

Scalable Complexity Control Facilitates Reasoning Ability of LLMs

Liangkai Hang^{1,†}, Junjie Yao^{1,†}, Zhiwei Bai¹, Tianyi Chen¹, Yang Chen¹, Rongjie Diao¹,
Hezhou Li¹, Pengxiao Lin¹, Zhiwei Wang¹, Cheng Xu¹, Zhongwang Zhang^{1,*}, Zhangchen Zhou¹,
Zhiyu Li^{4,5,*}, Zehao Lin^{4,5}, Kai Chen⁵, Feiyu Xiong^{4,5,*},
Yaoyu Zhang^{1,2,*}, Weinan E^{3,4}, Hongkang Yang^{5,*}, Zhi-Qin John Xu^{1,2,4*}

¹ Institute of Natural Sciences, School of Mathematical Sciences, Shanghai Jiao Tong University

² MOE-LSC, School of Artificial Intelligence, Shanghai Jiao Tong University

³ Center for Machine Learning Research, School of Mathematical Sciences, Peking University

⁴ Center for LLM, Institute for Advanced Algorithms Research, Shanghai

⁵ MemTensor (Shanghai) Technology Co., Ltd.

[†] Equal contribution, list in alphabetical order.

Abstract

The reasoning ability of large language models (LLMs) has been rapidly advancing in recent years, attracting interest in more fundamental approaches that can reliably enhance their generalizability. This work demonstrates that model complexity control, conveniently implementable by adjusting the initialization rate and weight decay coefficient, improves the scaling law of LLMs consistently over varying model sizes and data sizes. This gain is further illustrated by comparing the benchmark performance of 2.4B models pretrained on 1T tokens with different complexity hyperparameters. Instead of fixing the initialization std, we found that a constant initialization rate (the exponent of std) enables the scaling law to descend faster in both model and data sizes. These results indicate that complexity control is a promising direction for the continual advancement of LLMs.

1 Introduction

In recent years, large language models (LLMs) have achieved unprecedented progress, demonstrating impressive performance on a wide range of tasks [2, 36, 58, 66, 70, 77]. The key to this success is the improved generalizability of these models, in particular, their reasoning ability. Various approaches have been explored to enhance reasoning, such as post-training with reinforcement learning [30, 21], high-quality math and code data with reasoning traces [21], chain-of-thought and related prompting strategies [67], and the separation of reasoning and memory in pretraining [71].

This work identifies model complexity as the key factor in the development of the reasoning ability of LLMs. Complexity control can be implemented through various means, such as adjusting the rate of parameter initialization [39, 88, 89, 84, 73] or applying stronger penalties on parameter norms [84]. Similar to the gene that determines the characteristics of an organism, these designs directly regulate the reasoning ability of LLMs.

An illustration is provided in Figure 1. On the left, a pretrained model with large complexity (large initialization scale) fails to make meaningful next-token predictions on unseen test data. In the middle, with moderate complexity (the commonly used initialization scale), the model demonstrates basic knowledge of grammar (e.g. “finds that”) and vocabulary (e.g. “Jessica”), as well as the

*Corresponding author: xuzhiqin@sjtu.edu.cn, hongkang@alumni.princeton.edu, zhyy.sjtu@sjtu.edu.cn, xiongyf@iaar.ac.cn, lizy@iaar.ac.cn, 0123zzw666@sjtu.edu.cn

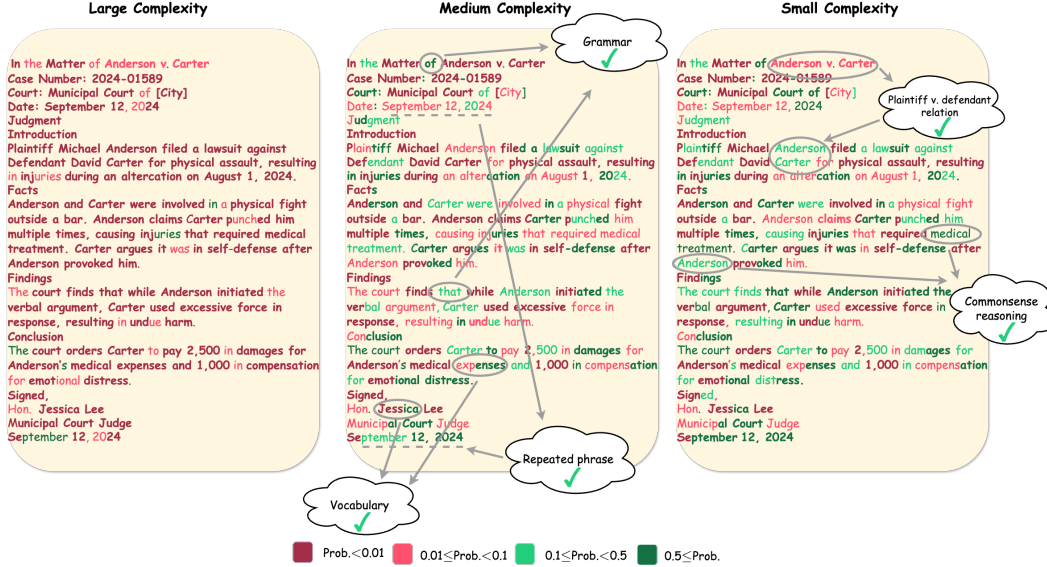


Figure 1: Next-token prediction accuracy with varying model complexity. The colors indicate the probability of each token predicted by the models. Model complexity decreases from left to right, as the initialization rate $\gamma = 0.1, 0.5, 1$ (see Section 3 for the definition of γ). These models have the same shape (180M parameters) and are trained the same dataset (40B tokens).

induction head mechanism [43] (e.g. the repeated phrase “September 12, 2024”). On the right, the low-complexity model (i.e. with small initialization scale) further captures sophisticated semantic reasoning, successfully predicting “Plaintiff Michael Anderson” given the context “Anderson v. Carter” despite that the full name “Michael Anderson” has not appeared before, and also predicting “required medical treatment” given that Anderson was punched by Carter.

Intuitively, smaller complexity forces the model to compress data into a smaller set of production rules, revealing the deep dependency among the tokens and preventing plain memorization. Previous studies [39, 88] have shown that with a smaller initialization scale, neurons within each layer tend to evolve within a few groups, a phenomenon known as condensation, which limits the effective number of neurons. Readers are referred to an overview of condensation [69]. Likewise, models trained with stronger penalty on parameter norm may converge to solutions with smaller norms, resulting in lower-complexity outputs. Thus, it is promising that these techniques can enhance the reasoning ability of LLMs.

As a verification of this intuition, our experiments exhibit improved scaling laws in data size and model size, as well as higher scores over almost all benchmarks (+4.6% for 0.9B model with 600B data, and +3.4% for 2.4B model with 1T data, averaged over 15 tasks). Compared with the standard deviation (std) of parameter initialization, the initialization rate (the exponent of std as a function of network width) turns out to be the right invariant for the scaling laws. This is in accordance with previous works [39, 88] on the phase diagram of neural network training. We provide some heuristic calculations to explain how complexity control facilitates the learning of multi-step reasoning.

2 Related works

LLMs reasoning ability Even advanced LLMs such as GPT-4 often struggle with implicit reasoning over parametric knowledge [56, 28, 47, 45, 3, 72], revealing their limited ability to internalize structured facts and rules. Verbalized reasoning strategies such as chain-of-thought can substantially boost performance, particularly for large models [67, 75, 54, 37, 74, 18, 33]. However, understanding the underlying capacity for implicit reasoning remains a critical challenge, often studied by controlled experiments [46, 13, 63]. This work takes the more intrinsic perspective of complexity control.

Effect of initialization Parameter initialization is known to be influential to the training and performance of classical neural networks [4, 10, 80, 14, 27, 40, 50, 52, 68]. For instance, distinct phases (the linear and condensed regimes) can be induced in wide ReLU networks by varying initialization rates [39, 88], and training in the condensed regime tend to fit data with lower-complexity functions [81, 85, 82, 86]. The particular case of Transformer language models has also been studied [25, 38, 59, 61, 78, 91, 83, 84], with particular interest on the impact of initialization on training stability and efficiency. Experiments on toy datasets [83, 84] show that small initialization scales assist Transformers to identify the elementary functions when fitting synthetic compositional data. However, the influence of initialization on the reasoning ability of Transformers trained on natural language data remains to be explored.

Weight decay [29] introduced weight decay as a method to improve the generalization of neural networks. Many subsequent works have explored its role in controlling model complexity and enhancing generalization [8, 7, 5, 42, 20, 65]. More recently, weight decay has been shown to be particularly important for achieving better generalization [44, 60].

