WHEN DOES SPARSITY MITIGATE THE CURSE OF DEPTH IN LLMS

Dilxat Muhtar[1,2,3,*,†] **Xinyuan Song**[4,*] **Sebastian Pokutta**[5,6] **Max Zimmer**[5,6]
Nico Pelleriti[5,6] **Thomas Hofmann**[7] **Shiwei Liu**[1,2,3,†]

[1]Max Planck Institute for Intelligent Systems
[2]ELLIS Institute Tübingen
[3]Tübingen AI Center
[4]Emory University
[5]Zuse Institute Berlin
[6]Technical University of Berlin
[7]ETH Zürich

pumpkindilxat@gmail.com, sliu@tue.ellis.eu

[*]Equal contribution [†]Corresponding author

ABSTRACT

Recent work has demonstrated the *curse of depth* in large language models (LLMs), where later layers contribute less to learning and representation than earlier layers. Such under-utilization is linked to the accumulated growth of variance in Pre-Layer Normalization, which can push deep blocks toward near-identity behavior. In this paper, we demonstrate that, ***sparsity***, beyond enabling efficiency, acts as a regulator of variance propagation and thereby ***improves depth utilization***. Our investigation covers two sources of sparsity: (i) implicit sparsity, which emerges from training and data conditions, including weight sparsity induced by weight decay and attention sparsity induced by long-context inputs; and (ii) explicit sparsity, which is enforced by architectural design, including key/value-sharing sparsity in Grouped-Query Attention and expert-activation sparsity in Mixture-of-Experts. Our claim is thoroughly supported by controlled depth-scaling experiments and targeted layer effectiveness interventions. Across settings, we observe a consistent relationship: sparsity improves layer utilization by reducing output variance and promoting functional differentiation. We eventually distill our findings into a practical rule-of-thumb recipe for training depth-effective LLMs, yielding a notable ***4.6% accuracy improvement*** on downstream tasks. Our results reveal sparsity, arising naturally from standard design choices, as a key yet previously overlooked mechanism for effective depth scaling in LLMs.

1 INTRODUCTION

Although LLMs exhibit remarkable capabilities, growing evidence shows that later Transformer layers are frequently under-utilized, contributing little to final performance (Gromov et al., 2024; Men et al., 2025; Csordás et al., 2025). For instance, recent studies demonstrate that skipping layers in LLMs incurs negligible performance degradation (Lad et al., 2024; Yang et al., 2024). This phenomenon reveals layer redundancy that, while enabling model compression through layer pruning (Li et al., 2025a; Dumitru et al., 2024; Yin et al., 2023), indicates inefficient utilization of training resources (Du et al., 2024; Kapl et al., 2025; Kamigaito et al., 2025).

Sun et al. (2025) have recently summarized this phenomenon as the *curse of depth (CoD)* and identified variance propagation as an underlying cause of this ineffectiveness. In widely adopted Pre-Layer Normalization (Pre-LN) architectures (Xiong et al., 2020; Kan et al., 2025; Wang et al., 2022), output variance tends to grow sub-exponentially with model depth (Sun et al., 2025; Takase et al., 2023). As variance accumulates, the magnitude of the residual stream dwarfs the updates provided by layers, causing deep layers to become functionally ineffective as their Jacobians approach the identity. Consequently, the community has largely focused on explicit variance control to mitigate this explosion, such as Scaled Initialization (Zhang et al., 2019; Luther & Seung, 2019; Takase et al., 2023), LayerNorm Scaling (Sun et al., 2025), advanced residual connections Zhu et al. (2025); Xie et al. (2025), and normalization (Li et al., 2024; Cai et al., 2025; Ding et al., 2021; Wang et al., 2024).

In parallel, a second trend has emerged in modern LLMs: the widespread adoption of *sparse computation*. Contemporary architectures increasingly incorporate sparsity at multiple levels: Mixture of Experts (MoE) activates only parameter subsets (Liu et al., 2025; Yang et al., 2025), Grouped Query Attention (GQA) reduces attention density (Ainslie et al., 2023; Shazeer, 2019), and extended sequence lengths naturally induce sparse attention patterns (Yuan et al., 2025; Xiao et al., 2023). While these innovations are typically justified by efficiency, their impact on variance propagation remains poorly understood. Intriguingly, these two approaches may be more deeply connected than previously recognized. Prior study has documented "signal collapse" in certain sparse networks, where variance diminishes as connection density decreases (Dey et al., 2024; 2025), suggesting that sparsity might inherently regulate variance. This observation raises a compelling question: *could sparsity, whether explicitly enforced or implicitly induced through training, serve as an intrinsic mechanism for mitigating the CoD by regulating variance propagation?*

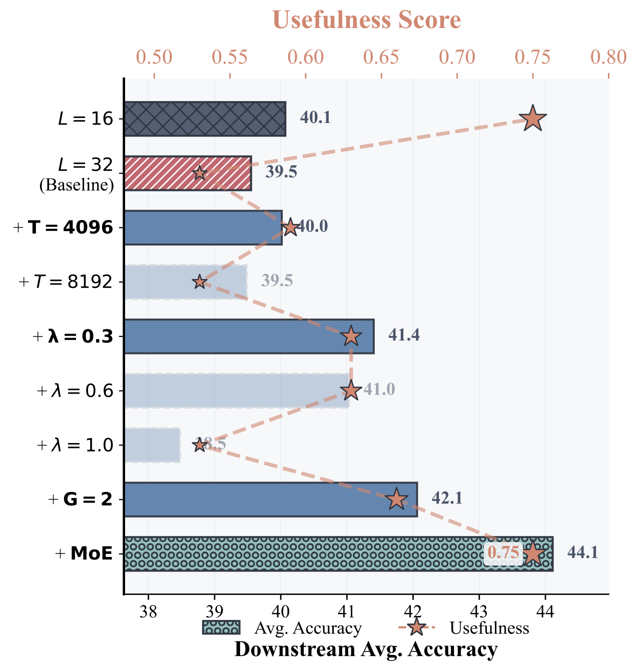
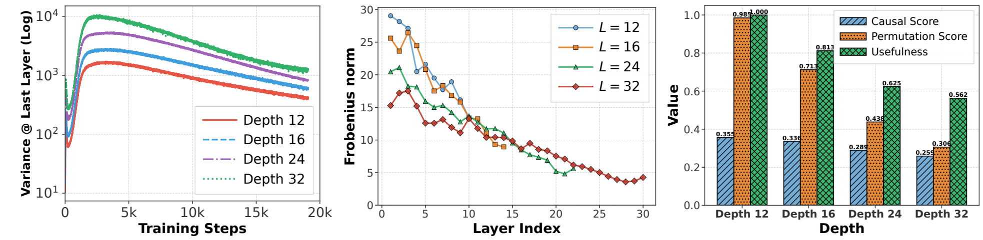


Figure 1: Progressive improvements in scaling 1.2B model to $L = 32$ depth through sparsity regularization.

In this work, we provide both theoretical and empirical evidence that sparsity serves as an intrinsic variance regulator that mitigates the CoD. We begin by characterizing the CoD through controlled experiments. We train models from scratch across varying depths (12 to 32 layers) while holding all other hyperparameters constant. To quantify layer effectiveness, we introduce three metrics: (1) Causal Score measures how much removing a layer disrupts subsequent layer representations; (2) Permutation Score quantifies layer interchangeability; and (3) Usefulness Score evaluates each layer's contribution to final performance. We then provide a formalization of how sparsity counteracts depth-induced variance explosion. We analyze two distinct paradigms: ***implicit sparsity***, i.e., weight sparsity induced through weight decay and attention sparsity caused by long-context input; and ***explicit sparsity***, i.e., enforced via GQA-style key/value sharing and sparse MoE routing. We systematically compare variance propagation and layer effectiveness across: (a) weight decay strengths, (b) sequence length scaling, (c) different GQA configurations (varying number of query groups), and (d) two MoE model scales (2B and 7B parameters) alongside their dense counterparts. Across all settings, we observe a consistent pattern: *increased sparsity correlates with reduced output variance and improved layer effectiveness.* Finally, we distill our findings into a practical rule of thumb for training depth-effective LLMs. By integrating complementary sparsity mechanisms, we train a 32-layer, 1.2B-parameter model that achieves stronger performance and improved layer effectiveness compared with a naively trained 32-layer baseline (Figure 1).

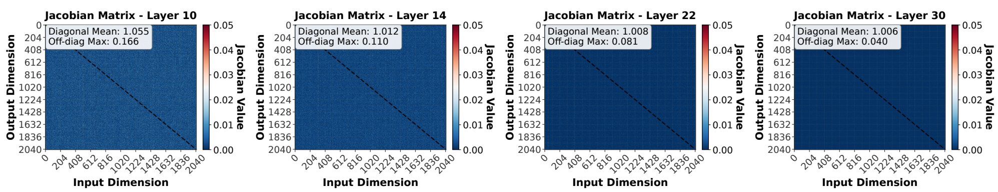
The main contributions are summarized as follows:

- We leverage three metrics, i.e., Causal Score, Permutation Score, and Usefulness Score, to quantify layer effectiveness. Using controlled depth-scaling experiments, we show that deeper models exhibit degraded layer utilization, providing empirical evidence of the CoD.
- We show that both implicit sparsity (e.g., weight decay and long-context inputs) and explicit sparsity (e.g., Mixture of Experts and Grouped Query Attention) mitigate residual-stream variance propagation, consistently reducing variance accumulation and improving layer effectiveness.
- We distill our findings into a simple rule-of-thumb for training depth-effective LLMs. combining complementary sparsity mechanisms yields a notable **4.6% accuracy gain** on downstream tasks.



(a) Last-layer variance (b) $||J - I||_F$ (Frobenius norm) (c) Layer effectiveness

Figure 2: **(a)** Last-layer variance increases with depth. **(b)** Jacobian Frobenius norm $||J - I||_F$. **(d)** Layers in deeper models exhibit lower effectiveness (usefulness and causality) and higher redundancy (permutation scores).



(a) $L = 12$ (b) $L = 16$ (c) $L = 24$ (d) $L = 32$

Figure 3: Jacobian matrices at layers $\{10, 14, 22, 30\}$ for $L \in \{12, 16, 24, 32\}$.

2 Variance Propagation and the Curse of Depth

In a Pre-LN (Xiong et al., 2020; Wang et al., 2024) Transformer block at layer ℓ, the forward pass applies layer normalization before the transformation:

$$\mathbf{x}_{\ell+1} = \mathbf{x}_\ell + \mathcal{F}(\mathrm{LN}(\mathbf{x}_\ell)), \tag{1}$$

where $\mathbf{x}_\ell \in \mathbb{R}^d$ is the input to layer ℓ, $\mathcal{F}(\cdot)$ denotes either a Multi-Head Attention (MHA) or FFN module, and $\mathrm{LN}(\cdot)$ is layer normalization.

Lemma 1 (gradient converge to identity (Sun et al., 2025) Theorem 3.3)**.** *For a Pre-LN Transformer with L layers using Equation* (1)*, assume $\sigma^2_{x_\ell}$ grows exponentially, then the partial derivative $\frac{\partial y_L}{\partial x_1}$ can be written as:*

$$\frac{\partial y_L}{\partial x_1} = \prod_{\ell=1}^{L-1} \left(\frac{\partial y_\ell}{\partial x'_\ell} \cdot \frac{\partial x'_\ell}{\partial x_\ell} \right). \tag{2}$$

We define the intermediate state $x'_\ell \in \mathbb{R}^d$ as the post-attention, pre-FFN residual state: $x'_\ell := x_\ell + \mathrm{Attn}_\ell(x_\ell)$, and the block output as $y_\ell := x'_\ell + \mathrm{FFN}_\ell(x'_\ell)$. The Euclidean norm of the upper bound for Equation (2) *is given as follows:*

$$\sigma^2_{x_\ell} \sim \exp(\ell), \quad \left\| \frac{\partial y_L}{\partial x_1} \right\|_2 \leq M, \tag{3}$$

From Lemma 1, under exponential variance growth, the Jacobian norm $\left\| \frac{\partial y_L}{\partial x_1} \right\|_2$ remains uniformly bounded by M as $L \to \infty$. Hence, depth alone does not cause instability: even at infinite depth, the Transformer stays stable, and the Weierstrass theorem guarantees convergence. Consequently, when L is very large, deeper layer transformations approach **identity mappings** from x_ℓ to y_ℓ, restricting expressivity and the model's ability to learn nontrivial mappings.

Experimental Setup To verify the accumulation of variance with model depth and examplify the CoD, we conduct controlled experiments using standard Pre-LN architectures. To isolate depth effects, we modify only the number of layers from 12 to 32, holding all other hyperparameters fixed and training on identical data. For each configuration, we perform a learning rate sweep and report results for the best-performing setting on validation data. We track last-layer output variance during training and define three metrics to quantify layer effectiveness. To verify Jacobian convergence to identity, we measure each layer's Frobenius deviation $\|J - I\|_F$. Details of each metric and experimental settings are provided in Sections B and E.1.

Main Observation We observe a clear causal chain from variance explosion to diminished layer effectiveness. Figure 2a shows that variance grows substantially with depth, aligning with (Sun et al., 2025). This variance explosion drives the Jacobian matrices toward identity mapping: Figure 2b reveals that the Frobenius norm $||J - I||_F$ decreases with depth, while Figure 3 shows increasingly diagonal-dominant Jacobian patterns in deeper models, supporting Lemma 1. Consequently, layer effectiveness deteriorates: Figure 2c demonstrates that all three score progressively decline, with Usefulness dropping from 0.75 ($L = 12$) to 0.53 ($L = 32$). While $L = 32$ achieves better performance (Table 1) with more effective layers (18 vs. 12 for $L = 12$), it exhibits severe inefficiency: using $2.56\times$ more parameters while wasting 14 layers, exemplifying CoD where most layers contribute minimally despite consuming substantial training compute. Additional analyses are provided in Sections C.1 to C.4.

Table 1: Evaluation results for models with different depths. E/W represents effective layers (E) versus wasted layers (W) as measured by the usefulness score (Section B).

Depth$L$	# of Param.	E/W	PPL↓	MMLU	ARC-C	ARC-E	Hellaswag
L=12	900M	12/0	13.72	24.20	30.89	58.96	47.24
L=16	1.2B	13/3	13.65	25.20	31.72	60.45	47.24
L=24	1.7B	15/9	13.53	26.39	32.59	61.24	47.50
L=32	2.3B	18/14	**13.47**	**27.52**	**34.22**	**61.74**	**48.63**

3 Sparsity as Variance Regularizer

Having established that variance propagation is a contributor to CoD, we now investigate sparsity as a mechanism to control variance accumulation.

3.1 Theoretical Analysis

Sparsity acts as a variance regularizer in residual stacks by attenuating the energy passed to each layer update. The following lemma quantifies this effect: the per-layer variance gain scales as $\alpha_\ell \rho_\ell$, so smaller mask density ρ_ℓ yields slower variance growth with depth.

