A CONDITIONING LENS ON MULTI-HEAD ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers have reshaped machine learning in large part through multi-head attention, yet the optimization role of multiple heads remains less understood. In this work, we provide a theoretical and empirical perspective on multi-head attention through the conditioning of attention-layer Jacobians. Under explicit assumptions on the statistical structure of attention-head weights, we show that increasing attention heads can improve the conditioning of the attention Jacobian, suggesting a conditioning-like mechanism for multi-head attention. This analysis motivates the hypothesis that allocating capacity to additional heads can improve trainability more effectively than simply increasing depth. We test this hypothesis across a range of transformer architectures and domains, and find that head-rich configurations consistently exhibit better-conditioned attention Jacobians and improved optimization behaviour. Building on these findings, we redesign standard transformer models to use more heads and fewer layers, achieving reductions of 30–50% in parameter count, TFLOPs, and memory while preserving accuracy. These leaner architectures also converge in fewer training steps than commonly used transformer baselines. Overall, our results identify attention heads as an important architectural factor in transformer conditioning and provide a practical route to more efficient transformer design without degrading downstream performance.

1 INTRODUCTION

Transformers have become the dominant modeling architecture across a broad range of domains, including natural language processing (NLP) (Vaswani, 2017; Devlin et al., 2018; Zhuang et al., 2021; Zhen et al., 2022), computer vision (Dosovitskiy et al., 2020; Carion et al., 2020; Liu et al., 2021; Touvron et al., 2021), and robotics (Fu et al., 2024; Maiti et al., 2023; Salzmann et al., 2020). Central to their success is the attention mechanism, which enables models to dynamically weight input elements and capture long-range dependencies. A defining feature of modern transformers is multi-head attention, in which several attention heads operate in parallel over different representation subspaces.

Despite its ubiquity, the optimization role of multi-head attention remains poorly understood. Attention is usually motivated as a representational mechanism, yet its effect on the gradient-based optimization dynamics of transformers has received comparatively little theoretical scrutiny. In contrast, for classical architectures such as multilayer perceptrons (MLPs), optimization behavior has been fruitfully studied through frameworks such as the Neural Tangent Kernel (NTK) (Jacot et al., 2018), where the conditioning of layer Jacobians is known to influence learning dynamics (Liu et al., 2022). An analogous conditioning-based understanding of multi-head attention is still lacking.

In this work, we develop a theoretical framework, under explicit statistical assumptions on attention-head weights, that studies multi-head attention through the conditioning of attention-layer Jacobians. We show that increasing the number of attention heads can improve the conditioning of the attention Jacobian by lowering its condition number, suggesting a conditioning-like mechanism through which multi-head attention can facilitate more stable and efficient gradient-based optimization.

Guided by this insight, we investigate a fundamental architectural trade-off in transformers: depth versus number of attention heads. Through controlled parameter-matched experiments, we compare allocating additional capacity to more heads against allocating it to more layers. We empirically find that head-rich configurations exhibit better-conditioned attention Jacobians and faster convergence, requiring fewer training steps to reach comparable or better performance. This suggests that attention heads and depth play distinct, non-interchangeable roles in transformer optimization.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Building on this observation, we show that depth can often be traded for additional attention heads, yielding leaner transformer architectures that preserve accuracy while improving optimization and resource efficiency. Since each transformer layer contributes substantially more to parameter count, computation, and memory than an additional attention head, reducing depth can lead to significant savings in model size, TFLOPs, and training memory.

We validate the generality of this design principle across a wide range of models and tasks. Specifically, we modify and retrain transformer architectures for ImageNet-1k classification with ViTs, self-supervised learning with DINO (Caron et al., 2021), object detection and instance segmentation (Ali et al., 2021), language modeling on Wikitext (Merity et al., 2016), The Pile (Gao et al., 2021), and GLUE (Wang et al., 2018). Across these settings, head-enriched, depth-reduced transformers converge in fewer training steps, use fewer parameters, TFLOPs, and memory, and match or exceed the performance of their original counterparts.

Our contributions are summarized as follows:

1. We develop a theoretical framework, under explicit assumptions on the statistical structure of attention-head weights, that shows how increasing the number of heads can improve the conditioning of attention-layer Jacobians.

2. We identify a practical architectural principle: depth can often be traded for additional attention heads, improving optimization efficiency while reducing parameter count, TFLOPs, and memory usage.

3. We validate this principle across a broad range of vision and language settings, demonstrating consistent improvements in convergence and resource efficiency without degrading downstream performance.

By revisiting the role of attention heads, we show that reallocating capacity from depth to heads can yield leaner, better-conditioned, and more resource-efficient transformers.

2 Related Work

Conditioning. A substantial body of work links optimization in multilayer perceptrons (MLPs) to the conditioning of network Jacobians. Within the Neural Tangent Kernel (NTK) framework (Jacot et al., 2018), improved Jacobian conditioning leads to better-conditioned kernels and faster convergence. In particular, Liu et al. (2022) show that increasing width reduces the condition number of the network Jacobian and NTK. Related work has also studied the complementary role of depth: in linear networks, depth can act as an implicit preconditioner for stochastic gradient descent (Arora et al., 2018), while in nonlinear MLPs, appropriately normalized depth can improve loss-landscape conditioning and stabilize convergence (Agarwal et al., 2021). Together, these results show that architectural choices such as width and depth can strongly shape optimization through conditioning (Poole et al., 2016; Vardi et al., 2022). In contrast, an analogous understanding for transformers remains limited (Levine et al., 2020b;a; Petty et al., 2023; Sanford et al., 2023). Although tangent-kernel analyses have been extended to transformers (Yang, 2020), the optimization role of multi-head attention itself remains largely uncharacterized.

Multi-head attention. Multi-head attention allows a model to process a token sequence through multiple attention heads, with each head learning a potentially distinct representation of the same input sequence (Vaswani, 2017). This design is commonly understood as enabling different heads to capture different relational or contextual patterns among tokens. However, prior work has shown that this multiplicity is often redundant at inference time: many heads can be removed after training with little or no loss in performance, and in some cases only a small subset of specialized heads is needed to preserve the final predictive function (Voita et al., 2019; Michel et al., 2019). Our results provide a complementary perspective: whereas these studies examine the role of heads after training, we study their effect during pre-training. We find that increasing the number of heads improves the conditioning of the attention Jacobian, which is associated with faster optimization and fewer training iterations. This suggests a dual role for attention heads: many heads may aid optimization during training, even if only a smaller set of specialized heads is ultimately necessary for inference.

3 THEORETICAL ANALYSIS

3.1 PRELIMINARIES AND NOTATION

Transformers. We briefly review the transformer architecture (Vaswani, 2017; Dosovitskiy et al., 2020; Prince, 2023). A transformer is composed of stacked layers, also known as transformer blocks. Each layer is formally represented as a mapping

$$\mathbf{T} : \mathbb{R}^{N\times D} \to \mathbb{R}^{N\times D} \tag{1}$$

defined by the expression

$$\mathbf{T}(X) = \mathbf{F}(\mathbf{A}(X) + X). \tag{2}$$

The dimension N denotes the number of tokens, while D is the token embedding dimension (Prince, 2023). The component $\mathbf{F}$ represents a multi-layer perceptron (MLP), typically consisting of a single hidden layer with a residual connection, and $\mathbf{A}$ denotes the attention mechanism. By stacking such components $\mathbf{T}$ in a feedforward manner, a transformer architecture is obtained, which may additionally include a final head $\mathbf{H}$ depending on the task. For example, in classification settings, $\mathbf{H}$ corresponds to the final classification head, see Prince (2023) for further details. Finally, while transformer architectures commonly incorporate normalization layers such as layer normalization after attention and MLP blocks, these components are omitted here for clarity.

The self-attention mechanism (Vaswani, 2017) has since been generalized to various forms (Touvron et al., 2021; Ali et al., 2021; Yuan et al., 2022; Ding et al., 2022; Wang et al., 2020; Choromanski et al., 2021). For clarity we go through the original definition of self-attention. The self-attention mechanism $\mathbf{A}$ uses three learnable matrices, the query (Q), key (K), and value (V) matrices. Given an input sequence $X \in \mathbb{R}^{N\times D}$, the matrices are first applied as follows: $q{=}QX$, $k{=}KX$, $v{=}VX$, where $Q, K \in \mathbb{R}^{D\times d}$ and $V \in \mathbb{R}^{D\times d}$. These are then combined to produce the output of the self-attention head as follows:

$$\mathbf{A}(X) := \mathbf{softmax}\left(\frac{q\,k^\top}{\sqrt{d}}\right) v \tag{3}$$

where the softmax is applied row-wise, d is known as the head dimension and the scaling $\frac{1}{\sqrt{d}}$ is so that the softmax values don't saturate (Vaswani, 2017). In general, Equation (3) computes the query-value product by a dot-product $qk^\top$. However, more general forms of attention exist that replace the dot-product with a general similarity measure ϕ and thus compute an attention block using $\phi(q,k)$: $\mathbf{A}(X) := \mathbf{softmax}(\phi(q,k))\, v$. In Section 4, we show our insights hold for such forms of attention. In this paper, our theoretical claims will be given in the context of self-attention so that we can provide concrete formulas. However, our insights hold true for general forms of attention as we will see in the experiments Section 4.

Multi-head attention. Equation (3) defines the formula for one attention head within a transformer block. In general, transformers use multiple heads leading to multi-head attention. Given an integer $h > 0$, known as the number of heads, we have an attention matrix $\mathbf{A}_i$ for each $1 \le i \le h$, each of dimension $N \times d$. Their outputs are then concatenated together to form a block of attention matrices

$$[\mathbf{A}_1, \cdots, \mathbf{A}_h] \tag{4}$$

where each $\mathbf{A}_i$ is known as an attention head. Note that as each head has dimension $N \times d$ and there are h heads we have that the block $[\mathbf{A}_1, \cdots, \mathbf{A}_h] \in \mathbb{R}^{N\times hd}$. Applying a projection matrix $\mathbf{P} \in \mathbb{R}^{hd\times D}$ we obtain multi-head attention

$$\mathbf{MA} = [\mathbf{A}_1, \cdots, \mathbf{A}_h] \cdot \mathbf{P} \tag{5}$$

giving $\mathbf{MA} \in \mathbb{R}^{N\times D}$. It's often the case that $d = \frac{D}{h}$ and thus h must divide D. With this relation between the head dimension d and the number of heads h, we see that $[\mathbf{A}_1, \cdots, \mathbf{A}_h] \in \mathbb{R}^{N\times D}$. In this paper, particularly in Section 3, we will often consider cases where the head dimension d does not depend on the number of heads h so that the equation $d = \frac{D}{h}$ does not hold. In such cases, after we fix d, we will have that $[\mathbf{A}_1, \cdots, \mathbf{A}_h] \in \mathbb{R}^{N\times hd}$ and as h gets larger the number of columns of $[\mathbf{A}_1, \cdots, \mathbf{A}_h]$, and the number of rows of $\mathbf{P}$, gets larger with each attention head $\mathbf{A}_i$ having a fixed dimension of $N \times d$.

The attention head of a layer in a transformer $\mathbf{A}(X) \in \mathbb{R}^{N\times d}$ has parameters given by those parameters in X from the previous layer and those given by Q, K and V that define $\mathbf{A}(X)$, see Equation (3). Our work will consider the Jacobian of $\mathbf{A}(X)$ with respect to the parameters within the layer of $\mathbf{A}(X)$, namely Q, K and V. Therefore, when we speak of the Jacobian of $\mathbf{A}(X)$ it will be with respect to Q, K, V. We will denote this Jacobian by $\mathbf{J}(\mathbf{A}(X))$ and note that it is defined by

$$\mathbf{J}(\mathbf{A}(X)) = \left[\frac{\partial \mathbf{A}(X)}{\partial Q}, \frac{\partial \mathbf{A}(X)}{\partial K}, \frac{\partial \mathbf{A}(X)}{\partial V}\right]^\top \tag{6}$$

where each of $\frac{\partial \mathbf{A}(X)}{\partial Q}$, $\frac{\partial \mathbf{A}(X)}{\partial K}$ and $\frac{\partial \mathbf{A}(X)}{\partial V}$ has dimension $(Nd)\cdot(Dd)$. Therefore, $J(\mathbf{A}(X))$ has dimension $(3Dd)\times(Nd)$.

Matrix notations. Given a matrix $M \in \mathbb{R}^{m\times n}$ we denote the vectorization of M by $\mathbf{vec}(M) \in \mathbb{R}^{mn\times 1}$ (Magnus & Neudecker, 2019). Note that for such a matrix there is a transformation $T_{mn} \in \mathbb{R}^{mn\times mn}$ such that $T_{mn}\mathbf{vec}(M) = \mathbf{vec}(M^\top)$ where $M^\top$ denotes the transpose of M. The matrix T_{mn} is known as a commutation matrix and is a permutation matrix (Magnus & Neudecker, 2019). The maximum singular value of a matrix M will be denoted by $\sigma_{\max}(M)$ and the minimum singular value by $\sigma_{min}(M)$. We will use the standard terminology SVD to denote the singular value decomposition of a matrix. Given a vector $z \in \mathbb{R}^n$ the notation $||z||_2$ will denote the vector 2-norm of z. Finally we will need the notion of a $\limsup$ which we remind the reader is the largest value that a sequence gets arbitrarily close to infinitely often, capturing its long-term upper bound behavior (Dym, 2004).

Condition number. The condition number of a matrix is the ratio of its largest to smallest singular values. Given a $n \times m$ matrix M, it is defined as $\kappa(M) := \sigma_{\max}(M) \,/\, \sigma_{\min}(M)$. It is well-defined, i.e. not infinite, when all singular values are positive. Since $\sigma_{\max}(M) \geq \sigma_{\min}(M)$, the condition number satisfies $\kappa(M) \geq 1$.

3.2 Main Theoretical Result

In this section we provide our main theorem of the paper. For this theorem, as remarked in Section 3.1, we assume the attention head dimension d is fixed and does not depend on the number of heads h so that the equality $d = \frac{D}{h}$ does not hold.

Theorem 3.1. *Let $\mathbf{MA} \in \mathbb{R}^{N\times D}$ denote a multi-head attention layer of a transformer block as in Equation (4) built via h attention heads $\mathbf{A}_i \in \mathbb{R}^{N\times d}$ for $1 \leq i \leq h$ and a projection $\mathbf{P} \in \mathbb{R}^{hd\times D}$. Assume, uniformly in $h > 0$, there exists constants $C_1, C_2 > 0$ and $C_3, C_4 > 0$ such that*

$$0 < C_1 \leq \sigma_{\min}(\mathbf{A}_i) \leq \sigma_{\max}(\mathbf{A}_i) \leq C_2 \text{ for } 1 \leq i \leq h \tag{7}$$

$$0 < C_3 \leq \sigma_{\min}(\mathbf{P}) \leq \sigma_{\max}(\mathbf{P}) \leq C_4 \tag{8}$$

and the columns of $[\mathbf{A}_1 \ldots, \mathbf{A}_h]$ are independent, mean zero, Gaussian distributed entries with covariance Σ satisfying $0 < \lambda_{\min}(\Sigma) \leq \lambda_{\max}(\Sigma)$. Then with probability approaching 1 we have the following asymptotic bound

$$\limsup_{h\to\infty} \kappa(\mathbf{J}(\mathbf{MA})) \leq \sqrt{\frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)}}. \tag{9}$$

Furthermore, if we assume that the columns of $[\mathbf{A}_1, \ldots, \mathbf{A}_h]$ are isotropic, so that $\Sigma = \sigma^2 I_N$, for some $\sigma > 0$, then with probability approaching 1 we have

$$\kappa(\mathbf{J}(\mathbf{MA})) = 1 + \mathcal{O}\left(\sqrt{\frac{N}{hd}}\right) \text{ as } h \to \infty. \tag{10}$$

In particular, for large h we have that $\kappa(\mathbf{J}(\mathbf{MA})) \approx 1$.

Remark 3.2. The assumptions in Theorem 3.1 should be interpreted as an idealized model for isolating the conditioning effect of multiple heads, rather than as an exact description of trained transformers. The bounded singular value assumptions in Equations (7) and (8) are nondegeneracy conditions: they rule out collapsed heads or exploding projections as the number of heads grows. For any fixed

finite number of heads, such constants exist whenever the head maps and output projection are nondegenerate. The uniform-in-h formulation is the corresponding asymptotic assumption. The covariance condition formalizes head diversity. In the anisotropic case, the limiting conditioning is controlled by $\lambda_{\max}(\Sigma)/\lambda_{\min}(\Sigma)$, while the isotropic case yields the clean limit $\kappa(\mathbf{J}(\mathbf{MA})) \to 1$. Thus, the theorem does not require trained heads to be exactly isotropic. Rather, it predicts improved conditioning when the concatenated head representations remain nondegenerate and sufficiently diverse. The Gaussian and independence assumptions are in the spirit of standard infinite-width and NTK analyses (Jacot et al., 2018; Nguyen et al., 2021; Nguyen & Mondelli, 2020). We discuss the assumptions empirically in Section A.1.2.