In this work, we systematically investigate how controlling pre-training complexity, via initialization strategies and weight decay, affects the performance of LLMs. Unlike prior studies that focus on training stability or synthetic tasks, we evaluate the impact across a wide range of downstream benchmarks and analyze the underlying mechanisms, aiming to offer practical guidance for large-scale model pre-training.

3 Complexity control

In this section, we introduce the approach to modulate the model complexity.

Initialization rate γ Given any trainable parameter matrix $\mathbf{W} \in \mathbb{R}^{d_{in} \times d_{out}}$. We initialize its elements according to the following normal distribution:

$$\mathbf{W}_{i,j} \sim \mathcal{N}\left(0, (d_{in}^{-\gamma})^2\right),$$

where γ is the initialization rate. Specifically, the initialization scale decreases as γ increases. Note that $\gamma = 0.5$ is commonly used in many default initialization methods, such as LeCun initialization [31] and He initialization [22]. As the network width towards infinity [39, 88], the training of the network with $\gamma > 0.5$ exhibits significant non-linear characteristics, i.e., condensation. Therefore, initialization scales with $\gamma > 0.5$ are generally considered small. Tuning γ is a scalable approach for complexity control.

Weight decay coefficient λ Given any trainable parameter θ_t where t denotes the current training step. Define $\hat{\theta}_t$ as the parameter after optimizing by gradient and moment, the weight decay is implemented by

$$\theta_{t+1} \leftarrow \hat{\theta}_t - \lambda C \theta_t, \quad (1)$$

where λ is the weight decay coefficient.

4 Results

To evaluate the impact of complexity control, we train LLMs based on the Llama- architecture [58] under different levels of model complexity. We examine scaling laws with respect to both model size and training data size, and compare different models across a range of benchmarks. Detailed setup of training and evaluation is provided in Appendix B.

4.1 Scaling law

We first establish three model complexity configurations: (1) the small-complexity setup with $\gamma = 1, \lambda = 1$, (2) the large-complexity setup with $\gamma = 0.5, \lambda = 0.1$, and (3) the commonly used default configuration (e.g., GPT2, HuggingFace) with $\sigma = 0.02, \lambda = 0.1$. Under each configuration, we train 0.8B-parameter models with varying training data scales ranging from 0.2 billion to 1.4

billion tokens. Figure 2 (left) demonstrates the relationship between test loss and data scale across different complexity configurations. The curve of small-complexity (green) exhibits a distinct leftward shift relative to the large-complexity (purple), suggesting that complexity controlling can effectively improve the sample efficiency. Additionally, while the test loss of the large-complexity model demonstrates comparable to the standard configuration at smaller data scales, their performance diverges as data size increases. Specifically, the former configuration achieves progressively lower test loss than the latter, indicating superior scalability of the γ -initialization approach with steeper scaling slopes. Besides, we train models with distinct model sizes by 1 billion tokens. Figure 2 (right) reveals similar patterns in the relationship between test loss and model size. Although σ -initialization achieves lower test loss with smaller models, its performance plateaus as the model size increases, resulting in significantly higher loss compared to γ -initialized models at larger scales. These results illustrate the scalability potential of the γ -initialization method, maintaining performance advantages across expanding parameters and data scales.

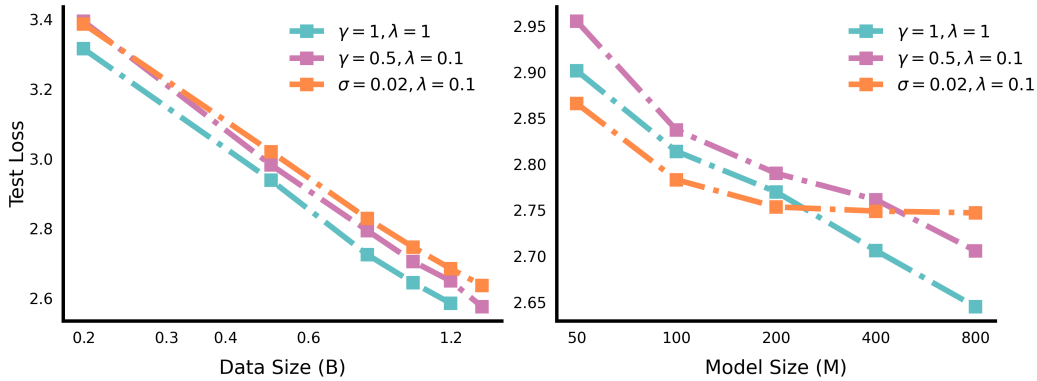


Figure 2: Test loss across varying data and model scales under different complexity configurations. Left: Test loss progression for 0.8B-parameter models trained with data scales ranging from 0.2B to 1.4B tokens. Right: Test loss versus model parameter counts (50M-0.8B) with fixed 1B training tokens. Line colors correspond to different complexity configurations.

4.2 Evaluation

Table 1: Evaluation of models with different model complexities.

Models ¹ (γ, λ)	0.9B Large (0.5, 0.1)	0.9B Small (1, 1)	2.4B Large (0.5, 0.1)	2.4B Small (0.58, 1)
MMLU	49.9	52.5 (+2.6)	60.4	64.4 (+4)
MMLU-Pro	17.6	21.5 (+3.9)	30.6	30.1 (-0.5)
BBH	33.3	34.9 (+1.6)	42.4	43.7 (+1.3)
ARC-C	46.1	49.8 (+3.7)	58.6	60.7 (+2.1)
TruthfulQA	53.4	56.8 (+3.4)	58.7	61.7 (+3)
WinoGrande	68.0	72.2 (+4.2)	73.1	76.9 (+3.8)
HellaSwag	63.0	67.3 (+4.3)	71.2	75.4 (+4.2)
AGIEval-EN	25.3	29.1 (+3.8)	33.2	35.6 (+2.4)
OpenBookQA	38.8	39.6 (+0.8)	41.8	43.4 (+1.6)
CommonsenseQA	57.7	67.2 (+9.5)	70.4	76.9 (+6.5)
GPQA	25.9	26.8 (+0.9)	29.0	31.7 (+2.7)
MATH	5.9	11.5 (+5.6)	34.3	35.3 (+1)
GSM8K	21.8	40.2 (+19.4)	52.8	63.8 (+11.0)
MBPP	6.6	10.8 (+4.2)	20.6	22.4 (+1.8)
IFEval	31.3	32.0 (+0.7)	34.7	40.9 (+6.2)

¹ “Large” and “Small” mean large complexity and small complexity, respectively.

To assess the impact of complexity control, we adopt reasonable complexity configurations and train LLMs with the following setup: (1) 0.9B-parameter models on 600B tokens from SlimPajama [53] and (2) 2.4B-parameter models trained with 1T high-quality corpus. We evaluate the performance

of our models across a comprehensive set of benchmark tasks. As shown in Table 1, the performance of small-complexity models improves significantly. Specifically, complexity controlling yields substantial gains in reasoning capabilities. On math-related tasks, two small-complexity models achieve improvements of 19.4 and 11.0 points on the GSM8K benchmark, respectively, as well as 5.6 and 1 points on the MATH dataset. Moreover, these small-complexity models demonstrate notable performance enhancements across other reasoning tasks, including Winogrande(+4.2, +3.8), HellaSwag(+4.3, +4.2), and CommonsenseQA(+9.5, +6.5). These results demonstrate that principled control of model complexity can significantly enhance the overall capabilities of LLMs, particularly in reasoning tasks. Note that small initialization may yield slower training. For 2.4B model, the performance of $\gamma = 0.58$ is slightly better than that of $\gamma = 1$ in Table 5 in Appendix.

Complexity control also significantly improves the performance of base models, with results summarized in Table 4 in Appendix C. As quantified in Figure 3, the small-complexity model attains greater score increments across most tasks at the SFT stage.

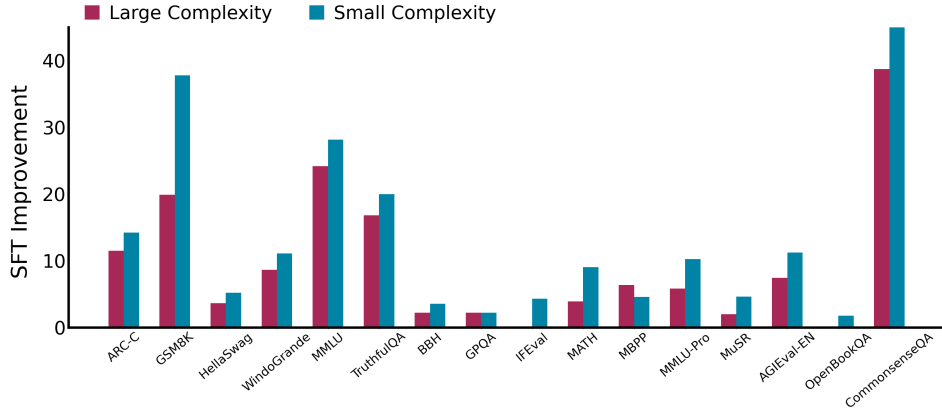


Figure 3: Performance improvement via SFT across complexity configurations(0.9B model). It is quantified as the performance gap between the SFT model and the corresponding base model.