Theorem 1 (Sparsity reduces variance propagation in residual-depth)**.** *Let $\{r_\ell\}_{\ell=0}^{L}$ follow the residual-depth recursion*

$$r_{\ell+1} = r_\ell + u_\ell, \quad u_\ell := W_\ell\big(D_\ell r_\ell\big), \qquad \ell = 0, \dots, L-1, \tag{4}$$

where $r_\ell \in \mathbb{R}^d$, $D_\ell \in \mathbb{R}^{d\times d}$ is a diagonal 0-1 mask, and $W_\ell \in \mathbb{R}^{d\times d}$ is a random linear map. Assume that for each ℓ, W_ℓ is independent of (r_ℓ, D_ℓ) and satisfies the second-moment bound and that the mask satisfies the density bound

$$\mathbb{E}\left[\|W_\ell x\|_2^2\right] \le \alpha_\ell \|x\|_2^2, \quad \mathbb{E}\left[\|D_\ell x\|_2^2\right] \le \rho_\ell \|x\|_2^2 \tag{5}$$

for all $x \in \mathbb{R}^d$, and for some $\rho_\ell \in [0,1]$. If $\alpha_\ell \le \alpha$ and $\rho_\ell \le \rho < 1$ for all ℓ, then the residual variance satisfies

$$\mathrm{Var}(r_L) \le \mathrm{Var}(r_0) \prod_{\ell=0}^{L-1} \Big(1 + \sqrt{\alpha_\ell \rho_\ell}\Big)^2 = O\Big(1 + \sqrt{\alpha\rho}\Big)^{2L}. \tag{6}$$

The proof is provided in D.1. Theorem 1 shows that the variance bound depends on sparsity only through ρ_ℓ: smaller ρ_ℓ (sparser D_ℓ) yields a smaller per-layer factor $\big(1 + \sqrt{\alpha_\ell \rho_\ell}\big)^2$, and therefore a smaller upper bound on $\mathrm{Var}(r_L)$. Hence, sparsity (captured by ρ_ℓ) directly controls variance propagation: smaller ρ_ℓ yields a smaller bound on $\mathrm{Var}(r_L)$ across depth. Since ρ_ℓ can be reduced not only by explicit masking but also by training-induced sparsity patterns, the resulting variance reduction can be viewed as an implicit regularization effect. The detailed result for a transformer based residual model is in Section D.2.

3.2 Implicit and Explicit Sparsity

Motivated by our theoretical analysis, we identify specific sparsity dimensions for experimental investigation. We categorize sparsity into implicit sparsity and explicit sparsity.

3.2.1 Implicit Sparsity

Implicit sparsity refers to sparsity induced dynamically during training. This includes sparsity from regularization (weight decay (Krogh & Hertz, 1991; Loshchilov & Hutter, 2019)), activation functions

(ReLU (Glorot et al., 2011; Hayou et al., 2019)), and input-dependent mechanisms (dropout (Srivastava et al., 2014), attention patterns (Vaswani, 2017)) that drive parameters or activations toward negligible values (Frankle & Carbin, 2018; Chen et al., 2020). We investigate two implicit sparsity dimensions: (1) weight decay, and (2) sequence length scaling.

Weight Decay Weight decay applies L_2 regularization to model parameters, adding penalty term $\lambda\|W\|_2^2$ to the loss function (Loshchilov & Hutter, 2019). This drives small-magnitude parameters toward zero, inducing sparsity without structural constraints. We quantify the induced sparsity by measuring the fraction of effectively zero parameters. For a trained model with parameter set Θ, we define:

$$\text{Sparsity}(\Theta;\epsilon) = \frac{1}{|\Theta|}\sum_{w\in\Theta} \mathbb{1}[|w| < \epsilon], \tag{7}$$

where ϵ is a threshold and $\mathbb{1}[\cdot]$ is the indicator function. We theoretically discuss the relationship of weight decay and sparsity in Appendix D.4.

Sequence Length Sequence length scaling induce implicit sparsity in attention mechanisms through positional bias and softmax normalization (Su et al., 2024; Xiao et al., 2023; Zhang et al., 2023). Position embeddings like RoPE (Su et al., 2024) introduce distance-dependent attention decay that the dot product decreases with relative distance. As sequence length T increases, this distance penalty causes attention to concentrate on a subset of positions with favorable relative distances. Softmax normalization over longer sequences produces more peaked distributions, concentrating attention on top-scoring positions while suppressing others toward zero (Xiao et al., 2023; Zhang et al., 2023; Yuan et al., 2025). We quantify attention sparsity by measuring the fraction of near-zero attention weights. For attention weights $\mathbf{A}_{\ell,h} \in \mathbb{R}^{T\times T}$ at layer ℓ and head h, we compute the sparsity at threshold ϵ as:

$$\text{Sparsity}_{\ell,h}(\epsilon) = \frac{1}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T} \mathbb{1}[A_{i,j} < \epsilon], \tag{8}$$

where $\mathbb{1}[\cdot]$ is the indicator function. This measures the percentage of attention weights below ϵ. The global attention sparsity aggregates across all layers and heads:

$$\text{Sparsity}_{\text{global}}(\epsilon) = \frac{1}{L\cdot H}\sum_{\ell=1}^{L}\sum_{h=1}^{H} \text{Sparsity}_{\ell,h}(\epsilon) \tag{9}$$

where L is the number of layers and H is the number of attention heads per layer. We discuss the relationship between sequence length and sparsity in Appendix D.5.

3.2.2 Explicit Sparsity

Explicit sparsity refers to architectural constraints that hard-code sparsity patterns into the model structure, ensuring that a predetermined fraction of connections or computational paths are absent by design (Ainslie et al., 2023; Dai et al., 2024; Fedus et al., 2022; Abnar et al., 2025). Unlike implicit sparsity that emerges dynamically during training, explicit sparsity is fixed at model initialization through architectural choices. In this study, we examine two prominent forsm of explicit sparsity in LLM design: (1) Grouped Query Attention (GQA), and (2) Mixture of Experts (MoE).

Grouped Query Attention GQA (Ainslie et al., 2023; Shazeer, 2019) reduces attention computation by sharing key-value heads across multiple query heads. In standard multi-head attention with H heads, each head maintains independent query ($\mathbf{Q}_h$), key ($\mathbf{K}_h$), and value ($\mathbf{V}_h$) projections. GQA partitions the H query heads into G groups, where all queries in a group share the same key-value:

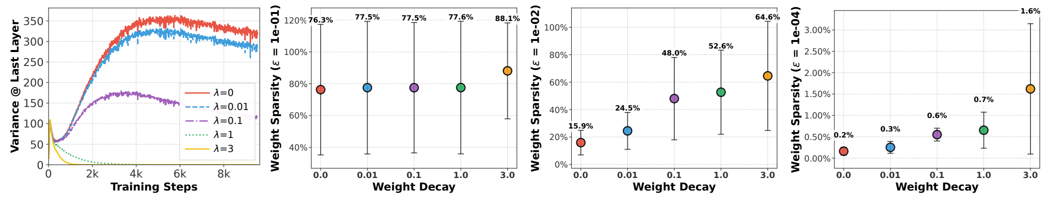
$$\text{Attention}_h(\mathbf{Q}_h, \mathbf{K}_{\lfloor h/G\rfloor}, \mathbf{V}_{\lfloor h/G\rfloor}). \tag{10}$$

This reduces the number of independent key-value computations from H to H/G, creating explicit sparsity. See Appendix D.6 for further discussion of GQA and sparsity.

Mixture of Experts MoE introduces sparsity by replacing dense FFN layers with multiple expert networks, activating only k out of E experts per token (Fedus et al., 2022; Bai et al., 2023; Dai et al., 2024; Bi et al., 2024). Specifically, a gating network routes each token to its top-k experts:

$$\mathbf{h}_{\text{out}} = \sum_{i\in\text{Top-}k(\mathbf{W}_g\mathbf{x})} g_i(\mathbf{x})\cdot \text{Expert}_i(\mathbf{x}) \tag{11}$$

where $g_i(\mathbf{x})$ are normalized gating weights and Expert_i are independent FFN networks. We provide an in-depth analysis of the relationship between sparsity and MoE in Appendix D.7.



(a) Last-layer variance (b) ϵ=1 $\times$ 10^{-1} (c) ϵ=1 $\times$ 10^{-2} (d) ϵ=1 $\times$ 10^{-4}

Figure 4: **(a)** Variance decreases with stronger weight decay. **(b)-(d)** Weight sparsity (fraction of weights $< \epsilon$) increases with weight decay at thresholds $\epsilon \in \{10^{-1}, 10^{-2}, 10^{-4}\}$.

4 Verification of Variance Dampening

4.1 Implicit Sparsity

4.1.1 Weight Decay

Experimental Settings We evaluate the influence of weight decay on model sparsity and variance propagation by training 1.2B-parameter models with varying weight decay coefficients. We employ the AdamW optimizer (Loshchilov & Hutter, 2019) with weight decay values of $\lambda \in \{0, 0.01, 0.1, 1.0, 3.0\}$, while keeping all other hyperparameters fixed. We analyze four key metrics: (1) model parameter sparsity as defined in Section 3.2.1; (2) the evolution of last-layer output variance throughout training; (3) validation perplexity and layer effectiveness scores. Detailed hyperparameters are provided in Section E.2.

Table 2: Validation perplexity and layer effectiveness scores across weight decay.

Weight Decay (λ)	PPL ($\downarrow$)	Effectiveness Score		
		Causal	Permutation	Usefulness
$\lambda = 0$	15.63	0.25	0.41	0.75
$\lambda = 0.01$	15.20	0.26	0.42	0.75
$\lambda = 0.1$	**14.83**	**0.31**	**0.52**	**0.81**
$\lambda = 1.0$	15.55	0.20	0.50	0.69
$\lambda = 3.0$	773.42	0.03	0.07	0.63

Main Observation Variance trajectories (Figure 4a) show consistent reduction with stronger weight decay, with last-layer variance decreasing as λ increases. At extreme values ($\lambda \in \{1.0, 3.0\}$), variance falls below 25, with $\lambda = 3.0$ driving most weights below 10^{-2} (Figure 4c). Weight sparsity increases correspondingly across all thresholds (Figures 4b to 4d), confirming that L_2 regularization induces parameter-level sparsity. Performance analysis (Table 2) reveals that within the optimal range ($\lambda \in [0, 0.1]$), both perplexity (15.63→14.83) and usefulness score (0.75→0.81) improve with weight decay, demonstrating that sparsification enhances model quality and depth utilization. However, excessive regularization ($\lambda \geq 1.0$) causes degradation despite continued variance reduction, with $\lambda = 3.0$ leading to collapse (perplexity 773.42). This *over-dampening phenomenon*, where excessive weight decay cripples model capacity despite achieving low variance, demonstrates that variance control must be balanced with model capacity.

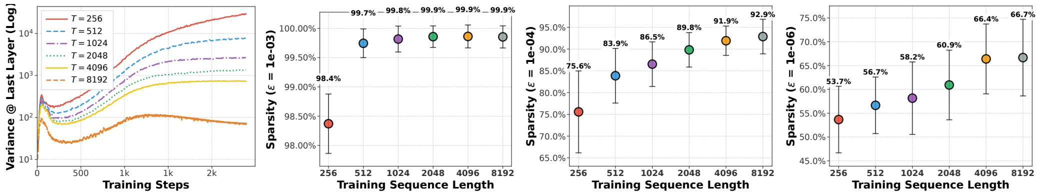
4.1.2 Sequence Length

Experimental Settings To evaluate the effectiveness of sequence length scaling for variance dampening, we train 1.2B models with varying maximum sequence lengths ranging from $T = 256$ to $T = 8192$ tokens. To isolate the effect of sequence length, we maintain constant computational budget across all configurations by adjusting the number of training steps inversely proportional to sequence length. All other hyperparameters remain identical across experiments. We analyze three key metrics: (1) the evolution of last-layer output variance throughout training, (2) the induced sparsity (defined in Section 3.2.1), and (3) validation perplexity and layer effectiveness scores. Detailed hyperparameters and training configurations are provided in Section E.3.

Table 3: Validation perplexity and layer effectiveness scores across training sequence lengths.

Sequence Length (T)	PPL ($\downarrow$)	Effectiveness Score		
		Causal	Permutation	Usefulness
$T = 256$	18.51	0.24	0.41	0.69
$T = 512$	15.71	0.25	0.45	0.75
$T = 1024$	14.77	0.28	0.46	0.75
$T = 2048$	**14.51**	0.30	0.50	**0.81**
$T = 4096$	14.52	**0.31**	**0.81**	**0.81**
$T = 8192$	16.30	0.29	0.49	0.75

Main Observation Variance trajectories (Figure 5a) show consistent reduction with longer training sequences, with $T = 256$ exhibiting the highest variance. Measuring attention sparsity on held-out



(a) Last-layer variance (b) ϵ=1 $\times$ 10^{-3} (c) ϵ=1 $\times$ 10^{-4} (d) ϵ=1 $\times$ 10^{-6}

Figure 5: **(a)** Last-layer output variance decreases with longer sequences. **(b)-(d)** Attention sparsity (fraction of weights below threshold ϵ) increases with sequence length across all thresholds. Results averaged over 3 random seeds.

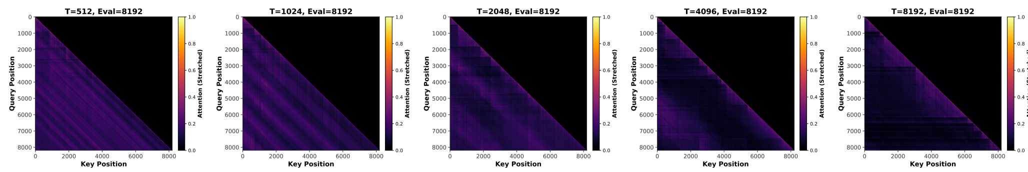


Figure 6: Average attention maps from head 0 across all layers for models trained with different sequence lengths (averaged over 100 random held-out samples). Attention weights are power-scaled ($A^{0.2}$) for visualization.

samples at $T = 8192$ (Figures 5b to 5d and 6), we observe increasing sparsity across all thresholds ($\epsilon \in \{10^{-3}, 10^{-4}, 10^{-6}\}$) as training length grows, confirming that longer contexts induce stronger implicit sparsity. Performance analysis (Table 3) reveals optimal scaling from $T = 256$ to $T = 2048$: perplexity improves by 4+ points while usefulness increases (0.69→0.81). However, further scaling to $T = 8192$ yields diminishing returns despite lower variance, suggesting *over-dampening* that restricts model capacity similar to excessive weight decay (Li et al., 2025b; Izzo et al., 2025; Saada et al., 2024). Additional analysis of attention entropy and sparsity at different evaluation lengths is provided in Sections C.6 and C.7.

4.2 Explicit Sparsity

4.2.1 Grouped Query Attention

Experimental Settings We analyze the variance dampening effect of GQA (Ainslie et al., 2023) by varying the number of key-value head groups G. We train 1.2B models with equal training FLOPs using group sizes $G \in \{1, 4, 16\}$ (MHA, GQA, and MQA (Shazeer, 2019), respectively). We evaluate the last-layer output variance, validation perplexity, and layer effectiveness scores. Detailed model definition and training configurations are provided in Section E.4.

Table 4: Impact of GQA on model performance and layer-wise effectiveness.

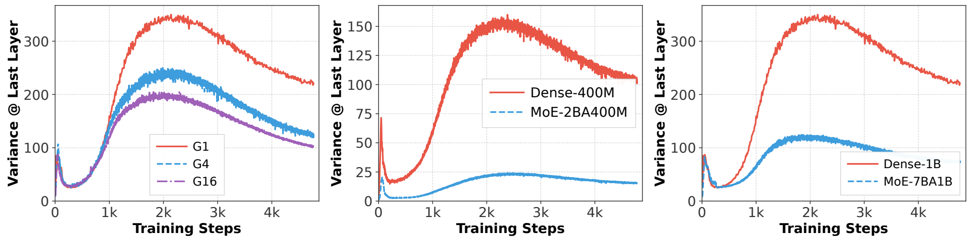
Group Size (G)	PPL (↓)	Effectiveness Score		
		Causal	Permutation	Usefulness
$G = 1$	14.52	0.31	0.51	0.81
$G = 4$	14.50	0.31	0.57	**0.87**
$G = 16$	**14.47**	**0.34**	**0.63**	**0.87**

Main Observation Results in Figure 7a show that variance decreases monotonically with group size. MQA ($G = 16$) exhibits 2× lower variance than MHA ($G = 1$). Under equal training FLOPs (Table 4), MQA achieves both better performance (PPL: 14.52 → 14.47) and higher layer effectiveness: usefulness score increases 7% (0.81 → 0.87), yielding 13 effective layers for MHA versus 14 for MQA. This confirms that structural sparsity through key-value sharing dampens variance and improves layer utilization.

