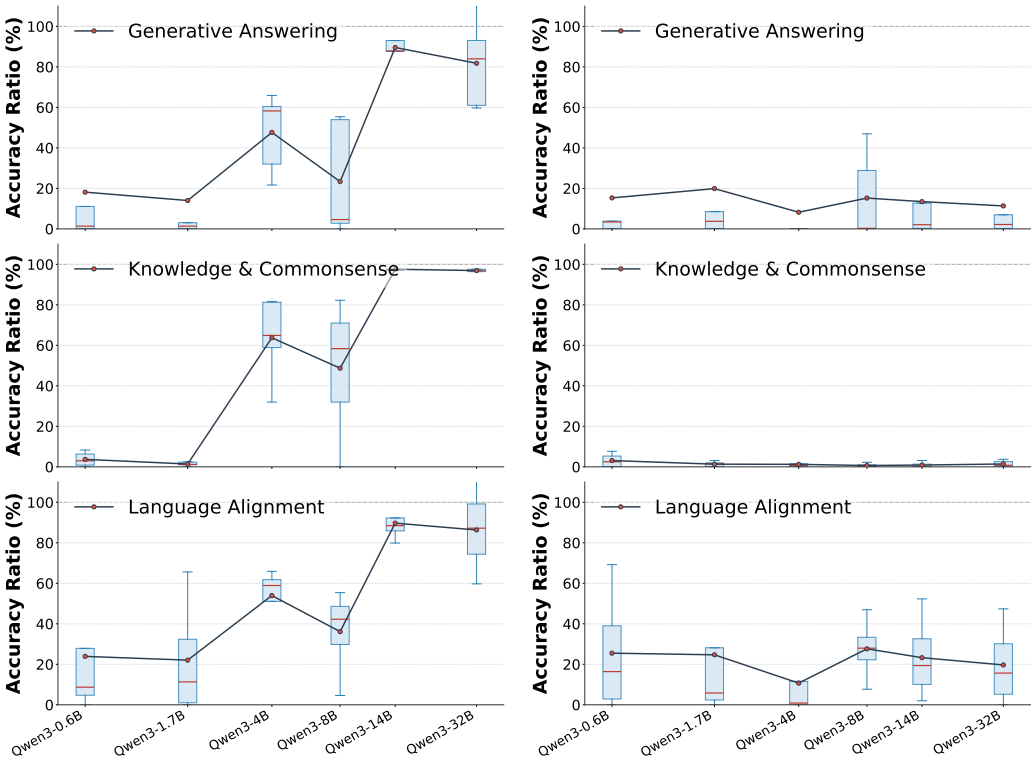
Relation to parallel-trained Transformers. A parallel-trained Transformer uses the residual $f_L^{\text{para}} = \Phi_{L,0}$, whereas a serial Transformer uses $f_L = \Phi_{L,h}$. Setting $M = 1$ in equation 17 gives an $\mathcal{O}(h)$ field discrepancy, so the corresponding one-step updates differ by $\mathcal{O}(h^2)$ and their accumulated fixed-parameter discrepancy is $\mathcal{O}(L^{-1})$ by equation 18. Parallel training optimizes the parameters directly under f_L^{para}, while our conversion evaluates this field at the serial-trained parameters $\theta_k^{L,*}$, thereby isolating the higher-order architectural difference before parameter adaptation.

This connection is also reflected in existing parallel-trained Transformers. Parallel MHA-FFN blocks are used by GPT-J (Wang & Komatsuzaki, 2021), GPT-NeoX-20B (Black et al., 2022), Pythia (Biderman et al., 2023), and PaLM (Chowdhery et al., 2023). Most directly, PaLM reports a small quality degradation for its 8B, 32-layer model but no degradation for its 62B, 64-layer model when comparing the parallel and serial formulations. This agreement in evaluation performance at the larger, deeper model scale matches the vanishing serial-parallel discrepancy predicted above and provides independent empirical support for Assumption 2.