5 Analysis

To investigate the effect of initialization scale and weight decay in complexity control, we trained 180M-parameter models with different complexity configurations. We conduct analysis from evaluation and parameter analysis, which yield mechanistic insights into complexity control.

5.1 Influence of initialization scale and weight decay

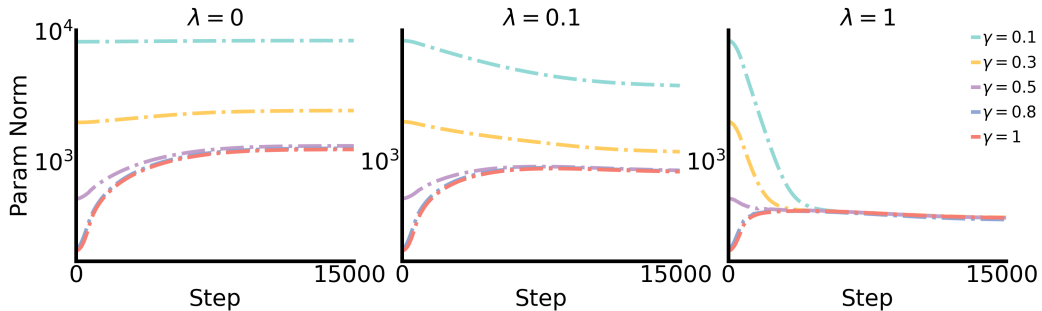


Figure 4: Parameter norm evolution across complexity configurations. Left to right: $\lambda = 0, 0.1, 1$; Line colors correspond to $\gamma = 0.1, 0.3, 0.5, 0.8, 1$.

Examining the interplay between initialization scale γ and weight decay λ on model complexity is critical. Figure 4 visualizes the temporal evolution of parameter norms under varying $\gamma - \lambda$,

demonstrating that large γ coupled with large λ systematically induces lower model complexity. Specifically, with small λ ($\lambda = 0, 0.1$), configurations with larger γ converge to smaller complexities. However, with large λ ($\lambda = 1$), models with different γ converge to comparable small norm.

5.2 Alignment between complexity and capability

We perform evaluations of models across multiple benchmark tasks. Figure 5A depicts the performance distribution of the average score, GSM8K, and HellaSwag, showing better performance as stronger complexity control. As shown in Figure 5B, model performance demonstrates a strong inverse correlation with model complexity. Full evaluation results are provided in Appendix E.

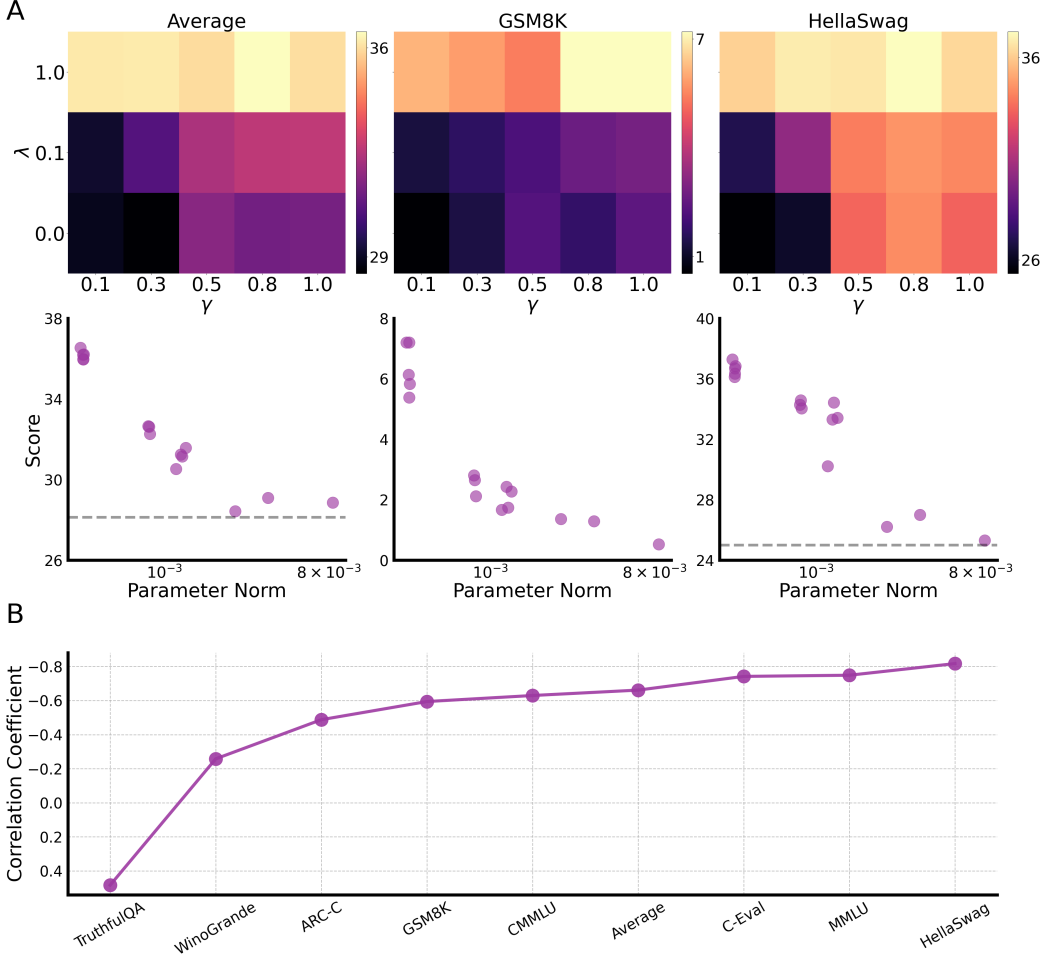


Figure 5: (A) Evaluation scores (average, GSM8K, HellaSwag) under varying complexities. Top: Performance landscape across $\gamma - \lambda$ with color indicating score (dark: low, light: high). Bottom: Score-complexity relationships with points indicating the models and the dashed lines denoting baseline performance levels. (B) Task-specific Spearman correlations between model complexity and task score. Stronger negative correlations (approaching -1) indicate greater performance enhancement through complexity control.

5.3 Model Analysis

Embedding space The embedding space reflects the model’s representation of the vocabulary and its learning patterns. We compare the cosine similarity of the 350 most frequent embedding vectors under different initialization scales. The results in Figure 6 demonstrate that with large complexity ($\gamma = 0.1$), the embeddings are pairwise orthogonal, indicating the model ignores their relationship. In

contrast, controlling complexity ($\gamma = 0.5, 1$) significantly increases the similarity among embeddings, consistent with previous study on condensation phenomenon [39]. This analysis suggests that small complexity encourages the model to find associations among tokens.

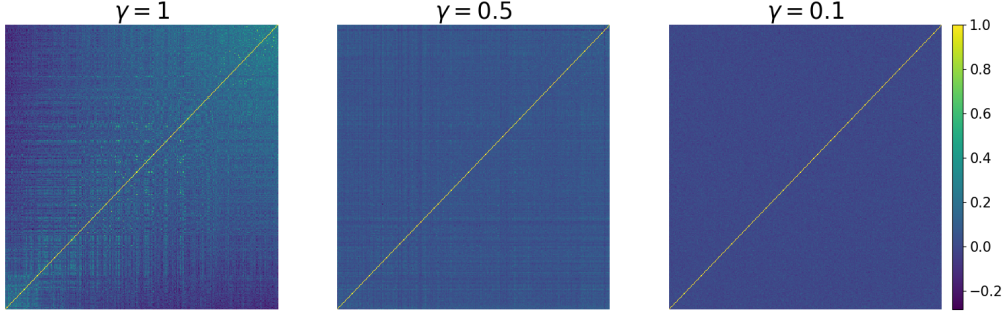


Figure 6: Cosine similarity among 350 embedding vectors which occur most frequently in training dataset under initialization scales $\gamma = 1, 0.5, 0.1$ ($\lambda = 0$).

