A Study on Transformer Configuration and Training Objective

Fuzhao Xue [1] **Jianghai Chen** [1] **Aixin Sun** [2] **Xiaozhe Ren** [3] **Zangwei Zheng** [2] **Xiaoxin He** [1] **Yongming Chen** [4] **Xin Jiang** [3] **Yang You** [1]

Abstract

Transformer-based models have delivered impressive results on many tasks, particularly vision and language tasks. In many model training situations, conventional configurations are often adopted. For example, we usually set the base model with hidden size (*i.e.,* model width) to be 768 and the number of transformer layers (*i.e.,* model depth) to be 12. In this paper, we revisit these conventional configurations by studying the the relationship between transformer configuration and training objective. We show that the optimal transformer configuration is closely related to the training objective. Specifically, compared with the simple classification objective, the masked autoencoder is effective in alleviating the over-smoothing issue in deep transformer training. Based on this finding, we propose "Bamboo", a notion of using deeper and narrower transformer configurations, for masked autoencoder training. On ImageNet, with such a simple change in configuration, the re-designed Base-level transformer achieves 84.2% top-1 accuracy and outperforms SoTA models like MAE by 0.9%. On language tasks, re-designed model outperforms BERT with the default setting by 1.1 points on average, on GLUE benchmark with 8 datasets.

1. Introduction

Transformer-based language models have achieved promising results on natural language understanding tasks, *e.g.,* Q&A (Qu et al., 2019; Yang et al., 2020), relation extraction (Xue et al., 2020; Zhou et al., 2020) and dialogue system (Ni et al., 2021). Recently, on vision tasks, transformers (Dosovitskiy et al., 2020; Zhou et al., 2021a; Xue et al., 2021; 2022) also outperform convolution-based models by a large margin. With sufficient training data, transformer-based models can be scaled to trillions of trainable parameters (Fedus et al., 2021; Du et al., 2021). Through scaling along the width (*i.e.,* hidden dimension) and depth (*i.e.,* number of transformer blocks), these huge transformers show effectiveness across various tasks and even areas.

Where are the configurations from? When using transformer, we typically follow the existing work to set the same width and depth for a "fair" comparison. For instance, we usually set the width of transformer-base model as 768 and the depth as 12. An interesting question here is: *Why do we select these hyper-parameters, even for problems in different areas?* To answer this question, we revisit the conventional configurations from some representative studies. For vision transformer (Dosovitskiy et al., 2020), authors set the base ViT configuration according to those used in BERT (Devlin et al., 2018). BERT selects such configuration following OpenAI GPT (Radford et al., 2018). OpenAI also follows the original transformer paper (Vaswani et al., 2017). In the original transformer paper, Vaswani et al. (2017) conduct a set of ablation studies on machine translation task to find the optimal configurations. That is, for a good range of tasks, we have largely followed the transformer configuration based on an ablation study on machine translation task, *i.e.,* a sequence-to-sequence task.

Should we use the same configuration for different training objectives? Nowadays, transformer-based models can be trained with various training objectives or strategies (Tay et al., 2022a;b). Taking the vision transformer (Dosovitskiy et al., 2020) as an example, we can train transformer from scratch with a supervised learning setting for image classification. In this straightforward image classification task, each image is modeled as a sequence of tokens, and each token corresponds to a patch in the image. We use the global information (from all tokens/patches of the image) to predict a single label, the category of the image. Here, as the training objective is to capture the global information of an image, the differences between token representations would not be considered directly. This image classification task is quite different from machine translation task, which requests for a strong understanding of a token sequence and

[1]School of Computing, National University of Singapore [2]School of Computer Science and Engineering, Nanyang Technological University [3]Huawei Noah's Ark Lab [4]School of Electrical and Electronic Engineering, Nanyang Technological University. Correspondence to: Fuzhao Xue <f.xue@u.nus.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

generating another sequence. Hence, intuitively, it is natural to assume that different optimal transformer configurations exist for these two different tasks.

Over-smoothing issue of the simple classification training objective. Previous work has tried to train a deeper transformer from scratch. However, as reported in (Zhou et al., 2021a; Gong et al., 2021), training by classification task (*i.e.,* using the global signal of the input sequence) has the over-smoothing problem. That means, at the deeper transformer layers, all token representations tend to be identical (Brunner et al., 2020). Such issue harms the scalability of training vision transformer, especially the scaling along depth. When scaling to a larger model, we only get a slight improvement or even poorer accuracy. Recently, Zhou et al. (2021a); Gong et al. (2021) show that, when adding special-designed regularization to avoid the "uniform tokens" (*i.e.,* the over-smoothing problem), it is possible to train a deeper transformer on the sequence (image) classification setting.

Masked autoencoder can scale to deeper and wider models without additional training data. Different from training from scratch above, the masked autoencoder is a two-stage training framework, including pre-training and fine-tuning. Given a partially masked input sequence, the pre-training stage aims to recover the original unmasked sequence. The fine-tuning is similar to the aforementioned training from scratch but requires much fewer training epochs. With the masked autoencoder, recent studies (Bao et al., 2021; He et al., 2021) successfully train large-scale transformers, even without using additional training data compared to supervised learning. This is counterintuitive because we usually assume more available training data is the key of self-supervised learning to improve effectiveness. This result motivates us to rethink the reason behind it.

Masked autoencoder can alleviate the over-smoothing issue. Intuitively, in masked autoencoder frameworks (*e.g.,* BERT, BEiT), the target is to recover the masked tokens based on the unmasked tokens. Compared to training transformer from scratch under supervised learning, whose target is a simple classification task, the masked autoencoder framework adopts a sequence labeling target. We hypothesize that the masked autoencoder training can alleviate the over-smoothing issue, which is a possible reason why the masked autoencoder can help to scale transformer up. Specifically, the sequence labeling task requires the model to learn semantic information from neighboring unmasked tokens. Since different masked tokens have different unmasked neighboring tokens, the unmasked token representations must carry their corresponding and sufficient semantics for the accurate prediction of the masked tokens, which in turn prevents the token representations to become identical (or very similar to each other). In a word, we may infer that the masked autoencoder's training objective helps to alleviate the over-smoothing problem by its regularization on token differences. To justify the reasoning above, we conduct an experimental investigation in Section 2.1. The results show that the over-smoothing issue is indeed alleviated in the masked autoencoder. Compared to training under masked autoencoder, training transformer by a simple classification task (*e.g.,* training vision transformer from scratch) does not have such benefit.