4.2.2 Mixture of Experts

Experimental Settings We adopt fine-grained experts (Dai et al., 2024; Muennighoff et al., 2024) and evaluate two configurations: (1) 7B total parameters (1B active) with 64 experts and top-8 routing; (2) 2B total parameters (400M active) with 32 experts, top-4 routing, and one shared expert. We benchmark against dense baselines matched by active parameter count. Details are in Section E.5.

Main Observation We compare the variance of dense and MoE models in Figure 7.



(a) GQA Variants. (b) 400M activated param. (c) 1B activated param.

Figure 7: **(a)** Output variance with different GQA configurations. **(b)-(c)** Comparison of last-layer variance between MoE and dense models with different activated parameters

Table 5: Validation perplexity and comparison of layerwise effectiveness between dense and MoE models.

	PPL (↓)	Effectiveness Score		
		Causal	Permutation	Usefulness
Dense-400M	17.56	0.30	0.54	0.87
MoE-2BA400M	**15.89**	**0.36**	**0.59**	**0.94**
Dense-1B	14.52	0.31	0.51	0.81
MoE-7BA1B	**13.82**	**0.34**	**0.61**	**0.94**

Results for both the 400M-active and 1B-active configurations show that the MoE architecture dampens variance by approximately 6× and 3×, respectively. Moreover, the MoE variants not only outperform their dense counterparts by over 2 perplexity but also exhibit higher layer effectiveness (e.g., usefulness score increases from 0.81 to 0.94 for the 1B configuration) (Table 5). These results demonstrate that explicit sparsity through MoE improves layer effectiveness.

5 Sparsity as an Enabler for Depth

Observing both implicit and explicit sparsity reduce variance, we explore scaling from $L = 16$ to $L = 32$ (at constant 1.2B parameters) while integrating different sparsity strategies to achieve better performance and layer utilization. All experiments use identical training data and parameter budgets, with learning rate sweeps for each configuration. We evaluate using average accuracy on ARC-C (Clark et al., 2018) and HellaSwag (Zellers et al., 2019), and Usefulness Score. Model and training details are in Section E.6.

Figure 1 reveals the solution for effective depth scaling. Naively increasing depth from $L = 16$ to $L = 32$ degrades performance: accuracy drops by 0.5 while Usefulness Score plummets from 0.75 to 0.53, indicating approximately 50% layer underutilization. However, progressively integrating sparsity recovers and surpasses the baseline. Extending context to $T = 4096$ recovers accuracy (40.0) and improves utilization (0.59), though $T = 8192$ shows diminishing returns due to over-dampening. Introducing implicit sparsity via weight decay also proves effective: $\lambda = 0.3$ increases accuracy to 41.4 with Usefulness Score rising to 0.63 (representing 20% improvement over the naive baseline). This trend continues with $\lambda = 0.6$ achieving competitive results, though excessive regularization ($\lambda = 1.0$) degrades both performance and utilization. Finally, combining explicit sparsity through GQA ($G = 2$) and MoE yields the best configuration: average accuracy of 44.1 and Usefulness Score of 0.75, representing a 4+ point accuracy gain over the $L = 16$ baseline while maintaining better layer utilization than the naive $L = 32$ model (0.75 vs. 0.53). These results demonstrate that appropriate sparsity mechanisms are essential for effective depth scaling.

6 Conclusion

We show that variance growth in deep networks leads to a *curse of depth*, where layers become increasingly underutilized as Jacobians drift toward identity mappings. We theoretically and empirically validate that sparsity, implicit or explicit, damps variance propagation, improving layer utilization and enabling effective depth scaling with better performance. This reframes sparsity as not only a compute-saving tool, but also an optimization mechanism that improves compute utilization by controling variance growth.

REFERENCES

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

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Wenrui Cai, Defa Zhu, Qingjie Liu, and Qiyang Min. Seednorm: Self-rescaled dynamic normalization. *arXiv preprint arXiv:2510.22777*, 2025.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained bert networks. *Advances in neural information processing systems*, 33:15834–15846, 2020.

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, 2018. URL `https://arxiv.org/abs/1803.05457`.

Róbert Csordás, Christopher D Manning, and Christopher Potts. Do language models use their depth efficiently? *arXiv preprint arXiv:2505.13898*, 2025.

Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Yu Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

Nolan Dey, Shane Bergsma, and Joel Hestness. Sparse maximal update parameterization: A holistic approach to sparse training dynamics. *Advances in Neural Information Processing Systems*, 37: 33836–33862, 2024.

Nolan Dey, Bin Claire Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: Completep enables compute-efficient deep transformers. *arXiv preprint arXiv:2505.01618*, 2025.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *NeurIPS*, 34:19822–19835, 2021.

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. *Advances in Neural Information Processing Systems*, 37:10491–10540, 2024.

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

Razvan-Gabriel Dumitru, Vikas Yadav, Rishabh Maheshwary, Paul-Ioan Clotan, Sathwik Tejaswi Madhusudhan, and Mihai Surdeanu. Layer-wise quantization: A pragmatic and effective method for quantizing llms beyond integer bit-levels. *arXiv preprint arXiv:2406.17415*, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022. URL `https://arxiv.org/abs/2101.03961`.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. Megablocks: Efficient sparse training with mixture-of-experts, 2022. URL `https://arxiv.org/abs/2211.15841`.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323. JMLR Workshop and Conference Proceedings, 2011.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*, 2024.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. In *International conference on machine learning*, pp. 2672–2680. PMLR, 2019.

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.

Zachary Izzo, Eshaan Nichani, and Jason D Lee. Quantitative bounds for length generalization in transformers. *arXiv preprint arXiv:2510.27015*, 2025.

Hidetaka Kamigaito, Ying Zhang, Jingun Kwon, Katsuhiko Hayashi, Manabu Okumura, and Taro Watanabe. Diversity of transformer layers: One aspect of parameter scaling laws. *arXiv preprint arXiv:2505.24009*, 2025.

Kelvin Kan, Xingjian Li, Benjamin J Zhang, Tuhin Sahai, Stanley Osher, Krishna Kumar, and Markos A Katsoulakis. Stability of transformers under layer normalization. *arXiv preprint arXiv:2510.09904*, 2025.

Ferdinand Kapl, Emmanouil Angelis, Tobias Höppe, Kaitlin Maile, Johannes von Oswald, Nino Scherrer, and Stefan Bauer. Do depth-grown models overcome the curse of depth? an in-depth analysis. *arXiv preprint arXiv:2512.08819*, 2025.

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

Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. The remarkable robustness of llms: Stages of inference? *arXiv preprint arXiv:2406.19384*, 2024.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Pengxiang Li, Lu Yin, and Shiwei Liu. Mix-ln: Unleashing the power of deeper layers by combining pre-ln and post-ln. *arXiv preprint arXiv:2412.13795*, 2024.

Pengxiang Li, Lu Yin, Xiaowei Gao, and Shiwei Liu. Owlore: Outlier-weighed layerwise sampled low-rank projection for memory-efficient llm fine-tuning, 2024. *URL https://arxiv.org/abs/2405.18380*, 2025a.

Ruining Li, Gabrijel Boduljak, et al. On vanishing variance in transformer length generalization. *arXiv preprint arXiv:2504.02827*, 2025b.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration, 2024. URL `https://arxiv.org/abs/2306.00978`.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL `https://arxiv.org/abs/1711.05101`.

Kyle Luther and H Sebastian Seung. Variancepreserving initialization schemes improve deep network training: But which variance is preserved. *arXiv preprint arXiv:1902.04942*, 2019.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20192–20204, 2025.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. Olmoe: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

Thiziri Nait Saada, Alireza Naderi, and Jared Tanner. Mind the gap: a spectral analysis of rank collapse and signal propagation in attention layers. *arXiv preprint arXiv:2410.07799*, 2024.

Noam Shazeer. Fast transformer decoding: One write-head is all you need, 2019. URL `https://arxiv.org/abs/1911.02150`.

Noam Shazeer. Glu variants improve transformer, 2020. URL `https://arxiv.org/abs/2002.05202`.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL `https://arxiv.org/abs/1701.06538`.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Wenfang Sun, Xinyuan Song, Pengxiang Li, Lu Yin, Yefeng Zheng, and Shiwei Liu. The curse of depth in large language models. *arXiv preprint arXiv:2502.05795*, 2025.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. Spike no more: Stabilizing the pre-training of large language models. *arXiv preprint arXiv:2312.16903*, 2023.

A Vaswani. Attention is all you need. *NeurIPS*, 2017.

Hongyu Wang, S Ma, L Dong, S Huang, D Zhang, and F Wei. Deepnet: Scaling transformers to 1000 layers. arxiv 2022. *arXiv preprint arXiv:2203.00555*, 2022.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *TPAMI*, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Kuai Yu, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. mhc: Manifold-constrained hyper-connections, 2025. URL `https://arxiv.org/abs/2512.24880`.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *ICML*, pp. 10524–10533. PMLR, 2020.

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.

Yifei Yang, Zouying Cao, and Hai Zhao. Laco: Large language model pruning via layer collapse. *arXiv preprint arXiv:2402.11187*, 2024.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Jaiswal, Mykola Pechenizkiy, Yi Liang, et al. Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity. *arXiv preprint arXiv:2310.05175*, 2023.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23078–23097, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019. URL `https://arxiv.org/abs/1905.07830`.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *NeurIPS*, 32, 2019.

Biao Zhang, Ivan Titov, and Rico Sennrich. Improving deep transformer with depth-scaled initialization and merged attention. *arXiv preprint arXiv:1908.11365*, 2019.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections, 2025. URL `https://arxiv.org/abs/2409.19606`.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models, 2022. URL `https://arxiv.org/abs/2202.08906`.

A Related Work

A.1 The Curse of Depth

While deeper networks achieve superior performance with sufficient data and compute (Kaplan et al., 2020; Hoffmann et al., 2022), recent studies reveal that modern Pre-LN LLMs (Yang et al., 2025; Bi et al., 2024; Dubey et al., 2024) suffer from increasing layer redundancy (Gromov et al., 2024; Men et al., 2025; Lad et al., 2024; Csordás et al., 2025). This degradation, termed the *Curse of Depth (CoD)* (Sun et al., 2025), stems from exponential variance growth that forces deeper layers toward identity mappings. Prior work addresses CoD through explicit variance control via scaled initialization (Takase et al., 2023; Zhang et al., 2019; Luther & Seung, 2019), normalization scaling (Sun et al., 2025; Zhu et al., 2025), or alternative normalization (Li et al., 2024; Cai et al., 2025). In contrast, we investigate whether sparsity from different choices can naturally mitigate CoD without explicit modifications.

A.2 Sparsity in Neural Networks

Sparsity has been central to neural network research for both biological inspiration (LeCun et al., 1989) and computational efficiency (Han et al., 2015). We categorize sparsity into two paradigms. **Implicit sparsity** emerges naturally from training dynamics and architectural choices: ReLU activations zero negative values (Glorot et al., 2011; Hayou et al., 2019), attention mechanisms concentrate on token subsets (Su et al., 2024; Xiao et al., 2023; Zhang et al., 2023; Yuan et al., 2025), and weight decay drives small parameters toward zero (Krogh & Hertz, 1991; Loshchilov & Hutter, 2019; Frankle & Carbin, 2018). **Explicit sparsity** is architecturally enforced to decouple capacity from computation: Mixture-of-Experts (MoE) activates only k of E experts per token (Fedus et al., 2022; Dai et al., 2024; Liu et al., 2025), while Grouped Query Attention (GQA) shares key-value projections across query heads (Ainslie et al., 2023; Shazeer, 2019; Dubey et al., 2024; Yang et al., 2025). While sparsity's efficiency benefits are well-established, its impact on variance propagation remains unexplored. We systematically demonstrate that both sparsity paradigms constrain variance propagation and mitigate CoD in LLMs.

B Definition of Each Score

Last-Layer Variance For final hidden states $\mathbf{h}_L \in \mathbb{R}^{n\times d}$, we compute per-token variance across dimensions, averaged over tokens:

$$\text{Var}(L) = \frac{1}{n}\sum_{i=1}^{n} \text{Var}(\mathbf{h}_{L,i,:}) = \frac{1}{nd}\sum_{i=1}^{n}\sum_{j=1}^{d}\left(h_{L,i,j} - \bar{h}_{L,i}\right)^2, \tag{12}$$

where $\bar{h}_{L,i} = \frac{1}{d}\sum_j h_{L,i,j}$ is the per-token mean. High variance indicates signal accumulation across depth, causing layer gradient to become negligible (Sun et al., 2025)

Causal Score The causal score measures how much each layer influences the computations of all subsequent layers (Csordás et al., 2025). For a model with N layers and hidden states $\mathbf{h}_\ell$ at layer ℓ, we define the causal effect of layer s on layer $\ell > s$ as:

$$C(s,\ell) = \frac{\|(\mathbf{h}_{\ell+1} - \mathbf{h}_\ell) - (\bar{\mathbf{h}}_{\ell+1} - \bar{\mathbf{h}}_\ell)\|_2}{\|\mathbf{h}_{\ell+1} - \mathbf{h}_\ell\|_2}, \tag{13}$$

where $\mathbf{h}_\ell$ denotes hidden states in the baseline model, and $\bar{\mathbf{h}}_\ell$ denotes hidden states when layer s is skipped. The global causal score aggregates these effects across all layer pairs:

$$S_{\text{causal}} = \frac{1}{\sqrt{N}} \cdot \frac{1}{N}\sum_{s=0}^{N-1}\left[\frac{1}{N-s-1}\sum_{\ell=s+1}^{N-1} C(s,\ell)\right], \tag{14}$$

where $1/\sqrt{N}$ normalizes for model depth. Higher causal scores indicate critical layers whose removal affects subsequent layers, while lower scores suggest minimal impact and potential redundancy.

Permutation Score The permutation score quantifies layer specialization by measuring performance degradation when layer positions are swapped Kapl et al. (2025). For layers ℓ_1 and ℓ_2, the pairwise permutation score is:

$$P(\ell_1, \ell_2) = \frac{|\mathcal{L}(M) - \mathcal{L}(M_{\text{swap}(\ell_1,\ell_2)})|}{|\mathcal{L}(M)|}, \tag{15}$$

where $\mathcal{L}(M)$ is the baseline loss and $\mathcal{L}(M_{\text{swap}(\ell_1,\ell_2)})$ is the loss after swapping. The global permutation score averages over all possible layer pairs:

$$S_{\text{permutation}} = \frac{2}{N(N-1)} \sum_{\ell_1=0}^{N-1} \sum_{\ell_2=\ell_1+1}^{N-1} P(\ell_1, \ell_2). \tag{16}$$

Higher scores indicate that layers are less interchangeable, while scores near zero suggest redundancy.

Usefulness Score The usefulness score quantifies each layer's contribution through linear approximation Kapl et al. (2025); Csordás et al. (2025); Sun et al. (2025). This approach measures the degree of nonlinearity each layer contributes, which is a fundamental indicator of computation beyond linear mappings.