B.5 Additional Experiments

B.5.1 Linear-Interpolation Continuations

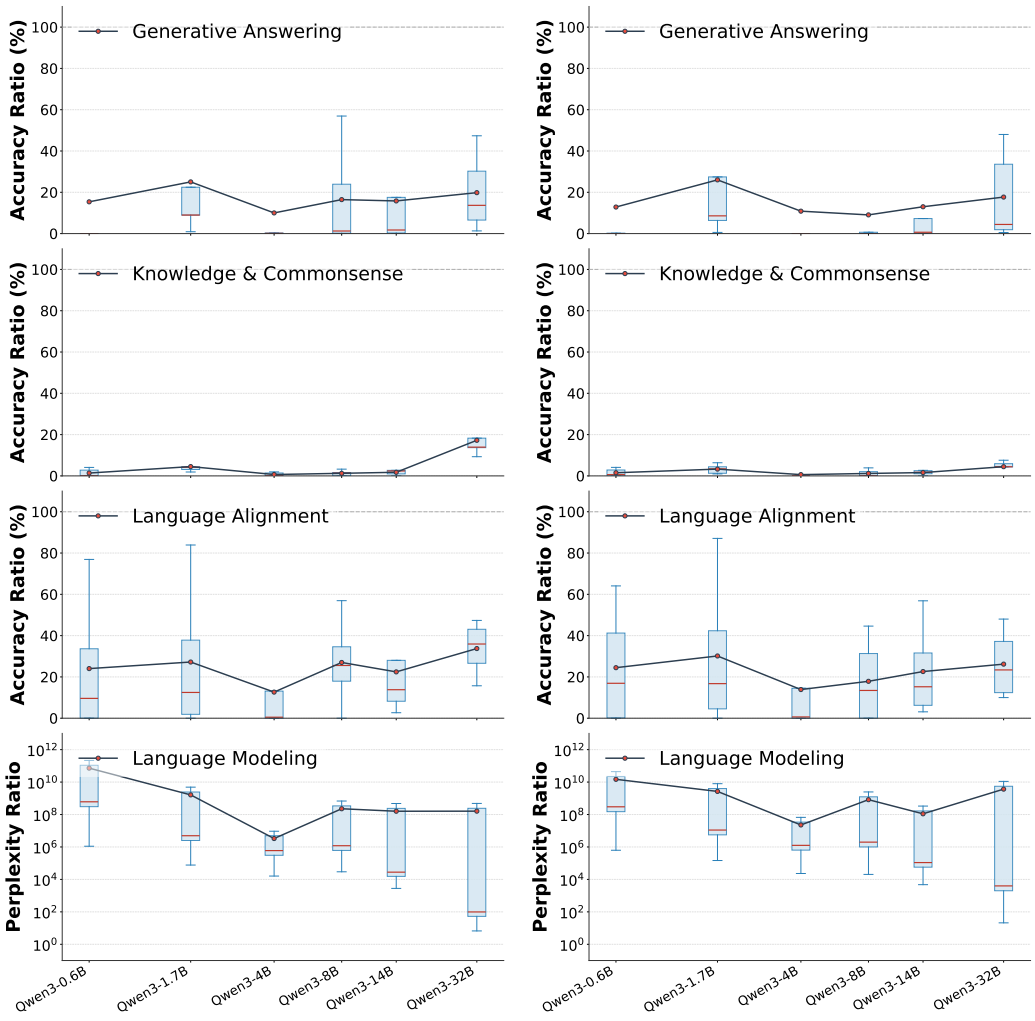
The accuracy-based metrics reinforce the perplexity comparison in Figure 2. Simple-continuation is inaccurate at the shallow end but approaches the base-model reference for the two deepest checkpoints. Linear interpolation shows no comparable recovery, consistent with the nonlinear failure mode isolated by Example 1.



(a) Simple-continuation. (b) Linear-interpolation continuation.

Figure 10: Immediate performance of simple-continuation (10(a)) and linear-interpolation (10(b)). Simple-continuation recovers sharply for the 14B and 32B checkpoints, whereas linear interpolation remains severely degraded across the evaluated range.

Figure 11 performs practical depth up-scaling with linear-interpolation continuation using scaled and direct residual implementations analogous to those of simple-continuation. Both variants exhibit large perplexity inflation and low normalized accuracy or exact match across the checkpoint family.