Why and how masked autoencoder alleviates over-smoothing? We further explore the reason behind this phenomenon via Fourier domain analysis in Section 2.2. First, self-attention layer in transformer will decay the high-frequency component of input signal (Wang et al., 2022b). When all high-frequency components are erased, all token representations would be identical. We find the masked autoencoder training objective can be seen as reconstructing the high-frequency components (HC) of input signal from the HC of the noisy masked input signal. Therefore, masked autoencoder can alleviate over-smoothing via learning a slower HC decay rate. Such ability is achieved by training the weights in self-attention layer. To further verify this finding, we conduct quantitative analysis in Section 2.3 and results show that, compared with the model trained with simple classification objective, the trainable matrices in model trained with masked autoencoder objective indeed has slower HC decay.

Potential of masked autoencoder with deeper configurations. If the masked autoencoder alleviates the over-smoothing issue (which is a challenge for scaling transformer along depth), does this mean the masked autoencoder can get more benefits from deep configurations? To answer this question, we re-visit the configurations for different training objectives, especially for the masked autoencoder. Accordingly, we conduct experiments to investigate the masked autoencoder configurations and propose our idea, Bamboo[1]. When training transformer with masked autoencoder, we suggest using deeper and narrower configurations with comparable computation budget as a typical setting, to achieve better effectiveness. To evaluate our new configurations, we conduct comprehensive experiments on computer vision and natural language processing tasks. On vision tasks, we evaluate our configuration on large-scale vision transformer training. With Bamboo configuration, the masked autoencoder outperforms baseline by a large margin. For instance, on ImageNet, with a comparable number of trainable parameters and computational cost, our narrower and deeper base-scale masked autoencoder, Bamboo-B, outperforms MAE-B by 0.9% in terms of top-1 accuracy. On natural language processing tasks, we conduct experiments on BERT. Results show that our configurations can improve

[1]The narrower and taller shape of the re-designed transformer looks like bamboo.

BERT-L by 1.1 points on GLUE datasets.

Contributions In summary, our main contributions are three folds: 1) We first study the relationship between transformer configuration and training objective, and then propose the insight that the masked autoencoder helps transformer to handle over-smoothing, although there can be other cofounders like training stability. We show this finding by experimental investigation, and more importantly, by theoretical reasoning on Fourier domain and verify our reasoning via quantitative analysis; 2) We argue that the existing transformer configurations cannot fully use the strength of masked autoencoder. To this end, we propose Bamboo, an idea to scale transformer along depth when training with masked autoencoder. We show that the narrower and deeper versions overperform existing configurations, in a plug-and-play manner; 3) We further verify our Bamboo configurations on larger scale vision transformer pre-training and natural language tasks. Results show that our Bamboo achieves state-of-the-art top-1 accuracy on image classification, and outperforms the original BERT configurations by 1.1 points on GLUE.

TL;DR for Practitioners In this paper, the most important thing we want to highlight is, not to underestimate the training objective before tuning model configuration. Usually, for a fair comparison, we simply adopt the previous configurations. However, sometimes, one training objective may look decent if it wins the "configuration lottery"[2]. However, for a different objective, the effectiveness would be underestimated without a configuration sweep. We may then miss a good training objective for the community. Therefore, to know about the potential of each novel training objective design, we strongly suggest practitioners analyze the inductive bias and customize configurations. Our paper shows one example of such analysis on MAE.

The following sections are organized based on the analysis process of this work. In Section 2, we briefly review the over-smoothing problem in transformer and show the strength of masked autoencoder in handling this issue. We then conduct experiments to investigate scaling masked autoencoder along the depth in Section 3. Based on the consistent sweet depth across scales, we suggest the Bamboo idea, using narrower and deeper configurations in masked autoencoder training. Then, we adapt the new configurations to a larger scale, and conduct evaluations across different areas, vision tasks in Section 4 and NLP tasks in Section 5. Finally, we discuss the difference between this work and the related work in Section 6.

[2]The previous used configuration matches well with the new training objective.

2. Over-smoothing under Different Training Objectives

The over-smoothing issue is well noted in graph neural networks. When we stack many graph convolution networks, the node representations tend to be identical (Chen et al., 2020). Recent studies show that transformer has a similar problem, known as "uniform tokens" (Shi et al., 2021). In deep transformer, each token representation can be seen as a node in a graph, and each attention score is an edge. Different token representations tend to be identical when we scale transformer across depth. In this work, we mainly focus on the over-smoothing problem under different training objectives, *i.e.,* sequence-level supervised learning, and masked antoencoder-based self-supervised learning. To compare these two objectives, we use supervised vision transformer (ViT) (Dosovitskiy et al., 2020) and vision masked autoencoder (MAE) (He et al., 2021) as two representative platforms to show our insights.

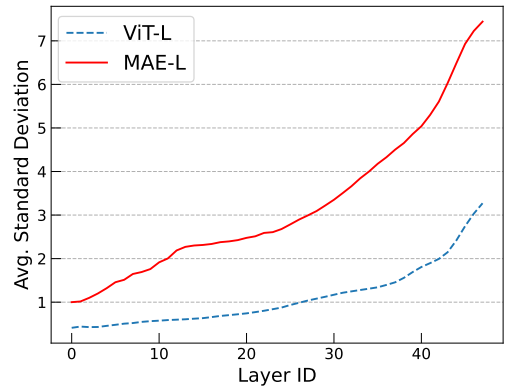
2.1. Experimental Investigation

Standard deviation. We first conduct two sets of quantitative analysis to study the over-smoothing behaviour under the two training objectives. For the first set of quantitative analysis, we focus on the standard deviation of token representations at different transformer blocks. Formally, transformer block's output (*i.e.,* token representations) can be written as $h = \{h_0, h_1, ..., h_{T-1}\}$, where $h_t \in \mathbb{R}^d$, d is the hidden dimension, T is the number of tokens at different transformer blocks. Usually, T is fixed across transformer blocks. For the token representations after the l^{th} transformer block, we denote the token representations as h^l. To compare the over-smoothing issue between supervised learning and masked autoencoder, we calculate the mean standard deviation of the token representations at different transformer layers:

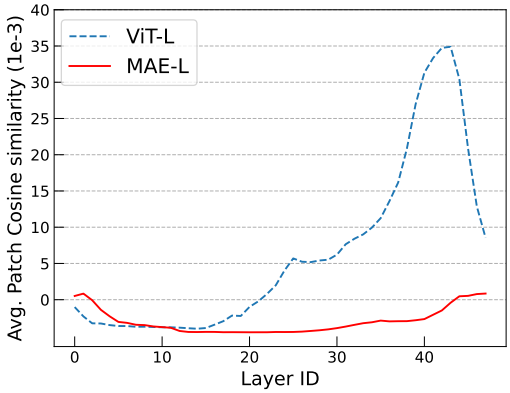
$$ms^i = \sqrt{\frac{1}{T-1}\sum_{t=0}^{T-1}\left(h_t^i - \bar{h^i}\right)^2} \tag{1}$$

where $\bar{h^i} = \frac{1}{T-1}\sum_{t=0}^{T-1} h_t^i$ is the mean vector of token representations at the i^{th} transformer block. We use a deeper version (48 layers) of ViT-L and MAE-L as platform[3] to validate our analysis above. Specifically, the ViT-L is trained on the supervised learning setting and treats image classification as a simple classification task, by average pooling prediction head. The MAE-L is trained on the masked autoencoder setting, which means the pre-training target is close to a sequence labeling task. We then fine-tune the

[3]The default configuration of ViT-L and MAE-L includes only 24 layers. We use deeper configuration to expose the over-smoothing issue more clearly.



(a) Average standard deviation



(b) Average patch-pair cosine similarity

Figure 1: Over-smoothing analysis of ViT and MAE with the same configuration (48 layers). We calculate the average standard deviation and average patch cosine similarity at every layer to compare the over-smoothing level.

pre-trained model for image classification. We set the width as 768 and the depth as 48. The total number of trainable parameters and FLOPs is comparable to the original configuration (*i.e.,* width is 1024, depth is 24). The the mean standard deviation of the token representations is shown in Figure 1(a).

Observe that the mean standard deviations of ViT-L and MAE-L both increase along depth. Intuitively, this contradicts with our expectation as the over-smoothing leads to similar token representations. As stated in (Dong et al., 2021), residual connection within transformer indeed alleviates over-smoothing (to some extent). Nevertheless, the over-smoothing issue still exists even if we have residual connection (Dong et al., 2021; Shi et al., 2021). We can observe that the mean standard deviation increasing on MAE-L is much faster than that on ViT-L. That means the deeper transformer blocks can learn different semantics for different token representations in MAE. In other words, the over-smoothing issue is much less pronounced in the transformer model pre-trained with the masked autoencoder setting.

Patch-pair cosine similarity. We further verify that the over-smoothing issue can be alleviated in masked autoencoder. Following Gong et al. (2021), we compare the patch-pair cosine similarity between ViT-L and MAE-L with deeper and narrower configuration. If there is an over-smoothing issue in the model, we should observe that the patch-pair cosine similarity increases along depth. The more serious the over-smoothing issue is, the faster the cosine similarity increases. To remove the impact from input representations (residual connection), we use the zero-centered token representations $\tilde{h^i} = h^i - \bar{h^i}$ for evaluation. The results are shown in Figure 1(b). Generally, the cosine similarity of ViT increases along the depth due to over-smoothing. However, for the model pre-trained by the masked autoencoder framework, the cosine similarity keeps constant along depth. This comparison is interesting, as we can barely observe the over-smoothing issue on the model pre-trained by the masked autoencoder, even if we are using a deeper model than usual.

2.2. Theoretical Analysis

The reason why over-smoothing happens in Transformer has been well-studied (Dong et al., 2021; Wang et al., 2022b). Conceptually, each token representation can be seen as node in a directed graph and each attention score is a weighted edge. Recent study (Wang et al., 2022b) proposes to understand the transformer over-smoothing issue via the Fourier domain analysis by giving a closer examination of model architecture. However, existing work ignores that training objective also has relation with over-smoothing. This paper adapts their theorem as basis to reason why MAE can alleviate over-smoothing.

Given a Discrete Fourier Transform (DFT) $\mathcal{F}$: $\mathbb{R}^N \rightarrow \mathbb{C}^N$, the Inverse Discrete Fourier Transform (IDFT) $\mathcal{F}^{-1}$: $\mathbb{C}^N \rightarrow \mathbb{R}^N$, and input signal $\mathbf{x} \in \mathcal{R}^N$, let $\mathbf{z} = \mathcal{F}\mathbf{x}$ be the spectrum of x. $\mathbf{z}_{\mathcal{DC}} \in \mathbb{C}$ and $\mathbf{z}_{\mathcal{HC}} \in \mathbb{C}^{N-1}$ take the first element and the rest elements of $\mathbf{z}$, respectively. The DFT here can be implemented by left multiplying a pre-defined DFT matrix whose k^{th} row is Fourier basis $f_k = [e^{2\pi j(k-1)\cdot 0}, \ldots, e^{2\pi j(k-1)\cdot(N-1)}]^T/\sqrt{N}$, where j denotes the imaginary unit and k denotes the k-th row of DFT matrix. Therefore, for signal $\mathbf{x}$, we have the Direct-Current (DC) component $\mathcal{DC}[\mathbf{x}] = f_1\mathbf{x}$ and the complementary high-frequency component $\mathcal{HC}[\mathbf{x}] = [f_2, \ldots, f_N]\mathbf{x}$.

Based on the definition above, given input token sequence $\mathbf{X} \in \mathcal{R}^{N\times d}$, its i-th channel $\mathbf{x}_i \in \mathcal{R}^N$, and radius of a ball $\gamma > 0$, assuming $||\mathbf{x}_i||_2 \leq \gamma^2$, Wang et al. (2022b) proposes and proves:

$$||\mathcal{HC}[\mathrm{SA}(\mathbf{X})]||_F \leq \tau ||\mathcal{HC}[\mathbf{X}]||_F \tag{2}$$

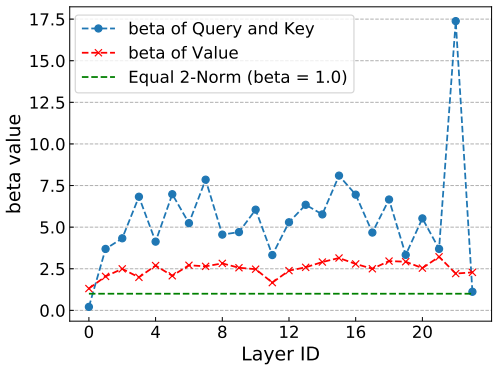
$$\tau = \sqrt{\frac{ne^{2\alpha}}{e^{2\alpha}+n-1}}||\mathbf{W}_V||_2 \tag{3}$$

$$\alpha \leq \frac{\gamma^2 ||\mathbf{W}_Q \mathbf{W}_K{}^T||_2}{\sqrt{d}} \tag{4}$$

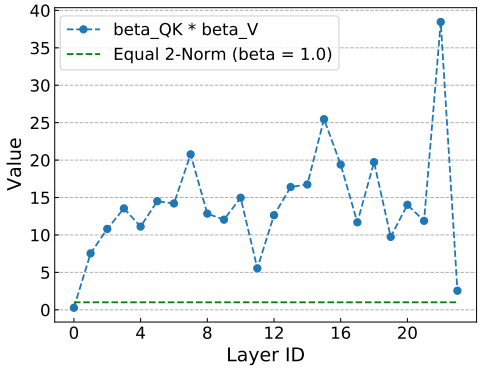
where $\mathrm{SA}(\cdot)$ denotes self-attention (Vaswani et al., 2017), $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$ are query, key and value trainable matrices. When $\tau < 1$, $\mathcal{HC}[\mathrm{SA}(\mathbf{X})]$ will decay to zero exponentially. As a comparison, the DC component would not be affected by the attention scores. After applying a number of attention layers, which is exactly what we do in Transformer, the DC component would dominate the hidden representations and thus over-smoothing happens.