Attention matrix For each trainable parameter matrix $\mathbf{W} \in \mathbb{R}^{d_{in} \times d_{out}}$, we define the following metrics to measure its condensation degree and low-rank degree, respectively.

$$D_c(\mathbf{W}) := \frac{1}{d_{in}d_{out}} \sum_{i,j} \frac{\mathbf{W}_i^T \mathbf{W}_j}{\|\mathbf{W}_i\|_2 \cdot \|\mathbf{W}_j\|_2}, \quad D_s = \frac{\max_i S_{\mathbf{W},i}}{\sum_i S_{\mathbf{W},i}}, \quad (2)$$

where \mathbf{W}_i and $S_{\mathbf{W},i}$ means the i -th row of \mathbf{W} and the i -th singular value of \mathbf{W} . Larger values of D_c and D_s indicate fewer effective directions in matrix \mathbf{W} , suggesting \mathbf{W} learns a smaller set of features for fitting dataset. Figure 7 exhibits the D_c and D_s of the query projection and key projection matrices of models with distinct complexities. The results demonstrate that controlling model complexity effectively increases D_c and D_s across the attention matrices, suggesting that the attention modules in small-complexity models focus more on the fundamental relationships between tokens within sequences, which results in stronger generalization and reasoning ability.

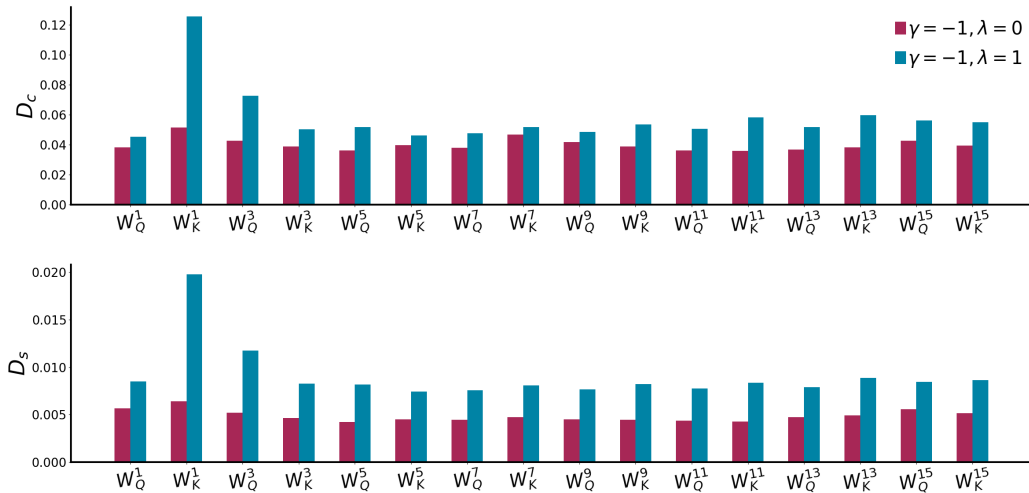


Figure 7: D_c and D_s of \mathbf{W}_Q and \mathbf{W}_K in each layer under different model complexity configurations.

6 Discussion

6.1 Training Stability

A potential concern in small complexity is training stability, particularly the emergence of loss spikes during optimization. This instability tends to escalate with increasing model scale, as observed in our experiments. To address this challenge in training our 2.4B parameter models, we adopt the $\gamma = 0.58$ which is not too large, and implement embedding normalization and sandwich normalization, which successfully mitigate the loss spike phenomenon. A detailed discussion is provided in Appendix D.

6.2 Training dynamics

Distinct $\gamma - \lambda$ configurations induce distinct learning dynamics and complexity trajectories, accompanied by different performance evolution patterns. We evaluate the two 2.4B models referenced in Table 1 every 5000 steps during pre-train and obtain the dynamics of the average score. Figure 8 characterizes the temporal evolution of model complexity and the average score. For the large-complexity model ($\gamma = 0.5, \lambda = 0.1$), complexity initially increases followed by progressive decay, during which performance exhibits slow improvement in the ascending phase but accelerates substantially after the complexity turning. Conversely, small-complexity configurations maintain monotonic complexity growth with steady performance gains throughout training.

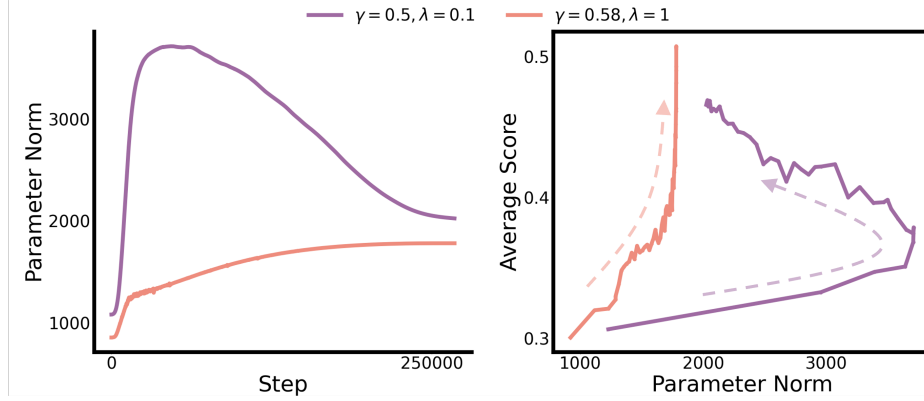


Figure 8: The evolution of parameter norm and average score with two distinct $\gamma - \lambda$ configurations.

6.3 Theoretical analysis

This section employs heuristic calculations to analyze how small initialization enhances the generalization of LLMs.

Recall that a 2-layer net $f : \mathbb{R}^d \rightarrow \mathbb{R}$ with 1-homogeneous activation (such as ReLU) can be written compactly as $f_\pi(\mathbf{x}) := \int \sigma(\mathbf{w} \cdot \mathbf{x}) d\pi(\mathbf{w})$ for some finite signed measure π over the parameter space \mathbb{R}^d (without loss of generality, we can replace \mathbf{x} by $[\mathbf{x}, 1]$ to account for the bias) [15, 17]. There are two common functional spaces for these functions. In the kernel regime [14, Theorem 3.9], the function space is characterized by the RKHS norm [48]

$$\|f\|_{\mathcal{H}} = \inf_{\pi} \left\| \frac{\delta\pi}{\delta\pi_0} \right\|_{L^2(\pi_0)} = \left(\int \left| \frac{\delta\pi}{\delta\pi_0}(\mathbf{w}) \right|^2 d\pi_0(\mathbf{w}) \right)^{1/2}, \quad \text{s.t. } f = f_\pi \text{ and } \pi \ll \pi_0$$

where π_0 is some base distribution (typically the initialization distribution of \mathbf{w}) and $\delta\pi/\delta\pi_0$ is the Radon-Nikodym derivative. In the mean-field regime [40, 50], the function space is characterized by the Barron norm [16, 17]

$$\|f\|_{\mathcal{B}} = \inf_{\pi} \|\pi\|_{\text{TV}} = \inf_{\pi} \int d|\pi|(\mathbf{w}), \quad \text{s.t. } f = f_\pi \text{ and } \text{supp}(\pi) \subseteq \mathbb{S}^d$$

where $\|\cdot\|_{\text{TV}}$ denotes total variation and π is supported on the unit sphere \mathbb{S}^d . Naturally, we can define an interpolation between these norms: for any $\gamma > -3/2$,

$$\|f\|_\gamma := \inf_{\pi} \left\| \frac{\delta\pi}{\delta\pi_0} \right\|_{L^{3+2\gamma}(\pi_0)}, \quad \text{s.t. } f = f_\pi \text{ and } \text{supp}(\pi) \subseteq \mathbb{S}^d$$

and by 1-homogeneity π_0 can be assumed to be supported on \mathbb{S}^d as well. One can check that $\|\cdot\|_{\gamma=-1/2} = \|\cdot\|_{\mathcal{H}}$ and $\|\cdot\|_{\gamma=-1} = \|\cdot\|_{\mathcal{B}}$. By Jensen's inequality, the function space of $\|\cdot\|_\gamma$ becomes larger as γ decreases. In particular, if $\gamma \leq -1$, then π does not have to be absolutely continuous with respect to π_0 , i.e. it can develop singular parts, sometimes known as ‘‘condensation’’ [9]. Else, $\gamma > -1$ and π must have the form $a\pi_0$.