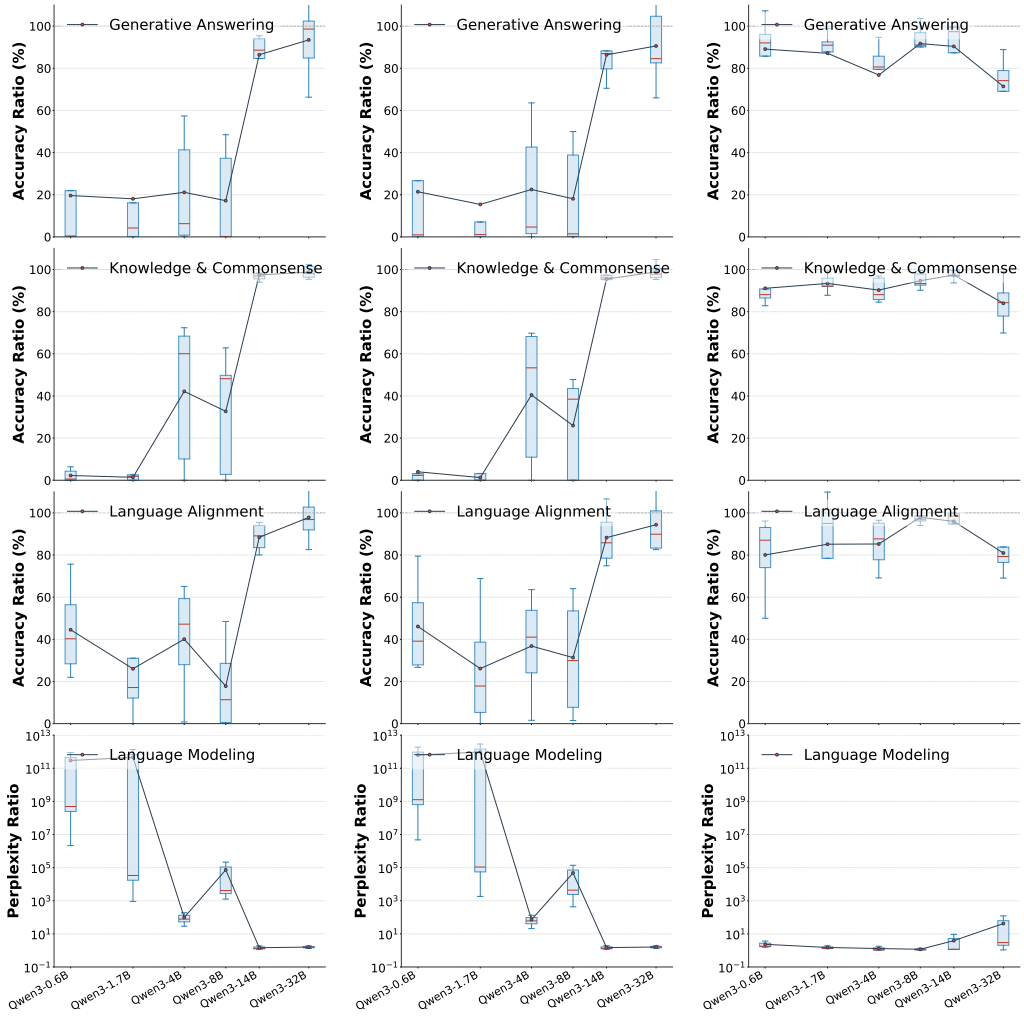


(a) Scaled interpolation. (b) Direct interpolation.

Figure 11: Practical depth up-scaling of linear-interpolation continuation, with scaled (11(a)) and direct (11(b)) residual implementations similar to those of simple-continuation. Both exhibit large perplexity inflation and low normalized accuracy or exact match across the checkpoint family.

B.5.2 Stacking and Concatenation across Dense Models

Figure 12 compares scaled stacking, direct stacking, and concatenation across the full family of dense Qwen3 checkpoints and traces their performance across depth. Concatenation is more stable at shallow checkpoints but degrades performance as depth increases. In contrast, both stacking variants recover sharply at 14B and 32B, producing the depth-dependent crossover.