Then, our first question is, can we understand the reason why MAE can alleviate the over-smoothing with Fourier analysis? For MAE, we mask parts of the ground truth $\mathbf{X}$ as $\mathbf{X}^m$. We define the mask function as $\mathbf{X}^m = \mathcal{M}(\mathbf{X})$. The masked parts in $\mathbf{X}^m$ are filled with the DC Component as what we usually do in popular MAE (*e.g.,* BERT). We usually use $\mathbf{X}^m$ as the model input. During training, model is minimizing the distance between $\mathbf{X}_L^m$ and $\mathbf{X}$, where $\mathbf{X}_L^m$ is the hidden representation after L transformer layers. If we transform $\mathbf{X}$, $\mathbf{X}^m$ and $\mathbf{X}_L^m$ to Fourier domain like $\mathbf{Z} = \mathcal{F}\mathbf{X}$, $\mathbf{Z}^m = \mathcal{F}\mathbf{X}^m$, and $\mathbf{Z}_L^m = \mathcal{F}\mathbf{X}_L^m$, the learning objective is minimizing the distance between $||\mathcal{HC}[\mathbf{Z}]||_F$ and $||\mathcal{HC}[\mathbf{Z}_L^m]||_F$. In $\mathcal{M}(\cdot)$, we replace the original signal including both DC and HC components with the constant mask signal with the DC component only, so we can assume $||\mathcal{HC}[\mathbf{Z}^m]||_F < ||\mathcal{HC}[\mathbf{Z}]||_F$. Then, we obtain $||\mathcal{HC}[\mathbf{Z}_L^m]||_F < ||\mathcal{HC}[\mathbf{Z}^m]||_F < ||\mathcal{HC}[\mathbf{Z}]||_F$. Obviously, the learning objective of MAE is pushing $||\mathcal{HC}[\mathbf{Z}_L^m]||_F \approx ||\mathcal{HC}[\mathbf{Z}]||_F$ and that would make the $||\mathcal{HC}[\mathbf{Z}_L^m]||$ closer to its upper bound (*i.e.,* $||\mathcal{HC}[\mathbf{Z}]||_F$) during MAE training, which means the smoothing rate τ in Eq. 2 would be pushed towards greater implicitly to avoid the decay of high-frequency information. As a comparison, the model trained via simple classification target can still finish the task using totally identical token representations with DC component only. The reason is that the high-frequency information decay is not regularized by training objective.

The next question is whether τ can be trained towards greater. The answer is yes. First, τ is positive related with $||\mathbf{W}_V||_2$. At the same time, α in Eq. 3 has an upper bound in Eq. 4. When model tends to provide more upside potential for α, it has to increase $||\mathbf{W}_Q\mathbf{W}_K^T||_2$ or at least ensure $||\mathbf{W}_Q\mathbf{W}_K^T||_2$ would not decay during training. Formally, let $n > 2$, we can easily find $\frac{\partial \tau}{\partial \alpha} > 0$ and $\frac{\partial \tau}{\partial ||\mathbf{W}_V||_2} > 0$. According to Eq. 4, we can increase the upper-bound of α via increasing $||\mathbf{W}_Q\mathbf{W}_K^T||_2$. Therefore, MAE can implicitly amplify the smoothing rate by increasing the value of $||\mathbf{W}_Q\mathbf{W}_K^T||_2$ and $||\mathbf{W}_V||_2$, and then alleviate the over-smoothing of Transformer.



(a) Quantitative verification of β_{QK} and β_V



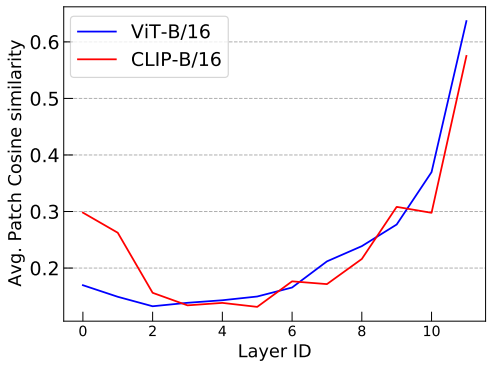
(b) Quantitative verification of combining β_{QK} and β_V

Figure 2: Quantitative verification of our theoretical reasoning. $beta > 1.0$ means the MAE is alleviating the over-smoothing issue at this layer.

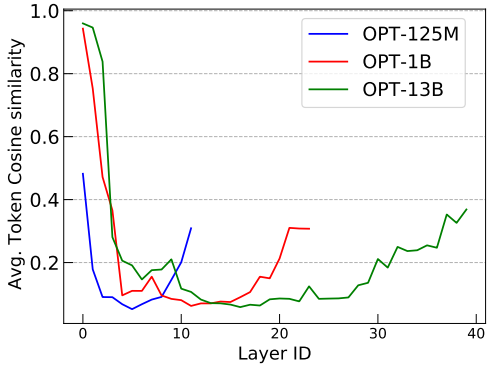
2.3. Quantitative Verification

We conduct quantitative verification to check our reasoning that MAE can implicitly maximize the smoothing rate. We initialize two large scale Transformer models (ViT-L/16 and MAE-L/16) with the same initializer but train them with ViT training objective and MAE pre-training objective, respectively. If our reasoning is correct, we can expect the model trained with MAE objective has greater $||\mathbf{W}_Q\mathbf{W}_K^T||_2$ and $||\mathbf{W}_V||_2$ than model trained with simple classification objective.