The question is how γ affects deep networks. Let $\mathcal{M}(\mathbb{R}^d, \mathbb{R}^k)$ denote \mathbb{R}^k -vector-valued finite measures over \mathbb{R}^d . For simplicity, consider deep residual nets, $\mathbf{x}^l = \mathbf{x}^{l-1} + f_l(\mathbf{x}^{l-1})$ for $l = 1, \dots, L$, where each block $f_l : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a 2-layer net parametrized by some $\pi_l \in \mathcal{M}(\mathbb{R}^d, \mathbb{R}^d)$. For any fixed sequence of input weights $(\mathbf{w}_l)_{l=1}^L \in \mathbb{R}^{L \times d}$, fixed (π_l) , and an input \mathbf{x}^0 drawn from some data distribution, the activation sequence $(\sigma(\mathbf{w}_l \cdot \mathbf{x}_l))$ is a stochastic process with a potentially complicated dependency structure. This is intended to capture cross-layer collaborations in LLMs or ‘‘circuits’’, such as the induction head [43], IOI circuit [62], and arithmetic circuits [35]. Therefore, we model the parameter distribution of the full network by $\pi \in \mathcal{M}(\mathbb{R}^{Ld}, \mathbb{R}^{Ld})$, instead of just the sequence (π_l) , in order to model the dependency among the activated parameters. A generalization of the norm $\|\cdot\|_\gamma$ from 2-layer nets to deep residual nets can be defined by

$$\|f\|_\gamma = \inf_{\pi} \left\| \frac{\delta\pi}{\delta\pi_0^{\otimes L}} \right\|_{L^{3+2\gamma}(\pi_0^{\otimes L})} = \inf_{\pi} \left(\int \left\| \frac{\delta\pi}{\delta\pi_0^{\otimes L}} (\oplus_{l=1}^L \mathbf{w}_l) \right\|^{3+2\gamma} \prod_{l=1}^L d\pi_0(\mathbf{w}_l) \right)^{1/(3+2\gamma)}$$

which ranges among all parametrizations (π_l) of f and the resulting dependency π with respect to the data distribution. To study $\|\cdot\|_\gamma$, we make the following assumptions:

1. Among the approximate global minimizers of the loss, training initialized with rate γ for any $\gamma > -3/2$ always converges to a minimizer f^* with the minimum $\|\cdot\|_\gamma$ norm. This assumption holds for 2-layer nets in the kernel regime ($\gamma = -1/2$) when trained by gradient descent [79], and we expect similar behavior in general settings.
2. The parameter distribution π^* of each approximate global minimizer f^* can always be decomposed into a weighted sum of product measures

$$\pi^* = \sum_i c_i \pi^i, \quad c_i \in \mathbb{R}, \quad \pi^i = \bigotimes_{l=1}^L \pi_l^i, \quad \|\pi_l^i\|_{\text{TV}} = 1$$

and these π^i have disjoint supports (up to π -negligible subsets). Furthermore, the variation of each π^i , namely $|\pi_l^i|$, is simply π_0 restricted to some subset $S_l^i \subseteq \mathbb{R}^d$, i.e. $|\pi_l^i| = \pi_0 \mathbf{1}_{S_l^i} / \pi_0(S_l^i)$, and there exists some constant $0 < \epsilon \ll 1$ such that either $\pi_0(S_l^i) = \epsilon$ or $\pi_0(S_l^i) = 0.9$. The purpose of this assumption is to simplify computation, and we expect our results to hold in much more general settings.

For each π^i , denote by L_i the number of π_l^i with $\pi_0(S_l^i) = \epsilon$. This L_i can be interpreted as the circuit depth, the number of layers where π^i is non-trivial. The norm of each minimizer f^* becomes

$$\begin{aligned} \|f^*\|_\gamma &= \left(\sum_i \int \left\| \frac{\delta(c_i \pi^i)}{\delta\pi_0^{\otimes L}} \right\|^{3+2\gamma} d\pi_0^{\otimes L} \right)^{1/(3+2\gamma)} = \left(\sum_i c_i^{3+2\gamma} \prod_{l=1}^L \int \left\| \frac{\delta\pi_l^i}{\delta\pi_0} \right\|^{3+2\gamma} d\pi_0 \right)^{1/(3+2\gamma)} \\ &= \left(\sum_i (c_i \epsilon^{-L_i} 0.9^{-(L-L_i)})^{3+2\gamma} \right)^{1/(3+2\gamma)} = 0.9^{-L} \|c_i (0.9/\epsilon)^{L_i}\|_{l^{3+2\gamma}} \end{aligned}$$

Hence, each minimizer can be viewed as a circuit ensemble, characterized by its circuit weights and circuit depths $\{(c_i, L_i)\}$. Recall that for general l^p norms, those with large p are sensitive to the maximum value, while those with small p are sensitive to the amount of non-zero elements. Thus, for large γ such as $-1/2$, the circuit ensemble chosen by training tends to be dense but uniformly shallow (many $c_i > 0$ but small $\max L_i$), whereas for small γ such as -1 , the chosen circuits are more likely to be sparse and deep (few $c_i > 0$ but L_i can be large).

Intuitively, sparse and deep circuits signify a generalizable solution, as the model has managed to compress the diverse training data into very few but flexible patterns. Meanwhile, dense and shallow circuits may imply that the model does not have a deep understanding of the data and has to memorize a lot. In conclusion, it is plausible that small initialization increases generalizability by shifting the preference for circuit ensembles. Although our calculation is based on simple residual networks, it is reasonable to expect that a similar mechanism applies to Transformers.

Acknowledgments and Disclosure of Funding

This work is supported by the National Key R&D Program of China Grant No. 2022YFA1008200, the National Natural Science Foundation of China Grant No. 92270001, 12371511, 12422119.

References

- [1] Megatron-deepspeed. <https://github.com/microsoft/Megatron-DeepSpeed>, 2022.
- [2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [3] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation. In *arXiv preprint: abs/2309.14402*, 2023.
- [4] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pages 8141–8150, 2019.
- [5] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International conference on machine learning*, pages 254–263. PMLR, 2018.
- [6] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.
- [7] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.
- [8] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.
- [9] Zheng-An Chen, Yuqing Li, Tao Luo, Zhangchen Zhou, and Zhi-Qin John Xu. Phase diagram of initial condensation for two-layer neural networks. *CSIAM Transactions on Applied Mathematics*, 5(3):448–514, 2024.
- [10] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.
- [11] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.
- [12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [13] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

- [14] Weinan E, Chao Ma, and Lei Wu. A comparative analysis of optimization and generalization properties of two-layer neural network and random feature models under gradient descent dynamics. *Sci. China Math.*, 63, 2020.
- [15] Weinan E, Chao Ma, and Lei Wu. Machine learning from a continuous viewpoint, I. *Science China Mathematics*, 63(11):2233–2266, 2020.
- [16] Weinan E, Chao Ma, and Lei Wu. The Barron space and the flow-induced function spaces for neural network models. *Constructive Approximation*, 55(1):369–406, 2022.
- [17] Weinan E and Stephan Wojtowytsch. Representation formulas and pointwise properties for Barron functions. *Calculus of Variations and Partial Differential Equations*, 61(2):1–37, 2022.
- [18] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [19] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.
- [20] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299. PMLR, 2018.
- [21] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-rl: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [23] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.
- [24] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.
- [25] Xiao Shi Huang, Felipe Perez, Jimmy Ba, and Maksims Volkovs. Improving transformer optimization through better initialization. In *International Conference on Machine Learning*, pages 4475–4483. PMLR, 2020.
- [26] Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*, 2023.
- [27] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.
- [28] Nora Kassner, Benno Krojer, and Hinrich Schütze. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online, November 2020. Association for Computational Linguistics.
- [29] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

- [30] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.
- [31] Yann LeCun, Leon Bottou, Genevieve B. Orr, and Klaus Robert Müller. *Efficient BackProp*, pages 9–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.
- [32] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese, 2023.
- [33] Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations*, 2024.
- [34] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2021.
- [35] Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermy, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025.
- [36] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
- [37] Jiacheng Liu, Ramakanth Pasunuru, Hannaneh Hajishirzi, Yejin Choi, and Asli Celikyilmaz. Crystal: Introspective reasoners reinforced with self-feedback. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11557–11572, Singapore, December 2023. Association for Computational Linguistics.
- [38] Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. *arXiv preprint arXiv:2004.08249*, 2020.
- [39] Tao Luo, Zhi-Qin John Xu, Zheng Ma, and Yaoyu Zhang. Phase diagram for two-layer relu neural networks at infinite-width limit. *Journal of Machine Learning Research*, 22(71):1–47, 2021.
- [40] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.
- [41] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.
- [42] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on learning theory*, pages 1376–1401. PMLR, 2015.
- [43] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022.
- [44] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. In *arXiv preprint: abs/2201.02177*, 2022.