Increasing head dimension. Theorem 3.1 studies the regime where the number of heads h increases while the head dimension d is fixed. An analogous result holds when d increases with h fixed, and is proved in Theorem A.3. These two regimes have different architectural interpretations: increasing h adds more attention patterns, while increasing d enriches the representation within a fixed set of heads.

Remark 3.3. Theorem 3.1 is a conditioning result for the attention-layer Jacobian. It does not by itself provide a complete characterization of optimization dynamics under adaptive optimizers such as Adam. Rather, it suggests a mechanism by which increasing heads can improve trainability. This mirrors related observations for MLPs, where increased width can yield better-conditioned Jacobians and improved optimization through the NTK lens (Liu et al., 2022). We therefore treat the theorem as a mechanistic prediction and test its optimization consequences empirically in Section 4.

Trading depth for heads. A transformer layer carries a large parameter and memory budget, largely due to its MLP block, whereas increasing the number of attention heads within a layer incurs a comparatively modest cost. Our analysis suggests that additional heads can improve attention-Jacobian conditioning, and our experiments show that this is accompanied by faster convergence in practice. This motivates the hypothesis that, under a fixed compute or training-budget regime, some depth can be traded for additional heads without harming final accuracy. Guided by this principle, we redesign standard architectures by increasing the number of heads while reducing depth, obtaining leaner models that preserve performance while reducing parameters, memory, and FLOPs. A complete theoretical characterization of when this depth-head trade-off is valid remains open. Our results in Section 4 provide empirical evidence for the trade-off and suggest concrete directions for more principled transformer architecture design.

4 Experiments

We perform extensive experiments with a variety of transformer-based models. Our goals are (1) to empirically verify the prediction of Theorem 3.1 about improvements in conditioning and (2) to evaluate the benefits on a variety of different transformer architectures and applications.

4.1 Image Classification with ViTs

We apply our insights from Section 3 to the problem of image classification on ImageNet-1k.

Validating the effects on conditioning. To validate Theorem 3.1, we begin with a ViT-Base (ViT-B) architecture (Dosovitskiy et al., 2020), whose standard architecture configuration is given by 12 layers, 12 heads, head dimension $d = 64$, embedding dimension 768, and MLP width 3072 (86M parameters). We first vary the number of heads while fixing the head dimension at $d = 64$, training models on ImageNet-1k with $h \in \{1, 4, 8, 12, 16, 20, 24\}$ and all other architectural components held fixed. Since the analysis also applies when fixing h and increasing d (Theorem A.3), we perform a complementary experiment with $h = 12$ and $d \in \{6, 18, 48, 64, 88, 104, 122\}$, chosen to approximately match parameter counts. Furthermore, to show that depth does not affect Theorem 3.1 we vary the number of layers from 12-17. All models are trained for 300 epochs using AdamW following Steiner et al. (2021); Dosovitskiy et al. (2020), details in Section A.2.1. Every 50 epochs, we compute the Jacobian condition number of the attention matrix, averaged across all layers, and then average this over 300 epochs of training. We ran five experiments with different random seeds. Figure 1 reports the mean over these runs. The standard deviations are smaller than the scale of the y-axis (log scale) in all cases and are therefore not visible on the plotted scale, which spans changes

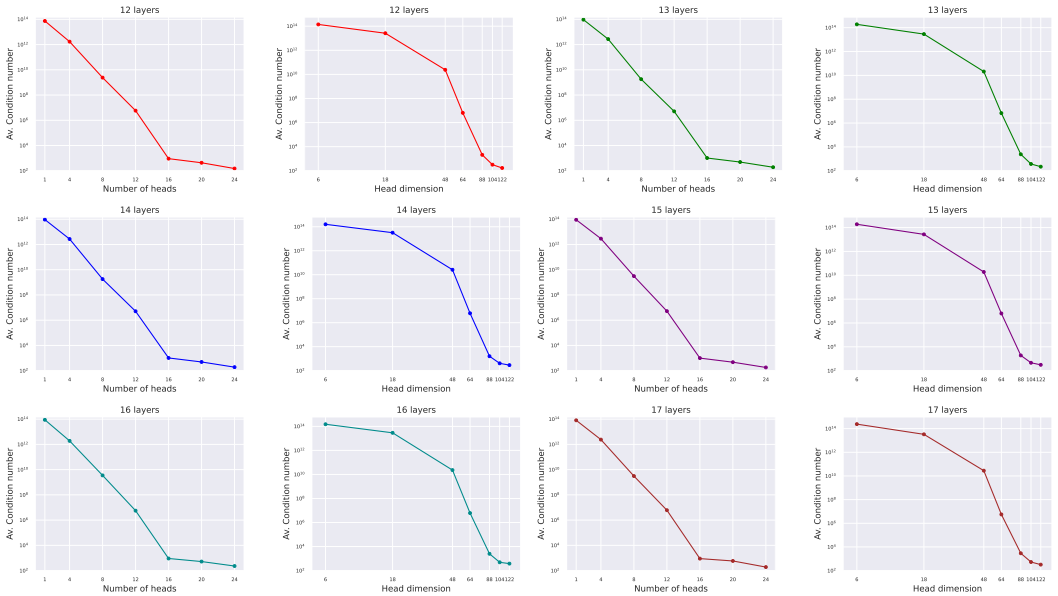


Figure 1: Empirical condition number of the attention-layer Jacobian in ViT-B models with varying numbers of heads and head dimensions, shown for layers 12-17 from left to right. In all cases, increasing either the number of heads or the head dimension improves Jacobian conditioning, consistent with the predictions of Theorem 3.1 and Theorem A.3.

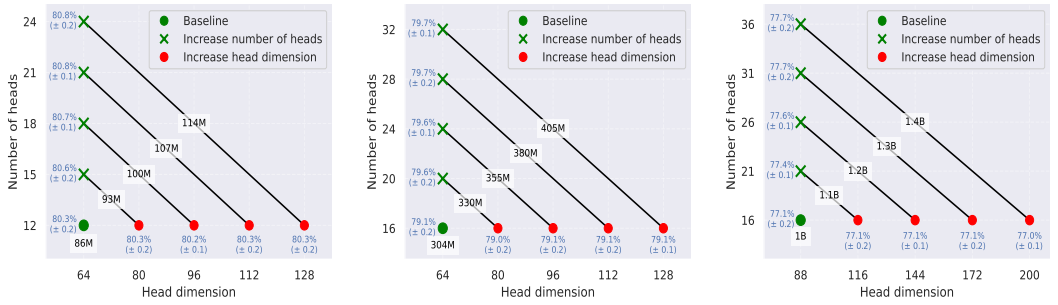


Figure 2: **Effect of increasing heads vs. head dimension in Vision Transformers.** ImageNet-1k Top-1% accuracy comparison (mean accuracy over 5 runs with standard deviations shown in brackets) for ViT-B (left), ViT-L (middle), and ViT-G (right) when increasing the number of heads (with fixed head dimension $d = 64$ for ViT-B and ViT-L, and $d = 88$ for ViT-G) versus increasing head dimension (with fixed number of heads $h = 12$ for ViT-B and $h = 16$ for ViT-L and ViT-G). All models are matched for parameter count. In all cases, increasing the number of heads with fixed head dimension yields slightly higher accuracy.

of two orders of magnitude. Across training, the average condition number decreases consistently as either the number of heads h or the head dimension d increases. This trend is stable across the layers considered, supporting the predictions of Theorem 3.1 and Theorem A.3 and showing they are independent of depth. Furthermore, the condition numbers were all finite as evidenced in Figure 1. Results for larger-scale ViTs (ViT-Large and ViT-Giant) are reported in Section A.2.1.

Number of heads vs. head dimension. Theorem 3.1 and Theorem A.3 identify two ways to improve the conditioning of the attention Jacobian: increasing the number of heads h or increasing the head dimension d. Beyond conditioning, increasing h introduces a useful inductive bias: with moderate head dimensions (e.g., $64 - 128$), additional heads capture a greater diversity of attention patterns, whereas increasing d enriches individual representations without increasing pattern diversity. Consequently, fixing d and increasing h is expected to be more effective. To test this hypothesis, we

conduct experiments on ViT-B, ViT-L and ViT-G architectures trained on ImageNet-1k. For ViT-B (86 million parameters), we fix $d = 64$ and vary h while keeping all other architectural components unchanged, and compare against models with fixed $h = 12$ and increasing d. We repeat the same comparison for ViT-L (304 million parameters), starting from the standard configuration with $d = 64$ and $h = 16$ and ViT-G (1 billion parameters), starting from the standard configuration with $d = 88$ and $h = 16$. We ran five trials for each case with different random seeds. In all cases, models are matched for parameter count, and increasing the number of heads consistently yields slightly higher accuracy (Figure 2).

Implementation. Motivated by the preceding analysis, throughout the remainder of the paper we increase the number of heads h while keeping the head dimension d fixed. This departs from standard transformer designs, which typically enforce the relation $d = D/h$, where D denotes the token embedding dimension (see Section 3.1). In our architectures this constraint is relaxed, and it holds only for the baseline models used for comparison.

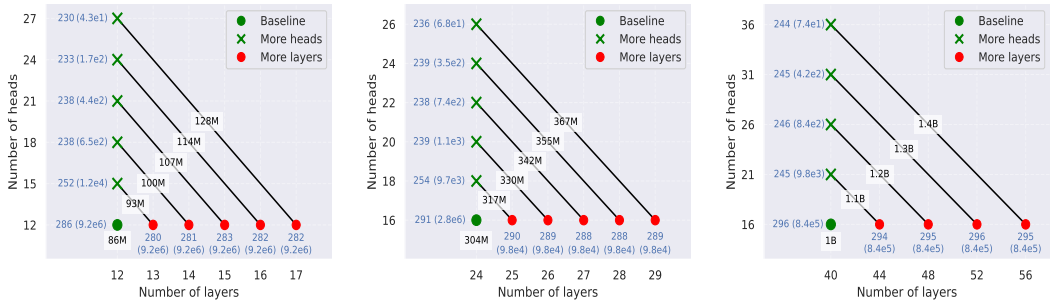


Figure 3: **Effect of increasing heads vs. depth in Vision Transformers.** Number of training epochs required to reach baseline Top-1% accuracy for ViT-B (left), ViT-L (middle), and ViT-G (right), averaged over five random seeds. We compare increasing heads at fixed head dimension ($d = 64$ for ViT-B and ViT-L, $d = 88$ for ViT-G) with increasing depth, using parameter-matched models. Increasing heads consistently reaches baseline accuracy in 10-20% fewer epochs than increasing depth. Bracketed values show average attention-Jacobian condition numbers: increasing heads at fixed depth reduces the condition number, while increasing depth at fixed head count leaves it largely unchanged.

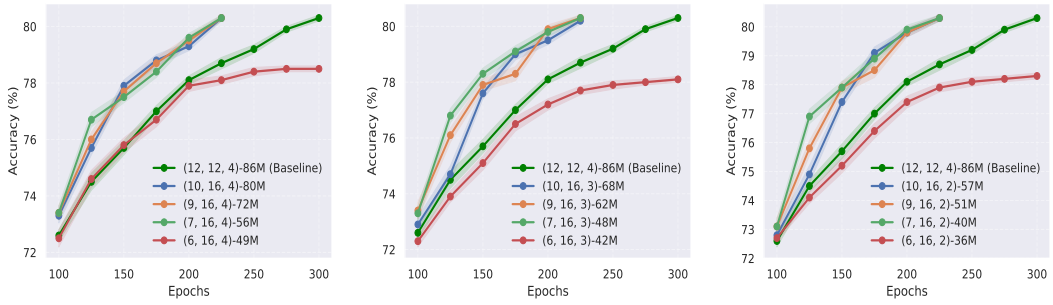


Figure 4: **Trading depth and MLP width for heads in ViT-B.** We compare the baseline ViT-B architecture (12 layers, 12 heads, MLP ratio 4) with shallower variants using 16 attention heads and MLP ratios of 4 (left), 3 (middle), and 2 (right). All models are trained on ImageNet-1k, and curves show the mean over five random seeds with standard deviations. Across MLP ratios, several head-enriched, depth-reduced configurations match the baseline accuracy while using fewer parameters and converging in approximately 10-20% fewer epochs. Legend entries indicate each model as (layers, heads, MLP ratio), with the corresponding parameter count shown alongside.

Increasing heads vs. increasing depth. Since increasing the number of heads also introduces additional parameters, we isolate this effect by comparing head increases against depth increases under

matched parameter budgets. We conduct this comparison on ViT-B, ViT-L, and ViT-G architectures (Dosovitskiy et al., 2020), measuring the number of training epochs required to reach the baseline ImageNet-1k Top-1% accuracy (80.3 for ViT-B, 79.1 for ViT-L, 77.1 for ViT-G) when increasing either the number of heads or the depth. All models are trained for 300 epochs using AdamW, following Steiner et al. (2021); Dosovitskiy et al. (2020), with full details provided in Section A.2.1. We run five trials with different random seeds and report the mean number of epochs required to reach the baseline accuracy, the standard deviation is between 2-4 epochs in all cases (see Figure 13 for a plot with standard deviations). As shown in Figure 3, across all model scales and parameter-matched comparisons, increasing the number of heads consistently reaches the baseline accuracy in 10-20% fewer training epochs than increasing depth.

Effect of MLP width. In transformer architectures, the hidden-layer width of the MLP blocks is typically set to $4D$, where D denotes the embedding dimension, and the factor 4 is commonly referred to as the MLP ratio. As an alternative means of increasing model width, we vary the MLP ratio and analyze its effect on performance. In contrast to increasing the number of attention heads, adjusting the MLP width yields only limited performance gains. A detailed ablation is provided in Section A.2.1.

Analysis of leaner transformer configurations. Figure 3 shows that increasing the number of attention heads leads to faster convergence than both the baseline ViT architecture and parameter-matched depth increases, albeit with additional parameters. Since Theorem 3.1 predicts improved Jacobian conditioning with more heads, we ask whether these optimization benefits can be retained while reducing model size. To test this, we increase the number of heads and then reduce depth and MLP width. Starting from ViT-B (Dosovitskiy et al., 2020), which uses 12 heads per layer, Figure 1 shows that increasing to 16 heads substantially lowers the average attention-Jacobian condition number. We therefore train ViT-B variants with 16 heads and fixed head dimension $d = 64$, reducing the depth from 12 to 6 layers and the MLP ratio from 4 to 2, while keeping the embedding dimension fixed at 768. All models are trained on ImageNet-1k for 300 epochs using AdamW with five random seeds; details are given in Section A.2.1. As shown in Figure 4, for MLP ratios 4, 3, and 2, several 16-head models with fewer than 12 layers reach baseline accuracy in fewer epochs. In particular, the 7-layer, 16-head ViT-B with MLP ratio 2 has only 40M parameters, matches baseline accuracy, and converges in approximately 15-20% fewer training epochs. Reducing depth below 7 layers leads to suboptimal performance and fails to reach baseline accuracy. Similar results for ViT-L and ViT-G are reported in Section A.2.1.

Other vision transformers. We extend our analysis to XCiT (Ali et al., 2021) (XCiT-M and XCiT-L), which uses cross-covariance attention, and DaViT (Ding et al., 2022) (DaViT-B, DaViT-L, and DaViT-G), which combines spatial and channel attention. To improve attention-Jacobian conditioning, we increase the number of heads at fixed head dimension, then reduce depth and MLP width. The resulting models have fewer parameters, lower per-iteration memory and computational costs, and faster training, while largely preserving accuracy. All models are trained on ImageNet-1k with AdamW, see Section A.2.1 for details and Table 1 for results. DaViT's four-stage architecture uses four-tuples to specify per-stage depth and head counts. Giant-scale results include only ViT and DaViT, as XCiT has no Giant variant.

4.2 Downstream Vision Tasks

We assess transfer beyond ImageNet classification using the lean ViT-B and XCiT-M architectures from Table 1. After DINO self-supervised pretraining on ImageNet-1k, we evaluate their representations with k-nearest-neighbor (k-NN) and linear probing Caron et al. (2021). We also evaluate XCiT-M on object detection and instance segmentation. Results are in Sections A.2.2 and A.2.3.

4.3 Language modeling.

GPT on WikiText and The Pile. We apply our conditioning-guided design to GPT-2 on WikiText (Merity et al., 2016) and The Pile (Gao et al., 2021), using five random seeds. The WikiText baseline has 303M parameters (20 layers, 16 heads of dimension 64, MLP ratio 4, embedding dimension 1024, context length 256), the billion-parameter Pile baseline has 24 layers, 16 heads of dimension 128, MLP ratio 4, embedding dimension 2048, and context length 2048. At fixed head dimension, we

Table 1: Lean ViT, XCiT, and DaViT configurations compared with their baselines. Each model is parameterized by (depth, heads, and MLP ratio). For DaViT, depth and heads are given per stage. Values in brackets denote stds over 5 random seeds for Top-1 accuracy. Lean configurations reduce parameters, per-iteration training time, memory, and TFLOPs while preserving Top-1 accuracy.