We define $\beta_{QK} = \frac{||\mathbf{W}_Q^{\mathrm{MAE}}\mathbf{W}_K^{\mathrm{MAE}\,T}||_2}{||\mathbf{W}_Q^{\mathrm{ViT}}\mathbf{W}_K^{\mathrm{ViT}\,T}||_2}$ and $\beta_V = \frac{||\mathbf{W}_V^{\mathrm{MAE}}||_2}{||\mathbf{W}_V^{\mathrm{ViT}}||_2}$. $\beta > 1$ means MAE tends to obtain larger smoothing rate than simple supervised learning. In that case, the MAE training objective is alleviating over-smoothing implicitly. We visualize the β value of different layers. The comparison

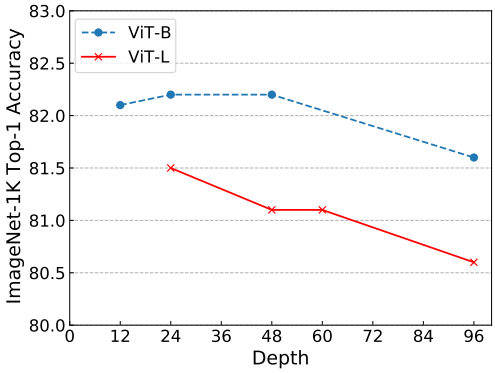


(a) CLIP patch-pair cosine similarity

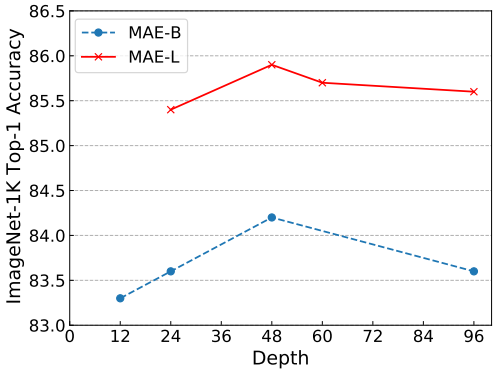


(b) OPT token-pair cosine similarity

Figure 3: Over-smoothing analysis of CLIP and OPT.



(a) Scaling supervised ViT along depth



(b) Scaling MAE along depth

Figure 4: Scaling ViT and MAE with comparable number of parameters and computation cost. For deeper model, we compress the width to ensure a fair comparison.

is shown in Figure 2. We found that, even if we initialize ViT and MAE with the same initializer, both β_{QK} and β_V are significantly greater than 1.0 (green line) for most layers after training, which matches well with our expectation.

2.4. Over-smoothing on Different Objectives

Based on the analysis presented above, we can conclude that token-level training objectives, such as next-token prediction in Language Modeling, exhibit a less severe over-smoothing issue. On the other hand, sequence-level objectives, like contrastive image pre-training, are more prone to over-smoothing. To validate this conclusion, we conducted cosine similarity experiments using CLIP (Radford et al., 2021) and OPT (Zhang et al., 2022). Figure 3(a) presents the results of the CLIP model, demonstrating a similar over-smoothing behavior to Vanilla ViT (Vision Transformer). This observation aligns with our expectations. Furthermore, to investigate whether the over-smoothing issue can be mitigated by next token prediction, a widely employed LLM pre-training objective, we evaluated OPT and found that it effectively addresses over-smoothing. This finding is significant as it helps to elucidate why LLM models exhibit greater scalability compared to numerous vision models.

3. Bamboo

Our analysis above shows that the masked autoencoder helps to alleviate the over-smoothing issue, which is a main challenge in scaling transformer along depth (Zhou et al., 2021a; Dong et al., 2021). To realize this potential, we suggest that we may obtain better performance by adapting deeper and narrower model configuration when training with MAE objective. We name this idea as Bamboo in this paper due to the shape of new transformer configuration. Certainly, infinite deeper and narrower configurations do not always improve performance, as there are a few other reasons hindering scaling transformer along depth. In this section, we devote to find a sweet point following the Bamboo idea, to achieve effective masked autoencoder by experiments.

To validate our reasoning above, we conduct experiments on training transformer on ILSVRC-2012 ImageNet (Deng et al., 2009) (ImageNet-1K), and report the top-1 accuracy. We use the original ViT trained by supervised learning and MAE pre-trained masked image modeling as the test platform. For a fair comparison with the original transformer configurations, when we scale transformer to deeper, we

Table 1: The configurations we used to scale transformer along depth. For a fair comparison, we keep a comparable computation cost with the original transformer configurations (*i.e.,* depth=12 for Base, depth=24 for Large).

Scale	Base				Large			
Depth	12	24	48	96	24	48	60	96
Width	768	512	384	256	1024	768	640	512
#Attention Heads	12	8	6	4	16	12	10	8
FLOPs	1×	0.9×	1×	0.9×	1×	1.1×	1×	1×

also reduce the width to keep comparable number of trainable parameters and computation cost. The configurations we used are summarized in Table 1.

For supervised learning models (*i.e.,* ViT), we train base scale models for 300 epochs and large scale models for 200 epochs following He et al. (2021). We use RandAugment (Cubuk et al., 2020), drop path (Huang et al., 2016), mixup, cutmix (Yun et al., 2019), label smoothing (Szegedy et al., 2016) for data augmentation. Detailed hyper-parameters are summarized in Appendix A. For the masked autoencoder, we pre-train the base scale models for 1600 epochs and fine-tune for 100 epochs. For large models, we pre-train for 800 epochs and fine-tune for 50 epochs.

The results of scaling to deeper transformers are summarized in Figure 4. For both the base-scale and large-scale models, we report the top-1 accuracy on ImageNet-1K dataset. We can observe the models that are trained by supervised image classification directly (*i.e.,* ViT-B and ViT-L) cannot improve accuracy when we use deeper architectures. For ViT-B, the top-1 accuracy on ImageNet-1K remains the same when the depth is smaller than 50, but after that, there is a significant drop on accuracy. For ViT-L, the accuracy decreases even faster than the ViT-B. There is a significant drop on accuracy when we use the narrower and deeper models. However, for masked autoencoders, we observe totally different patterns. Even if we keep the comparable trainable parameters and computation cost, with such a simple modification, the masked autoencoders gain significant improvements. More importantly, when we scale to 48 layers, both MAE-B and MAE-L reach sweet spots. When scaling to 96 layers, we observe the training is unstable compared with shallower models. This result matches well with our observation in Figure 1(b). At around 40th layer, the over-smoothing issue starts to happen slightly in masked autoencoder, which is much later than the ViT model. We suggest another reason of the unstable deep model training is the too-large model updates (Wang et al., 2022a). In this work, we focus on the over-smoothing issue of deep transformer training instead. The large model updates in deeper layers are out-of-scope. We leave that as our future work.