- [45] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore, December 2023. Association for Computational Linguistics.
- [46] Ben Prystawski, Michael Y. Li, and Noah Goodman. Why think step by step? reasoning emerges from the locality of experience. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [47] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.
- [48] Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In *2008 46th annual allerton conference on communication, control, and computing*, pages 555–561. IEEE, 2008.
- [49] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.
- [50] Grant Rotskoff and Eric Vanden-Eijnden. Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks. *Advances in neural information processing systems*, 31, 2018.
- [51] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.
- [52] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A central limit theorem. *Stochastic Processes and their Applications*, 130(3):1820–1852, 2020.
- [53] Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. <https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama>, June 2023.
- [54] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. Recitation-augmented language models. In *The Eleventh International Conference on Learning Representations*, 2023.
- [55] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.
- [56] Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. olympics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758, 2020.
- [57] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [58] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut

- Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. In *arXiv preprint: abs/2307.09288*, 2023.
- [59] Asher Trockman and J Zico Kolter. Mimetic initialization of self-attention layers. In *International Conference on Machine Learning*, pages 34456–34468. PMLR, 2023.
- [60] Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. In *arXiv preprint: abs/2309.02390*, 2023.
- [61] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [62] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023.
- [63] Xinyi Wang, Alfonso Amayuelas, Kexun Zhang, Liangming Pan, Wenhui Chen, and William Yang Wang. Understanding the reasoning ability of language models from the perspective of reasoning paths aggregation. In *arXiv preprint: abs/2402.03268*, 2024.
- [64] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.
- [65] Colin Wei and Tengyu Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. *Advances in neural information processing systems*, 32, 2019.
- [66] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. Survey Certification.
- [67] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [68] Francis Williams, Matthew Trager, Cláudio T. Silva, Daniele Panozzo, Denis Zorin, and Joan Bruna. Gradient dynamics of shallow univariate relu networks. *CoRR*, abs/1906.07842, 2019.
- [69] Zhi-Qin John Xu, Yaoyu Zhang, and Zhangchen Zhou. An overview of condensation phenomenon in deep learning. In *arXiv preprint arXiv:2504.09484*, 2025.
- [70] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- [71] Hongkang Yang, Zehao Lin, Wenjin Wang, Hao Wu, Zhiyu Li, Bo Tang, Wenqiang Wei, Jinbo Wang, Zeyun Tang, Shichao Song, Chenyang Xi, Yu Yu, Kai Chen, Feiyu Xiong, Linpeng Tang, and Weinan E. Memory³: Language modeling with explicit memory. *Journal of Machine Learning*, 3(3):300–346, 2024.
- [72] Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *arXiv preprint: abs/2402.16837*, 2024.
- [73] Junjie Yao, Zhongwang Zhang, and Zhi-Qin John Xu. An analysis for reasoning bias of language models with small initialization. *arXiv preprint arXiv:2502.04375*, 2025.

- [74] Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-star: Language models can teach themselves to think before speaking. In *arXiv preprint: abs/2403.09629*, 2024.
- [75] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STar: Bootstrapping reasoning with reasoning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- [76] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [77] Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, et al. PanGu- α : Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*, 2021.
- [78] Biao Zhang, Ivan Titov, and Rico Sennrich. Improving deep transformer with depth-scaled initialization and merged attention. *arXiv preprint arXiv:1908.11365*, 2019.
- [79] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.
- [80] Yaoyu Zhang, Zhi-Qin John Xu, Tao Luo, and Zheng Ma. A type of generalization error induced by initialization in deep neural networks. *arXiv:1905.07777 [cs, stat]*, 2019.
- [81] Yaoyu Zhang, Zhongwang Zhang, Leyang Zhang, Zhiwei Bai, Tao Luo, and Zhi-Qin John Xu. Linear stability hypothesis and rank stratification for nonlinear models. *arXiv preprint arXiv:2211.11623*, 2022.
- [82] Zhongwang Zhang, Yuqing Li, Tao Luo, and Zhi-Qin John Xu. Stochastic modified equations and dynamics of dropout algorithm. *arXiv preprint arXiv:2305.15850*, 2023.
- [83] Zhongwang Zhang, Pengxiao Lin, Zhiwei Wang, Yaoyu Zhang, and Zhi-Qin John Xu. Initialization is critical to whether transformers fit composite functions by reasoning or memorizing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [84] Zhongwang Zhang, Pengxiao Lin, Zhiwei Wang, Yaoyu Zhang, and Zhi-Qin John Xu. Complexity control facilitates reasoning-based compositional generalization in transformers. *arXiv preprint arXiv:2501.08537*, 2025.
- [85] Zhongwang Zhang and Zhi-Qin John Xu. Loss spike in training neural networks. *arXiv preprint arXiv:2305.12133*, 2023.
- [86] Zhongwang Zhang and Zhi-Qin John Xu. Implicit regularization of dropout. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [87] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.
- [88] Hanxu Zhou, Qixuan Zhou, Zhenyuan Jin, Tao Luo, Yaoyu Zhang, and Zhi-Qin John Xu. Empirical phase diagram for three-layer neural networks with infinite width. *Advances in Neural Information Processing Systems*, 2022.
- [89] Hanxu Zhou, Qixuan Zhou, Tao Luo, Yaoyu Zhang, and Zhi-Qin John Xu. Towards understanding the condensation of neural networks at initial training. *Advances in Neural Information Processing Systems*, 2022.
- [90] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023.
- [91] Chen Zhu, Renkun Ni, Zheng Xu, Kezhi Kong, W Ronny Huang, and Tom Goldstein. Gradinit: Learning to initialize neural networks for stable and efficient training. *Advances in Neural Information Processing Systems*, 34:16410–16422, 2021.

A Limitation and Future Work

While our methodology received thorough empirical validation, scaling to larger models and datasets remains constrained due to computational resource limitations. Future work will prioritize extending the framework to a larger training scale while exploring complexity control mechanisms during post-training.

B Experiments Setup

Model architecture Our models are based on the Llama-architecture [58]. Architectural configurations for different model scales are presented in Table 2. For the 2.4B variant, we incorporate embedding normalization and sandwich normalization techniques to enhance training stability.

Table 2: Training Configuration Across Model Scales

No.	Model Scale	Architecture Configuration	Pre-train Datasize
1	180M	Layers: 16 Head Dim: 80 Heads (KV): 16 (16)	40B
2	0.9B	Layers: 32 Head Dim: 64 Heads (KV): 32 (32)	600B
3	2.4B	Layers: 44 Head Dim: 80 Heads (KV): 40 (8)	1T

These configurations are systematically mapped as follows:

- **Configuration 1** governs results in:
 - Figure 1 and 4
 - Section 5
 - Appendix E
- **Configuration 2** applies to:
 - Figure 3, 13 and 14
 - Columns “0.9B Large” and “0.9B Small” in Table 1, 4
- **Configuration 3** encompasses:
 - Columns “2.4B Large” and “2.4B Small” in Table 1 and 4
 - Figure 8, 15 and 16
 - Appendix D

For Figure 2, the corresponding model specifications are detailed in Table 3.

Table 3: Architecture Configuration adopted in Figure 2.

Model Scale (M)	Layers	Heads (KV)	Head Dim
50	12	12	64
100	16	16	60
200	24	18	60
400	24	24	64
800	32	24	80

DataSet The 180M models are trained on a carefully curated 40B subset obtained through uniform sampling from the Memory³ training corpus. The 0.9B models undergo pre-training on a 600B subset derived from the SlimPajama corpus. Building upon the Memory³ foundation, we expand the training corpus through a rigorous data processing pipeline that integrates deduplication protocols, multi-dimensional quality assessment metrics, and optimized domain ratio adjustments. This systematic curation process yields a refined 1T training dataset, which is the basis for training our 2.4B models. For the SFT data, we also adopt the SFT data from Memory³.

Training setup The training is implemented using Microsoft’s Megatron-DeepSpeed framework [1], utilizing a mixed-precision configuration where model parameters, gradients, and activations were maintained in bfloat16 format while preserving optimizer states in float32 precision for the AdamW optimizer. The learning rate schedule adopted a cosine annealing strategy with linear warmup, where the warmup phase spanned the initial 5% of the total training iterations. The learning rate boundaries were configured with maximum and minimum values of 1×10^{-3} and 1×10^{-5} .

Training cost We present the computational costs for single training sessions across different model scales. The 180M model required 32 MX-C500 accelerators with a training duration of 12 hours. For the 0.9B architecture, the training process utilized 400 MX-C500 accelerators over 72 hours. 512 MX-C500 accelerators are employed for training a 2.4B model in one week.

Evaluation Details Our model are assessed across multiple open-source benchmarks via lm-eval-harness [19], covering areas such as factual knowledge: CMMLU [32], C-Eval [26], MMLU [23] and its enhanced version MMLU-Pro [64], OpenBookQA [41], and GPQA [49]. Language comprehension: BBH [55], ARC-C [11], TruthfulQA [34], WinoGrande [51], HellaSwag [76], AGIEval-EN [87], CommonsenseQA [57]. Code generation: MBPP [6] and IFEval [90]. Mathematical reasoning: MATH [24] and GSM8K [12].