Model	Config.	Top-1 (%)	Params	Time (s)	Mem. (GB)	TFLOPs
ViT-B	$(12, 12, 4)$ baseline	$80.3\,(\pm 0.2)$	86M	1.18	270.5	107.46
ViT-B	$(\mathbf{7}, \mathbf{16}, \mathbf{2})$	$\mathbf{80.6}\,(\pm \mathbf{0.2})$	**40M**	**0.68**	**145.6**	**50.32**
ViT-L	$(24, 16, 4)$ baseline	$79.1\,(\pm 0.2)$	304M	2.07	712.8	377.62
ViT-L	$(\mathbf{13}, \mathbf{20}, \mathbf{2})$	$\mathbf{79.2}\,(\pm \mathbf{0.1})$	**124M**	**1.08**	**343.2**	**156.78**
ViT-G	$(40, 16, 48/11)$ baseline	$77.1\,(\pm 0.2)$	1B	2.56	971.2	1242.95
ViT-G	$(\mathbf{21}, \mathbf{24}, \mathbf{2})$	$\mathbf{77.5}\,(\pm \mathbf{0.2})$	**351M**	**1.58**	**556.8**	**435.47**
XCiT-M	$(24, 8, 4)$ baseline	$84.4\,(\pm 0.2)$	84M	3.87	560.8	95.72
XCiT-M	$(\mathbf{13}, \mathbf{16}, \mathbf{2})$	$\mathbf{84.5}\,(\pm \mathbf{0.1})$	**47M**	**1.23**	**359.2**	**54.14**
XCiT-L	$(24, 16, 4)$ baseline	$84.9\,(\pm 0.2)$	189M	4.51	830.4	213.09
XCiT-L	$(\mathbf{13}, \mathbf{20}, \mathbf{2})$	$\mathbf{84.9}\,(\pm \mathbf{0.2})$	**99M**	**1.78**	**503.2**	**109.26**
DaViT-B	$((1, 1, 9, 1), (4, 8, 16, 32), 4)$ baseline	$84.6\,(\pm 0.1)$	87M	1.94	466.4	93.38
DaViT-B	$((\mathbf{1}, \mathbf{1}, \mathbf{5}, \mathbf{1}), (\mathbf{4}, \mathbf{8}, \mathbf{20}, \mathbf{32}), \mathbf{2})$	$\mathbf{84.6}\,(\pm \mathbf{0.2})$	**45M**	**1.48**	**287.2**	**45.71**
DaViT-L	$((1, 1, 9, 1), (6, 12, 24, 48), 4)$ baseline	$84.8\,(\pm 0.2)$	197M	2.07	702.4	208.73
DaViT-L	$((\mathbf{1}, \mathbf{1}, \mathbf{5}, \mathbf{1}), (\mathbf{6}, \mathbf{12}, \mathbf{28}, \mathbf{48}), \mathbf{2})$	$\mathbf{84.5}\,(\pm \mathbf{0.2})$	**100M**	**1.60**	**425.6**	**99.29**
DaViT-G	$((1, 1, 12, 3), (12, 24, 48, 96), 4)$ baseline	$82.6\,(\pm 0.3)$	1.4B	4.17	893.2	1178.16
DaViT-G	$((\mathbf{1}, \mathbf{1}, \mathbf{7}, \mathbf{2}), (\mathbf{6}, \mathbf{12}, \mathbf{52}, \mathbf{98}), \mathbf{2})$	$\mathbf{83.1}\,(\pm \mathbf{0.2})$	**621M**	**2.52**	**625.3**	**531.47**

Table 2: Leaner GPT-2 configurations on WikiText and The Pile compared with their baselines. Models are parameterized by (depth, number of attention heads, MLP ratio).

Dataset	Model	Perplexity	Params	Time (s)	Memory (GB)	TFLOPs
WikiText	GPT-2 $(20, 16, 4)$ baseline	$22.2\,(\pm 1.1)$	303M	1.71	404.0	254.76
WikiText	**GPT-2** $(\mathbf{11}, \mathbf{20}, \mathbf{2})$	$\mathbf{20.0}\,(\pm \mathbf{0.9})$	**155M**	**0.98**	**276.4**	**133.29**
The Pile	GPT-2 $(24, 16, 4)$ baseline	$9.0\,(\pm 0.8)$	1.3B	3.51	333.2	2378.92
The Pile	**GPT-2** $(\mathbf{13}, \mathbf{24}, \mathbf{2})$	$\mathbf{7.4}\,(\pm \mathbf{1.2})$	**652M**	**2.78**	**197.6**	**1277.39**

increase the head counts to 20 and 24, respectively, improving attention-Jacobian conditioning while reducing depth and MLP width. The lean models improve perplexity while reducing parameters, per-iteration time, memory, and TFLOPs (Table 2). Training details and further analysis are in Section A.2.4. **Commonsense reasoning results on HellaSwag, PIQA, and WinoGrande are also in Section A.2.4.** BERT results on GLUE are reported in Section A.2.4.

5 Limitations

While our theoretical analysis establishes that increasing the number of attention heads improves the conditioning of attention Jacobians, an understanding of how this conditioning interacts with optimization dynamics remains an open problem. Consequently, our explanation of the depth–head trade-off is primarily empirical. In addition, our main theorem (Theorem 3.1) relies on strict statistical assumptions which do not generally hold throughout training, further underscoring the need for more refined theoretical analyses that can deal with the actual optimization dynamics.

6 Conclusion

In this work, we reexamined the role of multi-head attention in transformer architectures. Our theoretical analysis shows that increasing the number of attention heads improves the conditioning of the attention Jacobian. This insight motivates the hypothesis that, when the attention Jacobian is well-conditioned, transformers may rely less on depth than is commonly assumed for effective optimization. Across a diverse set of transformer architectures, we validated this hypothesis empirically: shallower models with more attention heads achieve performance comparable to their deeper baselines while requiring substantially fewer parameters, reduced memory usage, and lower computational cost.

AI USE STATEMENT

Generative AI tools were used only to edit the manuscript's grammar, spelling, and wording, and to assist with visualizing results for submission. They were not used to generate synthetic data, design methods, implement code, translate text, clean data, conduct qualitative analysis, or interpret results. The authors checked every AI-assisted passage and figure against the underlying work. No experimental result, table entry, derivation, or scientific claim was produced by generative AI. We take full responsibility for the final manuscript and its associated artifacts.

REFERENCES

Naman Agarwal, Pranjal Awasthi, and Satyen Kale. A deep conditioning treatment of neural networks. In *Algorithmic Learning Theory*, pp. 249–305. PMLR, 2021.

Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International conference on machine learning*, pp. 244–253. PMLR, 2018.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Alex Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *International Conference on Learning Representations (ICLR)*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Mingyu Ding, Bin Xiao, Noel Codella, Ping Luo, Jingdong Wang, and Lu Yuan. Davit: Dual attention vision transformers. In *European conference on computer vision*, pp. 74–92. Springer, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Clive Dym. *Principles of mathematical modeling*. Elsevier, 2004.

Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 910–919, 2024.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Aadi Thite, Eric Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2021.

Jonas Geiping and Tom Goldstein. Cramming: Training a language model on a single gpu in one day. In *International Conference on Machine Learning*, pp. 11117–11143. PMLR, 2023.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Yoav Levine, Noam Wies, Or Sharir, Hofit Bata, and Amnon Shashua. The depth-to-width interplay in self-attention. *arXiv preprint arXiv:2006.12467*, 2020a.

Yoav Levine, Noam Wies, Or Sharir, Hofit Bata, and Amnon Shashua. Limits to depth efficiencies of self-attention. *NeurIPS*, 33:22640–22651, 2020b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59: 85–116, 2022.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Jan R Magnus and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 2019.

Abhisek Maiti, Sander Oude Elberink, and George Vosselman. Transfusion: Multi-modal fusion network for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6536–6546, 2023.

Matterport. Mask r-cnn implementation, 2017. URL `https://github.com/matterport/Mask_RCNN`.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

Quynh Nguyen, Marco Mondelli, and Guido F Montufar. Tight bounds on the smallest eigenvalue of the neural tangent kernel for deep relu networks. In *International Conference on Machine Learning*, pp. 8119–8129. PMLR, 2021.

Quynh N Nguyen and Marco Mondelli. Global convergence of deep networks with one wide layer followed by pyramidal topology. *Advances in Neural Information Processing Systems*, 33: 11961–11972, 2020.

Jackson Petty, Sjoerd van Steenkiste, Fei Sha, Ishita Dasgupta, Dan Garrette, and Tal Linzen. The impact of depth and width on transformer language model generalization. *openreview*, 2023.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. *Advances in neural information processing systems*, 29, 2016.

Simon JD Prince. *Understanding deep learning*. MIT press, 2023.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. *arXiv preprint arXiv:2001.03093*, 2, 2020.

Clayton Sanford, Daniel J Hsu, and Matus Telgarsky. Representational strengths and limitations of transformers. *NeurIPS*, 36:36677–36707, 2023.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Gal Vardi, Gilad Yehudai, and Ohad Shamir. Width is less important than depth in relu neural networks. In *Conference on learning theory*, pp. 1249–1281. PMLR, 2022.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 5797–5808, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):6575–6586, 2022.

Q Zhen, W Sun, H Deng, D Li, Y Wei, B Lv, J Yan, L Kong, and Y Zhong. cosformer: rethinking softmax in attention. In *International Conference on Learning Representations*, 2022.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized bert pre-training approach with post-training. In *Proceedings of the 20th chinese national conference on computational linguistics*, pp. 1218–1227, 2021.

A Appendix

A.1 Theoretical Results

A.1.1 Proof of results from Section 3

In this section we give the proof of Theorem 3.1. In order to prove Theorem 3.1, we will need an auxiliary theorem that provides bounds on the singular values of the Jacobian of a multi-head attention layer which is given by Theorem A.2.

The following lemma shows how the Jacobian of $\mathbf{MA}$ can be determined in terms of the Jacobian of each head $\mathbf{A}_i$ and the projection matrix $\mathbf{P}$.

Lemma A.1. *Let $\mathbf{MA} \in \mathbb{R}^{N\times D}$ denote a multi-head attention layer of a transformer block as in Equation* (4)*. Then*

$$\mathbf{J}(\mathbf{MH}) = \begin{bmatrix} \mathrm{Diag}(\mathbf{J}(\mathbf{A_1}), \ldots, \mathbf{J}(\mathbf{A_h}))(\mathbf{P} \otimes I_N) \\ ([\mathbf{A}_1, \ldots, \mathbf{A}_h]^\top \otimes I_D) \end{bmatrix}$$

with $\mathbf{J}(\mathbf{MH}) \in \mathbb{R}^{(3hDd+Dhd)\times ND}$

Proof. We break the proof up into a number of steps. We use two standard vectorized Kronecker identities. For arbitrary matrices A, B, C:

$$\mathrm{vec}(AB) \;=\; (I \otimes A)\,\mathrm{vec}(B), \qquad \mathrm{vec}(AB) \;=\; (B^\top \otimes I)\,\mathrm{vec}(A). \tag{11}$$

Step 1: Vectorize the multi-head output. Let $\mathbf{A}_{\mathrm{cat}} = [\mathbf{A_1}, \ldots, \mathbf{A}_h]$. By Equation (11),

$$y \;=\; \mathrm{vec}(\mathbf{A}_{\mathrm{cat}}\mathbf{P}) \;=\; (\mathbf{P}^\top \otimes I_N)\,\mathrm{vec}(\mathbf{A}_{\mathrm{cat}}) \;=\; (I_D \otimes \mathbf{A}_{\mathrm{cat}})\,\mathrm{vec}(\mathbf{P}). \tag{12}$$

The first equality will be used for derivatives w.r.t. the *head* parameters (which affect $\mathbf{A}_{\mathrm{cat}}$); the second for derivatives w.r.t. $\mathbf{P}$.

Step 2: Jacobian w.r.t. $\mathbf{P}$. From the second representation in Equation (12), $y = (I_D \otimes \mathbf{A}_{\mathrm{cat}})\,\mathrm{vec}(\mathbf{P})$. In row–Jacobian form this gives $\frac{\partial y}{\partial\,\mathrm{vec}(\mathbf{P})} = I_D \otimes \mathbf{A}_{\mathrm{cat}} \in \mathbb{R}^{(ND)\times(Dhd)}$. Transposing yields the column–Jacobian block

$$\frac{\partial\,\mathrm{vec}(\mathbf{P})}{\partial\, y} \;=\; \big(I_D \otimes \mathbf{A}_{\mathrm{cat}}\big)^\top \in \mathbb{R}^{(Dhd)\times(ND)}.$$

Step 3: Jacobian w.r.t. head parameters. From the first representation in Equation (12), $y = (\mathbf{P}^\top \otimes I_N)\,\mathrm{vec}(\mathbf{A}_{\mathrm{cat}})$. Differentiate w.r.t. the stacked head parameters $\big[\theta_1; \ldots; \theta_h\big]$. Only the i-th block $\mathbf{A}_i$ depends on θ_i, hence

$$\frac{\partial\, y}{\partial\,[\theta_1; \ldots; \theta_h]} \;=\; \big(\mathbf{P}^\top \otimes I_N\big)\,\mathrm{Diag}\big(\mathbf{J}(\mathbf{A}_1)^\top, \ldots, \mathbf{J}(\mathbf{A}_h)^\top\big) \;\in\; \mathbb{R}^{(ND)\times(3hDd)}.$$

Transposing gives the desired column–Jacobian block:

$$\frac{\partial\,[\theta_1; \ldots; \theta_h]}{\partial\, y} \;=\; \mathrm{Diag}\big(\mathbf{J}(\mathbf{A}_1), \ldots, \mathbf{J}(\mathbf{A}_h)\big)\,\big(\mathbf{P} \otimes I_N\big) \;\in\; \mathbb{R}^{(3hDd)\times(ND)}.$$

Step 4: Stack the blocks. By definition, $\theta = \big[\theta_1; \ldots; \theta_h;\ \mathrm{vec}(\mathbf{P})\big]$, so stacking the two column–Jacobian blocks from Steps 2–3 yields

$$\frac{\partial\,\theta}{\partial\, y} \;=\; \begin{bmatrix} \mathrm{Diag}\big(\mathbf{J}(\mathbf{A}_1), \ldots, \mathbf{J}(\mathbf{A}_h)\big)\,\big(P \otimes I_N\big) \\ \big(I_D \otimes \mathbf{A}_{\mathrm{cat}}\big)^\top \end{bmatrix}.$$

Step 5: Dimensions. Since $\mathbf{J}(\mathbf{A}_i) \in \mathbb{R}^{(3Dd)\times(Nd)}$, we have $\mathrm{Diag}(\mathbf{J}(\mathbf{A}_1), \ldots, \mathbf{J}(\mathbf{A}_h)) \in \mathbb{R}^{(3hDd)\times(Nhd)}$ and $\mathbf{P} \otimes I_N \in \mathbb{R}^{(Nhd)\times(ND)}$, hence the upper block is $(3hDd) \times (ND)$. Also $I_D \otimes \mathbf{A}_{\mathrm{cat}} \in \mathbb{R}^{(ND)\times(Dhd)}$, so its transpose is $(Dhd) \times (ND)$. Stacking gives a grand column–Jacobian of size $(3hDd + Dhd) \times (ND)$, as claimed. □

To establish the proof of Theorem 3.1, we first require an auxiliary result of our own. This theorem provides explicit bounds on the singular values of the Jacobian associated with the multi-head attention block, and will serve as a key technical tool in the argument that follows.