According to the experimental results above, we find that masked autoencoder can indeed scale transformer well along depth. Even if we keep comparable trainable parameters and computation cost with the original transformer, the model achieves better accuracy. Another observation is, both transformer-base and transformer-large reach their sweet spots at around 50 layers. We thus recommend a new set of transformer configurations in Table 2 following our Bamboo idea, which are deeper and narrower than the original transformer configurations.

4. Evaluation on Vision Tasks

4.1. Settings

We further evaluate our deeper and narrower configurations on vision task. We train different models with more training epochs and compare with SoTA vision models. We conduct experiments on ImageNet-1K and compare with recent supervised vision models *e.g.,* DeepViT (Zhou et al., 2021a) and DeiT (Touvron et al., 2021), and self-supervised vision models *e.g.,* DINO (Caron et al., 2021), MoCo v3 (Chen et al., 2021), BEiT (Bao et al., 2021) and MAE (He et al., 2021). Compared with the MAE, *the only difference is the configurations*. That is, MAE uses original transformer configurations and we use our Bamboo configurations. The experiments are to verify that such a simple modification can improve model effectiveness and show the potential of more reasonable configurations.

We evaluate the models on three different scales, *i.e.,* base, large, and huge. The data augmentation setting is exactly the same as MAE for a fair comparison. Again, the only difference is that we use the Bamboo configurations instead of the original transformer configurations. During fine-tuning, we use the same script with training ViT from scratch. We fine-tune base models for 100 epochs. For large and huge models, we fine-tune them for 50 epochs following existing work (Bao et al., 2021; He et al., 2021).

4.2. Results

Results on ImageNet-1K are reported in Table 3. For training ViT from scratch, we report the original results (Dosovitskiy et al., 2020) and the results with strong data augmentation (He et al., 2021). A few recent work (Zhou et al., 2021a; Touvron et al., 2021) focusing on training ViT from scratch on ImageNet-1K are also included. In general, we can find the models pre-trained by self-supervised learning (*e.g.,* DINO, MoCO v3, MAE) perform much better than training from scratch. If we only consider the self-supervised

Table 2: Re-designed configurations under Bamboo idea. The computation cost denotes the FLOPs compared with the original configuration.

Scale	Base		Large		Huge	
	Original	Bamboo	Original	Bamboo	Original	Bamboo
Depth	12	48	24	48	32	64
Width	768	384	1024	768	1280	896
#Attention Heads	12	6	16	12	16	14
Computation cost	1×	1×	1×	1.1×	1×	1×

Table 3: Top-1 accuracy on ImageNet-1K. We report two versions of ViT training from scratch. The first one is from original ViT paper (Dosovitskiy et al., 2020), and the second one is from He et al. (2021)'s re-implementation with strong data augmentation. For MAE-B, we reproduce the results by running the official code and obtain a slightly different result (denoted by 83.3*). The original result is 83.6.

Method	Pre-train Data	Base	Large	Huge
ViT from scratch (Dosovitskiy et al., 2020)	-	77.9	76.5	-
DeepViT (Zhou et al., 2021a)	-	80.9	-	-
DeiT (Touvron et al., 2021)	-	81.8	-	-
ViT from scratch (He et al., 2021)	-	82.1	81.5	80.9
DINO (Caron et al., 2021)	IN1K	82.8	-	-
MoCo v3 (Chen et al., 2021)	IN1K	83.2	84.1	-
BEiT (Bao et al., 2021)	IN1K + DALL-E	83.2	85.2	-
MaskFeat (Wei et al., 2021)	IN1K	84.0	85.7	-
IBOT (Zhou et al., 2021b)	IN1K	84.0	84.8	-
MAE (He et al., 2021) (Direct baseline)	IN1K	83.3*	85.9	86.9
Bamboo (Ours)	IN1K	**84.2**	**86.3**	**87.1**

learning approaches, masked image modeling-based methods (*e.g.,* BEiT, MaskFeat, IBOT, MAE) outperform the contrastive learning-based methods (*e.g.,* DINO, MoCo v3) significantly, especially on larger scale.

Since we are focusing on the scalability of training transformer with masked autoencoder, we choose MAE as our direct baseline. We train MAE with Bamboo configurations and report the results in Table 3. Observe that our Bamboo achieves the best top-1 accuracy on all scales. On the base scale, Bamboo achieves state-of-the-art performance, 84.2 top-1 accuracy, which is 0.9 (0.6) points higher than MAE-B. When we scale the model up to large scale and huge scale, Bamboo remains the best performer and achieves 86.3 and 87.1 top-1 accuracy respectively. Note that, compared to other scales, the improvement is not so significant on the huge scale. One reason is, the original huge configuration has been deep (*i.e.,* 32 layers), which is close to the sweet point. Similarly, this can also explain why the configurations designed under Bamboo can improve MAE-B significantly. Since the real run time may be influenced by many other factors (*e.g.,* GPU or TPU utilization), we report the real throughput in the Appendix B.

5. Evaluation on Language Tasks

We further evaluate our Bamboo configurations on language tasks. We select BERT (Devlin et al., 2018) as the platform to evaluate our Bamboo configuration because it is widely used on many language tasks. Note that BERT is a post-layer normalization transformer model. Better performance on BERT means our design can generalize to other architectures. We follow the BERT paper to use Wikipedia and bookscorpus to pre-train. During pre-training, we use LAMB optimizer and set batch size and learning rate as 4096 and 1.76e-3, respectively, following You et al. (2019).

During fine-tuning, we conduct experiments on General Language Understanding Evaluation (GLUE) benchmark. The GLUE benchmark (Wang et al., 2018) is widely used in natural language understanding tasks, which include 8 tasks, *i.e.,* CoLA, MNLI, MRPC, QNLI, QQP, RTE, SST-2 and STS-B. We set the learning rate as 1e-5 or 2e-5. Batch size is fixed as 32. For small datasets, *i.e.,* CoLA, MRPC, RTE and STS-B, we fine-tune for 10 epochs. For larger datasets, *i.e.,* MNLI, QNLI, QQP and SST-2, we fine-tune for 3 epochs. Matthew's correlation is used as metric for CoLA. For MNLI, we report the average accuracy on MNLI-m and MNLI-mm. QNLI and RTE also adapt the accuracy as metric. The results on MRPC and QQP are reported with the average of F1 and accuracy. We use Spearman correlation on STS-B. We run the code 5 times and report

Table 4: Results of fine-tuning on GLUE benchmark.