For each layer ℓ, we collect input-output pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ and fit an optimal linear approximation via least-squares. We then measure performance degradation when replacing layer ℓ with this linear transformation:

$$\Delta\mathcal{L}_\ell = \mathcal{L}(M_\ell^{\text{linear}}) - \mathcal{L}(M). \tag{17}$$

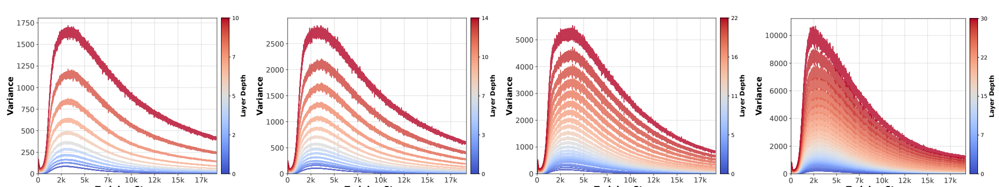
The global usefulness score measures the fraction of layers with significant ($> \alpha$) performance impact:

$$S_{\text{useful}} = \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\left[\frac{\mathcal{L}(M_\ell^{\text{linear}})}{\mathcal{L}(M)} > 1 + \alpha\right], \tag{18}$$

where we set $\alpha = 0.1$. This quantifies the model's *effective nonlinear depth*—the fraction of layers performing meaningful nonlinear transformations. Higher scores indicate efficient depth utilization; lower scores reveal redundancy.

C Additional Results

C.1 Variance Across All Depths



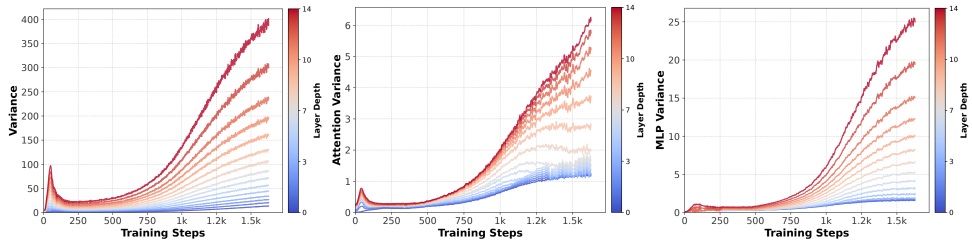
(a) $L = 12$ (b) $L = 16$ (c) $L = 24$ (d) $L = 32$

Figure 8: Variance of each layer's output during training for models of different depths in Section 2.

Instead of only the last layer's variance reported in previous settings, we also provide the variance of each layer during training for models with different depths in the depth control experiment in Section 2. The results are presented in Figure 8, where we can clearly see that the variance accumulates with depth.

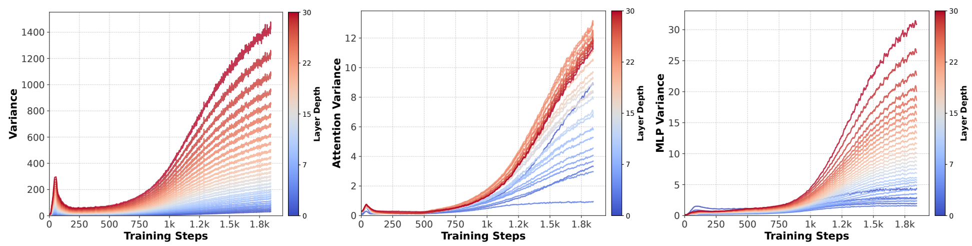
C.2 Variance of Different Blocks

Across all of our other experiments, we report the variance of each layer's output (e.g., the decoder layer in LLMs). In this subsection, we experiment with a model of $d = 1536$ hidden dimensions, MLP dimension of 4608, $h = 12$ attention heads, and maximum sequence length of 4096 with two depths: 16 and 32. We train for 1500+ steps using the FineWeb-Edu dataset with a learning rate of 1×10^3, weight decay of 0.1, and global batch size of 256. We report each layer's attention output variance and MLP output variance during training.



(a) Variance of decoder-block output. (b) Variance of attn. block output (c) Variance of MLP block output.

Figure 9: Variance of each layer's output during training with model $L = 16$.



(a) Variance of decoder-block output. (b) Variance of attn. block output (c) Variance of MLP block output.

Figure 10: Variance of each layer's output during training with model $L = 32$.

Results are presented in Figures 9 and 10. While output variance grows continuously with depth, MLP block variances increase more rapidly and show consistently stronger growth across depth compared to attention blocks. Interestingly, although the overall output variance exhibits substantial growth, the variance within the residual components (attention and MLP blocks) themselves grows only moderately. This suggests that the shortcut paths may also contribute significantly to variance accumulation and may require regularization methods to control the variance growth, which we leave for future work.

C.3 Jacobian Matrices of Different Layers

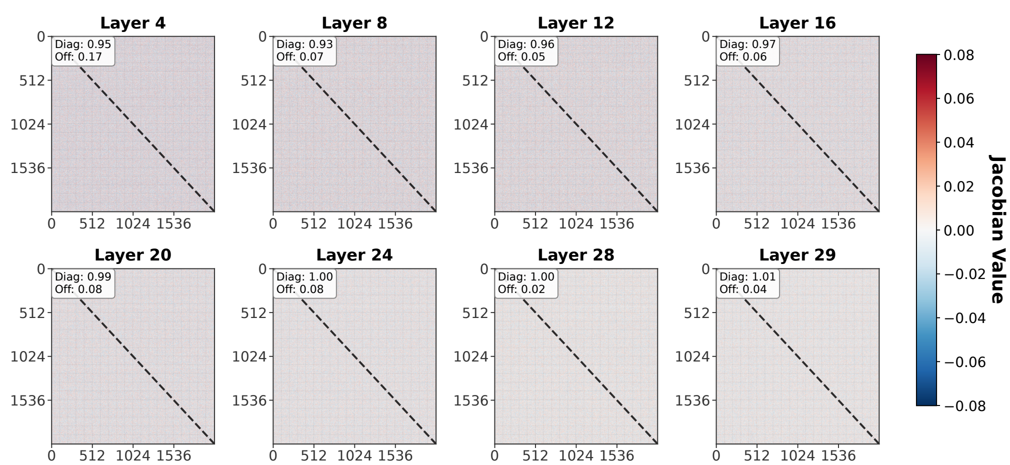
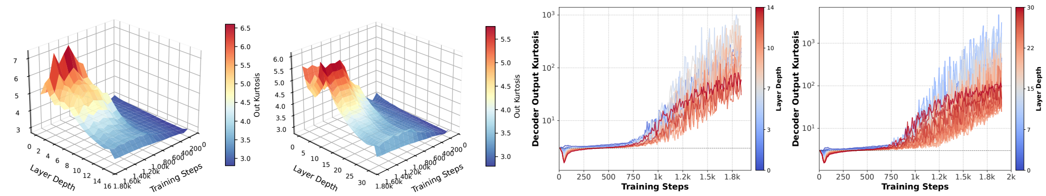


Figure 11: Jacobian matrices at different layers for $L = 32$. Results are transformed with a power of 0.3 for better visualization.

In Section 2, we prove that the Jacobian of the residual block approaches identity mapping with increased depth due to variance explosion. In Figure 3, we plot the Jacobian matrix at layer 30 for

$L = 32$. Here, we plot Jacobian matrices at different layers for $L = 32$ in Figure 11, where we can see the off-diagonal components gradually decrease as depth increases.

C.4 Kurtosis Analysis



(a) $L = 16$ (b) ($L = 32$) (c) $L = 16$ (d) $L = 32$

Figure 12: **(a)-(b):** per-dimension kurtosis. **(c)-(d)**: layer-level kurtosis.

To investigate whether variance accumulation is driven by outlier features, we analyze kurtosis (fourth standardized moment) of layer outputs.

Per-Dimension Kurtosis For each hidden dimension j at layer ℓ, we compute:

$$\text{Kurtosis}_{\ell,j} = \frac{\frac{1}{n}\sum_{i=1}^{n}(h_{\ell,i,j} - \bar{h}_{\ell,j})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(h_{\ell,i,j} - \bar{h}_{\ell,j})^2\right)^2} \tag{19}$$

where $\bar{h}_{\ell,j} = \frac{1}{n}\sum_{i=1}^{n} h_{\ell,i,j}$ is the mean across tokens for dimension j.

Layer-Level Aggregation We aggregate across dimensions to obtain layer-level kurtosis:

$$\text{Kurtosis}_{\ell} = \frac{1}{d}\sum_{j=1}^{d} \text{Kurtosis}_{\ell,j} \tag{20}$$

We compute both per-dimension kurtosis and layer-level kurtosis for all layers in the configurations from Section C.2 during training. The results are presented in Figure 12. From both the layer-level and per-dimension kurtosis across different depths, we can see that early layers show more outlier features (higher kurtosis) compared to deeper layers, which demonstrates that variance explosion is not driven by outlier features. We hypothesize that since prior work has shown that outlier features are crucial for model performance (Lin et al., 2024), the lower outlier prevalence in deeper layers indicates that they learn less effectively compared to shallower layers.

C.5 Attention Sparsity Across Different Evaluation Lengths

In Figure 5 and Figure 6, we evaluate attention sparsity at a fixed evaluation length of 8192 tokens across all models trained with different sequence lengths. To further investigate how sparsity varies when evaluation length matches training length, we provide additional results at evaluation lengths of 512, 1024, 2048, and 4096 tokens (Figures 13 to 16). Across all evaluation lengths, sparsity increases consistently with training sequence length, validating that longer training sequences induce stronger implicit attention sparsity regardless of evaluation context.

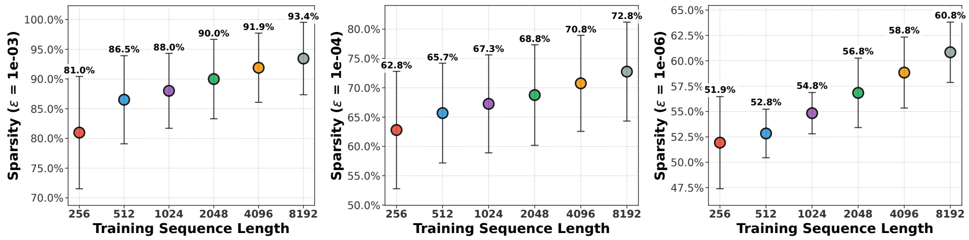
C.6 Attention Entropy

Complementing threshold-based sparsity (Equation (9)), we measure attention concentration using entropy. For attention weights $\mathbf{A}_{\ell,h} \in \mathbb{R}^{T\times T}$ at layer ℓ and head h, per-query entropy is:

$$\text{Entropy}_{\ell,h}(i) = -\sum_{j=1}^{T} A_{i,j} \log A_{i,j}. \tag{21}$$

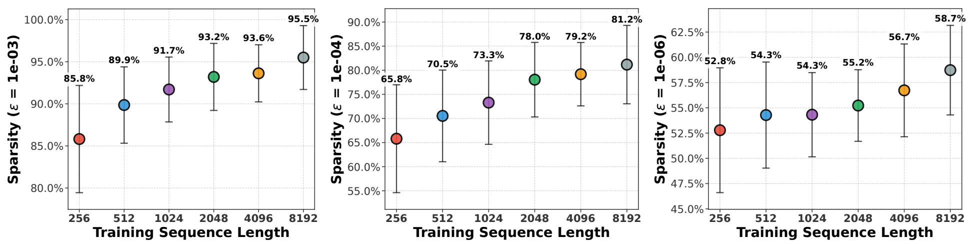
Head-level entropy averages across queries:

$$\text{Entropy}_{\ell,h} = \frac{1}{T}\sum_{i=1}^{T} \text{Entropy}_{\ell,h}(i). \tag{22}$$



(a) Attn. Sparsity (ϵ=1×10^{-3}) (b) Attn. Sparsity (ϵ=1×10^{-4}) (c) Attn. Sparsity (ϵ=1×10^{-6})

Figure 13: Attention sparsity with evaluation length of 512.



(a) Attn. Sparsity (ϵ=1×10^{-3}) (b) Attn. Sparsity (ϵ=1×10^{-4}) (c) Attn. Sparsity (ϵ=1×10^{-6})

Figure 14: Attention sparsity with evaluation length of 1024.

Global entropy averages across all layers (L) and heads (H):

$$\text{Entropy}_{\text{global}} = \frac{1}{L \cdot H} \sum_{\ell=1}^{L} \sum_{h=1}^{H} \text{Entropy}_{\ell,h} \tag{23}$$

Lower entropy indicates concentrated attention (sparse).

Attention entropy (Figure 17) decreases with training length across all evaluation length, mirroring the sparsity results and confirming that longer training induces stronger implicit sparsity.

C.7 Extended Variance Trajectories

Figures 5a and 7a show last-layer variance trajectories during training for models with different sequence lengths and group sizes. Since we control for total training FLOPs, models with shorter sequences require more training steps to match the compute of longer-sequence models, and models with larger group sizes (fewer KV heads) similarly require more steps to match FLOPs. For visualization clarity, we align trajectories to the run with fewest training steps in Figure 5a; complete unaligned trajectories are provided in Figure 18.

D Proofs and Mathematical Background

D.1 Proof of Theorem 1

Proof. For $\ell \in \{0, \dots, L-1\}$ and $r_{\ell+1} = r_\ell + u_\ell$ with $u_\ell = W_\ell(D_\ell r_\ell)$. Assume $\mathbb{E}[r_\ell] = 0$, because W_ℓ is independent of (r_ℓ, D_ℓ), we have:

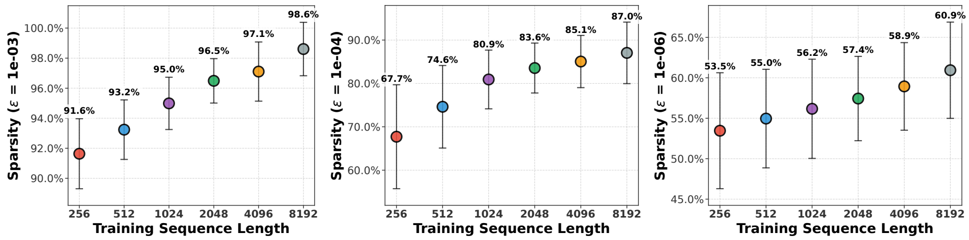
$$\mathrm{Var}(r_{\ell+1}) = \mathrm{Var}(r_\ell) + \mathrm{Var}(u_\ell) + 2\mathbb{E}\langle r_\ell, u_\ell \rangle. \tag{24}$$

By definition $u_\ell = W_\ell(D_\ell r_\ell)$. Condition on (r_ℓ, D_ℓ) and apply Equation (5) with $x = D_\ell r_\ell$:

$$\mathbb{E}\left[\|u_\ell\|_2^2 \mid r_\ell, D_\ell\right] = \mathbb{E}\left[\|W_\ell(D_\ell r_\ell)\|_2^2 \mid r_\ell, D_\ell\right] \le \alpha_\ell \|D_\ell r_\ell\|_2^2. \tag{25}$$

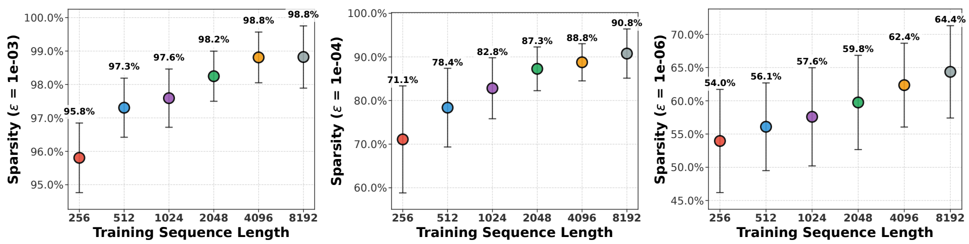
Take expectation over (r_ℓ, D_ℓ):

$$\mathrm{Var}(u_\ell) = \mathbb{E}\|u_\ell\|_2^2 \le \alpha_\ell \mathbb{E}\|D_\ell r_\ell\|_2^2. \tag{26}$$



(a) Attn. Sparsity (ϵ=1×10^{-3}) (b) Attn. Sparsity (ϵ=1×10^{-4}) (c) Attn. Sparsity (ϵ=1×10^{-6})

Figure 15: Attention sparsity with evaluation length of 2048.