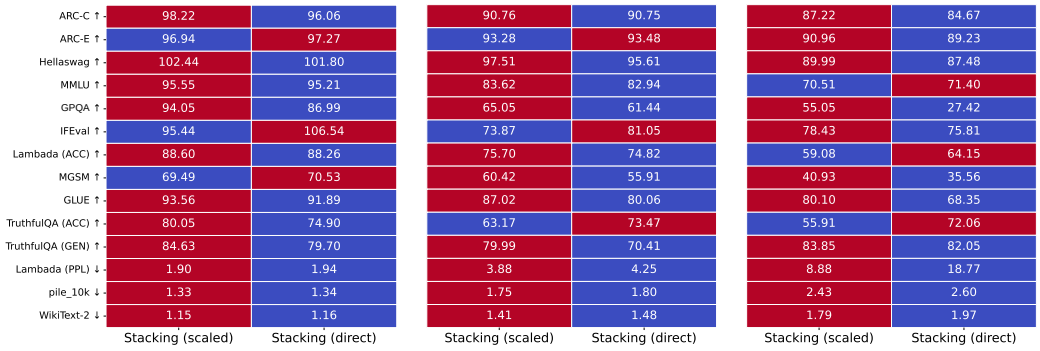


(a) Stacking (scaled). (b) Stacking (direct). (c) Concatenation.

Figure 12: Immediate performance of scaled stacking (12(a)), direct stacking (12(b)), and the concatenation baseline (12(c)) across the full family of dense Qwen3 checkpoints. The results exhibit a depth-dependent crossover: concatenation is more stable at shallow checkpoints but degrades performance as depth increases, whereas both stacking variants recover sharply at 14B and 32B.

B.5.3 Growth-Factor Sensitivity

Theorem 3 establishes terminal-state consistency for scaled stacking uniformly over the growth-factor M, whereas direct stacking has no corresponding guarantee. This uniformity controls the asymptotic error constant as L grows; for a fixed base depth, it does not imply that preservation improves as M increases. We compare the two variants on Qwen3-14B for $M \in \{2, 3, 4\}$ in Figure 13. At each growth-factor, scaled stacking attains better values on 11 of 14 metrics, including lower perplexity on all three language-modeling datasets. Both variants degrade as M increases, but residual scaling yields the more stable finite-depth refinement throughout the evaluated range.



(a) $M = 2$. (b) $M = 3$. (c) $M = 4$.

Figure 13: Immediate Qwen3-14B performance of scaled and direct stacking at growth-factors $M = 2, 3, 4$. Scaled stacking attains better values on 11 of 14 metrics at each growth-factor.

B.5.4 Mixture-of-Experts Model

We additionally evaluate scaled and direct stacking on the Qwen3-30B-A3B mixture-of-experts checkpoint with $M = 2$. Although the residual of MoE models is not smooth, both variants retain part of the base-model performance, with preservation slightly weaker than for dense models. Scaled stacking attains better values on 10 of 14 metrics, including every K&C and LM benchmark (Figure 14). This single-checkpoint result extends the empirical comparison to a sparse architecture whose routing dynamics do not satisfy the regularity assumptions of the theory.



Figure 14: Immediate performance of scaled and direct stacking on Qwen3-30B-A3B at growth-factor $M = 2$. Both variants retain part of the base-model performance, although preservation is slightly weaker than for dense models. Scaled stacking outperforms direct stacking on 10 of the 14 evaluated metrics, including all K&C and LM benchmarks.

B.5.5 Continual Training Experiments

We report the full training and validation trajectories underlying the first-order comparisons in Section 4. All panels plot loss and token accuracy against post-growth training tokens. These trajectories test consequences of retaining useful first-order information, rather than directly measuring adjoint or gradient error.

The finite-depth effect of residual scaling depends on the optimizer in Figure 15. The Muon curves favor scaled stacking, whereas the AdamW curves nearly overlap. This comparison is compatible with the asymptotic consistency result but does not imply that scaled stacking must improve every post-growth training trajectory.