C Evaluation of base models

Table 4 presents the evaluation results of the base models. The results reveal that the 0.9B model with small complexity does not demonstrate a significant advantage, while the 2.4B small-complexity model achieves measurable improvements. This disparity potentially stems from differences in pre-training data quality. Specifically, the 0.9B model’s training data (subsampling from the SlimPajama corpus) exhibits suboptimal quality, limiting its downstream task adaptability.

Table 4: Evaluation of base models

Models¹ (γ, λ)	0.9B Large (0.5,0.1)	0.9B Small (1,1)	2.4B Large (0.5,0.1)	2.4B Small (0.58,1)
MMLU	25.7	24.4 (-1.3)	36.5	41.1 (+4.6)
MMLU-Pro	11.8	11.3 (-0.5)	13.4	14.9 (+1.5)
BBH	31.1	31.3 (+0.2)	34.7	35.7 (+1.0)
ARC-C	34.6	35.6 (+1)	39.9	47.0 (+7.1)
TruthfulQA	36.6	36.8 (+0.2)	36.6	41.1 (+4.5)
WinoGrande	59.4	61.1 (+1.7)	64.0	68.0 (+4.0)
HellaSwag	59.4	62.0 (+2.6)	66.5	67.8 (+1.3)
AGIEval-EN	17.8	17.8	20.5	19.9 (-0.6)
OpenBookQA	39	37.8 (-1.2)	42.2	41.6 (-0.6)
CommonsenseQA	19	20.9 (+1.9)	38.2	48.7 (+10.5)
GPQA	23.7	24.6 (+0.9)	24.6	24.8 (+0.2)
MATH	2.0	2.5 (+0.5)	30.0	29.4 (-0.6)
GSM8K	1.9	2.4 (+0.5)	31.0	39.7 (+8.7)
MBPP	0.2	6.2 (+6)	19.2	22 (+2.8)
IFEval	27.2	27.7 (+0.5)	27.0	28.4 (+1.4)

¹ “Large” and “Small” mean large complexity and small complexity, respectively.

D Training Stability

We observe that controlling model complexity induces training instability with loss spikes as the model scale increases, particularly when employing extremely small initialization scales. The purple lines in Figure 9 demonstrate the training dynamics (loss and parameter norm evolution) of the 2.4B model with $\gamma = 1, \lambda = 1$, revealing severe instability that prevents convergence to small-complexity solutions. Contrastingly, Figures 4, 5 establish that sufficiently large λ values ($\lambda = 1$) enable small-complexity convergence regardless of initialization scale. This insight motivates our strategy: combining large λ with critical initialization scaling. Inspired by the initialization practices of GPT-2 and DeepSeek-V3, we adopt $\gamma = 0.58$ in our 2.4B model’s training. As shown in Figure 9 Table 7, this configuration achieves stabilized training and better performance.

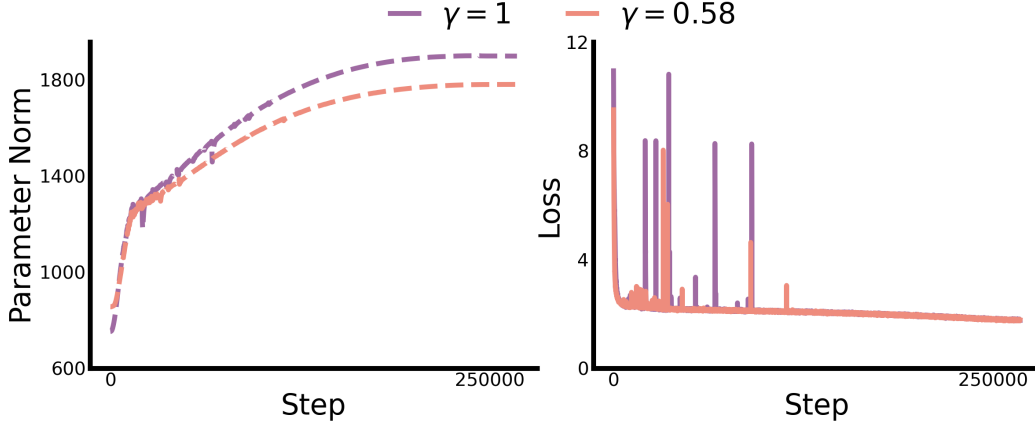


Figure 9: The dynamics of parameter norm (left) and loss (right) of 2.4B model with $\gamma = 1, \lambda = 1$ (purple) and $\gamma = 0.58, \lambda = 1$ (orange).

Table 5: Evaluation of 2.4B models with $\gamma = 1$ and $\gamma = 0.58$.

γ	1	0.58
MMLU	65.4	64.4
MMLU-Pro	30.1	30.1
BBH	45.0	43.7
ARC-C	59.0	60.7
TruthfulQA	64.1	61.7
WinoGrande	78.0	76.9
HellaSwag	74.6	75.4
AGIEval-EN	32.2	35.6
OpenBookQA	40.8	43.4
CommonsenseQA	77.7	76.9
GPQA	29.5	31.7
MATH	33.3	35.3
GSM8K	62.0	63.8
MBPP	21.6	22.4
IFEval	42.2	40.9
Average	50.4	50.9

E Evaluation of the 180M models

Table 6 comprehensively evaluates all 180M models referenced in Section 5.2, systematically illustrating performance improvements through complexity control. Additionally, Figure 10, 11 visualizes the performance-complexity correlation across all tasks, further validating our conclusions.

Table 6: Evaluation results of the 180M models under varying $\gamma - \lambda$ configurations

λ	γ	Average	Tasks							
			ARC-C	GSM8K	TruthfulQA	WinoGrande	HellaSwag	MMLU	CMMLU	C-EVAL
0	0.1	30.0	25.5	0.5	52.7	51.6	25.3	24.4	25.6	25.5
	0.3	29.1	22.6	1.4	50.4	48.3	26.2	25.6	25.5	27.5
	0.5	31.9	26.1	2.3	47.4	51.5	33.4	30.8	28.8	32.4
	0.8	31.4	23.7	1.7	47.3	50.4	34.4	31.1	29.5	31.1
	1.0	31.8	26.1	2.4	46.7	52.6	33.3	29.5	28.0	31.3
0.1	0.1	30.0	22.9	1.3	51.3	51.1	27.0	26.1	25.4	27.7
	0.3	31.1	25.9	1.7	48.7	50.9	30.2	29.4	27.6	29.9
	0.5	32.2	27.9	2.1	48.5	50.4	34.0	30.4	30.5	34.3
	0.8	32.4	25.4	2.7	48.6	51.5	34.6	31.9	31.1	35.2
	1.0	32.7	27.8	2.8	49.2	51.9	34.3	30.3	30.1	34.8
1.0	0.1	35.4	31.1	6.1	51.1	51.6	36.1	36.2	36.9	40.3
	0.3	35.8	30.0	5.8	50.6	55.4	36.9	36.1	37.1	37.7
	0.5	35.5	28.8	5.4	51.2	55.1	36.7	35.5	36.3	38.7
	0.8	35.5	29.4	7.2	47.9	53.8	37.3	37.4	38.3	41.1
	1.0	35.2	30.1	7.2	50.8	51.2	36.3	35.2	36.6	40.3

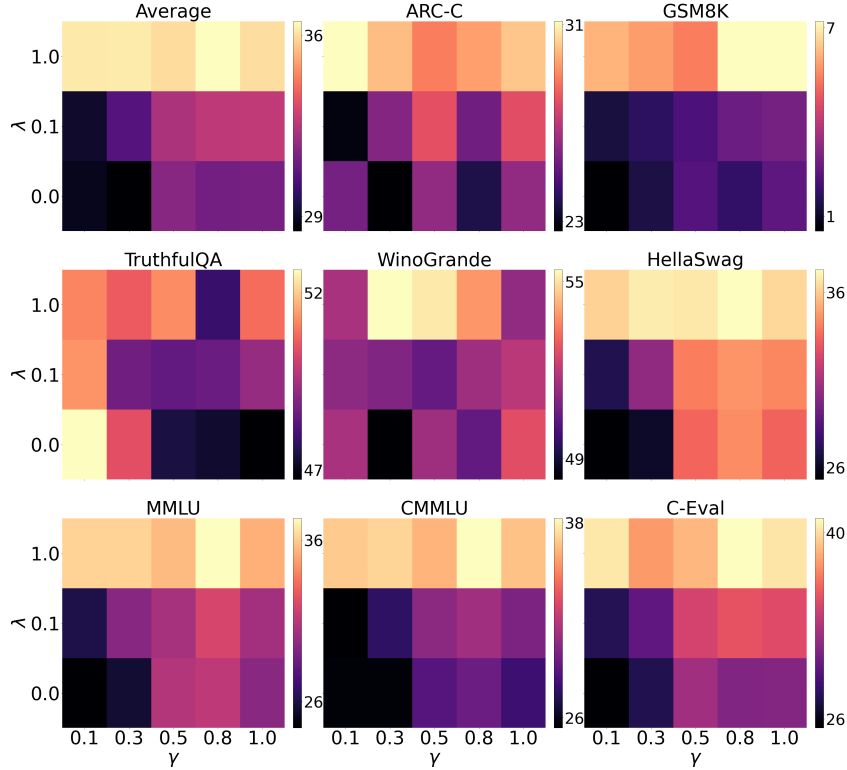


Figure 10: Performance landscape of all tasks across $\gamma - \lambda$ of the 180M models. The color indicates the score (dark: low, light: high).