(a) Attn. Sparsity (ϵ=1×10^{-3}) (b) Attn. Sparsity (ϵ=1×10^{-4}) (c) Attn. Sparsity (ϵ=1×10^{-6})

Figure 16: Attention sparsity with evaluation length of 4096.

Next, condition on r_ℓ and apply Equation (5) to $x = r_\ell$:

$$\mathbb{E}\left[\|D_\ell r_\ell\|_2^2 \mid r_\ell\right] \leq \rho_\ell \|r_\ell\|_2^2. \tag{27}$$

Take expectation over r_ℓ and combine with Equation (26):

$$\mathrm{Var}(u_\ell) \leq \alpha_\ell \rho_\ell \mathbb{E}\|r_\ell\|_2^2 = \alpha_\ell \rho_\ell \mathrm{Var}(r_\ell). \tag{28}$$

By Cauchy-Schwarz inequality,

$$\begin{aligned} \mathbb{E}\langle r_\ell, u_\ell \rangle &\leq \mathbb{E}\big[\|r_\ell\|_2 \|u_\ell\|_2\big] \leq \sqrt{\mathbb{E}\|r_\ell\|_2^2}\sqrt{\mathbb{E}\|u_\ell\|_2^2} \\ &= \sqrt{\mathrm{Var}(r_\ell)}\sqrt{\mathrm{Var}(u_\ell)}. \end{aligned} \tag{29}$$

Using Equation (28), hence

$$\mathbb{E}\langle r_\ell, u_\ell \rangle \leq \sqrt{\alpha_\ell \rho_\ell}\mathrm{Var}(r_\ell). \tag{30}$$

Plug Equation (28) and Equation (30) into Equation (24):

$$\begin{aligned} \mathrm{Var}(r_{\ell+1}) &\leq \mathrm{Var}(r_\ell) + \alpha_\ell \rho_\ell \mathrm{Var}(r_\ell) + 2\sqrt{\alpha_\ell \rho_\ell}\mathrm{Var}(r_\ell) \\ &= \Big(1 + \sqrt{\alpha_\ell \rho_\ell}\Big)^2 \mathrm{Var}(r_\ell). \end{aligned} \tag{31}$$

Iterate Equation (31) for $\ell = 0, \ldots, L-1$:

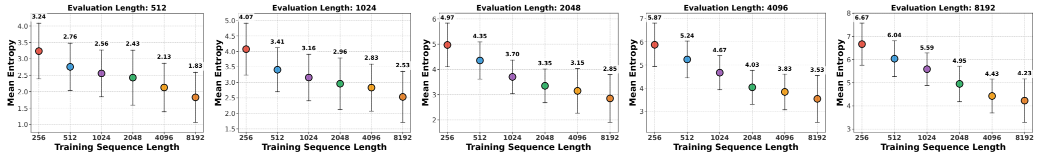
$$\mathrm{Var}(r_L) \leq \mathrm{Var}(r_0) \prod_{\ell=0}^{L-1} \Big(1 + \sqrt{\alpha_\ell \rho_\ell}\Big)^2 \tag{32}$$

In particular, if $\alpha_\ell \leq \alpha$ and $\rho_\ell \leq \rho < 1$ for all ℓ, then

$$\begin{aligned} \mathrm{Var}(r_L) &\leq \mathrm{Var}(r_0)\Big(1 + \sqrt{\alpha\rho}\Big)^{2L} \\ &= O\Big(1 + \sqrt{\alpha\rho}\Big)^{2L} \end{aligned} \tag{33}$$

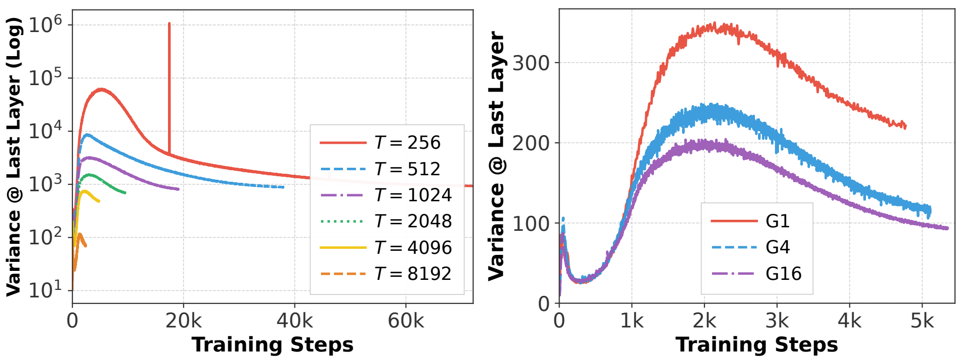
so smaller ρ (more sparsity) gives a smaller per-layer variance gain. which is Equation (6). The uniform bound with (α, ρ) follows directly. □

Theorem 1 depends on sparsity only through ρ_ℓ, the (second-moment) energy retention of the mask D_ℓ. Any mechanism that makes the effective mask sparser (smaller ρ_ℓ), even if it is not an explicit architectural constraint, yields a smaller per-layer gain $\big(1 + \sqrt{\alpha_\ell \rho_\ell}\big)^2$ and thus a smaller global variance growth across depth.



(a) Attn. Entropy (Eval. Length = 512) (b) Attn. Entropy (Eval. Length = 1024) (c) Attn. Entropy (Eval. Length = 2048) (d) Attn. Entropy (Eval. Length = 4096) (e) Attn. Entropy (Eval. Length = 8192)

Figure 17: Attention entropy with different evaluation length.



(a) Extended visualization of Figure 5a (b) Extended visualization of Figure 7a

Figure 18: Variance dynamics throughout training for models with different sequence lengths and group sizes. All models trained with equal FLOPs within its comparison group, leading to different training durations.

D.2 Attention Based Sparsity as variance Regularizer

Let $R_\ell \in \mathbb{R}^{n\times d}$ be the token matrix at 2 ℓ. Fix a *single* feature mask $D_\ell \in \mathbb{R}^{d\times d}$ (diagonal, 0–1), which is used by both the attention and FFN sublayers at depth ℓ. Define the two residual sublayers (no LayerNorm) in the standard sequential form

$$Z_\ell = R_\ell + \mathrm{MHA}_\ell(R_\ell D_\ell), \qquad R_{\ell+1} = Z_\ell + \mathrm{FFN}_\ell(Z_\ell D_\ell), \tag{34}$$

for $\ell = 0, \dots, L-1$.

Let the number of heads be H, head width be d_h with $Hd_h = d$. For $X \in \mathbb{R}^{n\times d}$, define for each head $h \in \{1, \dots, H\}$:

$$Q_{\ell,h}(X) = XW^Q_{\ell,h}, \quad K_{\ell,h}(X) = XW^K_{\ell,h}, \quad V_{\ell,h}(X) = XW^V_{\ell,h}, \tag{35}$$

with $W^Q_{\ell,h}, W^K_{\ell,h}, W^V_{\ell,h} \in \mathbb{R}^{d\times d_h}$. Let

$$S_{\ell,h}(X) = \frac{Q_{\ell,h}(X)K_{\ell,h}(X)^\top}{\sqrt{d_h}} \in \mathbb{R}^{n\times n}, \qquad P_{\ell,h}(X) = \mathrm{Softmax}\big(S_{\ell,h}(X)\big) \in \mathbb{R}^{n\times n}, \tag{36}$$

where $\mathrm{Softmax}$ applies softmax to each row. The head output is

$$H_{\ell,h}(X) = P_{\ell,h}(X)V_{\ell,h}(X) \in \mathbb{R}^{n\times d_h}. \tag{37}$$

Concatenate heads along features:

$$H_\ell(X) = \big[H_{\ell,1}(X); \dots; H_{\ell,H}(X)\big] \in \mathbb{R}^{n\times(Hd_h)} = \mathbb{R}^{n\times d}, \tag{38}$$

and apply the output projection $W^O_\ell \in \mathbb{R}^{d\times d}$:

$$\mathrm{MHA}_\ell(X) = H_\ell(X)W^O_\ell \in \mathbb{R}^{n\times d}. \tag{39}$$

Theorem 2 establishes that, for a residual recursion, variance propagation across depth is controlled by a per-layer gain factor that depends on sparsity only through a second-moment **energy retention** parameter. We now specialize this principle to a Transformer residual block composed of *multi-head self-attention* (MHA) and *feed-forward network* (FFN), *without LayerNorm*. Unlike the linear ResNet recursion, MSA includes the nonlinear $\mathrm{Softmax}$ coupling across tokens. Our goal is to isolate a sufficient condition under which the same conclusion holds: using a common feature mask D_ℓ for both MSA and FFN yields a variance bound whose dependence on sparsity appears only via the mask retention factor ρ_ℓ.

Assumption 1 (Common sparsity mask with bounded energy retention)**.** *For each layer ℓ, there exists a diagonal* 0–1 *mask $D_\ell \in \mathbb{R}^{d\times d}$ shared by the MSA and FFN sublayers, and a constant $\rho_\ell \in [0,1]$ such that for all $X \in \mathbb{R}^{n\times d}$,*

$$\mathbb{E}\|XD_\ell\|_F^2 \le \rho_\ell \|X\|_F^2. \tag{40}$$

Assumption 2 (MSA second-moment gain via explicit QKV/O control)**.** *For each layer ℓ and head $h \in \{1,\dots,H\}$, let*

$$Q_{\ell,h}(X) = XW^Q_{\ell,h}, \quad K_{\ell,h}(X) = XW^K_{\ell,h}, \quad V_{\ell,h}(X) = XW^V_{\ell,h}, \qquad P_{\ell,h}(X) = \mathrm{Softmax}\left(\frac{Q_{\ell,h}(X)K_{\ell,h}(X)^\top}{\sqrt{d_h}}\right), \tag{41}$$

and define $\mathrm{MHA}_\ell(X) = \big[P_{\ell,1}(X)V_{\ell,1}(X);\dots;P_{\ell,H}(X)V_{\ell,H}(X)\big]W^O_\ell$. *Assume there exist non-negative constants $\kappa_{\ell,h}$, $\nu_{\ell,h}$, and ω_ℓ such that for all $X \in \mathbb{R}^{n\times d}$,*

$$\|P_{\ell,h}(X)\|_2 \le \kappa_{\ell,h}, \qquad \|W^V_{\ell,h}\|_2 \le \nu_{\ell,h}, \qquad \|W^O_\ell\|_2 \le \omega_\ell. \tag{42}$$

Define

$$\alpha^{\mathrm{attn}}_\ell := \omega_\ell^2 \sum_{h=1}^{H} \kappa_{\ell,h}^2 \nu_{\ell,h}^2. \tag{43}$$

Assumption 3 (FFN second-moment gain)**.** *For each layer ℓ,* $\mathrm{FFN}_\ell : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ *satisfies: there exists $\alpha^{\mathrm{ffn}}_\ell \ge 0$ such that for all $X \in \mathbb{R}^{n\times d}$,*

$$\|\mathrm{FFN}_\ell(X)\|_F^2 \le \alpha^{\mathrm{ffn}}_\ell \|X\|_F^2. \tag{44}$$

For example, if $\mathrm{FFN}_\ell(X) = \sigma(XW_{\ell,1})W_{\ell,2}$ *with σ being L_σ-Lipschitz and $\|W_{\ell,1}\|_2 \le s_{\ell,1}$, $\|W_{\ell,2}\|_2 \le s_{\ell,2}$, then one may take $\alpha^{\mathrm{ffn}}_\ell = (L_\sigma s_{\ell,1}s_{\ell,2})^2$.*

Theorem 2 (Sparsity reduces variance propagation in Transformer)**.** *Let $\{R_\ell\}_{\ell=0}^L$ follow the no-LayerNorm Transformer recursion Equation* (34)*. Under Assumptions 1–3, the depth-L variance satisfies*

$$\mathrm{Var}(R_L) \le \mathrm{Var}(R_0) \prod_{\ell=0}^{L-1} \Big(1+\sqrt{\alpha^{\mathrm{attn}}_\ell \rho_\ell}\Big)^2 \Big(1+\sqrt{\alpha^{\mathrm{ffn}}_\ell \rho_\ell}\Big)^2. \tag{45}$$

In particular, if $\alpha^{\mathrm{attn}}_\ell \le \alpha^{\mathrm{attn}}$, $\alpha^{\mathrm{ffn}}_\ell \le \alpha^{\mathrm{ffn}}$ and $\rho_\ell \le \rho < 1$ for all ℓ, then

$$\begin{aligned}\mathrm{Var}(R_L) &\le \mathrm{Var}(R_0)\Big(1+\sqrt{\alpha^{\mathrm{attn}}\rho}\Big)^{2L}\Big(1+\sqrt{\alpha^{\mathrm{ffn}}\rho}\Big)^{2L} \\ &= O\Big(\Big(1+\sqrt{\alpha^{\mathrm{ffn}}\rho}\Big)^{2L}\Big(1+\sqrt{\alpha^{\mathrm{attn}}\rho}\Big)^{2L}\Big).\end{aligned} \tag{46}$$

The proof is provided in Section D.3. Theorem 2 shows that, even with the nonlinear $\mathrm{Softmax}$ in MHA, the depth-wise variance propagation can be controlled by a multiplicative gain whose dependence on sparsity appears only through the common retention factor ρ_ℓ. In this sense, feature sparsity acts as a variance regularizer: smaller ρ_ℓ reduces both the attention-branch gain $(1+\sqrt{\alpha^{\mathrm{attn}}_\ell\rho_\ell})^2$ and the FFN-branch gain $(1+\sqrt{\alpha^{\mathrm{ffn}}_\ell\rho_\ell})^2$, hence decreases the overall variance growth across depth. When effective sparsity is induced by training dynamics (rather than an explicit architectural mask), the same mechanism applies through a reduced empirical retention ρ_ℓ.

D.3 Proof of Theorem 2

Proof. Let $U^{\mathrm{attn}}_\ell := \mathrm{MHA}_\ell(R_\ell D_\ell)$, so $Z_\ell = R_\ell + U^{\mathrm{attn}}_\ell$. Expand

$$\mathrm{Var}(Z_\ell) = \mathbb{E}\|Z_\ell\|_F^2 = \mathbb{E}\|R_\ell\|_F^2 + \mathbb{E}\|U^{\mathrm{attn}}_\ell\|_F^2 + 2\mathbb{E}\langle R_\ell, U^{\mathrm{attn}}_\ell\rangle_F, \tag{47}$$

where $\langle A, B\rangle_F = \mathrm{trace}(A^\top B)$.