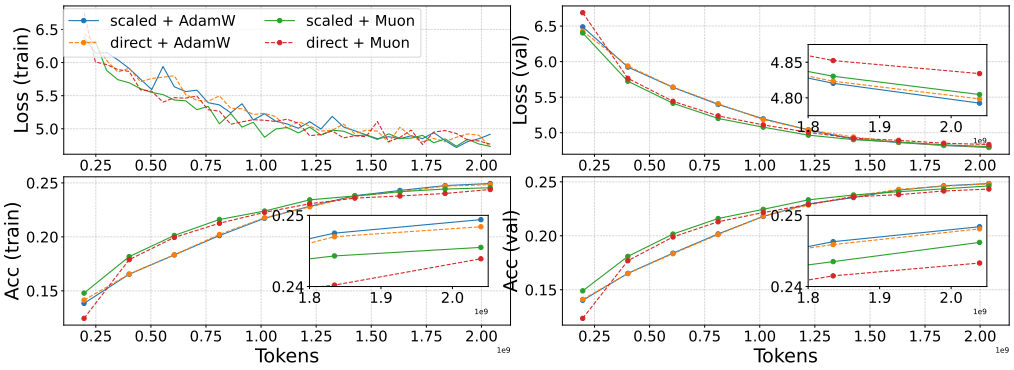


Figure 15: Direct and scaled stacking after expanding the 12.6M, 64-layer checkpoint to 25.2M parameters and 128 layers. The performance of scaled stacking with both AdamW and Muon is compared with that of direct stacking.

Inheriting optimizer state has a clearer effect for both growth-factors $M = 2$ (Figure 16) and $M = 4$ (Figure 17). Across both optimizers, the inherited-state runs retain an advantage throughout the displayed training horizon; the improvement is particularly pronounced for AdamW. This behavior supports the qualitative prediction that a depth refinement can reuse optimization information from the base model. The same results appear in the 100.8M target and in the more aggressive $M = 4$ expansion (Figures 18 and 19). In both cases, the configuration labeled stacking ends with better loss and accuracy than the corresponding concatenation configuration under AdamW and Muon.

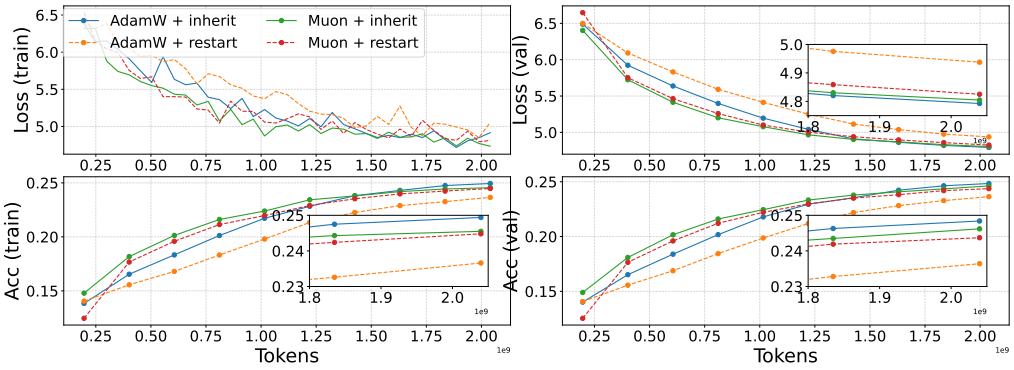


Figure 16: Optimizer-state inheritance versus restart after the 12.6M-to-25.2M depth expansion. Inheritance lowers loss and increases token accuracy for both AdamW and Muon, with a larger separation for AdamW. By the end of training, AdamW with inherited state attains the lowest validation loss among the four configurations.