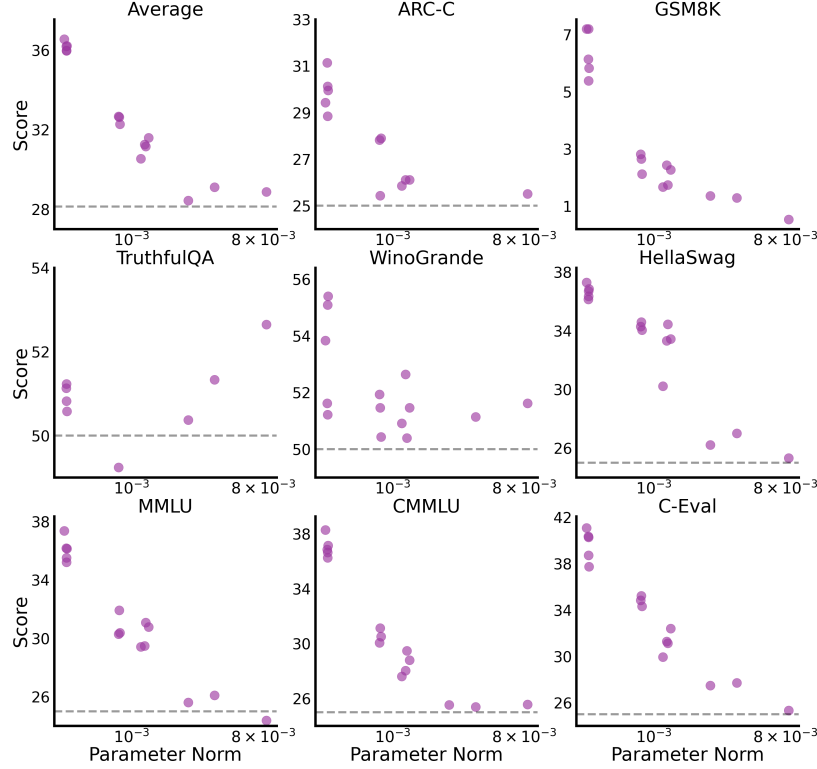


Figure 11: Score-complexity relationship with dashed lines denoting baseline performance levels.

F Model Analysis

F.1 Embedding Structure of 0.9B and 2.4B models

Figure 12 depicts the cosine similarity among 350 embedding vectors of 0.9B models and 2.4B models, with different model complexities. The results present a similar phenomenon with Figure 6, demonstrating that complexity control contributes to a focus on the association among different tokens.

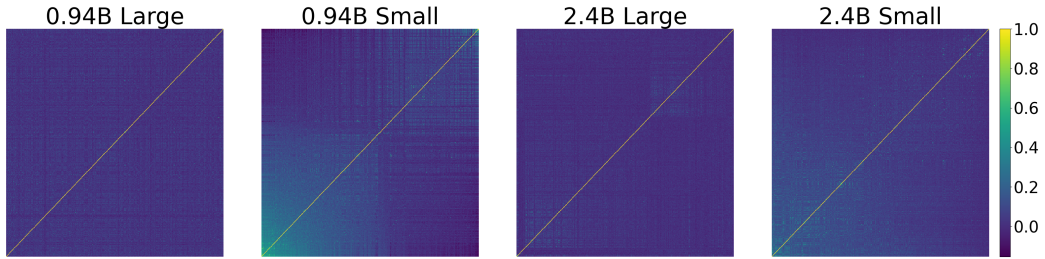


Figure 12: Cosine similarity among 350 embedding vectors which occur most frequently in training dataset under different complexities of 0.9B models and 2.4B models. The "Large" and "small" mean the large complexity and small complexity.

F.2 Attention module of 0.9B and 2.4B models

Figure 13, 14, 15, and 16 exhibit the D_c and D_s of query projection matrices and key projection matrices in the 0.9B models and 2.4B models, reveals a condensation and low-rank trend of the small-complexity models.



Figure 13: D_c of W_Q and W_K in 0.9B model's each layer under different model complexity configurations.



Figure 14: D_s of W_Q and W_K in 0.9B model's each layer under different model complexity configurations.

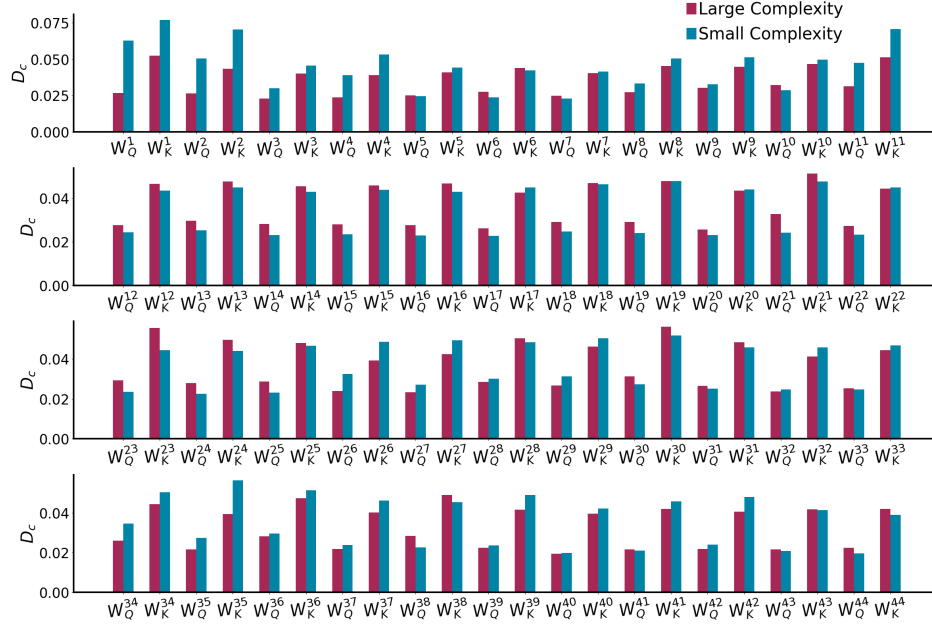


Figure 15: D_c of W_Q and W_K in 2.4B model's each layer under different model complexity configurations.

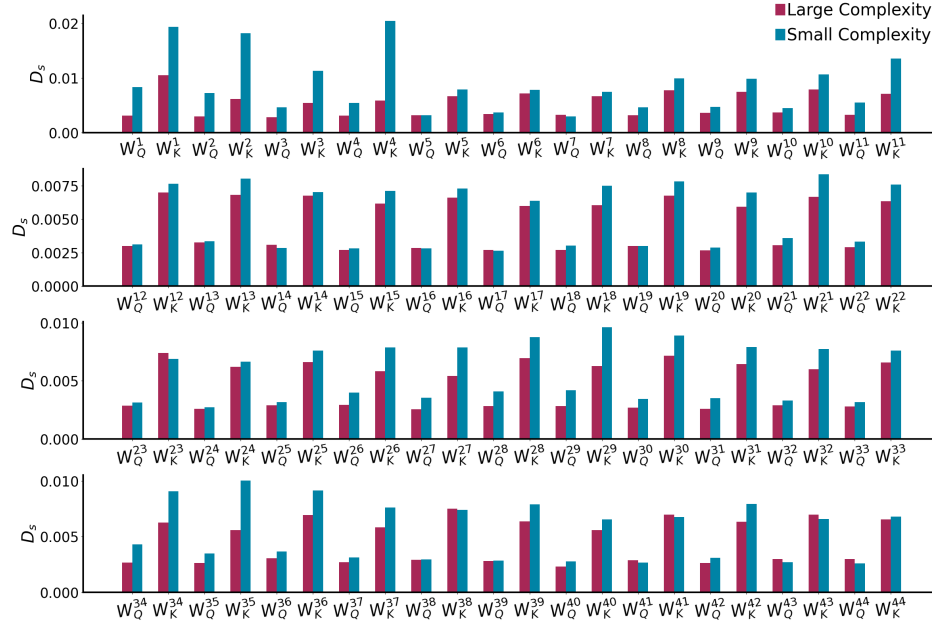


Figure 16: D_s of W_Q and W_K in 2.4B model's each layer under different model complexity configurations.