Theorem A.2. *Let $\mathbf{MA} \in \mathbb{R}^{N\times D}$ denote a multi-head attention layer of a transformer block as in Equation (4) built via h attention heads $\mathbf{A}_i \in \mathbb{R}^{N\times d}$ for $1 \le i \le h$ and a projection $\mathbf{P} \in \mathbb{R}^{hd\times D}$. Then we have that*

$$\sigma_{\min}(\mathbf{J}(\mathbf{MA})) \ge \sqrt{\sigma_{\min}(\mathbf{P})^2\left(\big(\min_{1\le i\le h}\sigma_{\min}(\mathbf{J}(\mathbf{A}_i))\big)^2 + (\sigma_{\min}([\mathbf{A}_1,\dots,\mathbf{A}_h])^2\right)} \tag{13}$$

$$\sigma_{\max}(\mathbf{J}(\mathbf{MA})) \le \sqrt{\sigma_{\max}(\mathbf{P})^2\left(\big(\max_{1\le i\le h}\sigma_{\max}(\mathbf{J}(\mathbf{A}_i))\big)^2 + (\sigma_{\max}([\mathbf{A}_1,\dots,\mathbf{A}_h])^2\right)}. \tag{14}$$

Therefore, we have the following condition number bound

$$\kappa(\mathbf{J}(\mathbf{MA})) \le \sqrt{\frac{\sigma_{\min}(\mathbf{P})^2\left(\big(\min_{1\le i\le h}\sigma_{\min}(\mathbf{J}(\mathbf{A}_i))\big)^2 + (\sigma_{\min}([\mathbf{A}_1,\dots,\mathbf{A}_h])^2\right)}{\sigma_{\max}(\mathbf{P})^2\left(\big(\max_{1\le i\le h}\sigma_{\max}(\mathbf{J}(\mathbf{A}_i))\big)^2 + (\sigma_{\max}([\mathbf{A}_1,\dots,\mathbf{A}_h])^2\right)}} \tag{15}$$

Proof. We break the proof up into a number of steps. Let

$$\mathbf{A}_{\text{cat}} = [\mathbf{A}_1,\dots,\mathbf{A}_h] \in \mathbb{R}^{N\times(hd)}, \qquad \mathbf{P} \in \mathbb{R}^{(hd)\times D}, \qquad \mathbf{MA} = \mathbf{A}_{\text{cat}}\mathbf{P} \in \mathbb{R}^{N\times D},$$

and for each head i,

$$\theta_i = \big[\text{vec}(W_{Q_i}); \text{vec}(W_{K_i}); \text{vec}(W_{V_i})\big] \in \mathbb{R}^{3Dd}, \qquad \mathbf{J}(\mathbf{A}_i) \in \mathbb{R}^{(3Dd)\times(Nd)}.$$

Let $y := \text{vec}(\mathbf{MA}) \in \mathbb{R}^{ND}$ and $\theta := [\theta_1;\dots;\theta_h;\ \text{vec}(\mathbf{P})] \in \mathbb{R}^{3hDd+Dhd}$.

Step 1: A block formula for the Jacobian. We use the standard vectorization identities, valid for arbitrary matrices A, B, C:

$$\text{vec}(AB) = (I\otimes A)\,\text{vec}(B), \qquad \text{vec}(AB) = (B^\top\otimes I)\,\text{vec}(A). \tag{16}$$

Applying Equation (16) twice gives two equivalent expressions for y:

$$y = \text{vec}(\mathbf{A}_{\text{cat}}\mathbf{P}) = (\mathbf{P}^\top\otimes I_N)\,\text{vec}(\mathbf{A}_{\text{cat}}) = (I_D\otimes\mathbf{A}_{\text{cat}})\,\text{vec}(\mathbf{P}). \tag{17}$$

Only the i-th block of $\text{vec}(\mathbf{A}_{\text{cat}}) = [\text{vec}(\mathbf{A}_1);\dots;\text{vec}(\mathbf{A}_h)]$ depends on θ_i; differentiating Equation (17) yields the Jacobian block

$$\text{Diag}\big(\mathbf{J}(\mathbf{A}_1),\dots,\mathbf{J}(\mathbf{A}_h)\big)\,(\mathbf{P}\otimes I_N)$$

Likewise, differentiating $y = (I_D\otimes\mathbf{A}_{\text{cat}})\text{vec}(\mathbf{P})$ w.r.t. $\text{vec}(\mathbf{P})$ gives the second Jacobian block $\mathbf{A}_{\text{cat}}^\top\otimes I_D$. Altogether,

$$\mathbf{J}(\mathbf{MA}) = \begin{bmatrix}\text{Diag}\big(\mathbf{J}(\mathbf{A}_1),\dots,\mathbf{J}(\mathbf{A}_h)\big)\,(\mathbf{P}\otimes I_N)\\ \big(I_D\otimes\mathbf{A}_{\text{cat}}\big)^\top\end{bmatrix} \in \mathbb{R}^{(3hDd+Dhd)\times(ND)}. \tag{18}$$

Step 2: Singular-value bounds for a vertical stack. For ease of exposition, write $\mathbf{J} := \mathbf{J}(\mathbf{MA}) = \begin{bmatrix}U\\V\end{bmatrix}$ with

$$U := \text{Diag}\big(\mathbf{J}(\mathbf{A}_1),\dots,\mathbf{J}(\mathbf{A}_h)\big)\,(\mathbf{P}\otimes I_N), \qquad V := \big(I_D\otimes\mathbf{A}_{\text{cat}}\big)^\top.$$

Then $\mathbf{J}\mathbf{J}^\top = UU^\top + VV^\top$. For any unit vector x,

$$x^\top(UU^\top+VV^\top)x = x^\top UU^\top x + x^\top VV^\top x \ge \lambda_{\min}(UU^\top) + \lambda_{\min}(VV^\top),$$

hence

$$\sigma_{\min}(\mathbf{J})^2 = \lambda_{\min}(\mathbf{J}\mathbf{J}^\top) \ge \sigma_{\min}(U)^2 + \sigma_{\min}(V)^2. \tag{19}$$

By the operator-norm triangle inequality,

$$\sigma_{\max}(\mathbf{J})^2 = \|UU^\top+VV^\top\|_2 \le \|UU^\top\|_2 + \|VV^\top\|_2 = \sigma_{\max}(U)^2 + \sigma_{\max}(V)^2. \tag{20}$$

Step 3: Bounding the factors U and V. We invoke standard singular-value facts: for any conformable A, B,

$$\sigma_{\min}(AB) \geq \sigma_{\min}(A)\,\sigma_{\min}(B), \qquad \sigma_{\max}(AB) \leq \sigma_{\max}(A)\,\sigma_{\max}(B). \tag{21}$$

For a block diagonal matrix,

$$\sigma_{\min}\big(\mathrm{Diag}(M_1,\ldots,M_h)\big) = \min_i \sigma_{\min}(M_i), \quad \sigma_{\max}\big(\mathrm{Diag}(M_1,\ldots,M_h)\big) = \max_i \sigma_{\max}(M_i). \tag{22}$$

For Kronecker products (with I denoting an identity),

$$\sigma_{\min}(A \otimes B) = \sigma_{\min}(A)\,\sigma_{\min}(B), \qquad \sigma_{\max}(A \otimes B) = \sigma_{\max}(A)\,\sigma_{\max}(B). \tag{23}$$

Applying Equation (21)–Equation (23) to U gives

$$\sigma_{\min}(U) \;\geq\; \sigma_{\min}\big(\mathrm{Diag}(\mathbf{J}(\mathbf{A}_i)^\top)\big)\,\sigma_{\min}(\mathbf{P} \otimes I_N) = \big(\min_i \sigma_{\min}(\mathbf{J}(\mathbf{A}_i))\big)\,\sigma_{\min}(\mathbf{P}),$$

$$\sigma_{\max}(U) \;\leq\; \sigma_{\max}\big(\mathrm{Diag}(\mathbf{J}(\mathbf{A}_i)^\top)\big)\,\sigma_{\max}(\mathbf{P} \otimes I_N) = \big(\max_i \sigma_{\max}(\mathbf{J}(\mathbf{A}_i))\big)\,\sigma_{\max}(\mathbf{P}).$$

For $V = (I_D \otimes \mathbf{A}_{\mathrm{cat}})^\top$, singular values are invariant under transpose and

$$\sigma_{\min}(V) = \sigma_{\min}(I_D \otimes \mathbf{A}_{\mathrm{cat}}) = \sigma_{\min}(\mathbf{A}_{\mathrm{cat}}), \qquad \sigma_{\max}(V) = \sigma_{\max}(I_D \otimes \mathbf{A}_{\mathrm{cat}}) = \sigma_{\max}(\mathbf{A}_{\mathrm{cat}}).$$

Step 4: Combine. Substituting these bounds into Equation (19)–Equation (20) yields

$$\sigma_{\min}(J) \;\geq\; \sqrt{\sigma_{\min}(\mathbf{P})^2\Big(\min_i \sigma_{\min}(\mathbf{J}(\mathbf{A}_i))\Big)^2 \;+\; \sigma_{\min}(\mathbf{A}_{\mathrm{cat}})^2}\,,$$

$$\sigma_{\max}(J) \;\leq\; \sqrt{\sigma_{\max}(\mathbf{P})^2\Big(\max_i \sigma_{\max}(\mathbf{J}(\mathbf{A}_i))\Big)^2 \;+\; \sigma_{\max}(\mathbf{A}_{\mathrm{cat}})^2}\,.$$

Dividing the second inequality by the first proves the stated condition-number bound for $\mathbf{J}(\mathbf{MA})$ and completes the proof. □

Using the above Theorem A.2 we can give the proof of Theorem 3.1.

Proof of the Theorem 3.1. We will break the proof up into a number of steps. We start by recalling some notation we have been using.

Setup and notation. Fix the per–head width d and let $h \in \mathbb{N}$, so that

$$\mathbf{A}_{\mathrm{cat}} = [\mathbf{A}_1,\ldots,\mathbf{A}_h] \in \mathbb{R}^{N\times(hd)}, \qquad \mathbf{P} \in \mathbb{R}^{(hd)\times D}, \qquad \mathbf{MA} = \mathbf{A}_{\mathrm{cat}}\mathbf{P} \in \mathbb{R}^{N\times D}.$$

For each head i, collect parameters $\theta_i \in \mathbb{R}^{3Dd}$ and define the (row) Jacobian $\mathbf{J}(\mathbf{A}_i) := \frac{\partial\,\mathrm{vec}(\mathbf{A}_i)}{\partial\,\theta_i} \in \mathbb{R}^{(Nd)\times(3Dd)}$. Let $y := \mathrm{vec}(\mathbf{MA}) \in \mathbb{R}^{ND}$ and $\theta := [\theta_1;\ldots;\theta_h;\,\mathrm{vec}(\mathbf{P})] \in \mathbb{R}^{3hDd+Dhd}$. Recall $\mathbf{J}(\mathbf{MA})$ is the Jacobian we are analyzing.

Step 1: Invoking Theorem A.2 By the Theorem 3.1 we have the following bounds

$$\sigma_{\min}(\mathbf{J}) \;\geq\; \sqrt{\sigma_{\min}(\mathbf{P})^2\Big(\min_i \sigma_{\min}(\mathbf{J}(\mathbf{A}_i))\Big)^2 \;+\; \sigma_{\min}(\mathbf{A}_{\mathrm{cat}})^2} \tag{24}$$

$$\sigma_{\max}(\mathbf{J}) \;\leq\; \sqrt{\sigma_{\max}(\mathbf{P})^2\Big(\max_i \sigma_{\max}(\mathbf{J}(\mathbf{A}_i))\Big)^2 \;+\; \sigma_{\max}(\mathbf{A}_{\mathrm{cat}})^2}\,. \tag{25}$$

Step 2: Uniform bounds on per–head Jacobians and the projection. By the assumptions Equation (7) and Equation (8), there exist $0 < C_1 \leq C_2 < \infty$ and $0 < C_3 \leq C_4 < \infty$, independent of h, such that for all i,

$$C_1 \;\leq\; \sigma_{\min}(\mathbf{J}(\mathbf{A}_i)) \;\leq\; \sigma_{\max}(\mathbf{J}(\mathbf{A}_i)) \;\leq\; C_2, \qquad C_3 \;\leq\; \sigma_{\min}(\mathbf{P}) \;\leq\; \sigma_{\max}(\mathbf{P}) \;\leq\; C_4.$$

Substituting these into Equation (24) yields

$$\sigma_{\min}(\mathbf{J}) \;\geq\; \sqrt{C_3^2C_1^2 + \sigma_{\min}(\mathbf{A}_{\mathrm{cat}})^2}\,, \qquad \sigma_{\max}(\mathbf{J}) \;\leq\; \sqrt{C_4^2C_2^2 + \sigma_{\max}(\mathbf{A}_{\mathrm{cat}})^2}\,. \tag{26}$$

Hence

$$\kappa(\mathbf{J}) \;=\; \frac{\sigma_{\max}(\mathbf{J})}{\sigma_{\min}(\mathbf{J})} \;\leq\; \sqrt{\frac{C_4^2C_2^2 + \sigma_{\max}(\mathbf{A}_{\mathrm{cat}})^2}{C_3^2C_1^2 + \sigma_{\min}(\mathbf{A}_{\mathrm{cat}})^2}}\,. \tag{27}$$

Step 3: Random–matrix control of $\sigma_{\min\,/\,\max}(\mathbf{A}_{\text{cat}})$. Recall that we assumed: the $m := hd$ columns of $\mathbf{A}_{\text{cat}}$ are independent, mean–zero, Gaussian in $\mathbb{R}^N$ with common covariance Σ satisfying $0 < \lambda_{\min}(\Sigma) \le \lambda_{\max}(\Sigma) < \infty$, and $N/m \to 0$ as $h \to \infty$ (with d fixed). Standard results for tall random matrices with independent Gaussian columns, see Vershynin (2018), give constants $C, c > 0$ such that, for every $\tau \in (0,1)$, with probability at least $1 - 2\exp(-c\tau^2 m)$,

$$\begin{aligned}\sqrt{m}\,\sqrt{\lambda_{\min}(\Sigma)}\,\big(1 - C\sqrt{N/m} - \tau\big) \;&\le\; \sigma_{\min}(\mathbf{A}_{\text{cat}}) &(28)\\ &\le\; \sigma_{\max}(\mathbf{A}_{\text{cat}}) &(29)\\ &\le\; \sqrt{m}\,\sqrt{\lambda_{\max}(\Sigma)}\,\big(1 + C\sqrt{N/m} + \tau\big). &(30)\end{aligned}$$

Define $\varepsilon_m := C\sqrt{N/m} + \tau$; then $\varepsilon_m \to 0$ as $m = hd \to \infty$ since $N/m \to 0$.

Step 4: Plug the concentration into Equation (27). Using Equation (30) in Equation (27) and writing $m = hd$, we obtain, with the same high probability,

$$\begin{aligned}\kappa(\mathbf{J}) &\le \sqrt{\frac{C_4^2 C_2^2 + m\,\lambda_{\max}(\Sigma)\,(1+\varepsilon_m)^2}{C_3^2 C_1^2 + m\,\lambda_{\min}(\Sigma)\,(1-\varepsilon_m)^2}}\\ &= \sqrt{\frac{\frac{C_4^2 C_2^2}{m} + \lambda_{\max}(\Sigma)\,(1+\varepsilon_m)^2}{\frac{C_3^2 C_1^2}{m} + \lambda_{\min}(\Sigma)\,(1-\varepsilon_m)^2}}. \qquad (31)\end{aligned}$$

Letting $h \to \infty$, so that $m \to \infty$, forces $\frac{C_4^2 C_2^2}{m}, \frac{C_3^2 C_1^2}{m} \to 0$ and $\varepsilon_m \to 0$, hence

$$\limsup_{h\to\infty}\; \kappa(\mathbf{J}) \;\le\; \sqrt{\frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)}}\,. \qquad (32)$$

Since $\kappa(\mathbf{J}) \ge 1$ always, Equation (32) shows that, in the general (non–isotropic) case, the limiting upper bound is $\sqrt{\kappa(\Sigma)}$.

Step 5: Isotropic specialization $\Rightarrow$ limit 1. If, moreover, the columns of $\mathbf{A}_{\text{cat}}$ are *isotropic*—i.e., $\Sigma = \sigma^2 I_N$ for some $\sigma^2 > 0$ (in particular, $\lambda_{\max}(\Sigma) = \lambda_{\min}(\Sigma) = \sigma^2$) then Equation (31) simplifies to

$$\kappa(\mathbf{J}) \;\le\; \sqrt{\frac{\frac{C_4^2 C_2^2}{m} + \sigma^2(1+\varepsilon_m)^2}{\frac{C_3^2 C_1^2}{m} + \sigma^2(1-\varepsilon_m)^2}} \;\xrightarrow[m\to\infty]{}\; \frac{1+\varepsilon_m}{1-\varepsilon_m} \;\xrightarrow[m\to\infty]{}\; 1,$$

again with probability tending to 1. Since $\kappa(\mathbf{J}) \ge 1$, we conclude $\kappa(\mathbf{J}) \to 1$ in probability (indeed, with high probability) as $h \to \infty$ with d fixed.