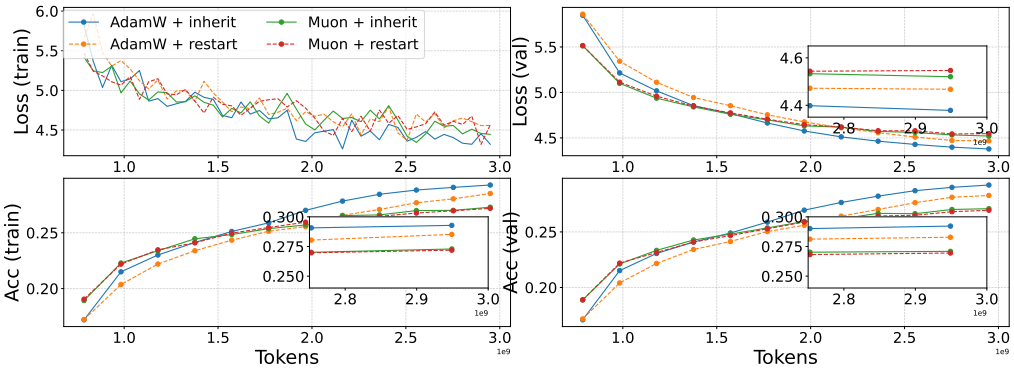


Figure 17: Optimizer-state inheritance versus restart after the 12.6M-to-50.4M depth expansion ($M = 4$). The advantages of inheriting optimizer state hold for a larger growth-factor. The benefit is still more pronounced for AdamW.

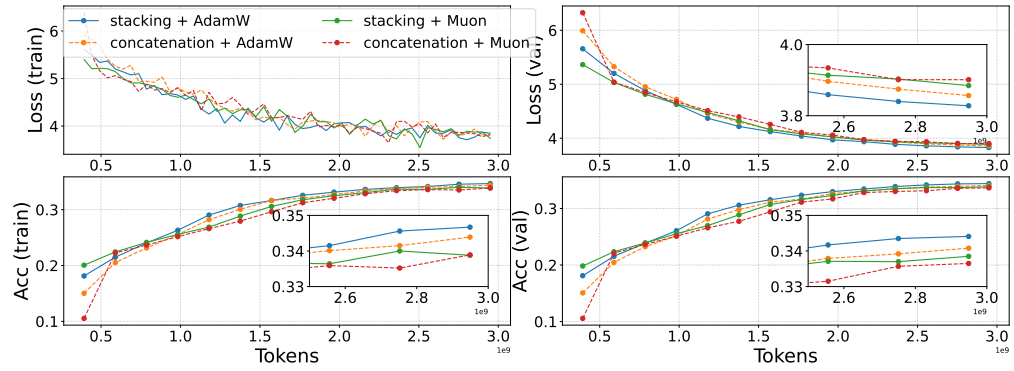


Figure 18: Stacking and concatenation for the 50.4M, 64-layer to 100.8M, 128-layer expansion. Stacking has lower loss and higher accuracy within each optimizer.

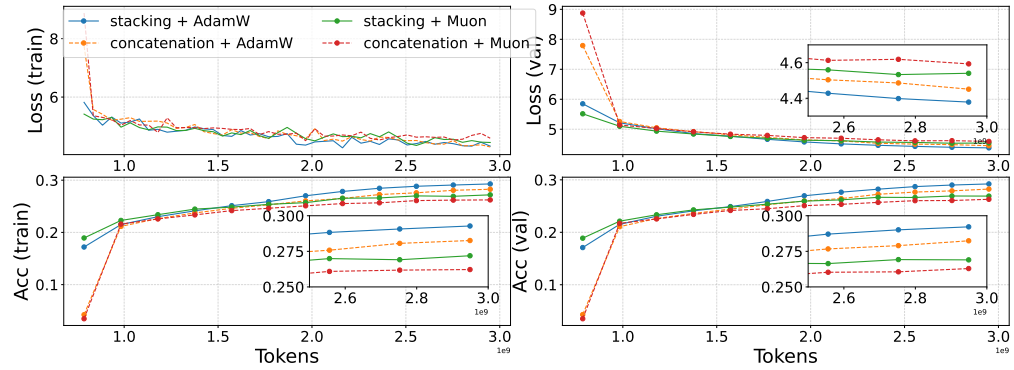


Figure 19: Stacking and concatenation for the 12.6M, 64-layer to 50.4M, 256-layer expansion ($M = 4$). The stacking configurations retain lower loss and higher accuracy at the end of training.