Method	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Avg
BERT-B	59.6	83.7	88.0	90.4	88.8	68.6	91.5	89.4	82.5
Bamboo-B	**60.5**	**84.3**	**88.4**	**90.5**	**89.0**	**70.0**	**92.2**	**89.5**	**83.1**
BERT-L	60.9	86.2	89.3	92.3	**89.6**	73.1	92.5	90.4	84.3
Bamboo-L	**62.9**	**87.1**	**89.8**	**92.4**	89.4	**77.3**	**93.8**	**90.6**	**85.4**

the median for fine-tuning.

The results on language tasks are reported in Table 4. Under the same pre-training and fine-tuning settings, models with Bamboo configurations outperform BERT significantly. Bamboo-L achieves the best performance in Table 4. Compared with BERT-L, our Bamboo-L wins on 7 out of 8 datasets, and surpassed BERT-L by 1.1 points on average. It is also notable Bamboo can outperform baselines on both large datasets (*e.g.,* MNLI) and small datasets (*e.g.,* CoLA).

6. Discussion

Compared with simply scaling along depth, this work maintains a comparable computation cost and the number of trainable parameters. When scaling along depth, we also make the deep transformer narrower. Under such setting, the deeper and narrower configurations re-designed under the Bamboo idea can still outperform the baseline configurations, suggesting that we should consider narrower and deeper transformer when training by masked autoencoder. One related work is (Tay et al., 2021), an empirical study of practical scaling of transformer, which has a similar observation, deeper and narrower can improve accuracy. On this basis, our finding is consistent with that in (Tay et al., 2021). However, we are not simply comparing different configurations (Tay et al., 2021) or training objectives (Voita et al., 2019). We are actually bridging the gap and study the relation between configurations and the training objective. Another related work (Levine et al., 2020) investigates the optimal depth-to-width ratio for different scales. However, they do not tackle the impact of training objectives.

Compared with brute-force hyper-parameter tuning, we provide both theoretical reasoning and quantitative analysis to justify our insight, *i.e.,* masked autoencoder alleviates over-smoothing issue in transformer. Motivated by this, we suggest using deeper and narrower model for masked autoencoder. There is no guarantee to ensure the configurations are always optimal. We believe it is impossible in deep learning to know the optimal configurations before experiments. However, we argue that the masked autoencoder gets more benefits from deeper configurations. Our insight may instruct future work to consider configurations according to the training objectives.

Instead of proposing a new approach or a new set of configuration, this work focuses on an existing but neglected problem. After revisiting, we find configurations should be re-designed for different training objectives. We highlight this is important as following the conventional configuration is not the real "fair" comparison. If we keep do this in the future, we will always pick the training objective that wins the "configuration lottery", and that would miss some real effective objective with great potential. The analysis sections in this paper can be seen as an example of how to design configurations for different objectives. After simply re-designing a set of configurations for masked autoencoder, we can see a significant improvement on both vision and language tasks. On vision tasks, we even achieve SoTA top-1 accuracy on ImageNet. However, note that we highlight our main contribution is an insight instead of a SoTA model, and it is orthogonal to the future transformer modifications.

7. Conclusion

In this work, we first study the relationship between transformer configuration and the training objective. Compared with supervised learning, training transformer with MAE can alleviate over-smoothing. We then explore the reason behind this finding through theoretical reasoning and quantitative verification via Fourier domain analysis. Under this insight, we rethink the widely used configurations in vision and language tasks, and suggest deeper and narrower configurations when training with MAE. To further verify the effectiveness of our configuration, we conduct comprehensive experiments on both large-scale vision and language tasks and achieve significant improvement with such a simple modification. More importantly, we argue that using a configuration for a fair comparison may not be really fair. That may underestimate the potential of a new training objective who does not win the "configuration lottery". We suggest analyzing the inductive bias of each objective and sweeping the configuration following the analysis.

Acknowledgement

This work is being sponsored by Huawei Noah's Ark Grant.

References

Bao, H., Dong, L., and Wei, F. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Brunner, G., Liu, Y., Pascual, D., Richter, O., Ciaramita, M., and Wattenhofer, R. On identifiability in transformers. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJg1f6EFDB.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., and Sun, X. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3438–3445, 2020.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9640–9649, 2021.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dong, Y., Cordonnier, J.-B., and Loukas, A. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, pp. 2793–2803. PMLR, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., Krikun, M., Zhou, Y., Yu, A. W., Firat, O., et al. Glam: Efficient scaling of language models with mixture-of-experts. *arXiv preprint arXiv:2112.06905*, 2021.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021.

Gong, C., Wang, D., Li, M., Chandra, V., and Liu, Q. Vision transformers with patch diversification. *arXiv preprint arXiv:2104.12753*, 2021.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. Deep networks with stochastic depth. In *European conference on computer vision*, pp. 646–661. Springer, 2016.

Levine, Y., Wies, N., Sharir, O., Bata, H., and Shashua, A. The depth-to-width interplay in self-attention. *arXiv preprint arXiv:2006.12467*, 2020.

Ni, J., Young, T., Pandelea, V., Xue, F., Adiga, V., and Cambria, E. Recent advances in deep learning based dialogue systems: A systematic survey. *arXiv preprint arXiv:2105.04387*, 2021.

Qu, C., Yang, L., Qiu, M., Croft, W. B., Zhang, Y., and Iyyer, M. Bert with history answer embedding for conversational question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1133–1136, 2019.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. 2018.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Shen, L., Wu, Z., Gong, W., Hao, H., Bai, Y., Wu, H., Wu, X., Xiong, H., Yu, D., and Ma, Y. Se-moe: A scalable and efficient mixture-of-experts distributed training and inference system. *arXiv preprint arXiv:2205.10034*, 2022.

Shi, H., Gao, J., Xu, H., Liang, X., Li, Z., Kong, L., Lee, S. M., and Kwok, J. Revisiting over-smoothing in bert from the perspective of graph. In *International Conference on Learning Representations*, 2021.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Tay, Y., Dehghani, M., Rao, J., Fedus, W., Abnar, S., Chung, H. W., Narang, S., Yogatama, D., Vaswani, A., and Metzler, D. Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*, 2021.

Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Bahri, D., Schuster, T., Zheng, H. S., Houlsby, N., and Metzler, D. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*, 2022a.

Tay, Y., Wei, J., Chung, H. W., Tran, V. Q., So, D. R., Shakeri, S., Garcia, X., Zheng, H. S., Rao, J., Chowdhery, A., et al. Transcending scaling laws with 0.1% extra compute. *arXiv preprint arXiv:2210.11399*, 2022b.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, 2017.