Asymptotic rate. In the isotropic case $\Sigma = \sigma^2 I$, the bound derived there gives (with high probability)

$$\kappa(\mathbf{J}) \;\le\; \sqrt{\frac{\frac{C_4^2 C_2^2}{m} + \sigma^2(1+\varepsilon_m)^2}{\frac{C_3^2 C_1^2}{m} + \sigma^2(1-\varepsilon_m)^2}} \;=\; \frac{1+\varepsilon_m}{1-\varepsilon_m}\,\big(1 + o(1)\big) \qquad (m \to \infty).$$

Using the expansion $\dfrac{1+\varepsilon_m}{1-\varepsilon_m} = 1 + 2\varepsilon_m + O(\varepsilon_m^2)$ and $\varepsilon_m = O\big(\sqrt{N/m}\big)$, we obtain

$$\kappa(\mathbf{J}) \;=\; 1 + O\Big(\sqrt{\tfrac{N}{m}}\Big) \;=\; 1 + O\Big(\sqrt{\tfrac{N}{hd}}\Big) \qquad \text{as } h \to \infty \text{ with } d \text{ fixed.}$$

□

A.1.2 Discussion of assumptions in main theorem

The assumptions of Theorem 3.1 should be understood as an idealized model for isolating the conditioning effect of multiple heads. The bounded singular value assumptions are nondegeneracy conditions on the per-head maps and output projection. At random initialization, standard continuous initializations make rank collapse a probability-zero event, while Xavier, He, and orthogonal initializations control operator norms at the relevant scale. LayerNorm further controls the scale of the inputs, and the row-wise softmax Jacobian has bounded operator norm. These facts make the upper-boundedness assumption natural near initialization. The lower-boundedness assumption is stronger: it rules out collapsed heads or nearly singular head Jacobians, and is therefore best viewed as an explicit nondegeneracy condition. We check that this holds empirically in our experiments below.

The covariance assumption formalizes head diversity. It should not be interpreted as claiming that trained head outputs are exactly independent, Gaussian, or isotropic. At random initialization, independent head parameters and high-dimensional concentration motivate a Gaussian random-feature approximation to the concatenated head representation. However, the shared input, the softmax nonlinearity, and training dynamics all introduce dependencies. The theorem therefore provides a tractable model predicting that adding nondegenerate and sufficiently diverse heads improves attention-Jacobian conditioning, with the anisotropic case controlled by the covariance condition number. Our empirical results test this prediction directly in finite-width trained transformers, where the assumptions are only approximate.

Empirically testing assumptions on singular values. We empirically examine the bounded-singular-value assumptions in Theorem 3.1, which require the attention-head maps $\mathbf{A}_i$ and output projection $\mathbf{P}$ to remain nondegenerate as the number of heads varies; that is, their minimum singular values should not collapse to zero and their maximum singular values should not grow without bound. For each model, layer, and training iterations, we compute the smallest and largest singular values of each attention-head representation $\mathbf{A}_i$, as well as the smallest and largest singular values of the corresponding output projection $\mathbf{P}$. We then average these quantities over layers, training checkpoints, and five random seeds, and report them as a function of the number of heads. This diagnostic is not intended to show that the idealized assumptions of Theorem 3.1 hold exactly, but rather to test whether the quantities appearing in the theorem remain bounded away from numerical collapse and do not exhibit unbounded growth during training. Figure 5 and Figure 6 show the empirical results for ViTs and GPTs, respectively. In all cases, the minimum singular values remain bounded away from numerical zero and the maximum singular values remain controlled as the number of heads increases, supporting the nondegeneracy assumptions on the attention-head maps and output projections in Theorem 3.1. We perform these experiments on fixed-depth architectures. Since the reported quantities are computed per layer and then averaged over layers, they measure the behavior of individual attention heads and projections rather than effects caused by changing the number of layers.

Empirically testing covariance assumptions. We next examine the covariance assumptions on the concatenated head representation. For each layer and checkpoint, we form

$$\mathbf{A}_{\mathrm{cat}} = [\mathbf{A}_1, \dots, \mathbf{A}_h] \in \mathbb{R}^{N \times hd}.$$

Let $a_j \in \mathbb{R}^N$ denote the j-th column of $\mathbf{A}_{\mathrm{cat}}$, for $1 \le j \le hd$, and define the empirical column mean

$$\bar{a} = \frac{1}{hd} \sum_{j=1}^{hd} a_j.$$

We compute the empirical covariance

$$\widehat{\Sigma} = \frac{1}{hd} \sum_{j=1}^{hd} (a_j - \bar{a})(a_j - \bar{a})^\top.$$

Equivalently, if

$$\widetilde{\mathbf{A}}_{\mathrm{cat}} = [a_1 - \bar{a}, \dots, a_{hd} - \bar{a}],$$

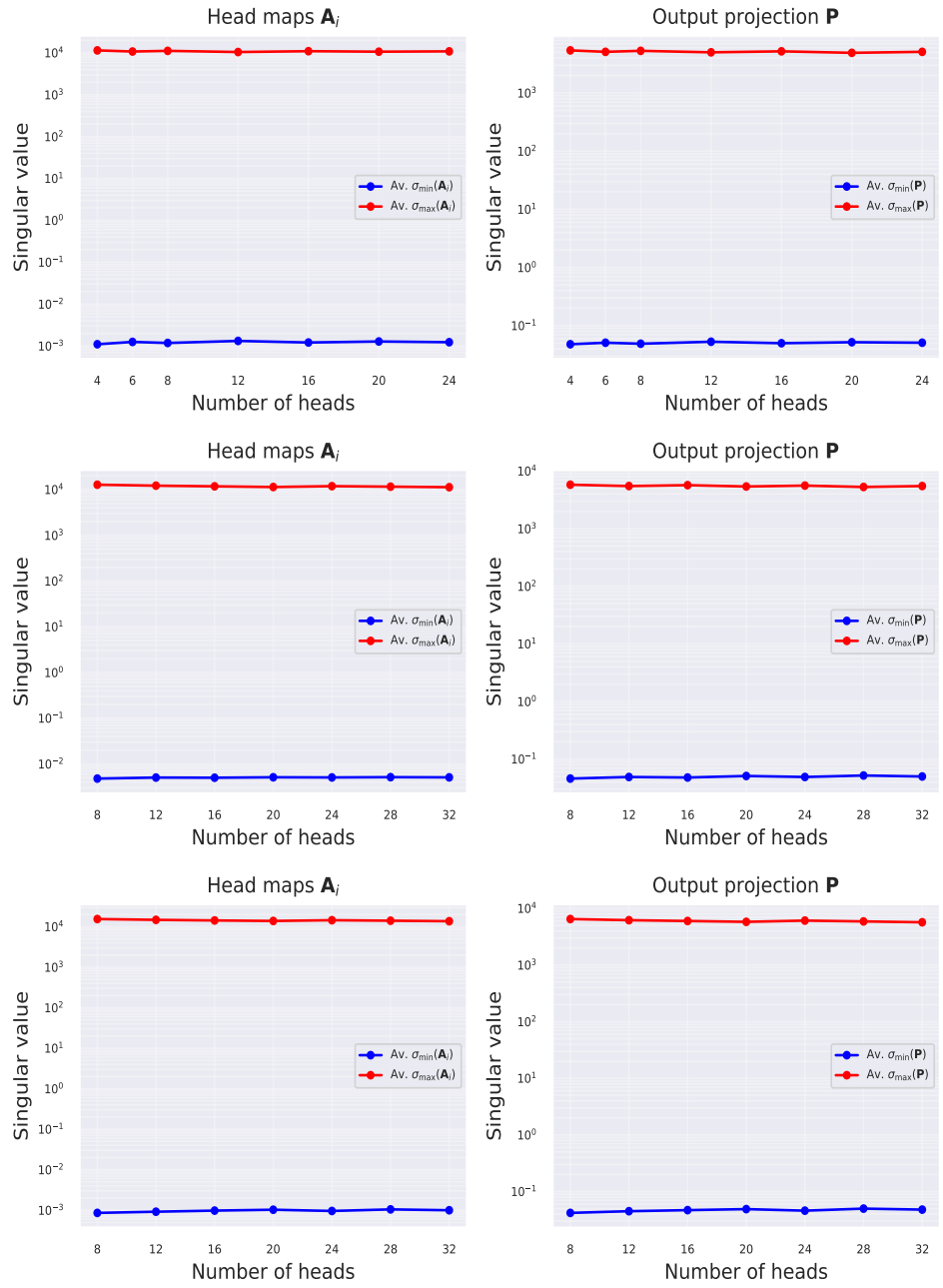


Figure 5: Empirical singular-value diagnostics for the assumptions in Theorem 3.1 on ViT-B with 12 layers (top), ViT-L with 24-layers (middle) and ViT-G with 40 layers (bottom) trained on ImageNet-1k. We report the average smallest and largest singular values of the attention-head maps $\mathbf{A}_i$ (left) and output projection $\mathbf{P}$ (right), averaged over layers, training, and five random seeds. The singular values remain bounded away from numerical zero and do not exhibit unbounded growth as the number of heads increases, supporting the nondegeneracy assumptions used in the theorem.

then

$$\widehat{\Sigma} = \frac{1}{hd} \widetilde{\mathbf{A}}_{\mathrm{cat}} \widetilde{\mathbf{A}}_{\mathrm{cat}}^{\top}.$$

For ViT-B, ViT-L, and ViT-G on ImageNet-1k, and GPT models on WikiText and The Pile, we track $\lambda_{\min}(\widehat{\Sigma})$ as the number of heads increases, averaging over layers and training checkpoints. This tests

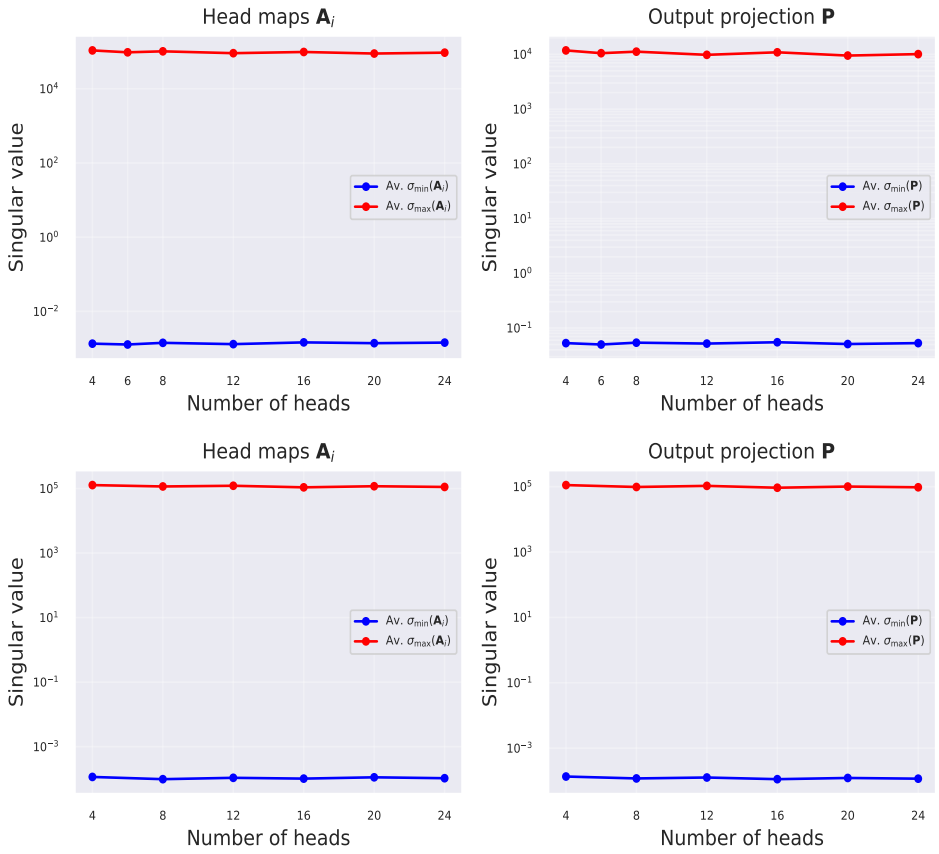


Figure 6: Empirical singular-value diagnostics for the assumptions in Theorem 3.1 on GPTs trained on WikiText with 20 layers (top) and The Pile with 24 layers (bottom). We report the average smallest and largest singular values of the attention-head maps $\mathbf{A}_i$ (left) and output projection $\mathbf{P}$ (right), averaged over layers, training, and five random seeds. The singular values remain bounded away from numerical zero and do not exhibit unbounded growth as the number of heads increases, supporting the nondegeneracy assumptions used in the theorem.

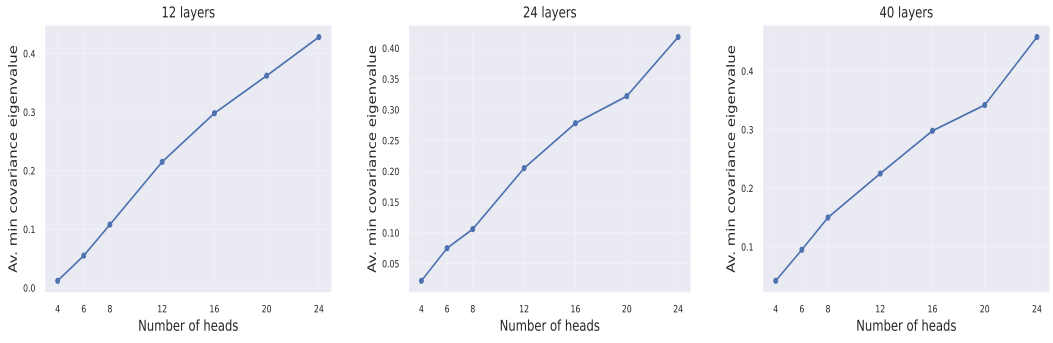


Figure 7: Average minimum eigenvalue of the covariance matrix over training as heads increase for a ViT-B (left), ViT-L (middle) and ViT-G (right) architectures on ImageNet-1k. In each case we see that the eigenvalue stays away from zero empirically demonstrating the assumption in Theorem 3.1.

whether the empirical covariance of the concatenated head representations remains nondegenerate during training. Figure 7 and Figure 8 show the results for the ViT and GPT models, respectively.

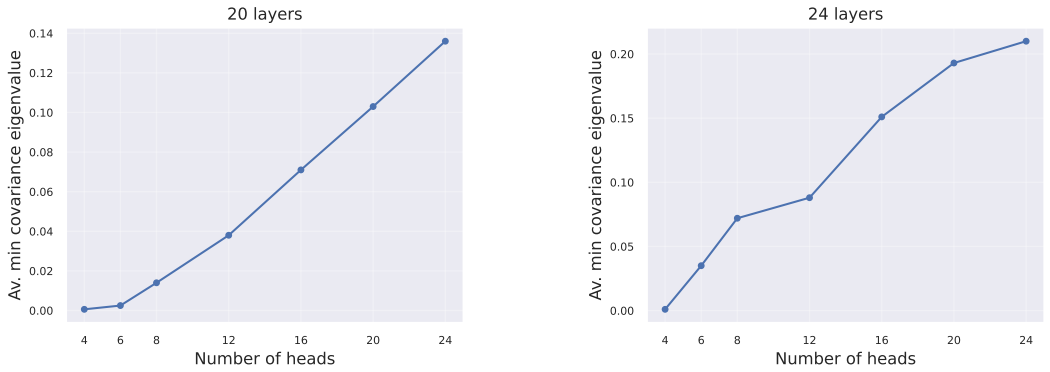


Figure 8: Average minimum eigenvalue of the covariance matrix over training as heads increase for a GPT trained on WikiText (left) and The Pile (right). In each case we see that the eigenvalue stays away from zero empirically demonstrating the assumption in Theorem 3.1.

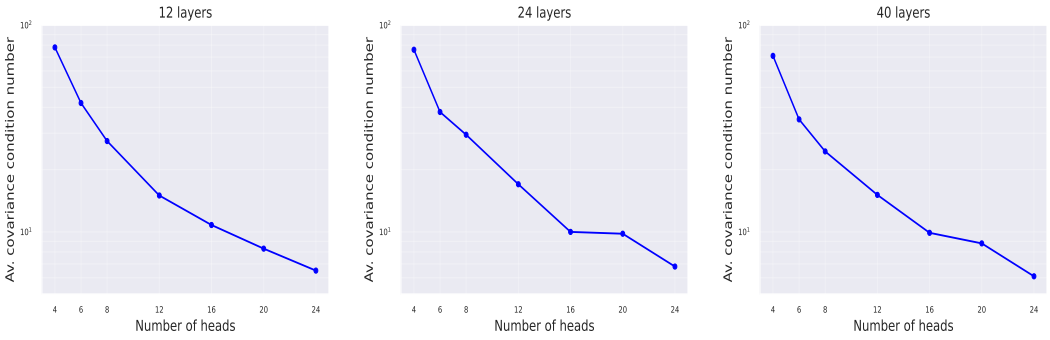


Figure 9: Covariance anisotropy of concatenated attention-head representations for ViT-B (left), ViT-L (middle), and ViT-G (right) trained on ImageNet-1k. We report $\eta(\widehat{\Sigma}_{\text{norm}})$, averaged over layers and training. Although $\eta(\widehat{\Sigma}_{\text{norm}}) \neq 1$ during training, it decreases toward 1 as the number of heads increases, indicating less anisotropy and greater head diversity. This supports the isotropic assumption in Theorem 3.1 as a useful large-head idealization.