We first bound $\mathbb{E}\|U_\ell^{\mathrm{attn}}\|_F^2$ using the explicit QKV/O structure. Fix any $X \in \mathbb{R}^{n\times d}$ and consider $\mathrm{MHA}_\ell(X)$. By Equation (39) and the spectral norm bound on W_ℓ^O,

$$\begin{aligned}\|\mathrm{MHA}_\ell(X)\|_F &= \|H_\ell(X)W_\ell^O\|_F \le \|W_\ell^O\|_2\|H_\ell(X)\|_F \\ &\le \omega_\ell\|H_\ell(X)\|_F.\end{aligned} \tag{48}$$

By Equation (38), $\|H_\ell(X)\|_F^2 = \sum_{h=1}^H \|H_{\ell,h}(X)\|_F^2$. For each head, using Equation (37) and $\|AB\|_F \le \|A\|_2\|B\|_F$,

$$\begin{aligned}\|H_{\ell,h}(X)\|_F &= \|P_{\ell,h}(X)V_{\ell,h}(X)\|_F \le \|P_{\ell,h}(X)\|_2\|V_{\ell,h}(X)\|_F \\ &\le \kappa_{\ell,h}\|V_{\ell,h}(X)\|_F.\end{aligned} \tag{49}$$

Also, $V_{\ell,h}(X) = XW_{\ell,h}^V$ and $\|XW\|_F \le \|W\|_2\|X\|_F$, so

$$\|V_{\ell,h}(X)\|_F \le \|W_{\ell,h}^V\|_2\|X\|_F \le \nu_{\ell,h}\|X\|_F. \tag{50}$$

Combining Equation (48)–Equation (50) yields

$$\begin{aligned}\|\mathrm{MHA}_\ell(X)\|_F^2 &\le \omega_\ell^2 \sum_{h=1}^H \kappa_{\ell,h}^2\nu_{\ell,h}^2\|X\|_F^2 \\ &= \alpha_\ell^{\mathrm{attn}}\|X\|_F^2,\end{aligned} \tag{51}$$

with $\alpha_\ell^{\mathrm{attn}}$ from Equation (43). Applying Equation (51) to $X = R_\ell D_\ell$ and taking expectation gives

$$\mathbb{E}\|U_\ell^{\mathrm{attn}}\|_F^2 = \mathbb{E}\|\mathrm{MHA}_\ell(R_\ell D_\ell)\|_F^2 \le \alpha_\ell^{\mathrm{attn}}\mathbb{E}\|R_\ell D_\ell\|_F^2. \tag{52}$$

Using the Assumption 1 with $X = R_\ell$ yields

$$\mathbb{E}\|R_\ell D_\ell\|_F^2 \le \rho_\ell\mathbb{E}\|R_\ell\|_F^2 = \rho_\ell\mathrm{Var}(R_\ell), \tag{53}$$

so

$$\mathbb{E}\|U_\ell^{\mathrm{attn}}\|_F^2 \le \alpha_\ell^{\mathrm{attn}}\rho_\ell\mathrm{Var}(R_\ell). \tag{54}$$

Using Cauchy–Schwarz gives

$$\begin{aligned}\mathbb{E}\langle R_\ell, U_\ell^{\mathrm{attn}}\rangle_F &\le \sqrt{\mathbb{E}\|R_\ell\|_F^2}\sqrt{\mathbb{E}\|U_\ell^{\mathrm{attn}}\|_F^2} \\ &\le \sqrt{\alpha_\ell^{\mathrm{attn}}\rho_\ell}\mathrm{Var}(R_\ell).\end{aligned} \tag{55}$$

Substitute Equation (54) and Equation (55) into Equation (47) to obtain

$$\mathrm{Var}(Z_\ell) \le \Big(1 + \sqrt{\alpha_\ell^{\mathrm{attn}}\rho_\ell}\Big)^2\mathrm{Var}(R_\ell). \tag{56}$$

Following Theorem 1, let $U_\ell^{\mathrm{ffn}} := \mathrm{FFN}_\ell(Z_\ell D_\ell)$. Repeating the same variance expansion and Cauchy–Schwarz argument, and using Assumption 3 plus Assumption 1, we get

$$\mathrm{Var}(R_{\ell+1}) \le \Big(1 + \sqrt{\alpha_\ell^{\mathrm{ffn}}\rho_\ell}\Big)^2\mathrm{Var}(Z_\ell). \tag{57}$$

Combine Equation (56) and Equation (57):

$$\begin{aligned}\mathrm{Var}(R_{\ell+1}) &\le \Big(1 + \sqrt{\alpha_\ell^{\mathrm{attn}}\rho_\ell}\Big)^2\Big(1 + \sqrt{\alpha_\ell^{\mathrm{ffn}}\rho_\ell}\Big)^2\mathrm{Var}(R_\ell) \\ &= O\Big(\Big(1 + \sqrt{\alpha^{\mathrm{ffn}}\rho}\Big)^{2L}\Big(1 + \sqrt{\alpha^{\mathrm{attn}}\rho}\Big)^{2L}\Big).\end{aligned} \tag{58}$$

Iterating Equation (58) for $\ell = 0, \ldots, L-1$ yields Equation (45). The uniform bound Equation (46) follows by $\alpha_\ell^{\mathrm{attn}} \le \alpha^{\mathrm{attn}}$, $\alpha_\ell^{\mathrm{ffn}} \le \alpha^{\mathrm{ffn}}$, and $\rho_\ell \le \rho$. □

D.4 Weight Decay Variance Control: Theory and Proof

We provide the formal theoretical analysis showing that weight decay contracts parameter variance and reduces layer output variance.

Theorem 3 (Weight decay contracts parameter variance and reduces layer-output variance)**.** *Let $W_t \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ follow the decoupled weight-decay update $W_{t+1} = (1 - \eta\lambda)W_t - \eta G_t$, Assume $0 < \eta\lambda < 2$, and that the gradient noise is independent of the weight. If $\mathrm{Var}(G_t) \leq \sigma_G^2$ for all t, then for a linear layer output u_t , we have*

$$\begin{aligned} \mathrm{Var}(u_t) &\leq \|\Sigma_x\|_2 \Big(\mathrm{Var}(W_t) + \|\mathbb{E}[W_t]\|_F^2 \Big), \\ &= O\left((1 - \eta\lambda)^{2t} + \frac{\eta\sigma_G^2}{\lambda} \right). \end{aligned} \tag{59}$$

where $\Sigma_x = \mathbb{E}[xx^\top]$ is the covariance matrix of x, and x independent of W_t

Under $0 < \eta\lambda \leq 1$, the bound in Equation (59) is decreasing in λ term-by-term: $(1 - \eta\lambda)^{2t}$ decreases monotonically with λ and captures contraction of the initialization contribution, while $\eta\sigma_G^2/\lambda$ decreases with λ and captures the steady-state variance induced by gradient noise. Hence, larger weight decay λ yields a smaller upper bound on $\mathrm{Var}(u_t)$. Since weight decay is applied during training and continuously shrinks parameter magnitude, this variance reduction can be viewed as an implicit regularization effect induced by optimization.

Proof. Let $\bar{W}_t := \mathbb{E}[W_t]$ and $\widetilde{W}_t := W_t - \bar{W}_t$, and similarly $\bar{G}_t := \mathbb{E}[G_t]$, $\widetilde{G}_t := G_t - \bar{G}_t$. Taking expectation of $W_{t+1} = (1 - \eta\lambda)W_t - \eta G_t$ gives

$$\bar{W}_{t+1} = (1 - \eta\lambda)\bar{W}_t - \eta\bar{G}_t. \tag{60}$$

Subtract Equation (60) from $W_{t+1} = (1 - \eta\lambda)W_t - \eta G_t$ to obtain the centered update

$$\widetilde{W}_{t+1} = (1 - \eta\lambda)\widetilde{W}_t - \eta\widetilde{G}_t. \tag{61}$$

By definition,

$$\mathrm{Var}(W_{t+1}) = \mathbb{E}\left[\|\widetilde{W}_{t+1}\|_F^2\right] = \mathbb{E}\left[\|(1 - \eta\lambda)\widetilde{W}_t - \eta\widetilde{G}_t\|_F^2\right]. \tag{62}$$

Thus we have:

$$\|(1 - \eta\lambda)\widetilde{W}_t - \eta\widetilde{G}_t\|_F^2 = (1 - \eta\lambda)^2\|\widetilde{W}_t\|_F^2 + \eta^2\|\widetilde{G}_t\|_F^2 - 2\eta(1 - \eta\lambda)\langle\widetilde{W}_t, \widetilde{G}_t\rangle_F. \tag{63}$$

Taking expectation and by the assumption that W_t and G_t are independent:

$$\begin{aligned} \mathrm{Var}(W_{t+1}) &= (1 - \eta\lambda)^2\mathbb{E}\left[\|\widetilde{W}_t\|_F^2\right] + \eta^2\mathbb{E}\left[\|\widetilde{G}_t\|_F^2\right] \\ &= (1 - \eta\lambda)^2\mathrm{Var}(W_t) + \eta^2\mathrm{Var}(G_t), \end{aligned} \tag{64}$$

Let $\rho := (1 - \eta\lambda)^2$. Under $0 < \eta\lambda < 2$, we have $0 \leq \rho < 1$. If $\mathrm{Var}(G_t) \leq \sigma_G^2$, then we have:

$$\mathrm{Var}(W_{t+1}) \leq \rho\mathrm{Var}(W_t) + \eta^2\sigma_G^2. \tag{65}$$

Iterating the inequality gives

$$\begin{aligned} \mathrm{Var}(W_t) &\leq \rho^t\mathrm{Var}(W_0) + \eta^2\sigma_G^2\sum_{s=0}^{t-1}\rho^s \\ &= \rho^t\mathrm{Var}(W_0) + \eta^2\sigma_G^2\frac{1 - \rho^t}{1 - \rho} \\ &\leq \rho^t\mathrm{Var}(W_0) + \frac{\eta^2\sigma_G^2}{1 - \rho}, \end{aligned} \tag{66}$$

Let $u_t = W_t x$. Since $\mathbb{E}[x] = 0$ and x is independent of W_t,

$$\begin{aligned} \mathbb{E}[u_t] &= \mathbb{E}[W_t]\mathbb{E}[x] = 0, \\ \mathrm{Var}(u_t) &= \mathbb{E}\left[\|u_t\|_2^2\right] = \mathbb{E}\left[x^\top W_t^\top W_t x\right]. \end{aligned} \tag{67}$$

Condition on W_t and use $\mathbb{E}[xx^\top] = \Sigma_x$:

$$\begin{aligned}\mathbb{E}\left[\|u_t\|_2^2 \mid W_t\right] &= \operatorname{tr}\left(W_t^\top W_t \Sigma_x\right) \\ &\leq \|\Sigma_x\|_2 \operatorname{tr}(W_t^\top W_t) \\ &= \|\Sigma_x\|_2 \|W_t\|_F^2.\end{aligned} \tag{68}$$

Decomposing $\|W_t\|_F^2$ into mean and variance terms,

$$\mathbb{E}\left[\|W_t\|_F^2\right] = \mathbb{E}\left[\|\widetilde{W}_t + \bar{W}_t\|_F^2\right] = \mathbb{E}\left[\|\widetilde{W}_t\|_F^2\right] + \|\bar{W}_t\|_F^2 = \operatorname{Var}(W_t) + \|\mathbb{E}[W_t]\|_F^2, \tag{69}$$

where the cross term vanishes since $\mathbb{E}[\widetilde{W}_t] = 0$. Therefore,

$$\operatorname{Var}(u_t) \leq \|\Sigma_x\|_2 \Big(\operatorname{Var}(W_t) + \|\mathbb{E}[W_t]\|_F^2\Big), \tag{70}$$

By $\mathbb{E}[G_t] = 0$, the mean recursion Equation (60) becomes

$$\mathbb{E}[W_{t+1}] = (1 - \eta\lambda)\mathbb{E}[W_t], \tag{71}$$

hence

$$\|\mathbb{E}[W_t]\|_F^2 = (1 - \eta\lambda)^{2t} \|\mathbb{E}[W_0]\|_F^2. \tag{72}$$

From the assumption in Theorem 3, $\operatorname{Var}(G_t) \leq \sigma_G^2$,

$$\operatorname{Var}(W_t) \leq (1 - \eta\lambda)^{2t} \operatorname{Var}(W_0) + \frac{\eta^2 \sigma_G^2}{1 - (1 - \eta\lambda)^2}. \tag{73}$$

Substituting Equation (73) and Equation (72) into Equation (70), we obtain

$$\begin{aligned}\operatorname{Var}(u_t) &\leq \|\Sigma_x\|_2 \left((1 - \eta\lambda)^{2t} \Big(\operatorname{Var}(W_0) + \|\mathbb{E}[W_0]\|_F^2\Big) + \frac{\eta^2 \sigma_G^2}{1 - (1 - \eta\lambda)^2}\right) \\ &= \|\Sigma_x\|_2 \left((1 - \eta\lambda)^{2t} \Big(\operatorname{Var}(W_0) + \|\mathbb{E}[W_0]\|_F^2\Big) + \frac{\eta \sigma_G^2}{\lambda(2 - \eta\lambda)}\right) \\ &= O\left((1 - \eta\lambda)^{2t} + \frac{\eta \sigma_G^2}{\lambda}\right),\end{aligned} \tag{74}$$

where the last step uses $0 < \eta\lambda \leq 1$ so that $2 - \eta\lambda = \Theta(1)$.

The bound is the sum of two terms with different decay behaviors in λ. First, $(1 - \eta\lambda)^{2t}$ decreases monotonically in λ and captures exponential contraction of the initialization contribution under weight decay. Second, $\frac{\eta\sigma_G^2}{\lambda}$ also decreases in λ and corresponds to the steady-state variance induced by gradient noise. Therefore, increasing λ tightens both parts of the upper bound, implying smaller $\operatorname{Var}(u_t)$ under the stated assumptions.

□

D.5 Sequence Length Variance Control: Theory and Proof

Theorem 4 (Sequence length reduces attention-output variance)**.** *Assume the attention output is the uniform average $x = \frac{1}{T}\sum_{i=1}^T h_i$, where $\{h_i\}_{i=1}^T$ are scalar random variables (e.g., one fixed coordinate of the value vectors) that are zero-mean and mutually independent, with* $\operatorname{Var}(h_i) = \sigma^2$ *for all i. Then* $\operatorname{Var}(x) = \frac{\sigma^2}{T}$.

Proof. Let $x = \sum_{i=1}^T a_i h_i$ with $a_i \geq 0$ and $\sum_{i=1}^T a_i = 1$. Assume $\{h_i\}_{i=1}^T$ are independent and $\mathbb{E}[h_i] = 0$ for all i, with $\operatorname{Var}(h_i) = \sigma^2$. Then $\mathbb{E}[x] = 0$ and

$$\begin{aligned}\operatorname{Var}(x) = \mathbb{E}\big[x^2\big] &= \mathbb{E}\left[\left(\sum_{i=1}^T a_i h_i\right)^2\right] \\ &= \sum_{i=1}^T a_i^2 \mathbb{E}[h_i^2] + 2 \sum_{1 \leq i < j \leq T} a_i a_j \mathbb{E}[h_i h_j].\end{aligned} \tag{75}$$

By independence and $\mathbb{E}[h_i] = 0$, for $i \neq j$ we have

$$\mathbb{E}[h_i h_j] = \mathbb{E}[h_i]\mathbb{E}[h_j] = 0, \tag{76}$$

so all cross terms vanish. Also $\mathbb{E}[h_i^2] = \mathrm{Var}(h_i) = \sigma^2$. Hence,

$$\mathrm{Var}(x) = \sum_{i=1}^{T} a_i^2 \sigma^2 = \sigma^2 \sum_{i=1}^{T} a_i^2. \tag{77}$$

If $a_i = 1/T$, then $\sum_{i=1}^{T} a_i^2 = 1/T$, giving $\mathrm{Var}(x) = \sigma^2/T$. $\square$

D.6 Grouped Query Attention Variance Analysis: Theory and Proof

We formally characterize how attention variants differ in output variance through their averaging mechanisms.