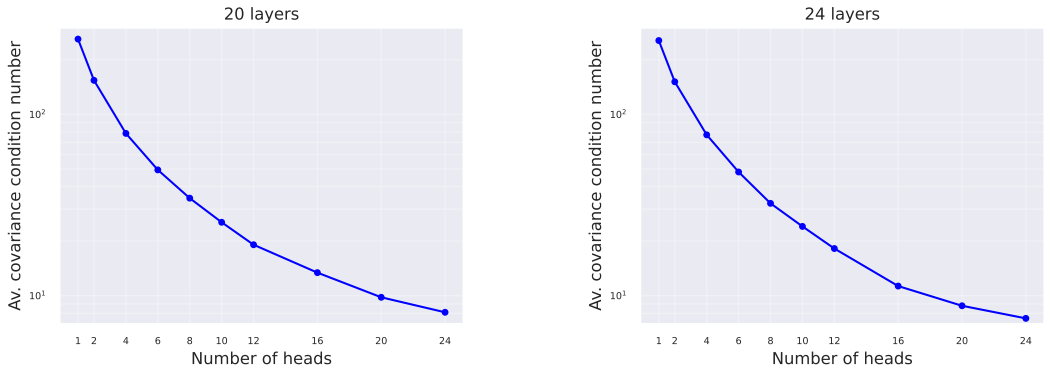


Figure 10: Covariance anisotropy of concatenated attention-head representations for GPT trained on WikiText (left) and The Pile (right). We report $\eta(\widehat{\Sigma}_{\text{norm}})$, averaged over layers and training checkpoints. While the covariance is not exactly isotropic during training, increasing the number of heads moves $\eta(\widehat{\Sigma}_{\text{norm}})$ closer to 1, indicating less anisotropy and greater head diversity.

In each case, $\lambda_{\min}(\widehat{\Sigma})$ remains bounded away from numerical zero, supporting the covariance nondegeneracy assumption in Theorem 3.1.

To assess the isotropy assumption used in the second part of Theorem 3.1, we compute the normalized empirical covariance

$$\widehat{\Sigma}_{\text{norm}} = \widehat{\Sigma}/(\text{tr}(\widehat{\Sigma})/N).$$

We then track the eigenvalue condition number

$$\eta(\widehat{\Sigma}_{\text{norm}}) = \lambda_{\max}(\widehat{\Sigma}_{\text{norm}})/\lambda_{\min}(\widehat{\Sigma}_{\text{norm}}).$$

Perfect isotropy corresponds to $\eta(\widehat{\Sigma}_{\text{norm}}) = 1$. As expected, trained transformers are not exactly isotropic. However, $\eta(\widehat{\Sigma}_{\text{norm}})$ remains controlled and generally decreases as the number of heads increases, indicating that the concatenated head representations become less anisotropic and more diverse. Figure 9 and Figure 10 report these quantities averaged over layers and training for the ViT and GPT models, respectively.

A.1.3 Further theoretical insights.

In this section, we give the analogue of Theorem 3.1 where we fix the number of heads h and allow the head dimension d to vary.

Theorem A.3. *Fix the number of heads $h \in \mathbb{N}$ and let $d \in \mathbb{N}$ vary. For each head $i = 1, \dots, h$, let*

$$\mathbf{A}_i \in \mathbb{R}^{N\times d}, \qquad \mathbf{A}_{\text{cat}} := [\,\mathbf{A}_1, \dots, \mathbf{A}_h\,] \in \mathbb{R}^{N\times(hd)}, \qquad \mathbf{P} \in \mathbb{R}^{(hd)\times D},$$

and define the multi-head output $\mathbf{MA} := \mathbf{A}_{\text{cat}}\mathbf{P} \in \mathbb{R}^{N\times D}$. Let

$$y := \text{vec}(\mathbf{MA}) \in \mathbb{R}^{ND}, \qquad \theta := [\,\theta_1; \dots; \theta_h;\, \text{vec}(\mathbf{P})\,] \in \mathbb{R}^{3hDd+Dhd},$$

where $\theta_i = [\text{vec}(W_{Q_i}); \text{vec}(W_{K_i}); \text{vec}(W_{V_i})] \in \mathbb{R}^{3Dd}$, and define the (row) per-head Jacobians

$$\mathbf{J}(\mathbf{A}_i) \in \mathbb{R}^{(3Dd)\times(Nd)}.$$

Let the Jacobian of the multi-head attention be given by

$$\mathbf{J} := \mathbf{J}(\mathbf{MA}) = \frac{\partial\theta}{\partial y} \in \mathbb{R}^{(3hDd+Dhd)\times(ND)}.$$

Assume the following bounds hold uniformly in d (for all sufficiently large d):

- (B1) *There exist constants $0 < C_1 \le C_2 < \infty$ such that $C_1 \le \sigma_{\min}(\mathbf{J}(\mathbf{A}_i))$ and $\sigma_{\max}(\mathbf{J}(\mathbf{A}_i)) \le C_2$ for all i.*
- (B2) *There exist constants $0 < C_3 \le C_4 < \infty$ such that $C_3 \le \sigma_{\min}(\mathbf{P})$ and $\sigma_{\max}(\mathbf{P}) \le C_4$.*
- (B3) *Writing $m := hd$, the m columns of $\mathbf{A}_{\text{cat}}$ are independent, mean-zero, Gaussian in $\mathbb{R}^N$ with common covariance Σ satisfying $0 < \lambda_{\min}(\Sigma) \le \lambda_{\max}(\Sigma) < \infty$.*

Then:

(i) ***Analogue of Theorem A.2.***

$$\sigma_{\min}(\mathbf{J}) \;\ge\; \sqrt{C_3^2C_1^2 + \sigma_{\min}(\mathbf{A}_{\text{cat}})^2}\,, \qquad \sigma_{\max}(\mathbf{J}) \;\le\; \sqrt{C_4^2C_2^2 + \sigma_{\max}(\mathbf{A}_{\text{cat}})^2}\,,$$

hence

$$\kappa(\mathbf{J}) \;\le\; \sqrt{\frac{C_4^2C_2^2 + \sigma_{\max}(\mathbf{A}_{\text{cat}})^2}{C_3^2C_1^2 + \sigma_{\min}(\mathbf{A}_{\text{cat}})^2}}\,.$$

(ii) ***Asymptotic high-probability bound (analogue of first part of Theorem 3.1.*** *With probability tending to 1 as $d \to \infty$ (with h fixed),*

$$\limsup_{d\to\infty}\; \kappa(\mathbf{J}) \;\le\; \sqrt{\frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)}}\,.$$

(iii) ***Isotropic specialization (analogue of second part of Theorem 3.1).*** *If the columns of* $\mathbf{A}_{\text{cat}}$ *are isotropic, i.e.* $\Sigma = \sigma^2 I_N$*, then with probability tending to* 1*,*

$$\kappa(\mathbf{J}) \longrightarrow 1 \qquad \text{as } d \to \infty \text{ with } h \text{ fixed,}$$

and, moreover,

$$\kappa(\mathbf{J}) = 1 + O\Big(\sqrt{\tfrac{N}{hd}}\Big).$$

Proof. The proof is identical in structure to how we proceeded with Theorem A.2 and Theorem 3.1, with $m := hd$ now growing via $d \to \infty$ (and fixed h).

Step 1. An analogue of Lemma A.1 yields the two-block bounds

$$\sigma_{\min}(\mathbf{J}) \geq \sqrt{\sigma_{\min}(\mathbf{P})^2\big(\min_i \sigma_{\min}(\mathbf{J}(\mathbf{A}_i))\big)^2 + \sigma_{\min}(\mathbf{A}_{\text{cat}})^2} \tag{33}$$

$$\sigma_{\max}(\mathbf{J}) \leq \sqrt{\sigma_{\max}(\mathbf{P})^2\big(\max_i \sigma_{\max}(\mathbf{J}(\mathbf{A}_i))\big)^2 + \sigma_{\max}(\mathbf{A}_{\text{cat}})^2}\,. \tag{34}$$

Step 2 (Uniform bounds). By (B1)–(B2), for all large d,

$$C_1 \leq \sigma_{\min}(\mathbf{J}(\mathbf{A}_i)) \leq \sigma_{\max}(\mathbf{J}(\mathbf{A}_i)) \leq C_2, \qquad C_3 \leq \sigma_{\min}(\mathbf{P}) \leq \sigma_{\max}(\mathbf{P}) \leq C_4,$$

which gives the deterministic bound in (i).

Step 3 (Random-matrix control). By (B3) and standard Gaussian singular value estimates for tall matrices with $m = hd$ independent columns, with high probability

$$\sqrt{m\,\lambda_{\min}(\Sigma)}\,(1 - C\sqrt{N/m} - \tau) \leq \sigma_{\min}(\mathbf{A}_{\text{cat}}) \leq \sigma_{\max}(\mathbf{A}_{\text{cat}}) \leq \sqrt{m\,\lambda_{\max}(\Sigma)}\,(1 + C\sqrt{N/m} + \tau),$$

for universal constants $C, c > 0$ and any $\tau \in (0, 1)$ (with probability $\geq 1 - 2e^{-c\tau^2 m}$).

Step 4 (Asymptotics). Plugging these into the bound from Step 2 and letting $m = hd \to \infty$ yields (ii). In the isotropic case $\Sigma = \sigma^2 I$, the ratio simplifies to $\frac{1+\varepsilon_m}{1-\varepsilon_m}$ with $\varepsilon_m = O(\sqrt{N/m})$, giving

$$\kappa(\mathbf{J}) = 1 + O\Big(\sqrt{\tfrac{N}{m}}\Big) = 1 + O\Big(\sqrt{\tfrac{N}{hd}}\Big),$$

which proves (iii). □

A.2 Experiments

A.2.1 Vision Transformers on ImageNet-1k

Training details. For the vision transformer experiments carried out in Section 4.1 and the further experiments carried out in this section we used the timm github code base for all vision transformer architectures: https://github.com/huggingface/pytorch-image-models/tree/main/timm. The vision transformers ViT-B, ViT-L, ViT-G (Dosovitskiy et al., 2020; Steiner et al., 2021), DaViT-B, DaViT-L, DaViT-G (Ding et al., 2022) were trained for 300 epochs, while while XCiT-M and XCiT-L (Ali et al., 2021) were trained for 400 epochs, all using the AdamW optimizer with a learning rate of 3e-4, with weight decay factor of 3e-1, and a batch size of 1024, following the cited references. We remind the reader that in the timm github codebase there is no XCiT-G scale, similar to ViT-G and DaViT-G, and this is why we only used XCiT-M and XCiT-L. Each experiment was run over 5 trials with different random seeds. When trained on ImageNet-1k we applied the data augmentation carried out in Steiner et al. (2021).

Hardware. The experiments in Section 4.1 and in this section were all carried out on 8 H200 GPUs using multi-GPU training.

Validating the effects on conditioning for ViT-L and ViT-G. We validated the conditioning effects on the attention Jacobian of increasing heads and head dimension for larger ViT-L and ViT-G models (Dosovitskiy et al., 2020). The baseline ViT-L architecture comprises of 24 layers, 16 attention heads each with head dimension $d = 64$, embedding dimension 1024 and an MLP hidden layer

width of $4 \times 1024 = 4096$. We first vary the number of heads while fixing the head dimension at $d = 64$, training models on ImageNet-1k with $h \in \{1, 4, 8, 12, 16, 20, 24\}$ and all other architectural components held fixed. Since the analysis also applies when fixing h and increasing d (Theorem A.3), we perform a complementary experiment with $h = 16$ and $d \in \{6, 18, 48, 64, 88, 104, 122\}$, chosen to approximately match parameter counts. We also did the same for a larger ViT-G model with the baseline admitting 40 layers, 16 heads, embedding dimension 1408 and an MLP hidden layer width of $48/11 \times 1408 = 6144$. All models are trained for 300 epochs using the above training regime for ViT-L, ViT-G architectures. Every 50 epochs, we compute the Jacobian condition number of the attention matrix, averaged across layers. We ran five experiments with different random seeds. As shown in Figure 11 and Figure 12, the average condition number over training decreases consistently with increasing h and d, confirming Theorem 3.1 and Theorem A.3 for these larger architectures. We plotted the mean over these runs. The standard deviations are much smaller than the y-axis scale and therefore not visible on the plotted scale, which spans changes of roughly two orders of magnitude.

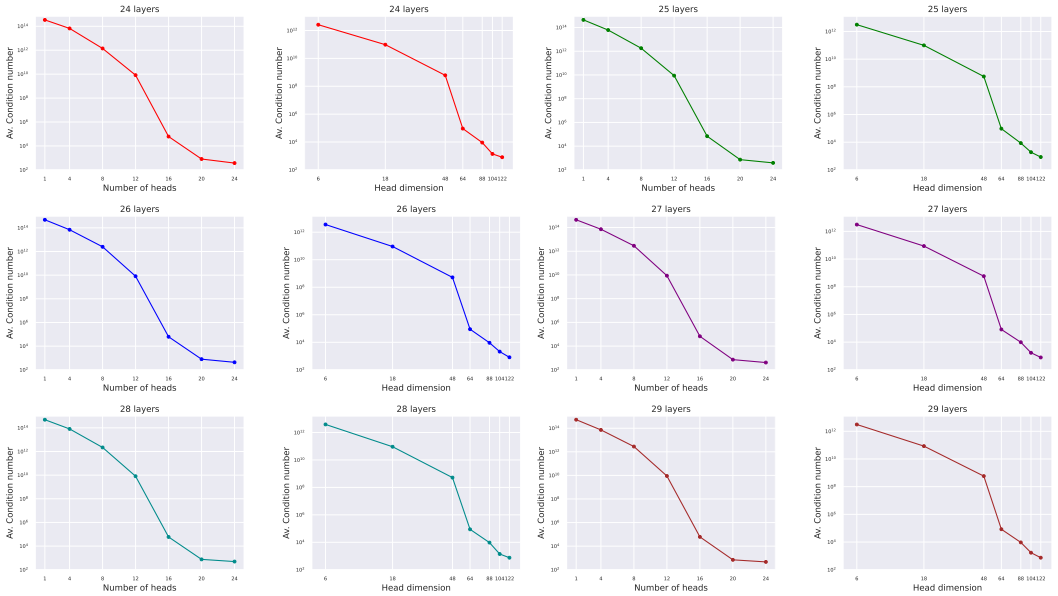


Figure 11: Empirical condition number of the attention-layer Jacobian in ViT-L models with varying numbers of heads and head dimensions, shown for layers 24–29 from left to right. In all layers, increasing either the number of heads or the head dimension improves Jacobian conditioning, consistent with the predictions of Theorem 3.1 and Theorem A.3.

Standard deviations for Figure 3. In Figure 13 we plot the associated standard deviations over 5 trials associated to Figure 3. These are shown in brackets and range from 2-4 epochs.

Effects of increasing MLP width. In transformer architectures, the hidden-layer width of the MLP blocks is typically set to $4D$, where D denotes the embedding dimension; the factor 4 is commonly referred to as the MLP ratio. We investigate the effect of increasing MLP width compared to increasing network depth in ViT-B, ViT-L, and ViT-G architectures. For each model, we train parameter-matched variants to reach the baseline accuracy and record the number of training epochs required for convergence. We ran five trials with different random seeds. Across all settings, we observe no meaningful reduction in training epochs (mean over five trials) when increasing the MLP ratio compared to increasing depth. We believe that this is because, at these scales, MLP widths are already sufficiently large, limiting the potential gains in training efficiency from further widening. Note that when changing the MLP width or depth we don't change the attention heads and hence the condition numbers for the attention Jacobian both never change from the baseline and that's why for this plot we don't show them. The results are summarized in Figure 14.