Theorem 5 (Output variance scaling for Attention / GQA / MQA ($G = H$))**.** *Assume (i) attention weights are uniform over the attended keys, (ii) the value rows $\{V_j\}_{j=1}^{n}$ are zero-mean, mutually independent, and satisfy $\mathrm{Var}(V_j) = \sigma_V^2$ (all j), and (iii) variance is measured per output coordinate. Then:*

$$\mathrm{Var}\big(\mathrm{Attn}\big) = \frac{\sigma_V^2}{n}. \tag{78}$$

*For **grouped query attention (GQA)** with G groups of equal size $n_g = n/G$, where each query attends only within its group, we have*

$$\mathrm{Var}\big(\mathrm{Attn}_{\mathrm{GQA}}\big) = \frac{1}{G} \cdot \mathrm{Var}\big(\mathrm{Attn}\big). \tag{79}$$

*For **multi-query attention (MQA)**, under the same uniform-weight assumption, each head output has*

$$\mathrm{Var}\big(\mathrm{Attn}_{\mathrm{MQA}}\big) = \frac{\sigma_V^2}{Gn} = \frac{1}{G}\mathrm{Var}\big(\mathrm{Attn}\big). \tag{80}$$

Theorem 5 shows that variance reduction comes from averaging: under uniform attention, the output averages over n values, so $\mathrm{Var}(\mathrm{Attn}) = \sigma_V^2/n$. GQA and MQA averages over only $n_g = n/G$ values and thus reduces variance by a factor G.

Proof. **Variance of the Attention.** The scaled dot-product attention mechanism is defined as:

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V. \tag{81}$$

The softmax function outputs a probability distribution over the keys. Let the softmax output be $A = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)$, where A is a matrix with each row summing to 1. The final attention output is obtained by multiplying the softmax output A with the value matrix V:

$$\mathrm{Attn}(Q, K, V) = AV. \tag{82}$$

To simplify the analysis, we make the following additional assumptions: The softmax output A is approximately uniform, meaning each element of A is roughly $1/n$, where n is the number of keys/values. Given this assumption, the variance of the attention is:

$$\mathrm{Var}(\mathrm{Attn}(Q, K, V)) \sim \mathrm{Var}(AV) = \frac{1}{n} \sum_{i=1}^{n} \mathrm{Var}(V_i) d_{\mathrm{head}} = \frac{1}{n} \cdot n \cdot \sigma_V^2 d = \sigma_V^2 d = \sigma_W^2 d. \tag{83}$$

where W is the universal weight matrix defined as before.

Variance under Grouped Query Attention. Suppose the n queries are partitioned into G disjoint groups of equal size $n_g = n/G$, and each query attends only to its own group. Then the softmax output A is uniform for all V_i, and roughly uniform of weight $1/n_g$. Writing V_i for the i-th row of V, we have

$$\mathrm{Attn}(Q,K,V) = AVd = \begin{pmatrix} \frac{1}{n_g}(\sum_{i\in\mathcal{G}_1} A_i)V_1 \\ \frac{1}{n_g}(\sum_{i\in\mathcal{G}_2} A_i)V_2 \\ \vdots \\ \frac{1}{n_g}(\sum_{i\in\mathcal{G}_G} A_i)V_G \end{pmatrix} d, \tag{84}$$

where $\{\mathcal{G}_1,\ldots,\mathcal{G}_G\}$ are the index sets of each group. Under the same variance assumption $\mathrm{Var}(V_i) = \sigma_V^2$ and independence across rows, each group's output has variance

$$\mathrm{Var}\Big(\tfrac{1}{G}\Big(\tfrac{1}{n_g}\sum_{i\in\mathcal{G}_g} A_i\Big)V_g\Big) = \tfrac{1}{G^2 n_g^2}\sum_{i\in\mathcal{G}_g}\mathrm{Var}(V_g) = \frac{n_g\sigma_V^2}{G^2 n_g^2} = \tfrac{\sigma_V^2}{G^2 n_g}. \tag{85}$$

Since there are G independent group outputs, the overall variance (per row) remains

$$\mathrm{Var}\big(\mathrm{Attn}(Q,K,V)\big) = \frac{\sigma_V^2 d}{G^2 n_g} = \frac{\sigma_V^2 d}{Gn}. \tag{86}$$

In particular, compared to the fully uniform case ($G=1$), grouped query attention reduces the output variance by a factor of G.

Variance under Multi-Query Attention. In Multi-Query Attention (MQA), we use n distinct query projections but share a single pair of keys and values across all heads. Let

$$Q^{(n)} \in \mathbb{R}^{n\times d_k} \quad , \quad K \in \mathbb{R}^{n\times d_k} \quad , \quad V \in \mathbb{R}^{n\times d_v} \tag{87}$$

where $Q^{(n)}$ is the n-th query projection and (K,V) are shared. The output of head g is

$$\mathrm{Attn}^{(n)}(Q^{(n)},K,V) = \mathrm{softmax}\Big(\tfrac{Q^{(g)}K^\top}{\sqrt{d_k}}\Big)V. \tag{88}$$

Assume each row V_i of V has variance $\mathrm{Var}(V_i) = \sigma_V^2$ and that the softmax weights are approximately uniform, i.e. each weight is about $1/n$. Then for any query i and head g,

$$\mathrm{Attn}_i^{(g)} = \tfrac{1}{n}\sum_{j=1}^{n} V_j, \qquad \mathrm{Var}\big(\mathrm{Attn}_i^{(g)}\big) = \frac{1}{n^2}\sum_{j=1}^{n_g}\mathrm{Var}(V_j)d = \frac{\sigma_V^2 d}{Gn}. \tag{89}$$

□

D.7 Mixture of Experts Variance Analysis: Theory and Proof

We formally characterize how Top-k MoE routing reduces variance through selective averaging.

We write the gating-based MoE update in Top-k set notation. Let $S(x) := \text{Top-}k(\mathbf{W}_g x)$ be the selected experts, so

$$\mathrm{MoE}(x) = \sum_{i\in S(x)} g_i(x)\mathrm{FFN}_i(x), \qquad |S(x)| = k. \tag{90}$$

In the uniform-gating case $g_i(x) = 1/k$ (for $i \in S(x)$), this reduces to $\mathrm{MoE}(x) = \frac{1}{k}\sum_{i\in S(x)}\mathrm{FFN}_i(x)$, which is the form used in Theorem 6.

Theorem 6 (Top-k MoE reduces layer-output and Jacobian variance)**.** *Fix a Top-k MoE layer with uniform gating*

$$\mathrm{MoE}(x) = \frac{1}{k}\sum_{i\in S(x)}\mathrm{FFN}_i(x), \qquad |S(x)| = k, \tag{91}$$

and assume the routing set $S(x)$ is locally constant around x.

(Output variance). *If $\{\mathrm{FFN}_i(x)\}_{i\in S(x)}$ are mutually independent with common mean and variance, then*

$$\mathrm{Var}\left[\mathrm{MoE}(x)\right] = \frac{1}{k}\mathrm{Var}\left[\mathrm{FFN}_1(x)\right]. \tag{92}$$

(Jacobian variance). *Let* $J_i(x) := \frac{\partial \mathrm{FFN}_i(x)}{\partial x}$ *and* $J^{\mathrm{MoE}}(x) := \frac{\partial \mathrm{MoE}(x)}{\partial x} = \frac{1}{k}\sum_{i\in S(x)} J_i(x)$*. If* $\{J_i(x)\}_{i\in S(x)}$ *are mutually independent with common mean and variance, then*

$$\mathrm{Var}\left[J^{\mathrm{MoE}}(x)\right] = \frac{1}{k}\mathrm{Var}\left[J_1(x)\right]. \tag{93}$$

Theorem 6 shows that Top-k MoE reduces variance by an averaging effect: under uniform gating, the MoE output and Jacobian are averages over k selected experts, and under independence the variance decreases by a factor $1/k$.

Proof. First we want to prove the **output variance** in Equation 92. Let $u_i(x_\ell) := \mathrm{FFN}_i(x_\ell) \in \mathbb{R}^d$ denote the i-th expert output, and

$$u^{\mathrm{MoE}}(x_\ell) := \mathrm{MoE}(x_\ell) = \frac{1}{k}\sum_{i\in\text{Top-}k} u_i(x_\ell). \tag{94}$$

By the assumption that the selected expert outputs $\{u_i(x_\ell)\}_{i\in\text{Top-}k}$ are mutually independent, with common mean $\mathbb{E}[u_i(x_\ell)] = m_\ell$ and common second moment, then $\mathbb{E}[u^{\mathrm{MoE}}(x_\ell)] = m_\ell$, and

$$u^{\mathrm{MoE}}(x_\ell) - m_\ell = \frac{1}{k}\sum_{i\in\text{Top-}k}\big(u_i(x_\ell) - m_\ell\big). \tag{95}$$

Therefore,

$$\begin{aligned}\mathrm{Var}\left[u^{\mathrm{MoE}}(x_\ell)\right] &= \mathbb{E}\left[\left\|\frac{1}{k}\sum_{i\in\text{Top-}k}\big(u_i(x_\ell)-m_\ell\big)\right\|_2^2\right] \\ &= \frac{1}{k^2}\mathbb{E}\left[\left\|\sum_{i\in\text{Top-}k}\big(u_i(x_\ell)-m_\ell\big)\right\|_2^2\right].\end{aligned} \tag{96}$$

Expanding the Equation Equation (96) yields

$$\left\|\sum_{i\in\text{Top-}k}\big(u_i(x_\ell)-m_\ell\big)\right\|_2^2 = \sum_{i\in\text{Top-}k}\|u_i(x_\ell)-m_\ell\|_2^2 + 2\sum_{\substack{i,j\in\text{Top-}k\\ i<j}}\langle u_i(x_\ell)-m_\ell, u_j(x_\ell)-m_\ell\rangle. \tag{97}$$

Using independence with zero-mean centered terms gives

$$\mathbb{E}\left[\langle u_i(x_\ell)-m_\ell, u_j(x_\ell)-m_\ell\rangle\right] = \langle \mathbb{E}[u_i(x_\ell)-m_\ell], \mathbb{E}[u_j(x_\ell)-m_\ell]\rangle = 0, \qquad (i\neq j), \tag{98}$$

so all cross terms vanish. Hence

$$\begin{aligned}\mathrm{Var}\left[u^{\mathrm{MoE}}(x_\ell)\right] &= \frac{1}{k^2}\sum_{i\in\text{Top-}k}\mathbb{E}\left[\|u_i(x_\ell)-m_\ell\|_2^2\right] \\ &= \frac{1}{k^2}\cdot k\mathrm{Var}\left[u_1(x_\ell)\right] \\ &= \frac{1}{k}\mathrm{Var}\left[\mathrm{FFN}_1(x_\ell)\right].\end{aligned} \tag{99}$$

Next we want to prove the **Jacobian variance** in Equation 93. For the variance, in a standard feed-forward network (FFN), the Jacobian variance at layer ℓ is

$$\mathrm{Var}\left[J_\ell\right] = \mathrm{Var}\left(\frac{\partial \mathrm{FFN}(x_\ell)}{\partial x_\ell}\right). \tag{100}$$

For a mixture-of-experts (MoE) layer with Top-k routing, the output is

$$\mathrm{MoE}(x_\ell) = \frac{1}{k}\sum_{i\in\text{Top-}k}\mathrm{FFN}_i(x_\ell), \tag{101}$$

with Jacobian

$$J_\ell^{\text{MoE}} = \frac{\partial \text{MoE}(x_\ell)}{\partial x_\ell} = \frac{1}{k} \sum_{i \in \text{Top-}k} \frac{\partial \text{FFN}_i(x_\ell)}{\partial x_\ell} = \frac{1}{k} \sum_{i \in \text{Top-}k} J_{\ell,i}. \tag{102}$$

Assume the selected experts have mutually independent Jacobians and identical second-moment structure, we have

$$\text{Var}\left[J_\ell^{\text{MoE}}\right] = \text{Var}\left(\frac{1}{k} \sum_{i \in \text{Top-}k} J_{\ell,i}\right) = \frac{1}{k^2} \sum_{i \in \text{Top-}k} \text{Var}\left[J_{\ell,i}\right]. \tag{103}$$

If each expert has the same variance as a standard FFN, then

$$\text{Var}\left[J_\ell^{\text{MoE}}\right] = \frac{1}{k} \text{Var}\left[J_\ell^{\text{FFN}}\right]. \tag{104}$$

Thus, under these assumptions, the MoE layer introduces a variance reduction factor $1/k$ compared with a single FFN layer.

By Sun et al. (2025), we can estimate the upper bound of the gradient norm of $\frac{\partial y_\ell}{\partial x_1}$:

$$\left\| \frac{\partial y_\ell}{\partial x'_\ell} \right\|_2 \leq 1 + \left\| \frac{\partial \text{FFN}(\text{LN}(x'_\ell))}{\partial \text{LN}(x'_\ell)} \right\|_2 \left\| \frac{\partial \text{LN}(x'_\ell)}{\partial x'_\ell} \right\|_2. \tag{105}$$

For MoE systems Shazeer et al. (2017), based on Equation Equation (105), we have

$$\begin{aligned} y_\ell &= x'_\ell + \text{MoE}\left(\text{LN}(x'_\ell)\right) \\ &= x'_\ell + \sum_{i \in \text{Top-}k} g_i \text{FFN}_i\left(\text{LN}(x'_\ell)\right) \\ &= x'_\ell + \frac{1}{k} \sum_{i \in \text{Top-}k} \text{FFN}_i\left(\text{LN}(x'_\ell)\right), \qquad (\text{since } g_i = \tfrac{1}{k}). \end{aligned} \tag{106}$$

Assuming the routing set $\text{Top-}k$ is locally constant around x'_ℓ, the Jacobian satisfies

$$\begin{aligned} \left\| \frac{\partial y_\ell}{\partial x'_\ell} \right\|_2 &= \left\| I + \frac{1}{k} \sum_{i \in \text{Top-}k} \frac{\partial \text{FFN}_i(\text{LN}(x'_\ell))}{\partial x'_\ell} \right\|_2 \\ &\leq 1 + \frac{1}{k} \left\| \sum_{i \in \text{Top-}k} \frac{\partial \text{FFN}_i(\text{LN}(x'_\ell))}{\partial x'_\ell} \right\|_2. \end{aligned} \tag{107}$$

We know that,

$$\frac{\partial \text{FFN}_i(\text{LN}(x'_\ell))}{\partial x'_\ell} = \frac{\partial \text{FFN}_i(\text{LN}(x'_\ell))}{\partial \text{LN}(x'_\ell)} \frac{\partial \text{LN}(x'_\ell)}{\partial x'_\ell}, \tag{108}$$

hence

$$\left\| \frac{\partial y_\ell}{\partial x'_\ell} \right\|_2 \leq 1 + \frac{1}{k} \left\| \sum_{i \in \text{Top-}k} \frac{\partial \text{FFN}_i(\text{LN}(x'_\ell))}{\partial \text{LN}(x'_\ell)} \right\|_2 \left\| \frac{\partial \text{LN}(x'_\ell)}{\partial x'_\ell} \right\|_2. \tag{109}$$

After adding a scaling factor, we obtain

$$\left\| \frac{\partial y_\ell}{\partial x'_\ell} \right\|_2 \leq 1 + \frac{1}{k} \left\| \sum_{i \in \text{Top-}k} \frac{\partial \text{FFN}_i(\text{LN}(x'_\ell))}{\partial \text{LN}(x'_\ell)} \right\|_2 \left\| \frac{1}{\sqrt{\ell}} \right\|_2 \left\| \frac{\partial \text{LN}(x'_\ell)}{\partial x'_\ell} \right\|_2. \tag{110}$$

For layers with MoE-FFN, we apply this MoE-specific scaling factor; for layers other than MoE-FFN, we use the original scaling factor. $\square$

E EXPERIMENTAL SETTINGS

Across all our experiments, we implement a standard LLM transformer architecture with RoPE (Su et al., 2024), SwiGLU (Shazeer, 2020), and RMSNorm (Zhang & Sennrich, 2019). We utilize the FineWeb-Edu dataset[1] for training models across all our experiments and tokenize the dataset with the GPT-NeoX tokenizer[2]. Although different experiment groups may utilize different data sizes, all our evaluation perplexities are computed on the same held-out evaluation set from FineWeb-Edu. In our experiments, we adapt the training framework from OLMo[3] and use the AdamW optimizer with mixed precision training. All our experiments are run on Hopper-series GPUs.