Analysis of leaner transformer configurations for ViT-L and ViT-G. As shown in Figure 3, increasing the number of attention heads leads to faster convergence than both the baseline ViT-L/ViT-

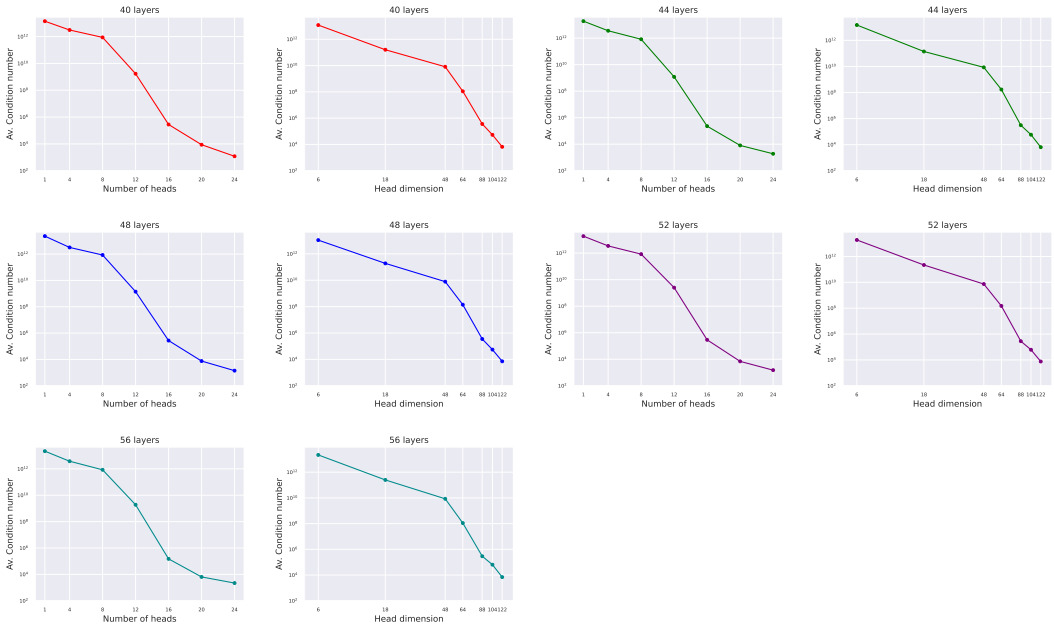


Figure 12: Empirical condition number of the attention-layer Jacobian in ViT-G models with varying numbers of heads and head dimensions, shown for layers 40–56 from left to right. In all layers, increasing either the number of heads or the head dimension improves Jacobian conditioning, consistent with the predictions of Theorem 3.1 and Theorem A.3.

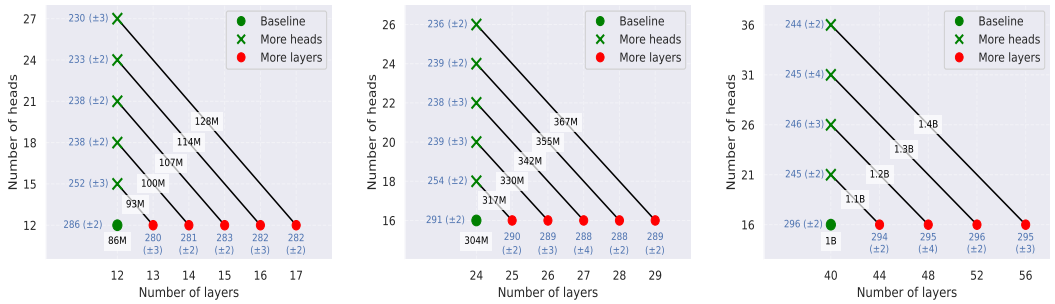


Figure 13: Standard deviations for convergence for plots in Figure 3 shown in brackets.

G architectures and depth-increased variants, albeit at the cost of additional parameters. Motivated by our theoretical prediction (Theorem 3.1) that increasing the number of heads improves attention Jacobian conditioning, we investigate whether these optimization benefits can be retained while reducing model size, following the same approach used for ViT-B in Section 4.1. For each case we ran a total of five trials over different random seeds. Specifically, we found increasing the number of attention heads to 20 for a ViT-L gave a well-conditioned attention Jacobian. We then progressively reduce transformer depth and MLP width.

We observe trends consistent with those seen for ViT-B. For ViT-L, increasing the number of heads to 20 enables a reduction in depth to 11 layers and the MLP ratio to 2, without loss in accuracy and with faster convergence. Further reducing the depth below 13 layers prevents convergence to the baseline accuracy across all tested MLP ratios. The corresponding results are shown in Figure 15.

Similarly, for ViT-G, increasing the number of heads to 24 allows the depth to be reduced to 21 layers and the MLP ratio to 2, while maintaining performance and reducing the number of training epochs. Reducing the depth below 21 layers again results in models that fail to converge to the baseline accuracy across all MLP ratios. These results are summarized in Figure 16.

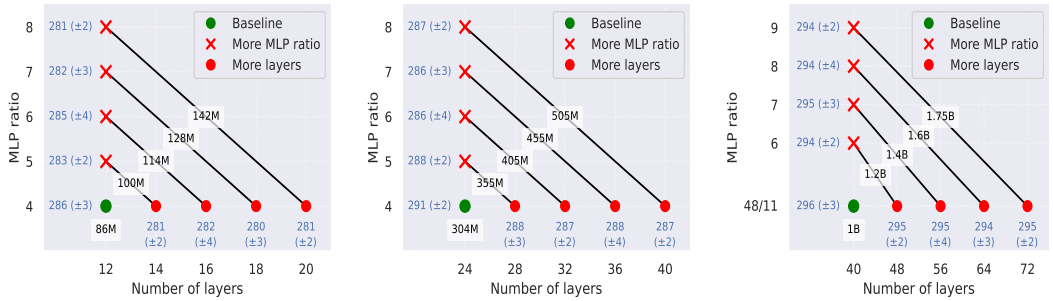


Figure 14: **Effect of increasing MLP ratio versus depth in Vision Transformers.** Comparison of the number of ImageNet-1k training epochs required for convergence for ViT-B (left), ViT-L (middle), and ViT-G (right), averaged over five random seeds. We compare parameter-matched increases in MLP ratio, starting from the baseline value of 4, against increases in network depth. Across all model scales, increasing the MLP ratio does not reduce the number of epochs (mean over 5 trials) required for convergence relative to increasing depth. Standard deviations range from 2 to 4 epochs, indicating that the observed trends are stable across random seeds.

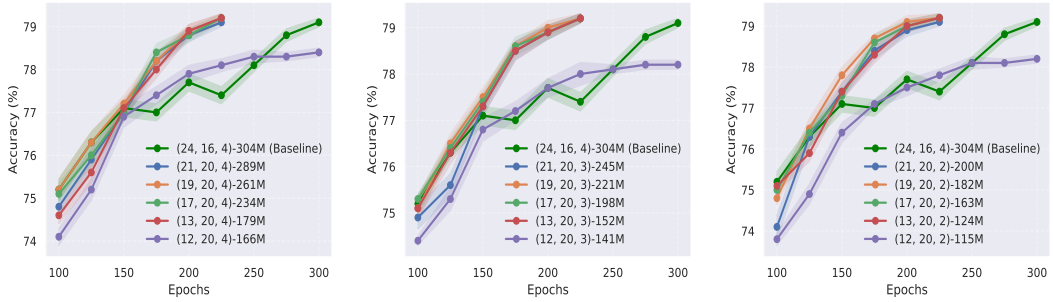


Figure 15: **Trading depth and MLP width for heads in ViT-L.** Comparison of the baseline ViT-L architecture (24 layers, 16 heads, MLP ratio 4) with shallower ViT-L variants using 20 attention heads and MLP ratios of 4 (left), 3 (middle), and 2 (right). All models are trained on ImageNet-1k to reach the baseline accuracy. For each MLP ratio, several configurations achieve baseline accuracy (mean over 5 trials with different random seeds) while using fewer parameters and converging in approximately 10-20% fewer training epochs. Configurations are annotated in the legend by (number of layers, number of heads, MLP ratio), with the corresponding parameter counts shown alongside.

A.2.2 Self-Supervised Learning with DINO

In this section, we evaluate the transformer configurations identified in Table 1 under a downstream evaluation protocol to assess their effectiveness in evaluation performance. Specifically, we perform self-supervised pretraining using DINO on ImageNet-1k, following the setup of Caron et al. (2021). The quality of the learned representations is then evaluated using standard k-nearest neighbor (k-NN) classification and linear probing protocols, as in Caron et al. (2021). We focus on ViT-B and XCiT-M architectures.

Training details. The experiments were run using the original DINO github codebase: https://github.com/facebookresearch/dino and we used the hyperparameters used in the original paper of Caron et al. (2021) and shown at https://github.com/facebookresearch/dino for both training and evaluation (k-nn and linear probing). Namely, the optimizer used was AdamW, training was done for 100 epochs with a learning rate of 5e-4 and a batch size of 512. Both ViT-B and XCiT-M architectures were used from the timm github code base: https://github.com/huggingface/pytorch-image-models/tree/main/timm.

Hardware. All experiments in this section were carried out on 8 H200 GPUs using multi-GPU training.

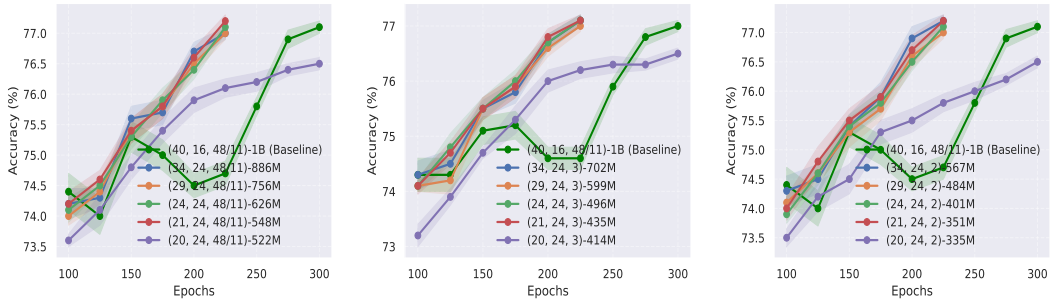


Figure 16: **Trading depth and MLP width for heads in ViT-G.** Comparison of the baseline ViT-G architecture (40 layers, 16 heads, MLP ratio 48/11) with shallower ViT-G variants using 24 attention heads and MLP ratios of 48/11 (left), 3 (middle), and 2 (right). All models are trained on ImageNet-1k to reach the baseline accuracy. For each MLP ratio, several configurations achieve baseline accuracy (mean over 5 trials with different random seeds) while using fewer parameters and converging in approximately 10-20% fewer training epochs. Configurations are annotated in the legend by (number of layers, number of heads, MLP ratio), with the corresponding parameter counts shown alongside.

The results, summarized in Table 3, show that the proposed configurations with increased attention heads and reduced depth and MLP width achieve improved downstream performance while using substantially fewer parameters. We followed the exact training recipe from Caron et al. (2021) using their associated GitHub. The training and evaluation were done on H200 GPUs.

Table 3: Downstream evaluation of representations learned via self-supervised DINO pretraining on ImageNet-1k, using k-nearest neighbor (k-NN) classification and linear probing. Results are reported for baseline ViT-B and XCiT-M architectures as well as their lean variants. Values in brackets denote standard deviations over random seeds. In both cases, the leaner configurations achieve improved downstream performance while using substantially fewer parameters.

	Params.	K-NN (%)	Linear (%)
ViT-B (12, 12, 4) baseline	86M	77.1 ($\pm$0.2)	78.6 ($\pm$0.3)
ViT-B (7, 16, 2)	**40M**	**77.4** ($\pm$**0.2**)	**78.9** ($\pm$**0.2**)
XCiT-M (24, 8, 4) baseline	84M	77.6 ($\pm$0.2)	78.8 ($\pm$0.3)
XCiT-M (13, 16, 2)	**47M**	**77.8** ($\pm$**0.3**)	**80.0** ($\pm$**0.2**)

A.2.3 Object Detection and Instance Segmentation

In this section, we evaluate whether the proposed lean transformers retain strong transfer performance on downstream dense prediction tasks. We consider object detection and instance segmentation using the XCiT-M architectures (Ali et al., 2021) trained on ImageNet-1k in Section 4.1, followed by fine-tuning on COCO 2017 (Lin et al., 2014). The XCiT models are used as backbones in Mask R-CNN (He et al., 2017) with a Feature Pyramid Network (FPN) for multi-scale feature representation. Results are reported in Table 4; each entry is the mean over five random seeds, with standard deviations shown in brackets. We report bounding-box average precision AP^b, AP^b_{50}, and AP^b_{75}, together with mask average precision AP^m, AP^m_{50}, and AP^m_{75}. The lean XCiT-M configuration from Section 4.1 matches, and in several metrics slightly improves upon, the original baseline while using substantially fewer parameters.

Hardware and Implementation: The experiments were carried out on Nvidia H200 GPUs. The implementation followed He et al. (2017). We used the code base given by the GitHub Matterport (2017) following their exact training regime.

A.2.4 Language Modeling

GPT on WikiText

Table 4: Object detection and instance segmentation results on COCO 2017 using XCiT-M backbones with Mask R-CNN and FPN. Values are averaged over five random seeds, with standard deviations shown in brackets. The lean XCiT-M configuration matches or improves the baseline while using fewer parameters.

Model	Params.	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
XCiT-M $(24, 8, 4)$ baseline	84M	$45.7\,(\pm 0.2)$	$66.8\,(\pm 0.3)$	$49.6\,(\pm 0.2)$	$40.8\,(\pm 0.1)$	$63.6\,(\pm 0.3)$	$43.3\,(\pm 0.2)$
XCiT-M $(13, 20, 2)$ ours	**47M**	$\mathbf{45.9\,(\pm 0.1)}$	$\mathbf{67.1\,(\pm 0.2)}$	$\mathbf{49.8\,(\pm 0.3)}$	$\mathbf{40.8\,(\pm 0.2)}$	$\mathbf{63.7\,(\pm 0.4)}$	$\mathbf{43.6\,(\pm 0.2)}$

Training details. The GPT-2 model trained on the Wikitext dataset was done so using an AdamW optimizer, with a learning rate of 3e-4, a batch size of 512, and for 32000 iterations.

Hardware. All experiments in this section were carried out on 8 H200 GPUs using multi-GPU training.

In Section 4.1 we gave a detailed analysis on the ViT-B, ViT-L and ViT-G architectures showing that they follow the insight derived in the theoretical section Section 3. We now want to repeat those experiments for the case of a language model to show that they are not intrinsic to vision transformers only.

Validating the effects on conditioning. To empirically validate Theorem 3.1, we consider a baseline medium GPT-2 model with 20 layers, 16 attention heads per layer, a head dimension of 64, an embedding dimension of 1024, a context length of 256, and an MLP ratio of 4. We first vary the number of attention heads while fixing the head dimension at $d = 64$, training models on the Wikitext dataset with $h \in \{1, 4, 8, 12, 16, 20, 24\}$ and keeping all other architectural components fixed. In addition, motivated by Theorem A.3, we perform a complementary experiment in which the number of heads is fixed at $h = 12$ while the head dimension is varied over $d \in \{6, 18, 48, 64, 88, 104, 122\}$, with values chosen to approximately match parameter counts across configurations. All models are trained for 32000 iterations. Every 4000 iterations, we compute the condition number of the attention Jacobian, averaged across layers. A total of five trials were run with different random seeds. As shown in Figure 17, the average condition number consistently decreases as either h or d increases, providing empirical support for both Theorem 3.1 and Theorem A.3.

Number of heads vs. head dimension. We next compare the effects of increasing the number of attention heads versus increasing the head dimension. Specifically, we vary the number of heads while fixing the head dimension at $d = 64$, and separately vary the head dimension while fixing the number of heads at $h = 16$. In both cases, models are trained to convergence and evaluated using perplexity. Five trials are run with different random seeds. The results, shown in the left panel of Figure 18 show the mean perplexity with standard deviation in brackets, indicate that increasing the number of heads consistently yields lower perplexity than increasing the head dimension. This suggests that allocating capacity to additional attention heads is more effective for improving performance than increasing per-head dimensionality.

Increasing heads vs. increasing depth. We next compare the effect of increasing the number of attention heads, while fixing the head dimension at $d = 64$, against increasing network depth. For a fair comparison, all models are parameter-matched and trained until they reach the baseline perplexity, and we record the number of training iterations required for convergence. We ran five trials with different random seeds. The middle panel of Figure 18 shows that, across all settings, increasing the number of attention heads consistently reduces the number of training iterations by approximately 10-20% compared to increasing depth (standard deviation ranged from 42-85 iterations showing stable training over different seeds). Note that in brackets we plot the condition numbers showing that as we increase heads the condition number of the attention Jacobian goes down but when we increase depth the condition numbers stays the same.