E.1 DEPTH CONTROL EXPERIMENTS

Table 6: Model configuration for Section 2.

Depth L	# of Param.	Hidden Dim	MLP Hidden Dim	Query Head	Key-Value Head	Max Training Length
L=12	900M	2048	8192	16	16	1024
L=16	1.2B					
L=24	1.7B					
L=32	2.3B					

In Section 2, we vary the model depths to verify the phenomenon of CoD. Specific details about each model are in Table 6. In all runs, we keep the weight decay at 0.1 and warmup iterations at 1000 with a warmup cosine learning rate scheduler. We sweep the learning rate over $\{5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 8 \times 10^{-4}, 1 \times 10^{-3}\}$ and choose 1×10^{-3} for $L \in \{12, 16, 24\}$, and 8×-4 for $L = 32$. All runs are trained on 10B tokens with global batch size of 256, and we evaluate the downstream performance on MMLU, ARC-Easy, ARC-Challenge, and Hellaswag using the implementation from lm-eval-harness[4] with zero-shot settings, reporting accuracy for MMLU and normalized accuracy (acc_norm) for other tasks.

E.2 WEIGHT DECAY EXPERIMENTS

All of the experiments in Section 4.1.1 are trained with a model of $L = 16$ layers, $d = 2048$ hidden dimensions, MLP dimension of 8192, and $h = 16$ attention heads (with the same number of key-value heads). We vary the weight decay strength of the AdamW optimizer and use the same learning rate of 1×10^{-3}, 1000 warmup iterations, and a warmup cosine learning rate scheduler. All models are trained on 5B tokens with a global batch size of 128.

E.3 SEQUENCE LENGTH EXPERIMENTS

All of the experiments regarding sequence length in Section 4.1.2 are trained with a model of $L = 16$ layers, $d = 2048$ hidden dimensions, MLP dimension of 8192, and $h = 16$ attention heads (with the same number of key-value heads). We keep the weight decay at 0.1, learning rate at 1×10^{-3}, 1000 warmup iterations, and a warmup cosine learning rate scheduler. All models are trained on 5B tokens with a global batch size of 256.

E.4 GROUP QUERY ATTENTION EXPERIMENTS

The experiments regarding GQA in Section 4.2.1 are trained with a model of $L = 16$ layers, $d = 2048$ hidden dimensions, MLP dimension of 8192, and $h = 16$ attention heads. The group size settings $G \in \{1, 4, 16\}$ correspond to key-value head sizes of 16, 4, and 1, respectively. With this configuration, the model sizes are approximately 1.2B (1.176B), 1.1B (1.076B), and 1.051B parameters. We keep the training sequence length at 4096, weight decay at 0.1, learning rate at 1×10^{-3}, 1000 warmup iterations, and a warmup cosine learning rate scheduler. The $G = 1$ models are trained on 5B tokens, while $G = 4$ and $G = 16$ are trained on 5.35B and 5.6B tokens, respectively, to keep the training FLOPs constant. All experiments are trained with a global batch size of 256.

[1] `https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu`
[2] `https://github.com/EleutherAI/gpt-neox`
[3] `https://github.com/allenai/OLMo`
[4] `https://github.com/EleutherAI/lm-evaluation-harness`

E.5 Mixture of Expert Experiments

Table 7: Model configuration for Section 4.2.2.

	Depth	Hidden Dim	MLP Hidden Dim	Query Head	Key-Value Head	Total Experts	Activated Experts	Shared Experts
Dense-400M	16	1536	3072	12		-	-	-
MoE-2BA400M		768	1536	12		32	4	1
Dense-1B		2048	8192	16		-	-	-
MoE-7BA1B		2048	2048	16		64	8	-

We adopt the kernel and MoE implementation from OLMo-core[5] and utilize the implementation of DropLess MLP (Gale et al., 2022). For both size configurations of MoE training, we set the load balancing loss (Shazeer et al., 2017) to 0.01 and router z-loss to 0.001 (Zoph et al., 2022) according to the practice of OLMoE (Muennighoff et al., 2024). The detailed configurations of the models for the two setups are presented in Table 7. For all runs, we set the training sequence length at 4096, weight decay at 0.1, and 1000 warmup iterations with a warmup cosine learning rate scheduler, training on 5B total tokens with a global batch size of 256. We set the learning rate to 1×10^{-3} for Dense-400M, MoE-2BA400M, and Dense-1B experimental runs, but set the learning rate for MoE-7BA1B to 4×10^{-4} according to learning rate sweep results.

E.6 Ablation Experiments

Table 8: Model configuration for Section 5

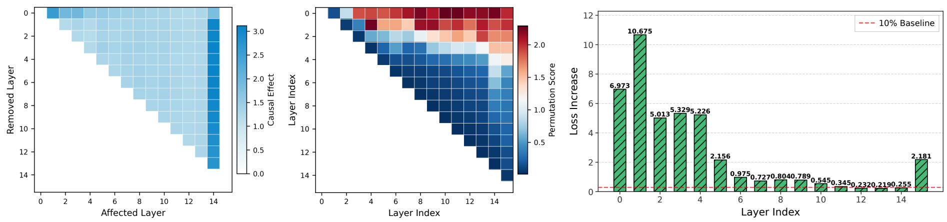
	Depth	Hidden Dim	MLP Hidden Dim	Query Head	Key-Value Head	Total Experts	Activated Experts	Shared Experts
$L = 16$	16	2048	8192	16		-	-	-
$L = 32$	32	1536	6144	16		-	-	-
MoE-$L = 32$	32	1120	1120	16		64	8	-

For all experiments in Section 5, we use 1.2B parameter models with configurations detailed in Table 8. We train on 10B tokens from FineWeb-Edu and report evaluation perplexity on a held-out dataset, along with zero-shot accuracy on ARC-Challenge (Clark et al., 2018) and HellaSwag (Zellers et al., 2019).

All experiments use the AdamW optimizer with a global batch size of 256. Unless otherwise specified, models are trained with weight decay of 0.1, context length of 1024, 1000 warmup iterations, and a cosine learning rate scheduler. We use a learning rate of 4×10^{-4} for MoE models and 1×10^{-3} for all other configurations.

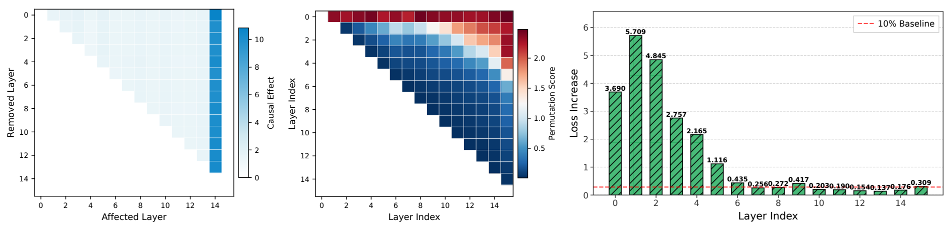
[5] `https://github.com/allenai/OLMo-core`

E.7 WEIGHT DECAY



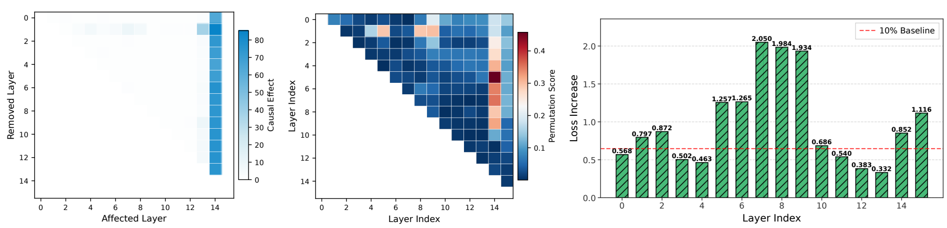
(a) Causal score (b) Permutation score (c) Loss increase

Figure 19: Score visualziation for weight decay $\lambda = 0.1$.



(a) Causal score (b) Permutation score (c) Loss increase

Figure 20: Score visualziation for weight decay $\lambda = 1$.

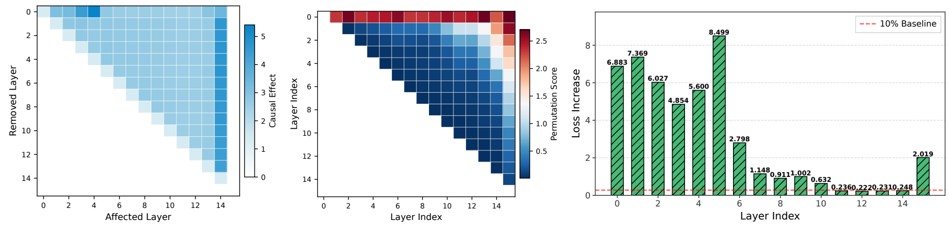


(a) Causal score (b) Permutation score (c) Loss increase

Figure 21: Score visualziation for weight decay $\lambda = 3$.

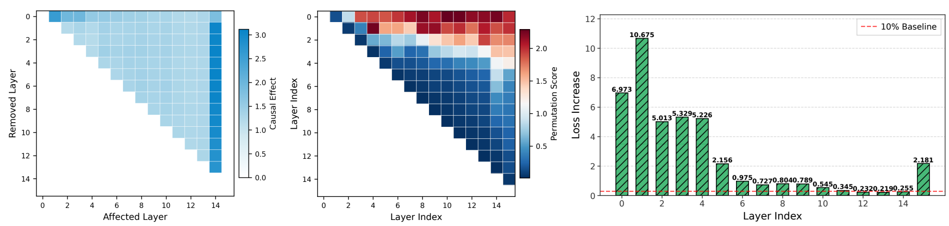
Visualizations of each score for $\lambda \in \{0.1, 1, 3\}$ are provided in Figures 19 to 21, respectively.

E.8 SEQUENCE LENGTH



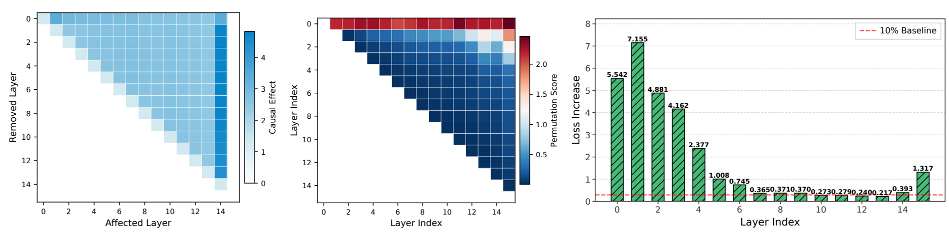
(a) Causal score (b) Permutation score (c) Loss increase

Figure 22: Score visualziation for sequence length $T = 1024$.



(a) Causal score (b) Permutation score (c) Loss increase

Figure 23: Score visualziation for sequence length $T = 4096$.

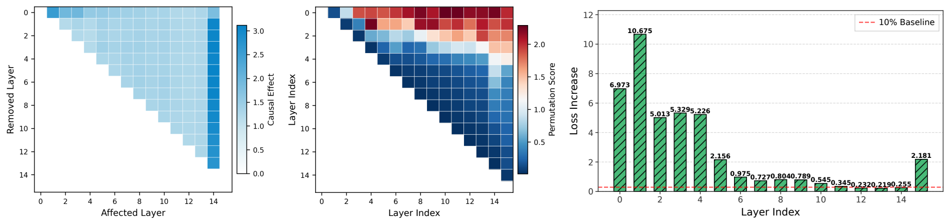


(a) Causal score (b) Permutation score (c) Loss increase

Figure 24: Score visualziation for sequence length $T = 8192$.

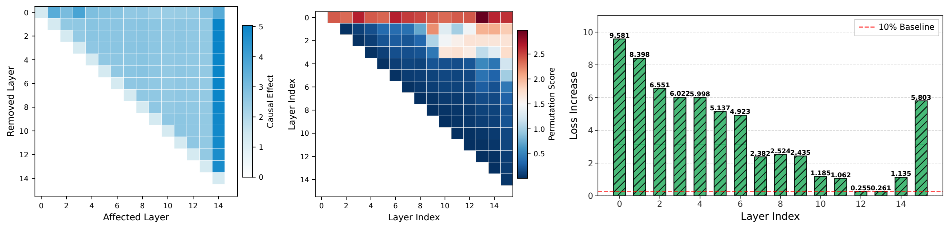
Visualizations of each score for $T \in \{1024, 4096, 8192\}$ are provided in Figures 22 to 24, respectively.

E.9 Grouped Query Attention



(a) Causal score (b) Permutation score (c) Loss increase

Figure 25: Score visualziation for MHA $G = 1$.

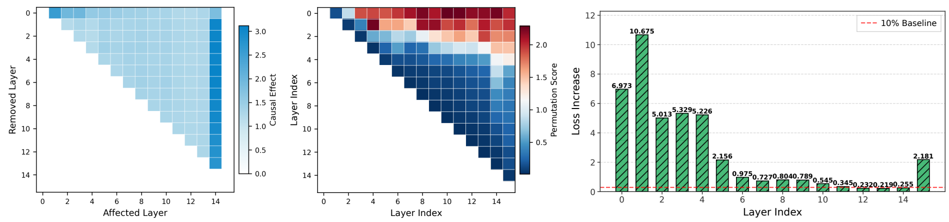


(a) Causal score (b) Permutation score (c) Loss increase

Figure 26: Score visualziation for MQA $G = 16$.

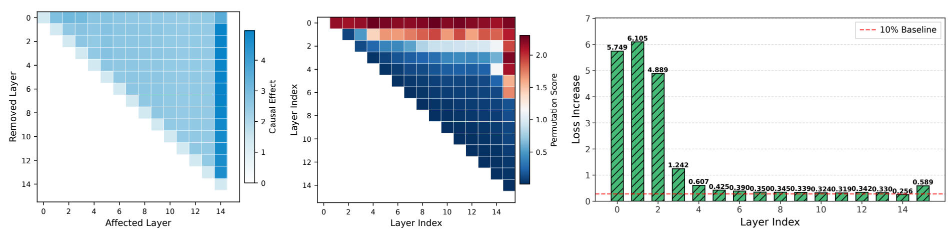
Visualizations of each score for $G \in \{1, 16\}$ are provided in Figures 25 and 26, respectively.

E.10 MIXTURE OF S



(a) Causal score (b) Permutation score (c) Loss increase

Figure 27: Score visualziation for Dense-1B.



(a) Causal score (b) Permutation score (c) Loss increase

Figure 28: Score visualziation for MoE-7BA1B.

Visualizations of Dense-1B and MoE-7BA1B are provided in Figures 27 and 28, respectively.