Increasing MLP width. We also compare increasing the MLP width against increasing network depth. Consistent with our observations for ViTs in Section A.2.1, we find that increasing the MLP width yields no meaningful improvement in training efficiency. The corresponding results are shown in the right panel of Figure 18. For each experiment, we ran five trials with different random seeds

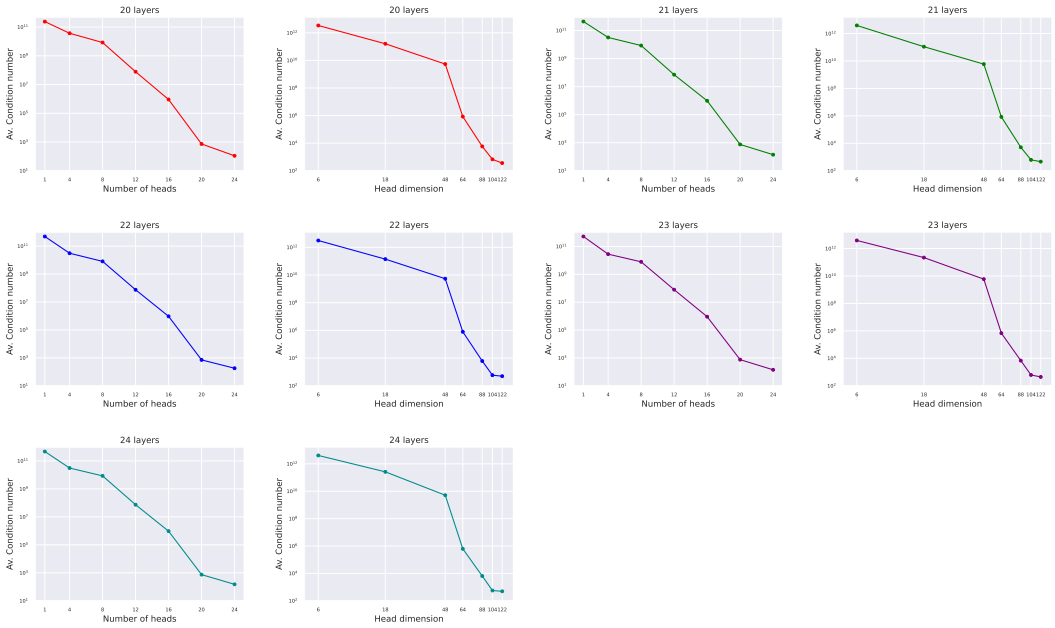


Figure 17: Empirical condition number (mean over 5 trials) of the attention-layer Jacobian in GPT-2 model on WikiText with varying numbers of heads and head dimensions, shown for layers 20–24 from left to right. In all layers, increasing either the number of heads or the head dimension improves Jacobian conditioning, consistent with the predictions of Theorem 3.1 and Theorem A.3.

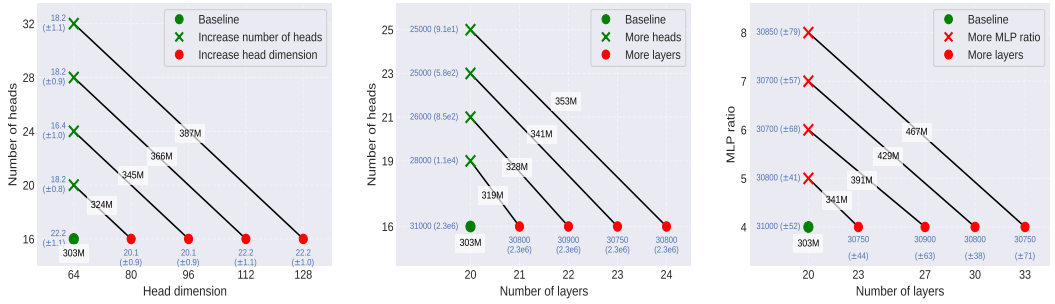


Figure 18: Left: Parameter–performance comparison for a medium-sized GPT-2 model trained on Wikitext when varying the number of attention heads versus the head dimension. Numbers adjacent to markers indicate mean perplexity (over 5 trials) after convergence. Middle: Parameter–efficiency comparison for increasing the number of attention heads versus increasing network depth. Models are trained to match the baseline perplexity, and numbers indicate the number of training iterations (mean over five trials) required for convergence. The numbers in brackets indicate the average condition number over training for the attention Jacobian. We see that as we increase heads, with fixed depth, the average condition number goes down but as we increase depth with fixed number of heads the average condition number stays the same. Right: Parameter–efficiency comparison for increasing the MLP ratio versus increasing network depth under the same parameter-matched setting (over 5 trials). Across all comparisons, increasing the number of attention heads yields the most favorable trade-off between performance and training efficiency.

(standard deviations ranged from 38-79 iterations). Note that when changing the MLP width or depth we don't change the attention heads and hence the condition numbers for the attention Jacobian never change from the baseline and that's why for this plot we don't show them.

Analysis of leaner configurations. The middle panel of Figure 18 shows that increasing the number of attention heads leads to more efficient training, albeit at the cost of increased parameter count relative to the baseline medium GPT-2 model. Motivated by our theoretical prediction (Theorem 3.1) that additional heads improve attention Jacobian conditioning, we investigate whether these optimization benefits can be preserved while reducing overall model size. To this end, we first increase the number of attention heads to 20 as we found this yielded a well-conditioned attention Jacobian and then progressively reduce transformer depth and MLP width. The results are shown in Figure 19. Across all tested MLP ratios, we identify configurations that reach the baseline perplexity in fewer training epochs. However, reducing the depth below 11 layers consistently prevents convergence to the baseline perplexity. Each exerpiment was run over five trials with different random seeds.

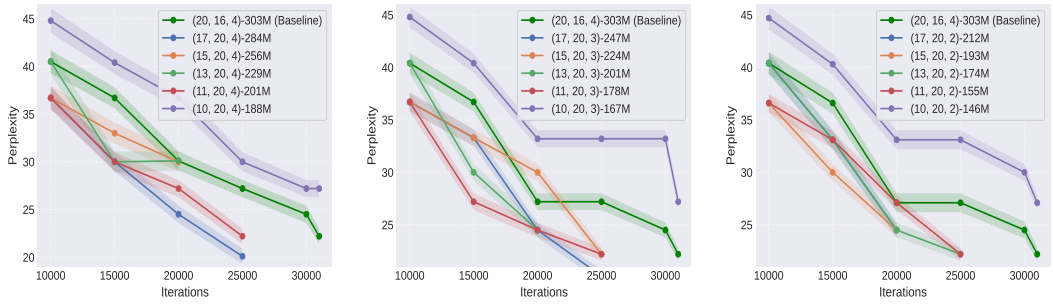


Figure 19: **Trading depth and MLP width for heads in a medium GPT-2.** Comparison of the baseline GPT-2 architecture (20 layers, 16 heads, MLP ratio 4) with shallower variants using 20 attention heads and MLP ratios of 4 (left), 3 (middle), and 2 (right). All models are trained on Wikitext to reach the baseline perplexity (mean over 5 trials). For each MLP ratio, several configurations achieve baseline perplexity while using fewer parameters and converging in approximately 10-20% fewer training epochs. Configurations are annotated in the legend by (number of layers, number of heads, MLP ratio), with the corresponding parameter counts shown alongside.

GPT-2 on The Pile.

Training details. The GPT-2 model trained on The Pile dataset was done so using an AdamW optimizer, with a learning rate of 3e-4, a batch size of 256, and for 201000 iterations.

Hardware. All experiments in this section were carried out on 8 H200 GPUs using multi-GPU training.

Validating the effects on conditioning. Similar to what we did in Section A.2.4 we validate Theorem 3.1 and Theorem A.3 for a large scale 1.3 billion parameter GPT-2 model trained on The Pile dataset. The baseline architecture uses 24 layers, 16 heads, an embedding dimension of 2048, a context length of 2048 and an MLP ratio of 4. For this experiment we vary the number of attention heads keeping the head dimension fixed at 128, which is the baseline head dimension. We also vary the head dimension keeping the number of heads fixed at 16. We ran five trials for each experiment with different random seeds. The results are shown in Figure 20 and once again we see that the average condition number consistently decreases as either h or d increases, providing empirical support for both Theorem 3.1 and Theorem A.3 on large scale GPT-2 model.

Number of heads vs. head dimension. We compare the effects of increasing the number of attention heads versus increasing the head dimension. We vary the number of heads while fixing the head dimension at $d = 128$, and separately vary the head dimension while fixing the number of heads at $h = 16$. In both cases, models are trained to convergence and evaluated using perplexity. A total of five trials per experiment with different seeds was run. The results, shown in the left panel of Figure 21, indicate that increasing the number of heads consistently yields lower perplexity than increasing the head dimension.

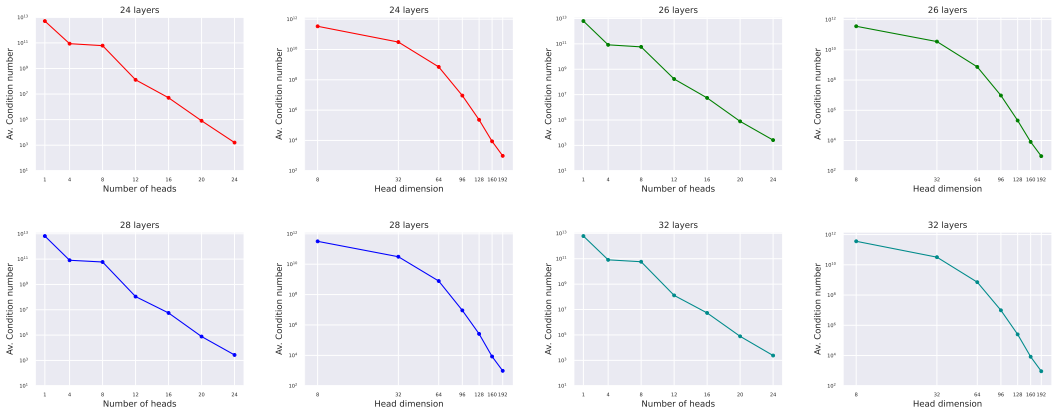


Figure 20: Empirical condition number (mean over 5 trials) of the attention-layer Jacobian in GPT-2 model on The Pile with varying numbers of heads and head dimensions, shown for layers 24-32 from left to right. In all layers, increasing either the number of heads or the head dimension improves Jacobian conditioning, consistent with the predictions of Theorem 3.1 and Theorem A.3.

Increasing heads vs. increasing depth. The middle panel of Figure 18 shows that, across all settings, increasing the number of attention heads consistently reduces the number of training iterations by approximately 10-20% compared to increasing depth. Each experiment was run over five trials with different random seeds (standard deviations ranged from 47-91 iterations showing stable training over different seeds).

Increasing MLP width. We also compare increasing the MLP width against increasing network depth. Consistent with our observations for ViTs in Section A.2.1, we find that increasing the MLP width yields no meaningful improvement in training efficiency. The corresponding results are shown in the right panel of Figure 18. Each experiment was run over five trials with different random seeds, with standard deviations ranging from 42-84 iterations. Note that when changing the MLP width or depth we don't change the attention heads and hence the condition numbers for the attention Jacobian never change from the baseline and that's why for this plot we don't show them.

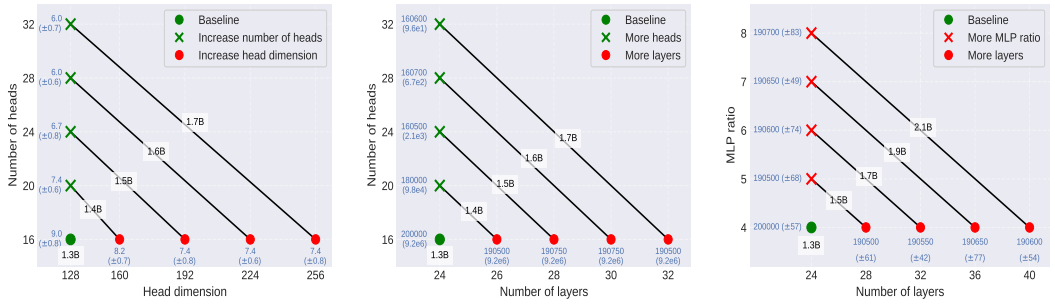


Figure 21: Left: Parameter–performance comparison for a large-sized GPT-2 model trained on The Pile when varying the number of attention heads versus the head dimension. Numbers adjacent to markers indicate perplexity (mean over 5 trials) after convergence. Middle: Parameter–efficiency comparison for increasing the number of attention heads versus increasing network depth. Models are trained to match the baseline perplexity, and numbers indicate the number of training iterations required for convergence (mean over 5 trials). Right: Parameter–efficiency comparison for increasing the MLP ratio versus increasing network depth under the same parameter-matched setting (mean over 5 trials). Across all comparisons, increasing the number of attention heads yields the most favorable trade-off between performance and training efficiency.

Analysis of leaner configurations. Motivated by the improved attention Jacobian conditioning observed with a larger number of heads, we increase the number of attention heads to 24 and then

progressively reduce transformer depth and MLP width. The results are shown in Figure 22. Each experiment was run over five trials with different random seeds. Across all tested MLP ratios, we identify configurations that reach the baseline perplexity in fewer training epochs. In contrast, reducing the depth below 13 layers consistently prevents convergence to the baseline perplexity.

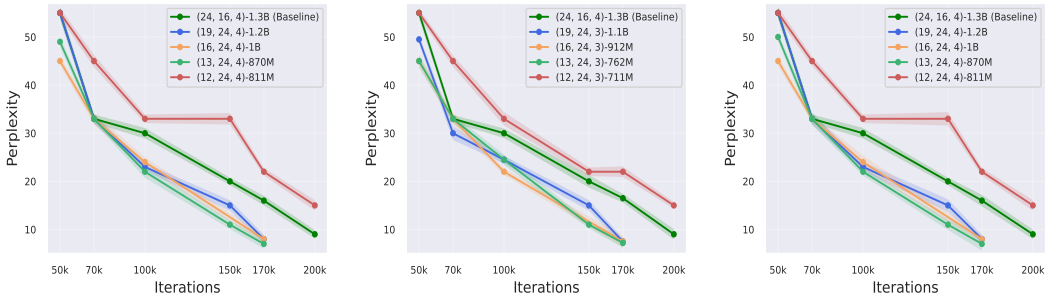


Figure 22: **Trading depth and MLP width for heads in a large GPT-2.** Comparison of the baseline GPT-2 architecture (24 layers, 16 heads, MLP ratio 4) with shallower variants using 24 attention heads and MLP ratios of 4 (left), 3 (middle), and 2 (right). All models are trained on The Pile to reach the baseline perplexity. For each MLP ratio, several configurations achieve baseline perplexity (mean over 5 trials) while using fewer parameters and converging in approximately 10-20% fewer training epochs. Configurations are annotated in the legend by (number of layers, number of heads, MLP ratio), with the corresponding parameter counts shown alongside.

Downstream performance. We evaluate downstream commonsense reasoning only for the GPT model trained on The Pile, since The Pile provides broad-coverage pretraining suitable for transfer evaluation. In contrast, WikiText is used as a controlled language-modeling benchmark. Its smaller size and narrow Wikipedia-style distribution makes downstream commonsense evaluation less informative.

For the downstream performance using the trained GPT models on The Pile from Section 4.3 we evaluated on the commonsense reasoning tasks HellaSwag, PIQA and WinoGrande. The results are shown in Table 5 where we see that our leaner configuration preserves downstream performance.

Table 5: Commonsense reasoning results for GPT-2 models pre-trained on The Pile. Values in brackets denote standard deviations over random seeds.

Model	HellaSwag	PIQA	WinoGrande
GPT (1.3B-baseline)	38.1 (±0.2)	70.1 (±0.2)	54.3 (±0.3)
GPT (652M-ours)	**38.3** (±**0.2**)	**70.3** (±**0.1**)	**54.3** (±**1.2**)

Table 6: Comparison of a pretrained BERT baseline (16 layers, 12 heads per layer) with our leaner variant (9 layers, 16 heads) on the GLUE benchmark (mean scores over 5 trials). Values in brackets denote standard deviations over random seeds. The leaner model achieves comparable performance while using substantially fewer parameters and less memory.

Model	MNLI	SST-2	STSB	RTE	QNLI	QQP	MRPC	CoLA	GLUE	Params.	Mem.
BERT (16,12,4)	83.8 (±0.3)	92.3 (±0.2)	86.3 (±0.4)	55.1 (±0.3)	90.1 (±0.2)	87.3 (±0.3)	85.0 (±0.4)	48.9 (±0.3)	78.6 (±0.2)	119M	13.8 GB
BERT (9,16,2)	**83.9** (±**0.2**)	**92.5** (±**0.3**)	**86.3** (±**0.3**)	**55.3** (±**0.4**)	**90.1** (±**0.2**)	**87.4** (±**0.3**)	**85.2** (±**0.2**)	**48.9** (±**0.4**)	**78.7** (±**0.3**)	**65M**	**5.8 GB**

BERT on the GLUE benchmark We consider the BERT architecture from Geiping & Goldstein (2023) trained on the Pile dataset (Gao et al., 2021) following the exact training recipe and hyperparameters given in Geiping & Goldstein (2023). The goal of this section is to see if increasing heads and reducing the depth and MLP ratio can still maintain good performance on a downstream task for a non auto regressive language model. We thus evaluate the models on the GLUE benchmark (Wang et al., 2018). The original model uses 16 layers, 12 heads, an embedding dimension of 768 and an MLP ratio of 4. We found that increasing the number of heads to 16 leads to better conditioned

attention Jacobian as claimed by Theorem 3.1. We then found we could reduce the depth to 9 layers and the MLP ratio to 2. The evaluation results are shown in Table 6 where we see that the leaner configuration outperforms the original with much less parameters. We ran five trials for each with different random seeds.