Voita, E., Sennrich, R., and Titov, I. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. *arXiv preprint arXiv:1909.01380*, 2019.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

Wang, H., Ma, S., Dong, L., Huang, S., Zhang, D., and Wei, F. Deepnet: Scaling transformers to 1,000 layers. *arXiv preprint arXiv:2203.00555*, 2022a.

Wang, P., Zheng, W., Chen, T., and Wang, Z. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. *arXiv preprint arXiv:2203.05962*, 2022b.

Wei, C., Fan, H., Xie, S., Wu, C.-Y., Yuille, A., and Feichtenhofer, C. Masked feature prediction for self-supervised visual pre-training. *arXiv preprint arXiv:2112.09133*, 2021.

Xue, F., Sun, A., Zhang, H., and Chng, E. S. An embarrassingly simple model for dialogue relation extraction. *arXiv preprint arXiv:2012.13873*, 2020.

Xue, F., Shi, Z., Wei, F., Lou, Y., Liu, Y., and You, Y. Go wider instead of deeper. *arXiv preprint arXiv:2107.11817*, 2021.

Xue, F., He, X., Ren, X., Lou, Y., and You, Y. One student knows all experts know: From sparse to dense. *arXiv preprint arXiv:2201.10890*, 2022.

Yang, Z., Garcia, N., Chu, C., Otani, M., Nakashima, Y., and Takemura, H. Bert representations for video question answering. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1556–1565, 2020.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C.-J. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q., and Feng, J. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021a.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021b.

Zhou, W., Huang, K., Ma, T., and Huang, J. Document-level relation extraction with adaptive thresholding and localized context pooling. *arXiv preprint arXiv:2010.11304*, 2020.

A. Fine-tuning Hyper-parameters

Table 5: Hyper-parameters on ImageNet fine-tuning

Parameter	Base	Large	Huge
Epoch	100	50	50
Warmup Epochs		5	
Batch Size		1024	
Learning rate		2e-3	
Layer-wise learning rate decay	0.65	0.75	0.75
Weight Decay		0.05	
DropPath	0.1	0.2	0.2
Label smoothing		0.1	
Erasing prob.		0.25	
RandAug		9/0.5	
Mixup prob.		0.8	
Cutmix prob.		1.0	

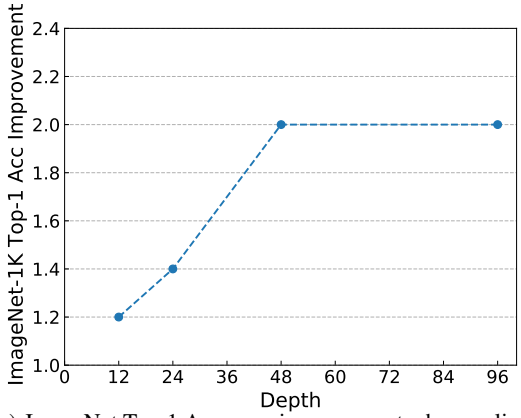
B. Throughput Comparison

Table 6: Throughput comparison of re-designed configurations under Bamboo idea. The throughput here means the image precessed per second by one TPU core. is measured during MAE pre-training. For base and large-level models, we use 128 TPUv3 cores in parallel. For the huge models, we use 256 TPUv3 cores.

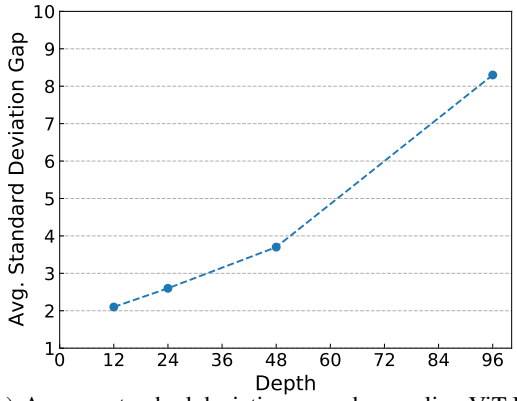
Scale	Base		Large		Huge	
	Original	Bamboo	Original	Bamboo	Original	Bamboo
Computation Cost	1×	1×	1×	1.1×	1×	1×
Throughput	1×	0.9×	1×	0.9×	1×	0.9×

From Table 6, we can see the deeper configurations are slightly slower, although they have comparable computation cost. However, we highlight that is fine. There is a trade-off instead of a limitation when using narrow configuration. During inference, we can actually do the inference layer by layer and only load one transformer layer into memory. After using, we can offload the layer and load the next one in. Such a design can be found in Figure 5 of SE-MoE paper (Shen et al., 2022). Since our single layer is narrower and includes fewer parameters, our model is more memory-efficient during inference.

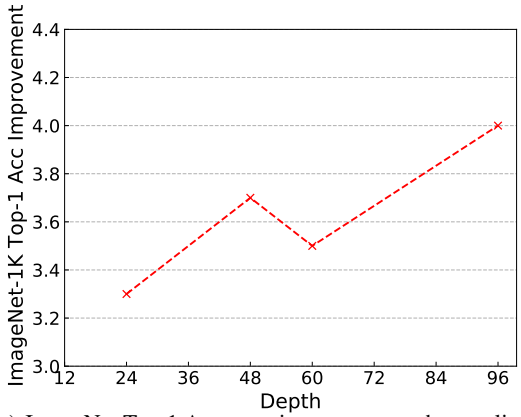
C. More Figures



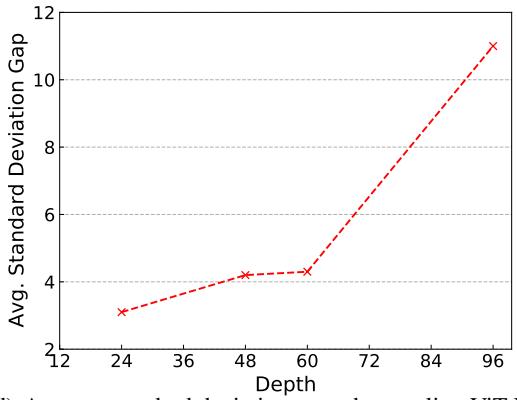
(a) ImageNet Top-1 Accuracy improvement when scaling ViT-B and MAE-B across depth.



(b) Average standard deviation gap when scaling ViT-B and MAE-B across depth.



(c) ImageNet Top-1 Accuracy improvement when scaling ViT-L and MAE-L across depth.



(d) Average standard deviation gap when scaling ViT-L and MAE-L across depth.

Figure 5: We compare the ImageNet Top-1 accuracy and average standard deviation of ViT and MAE models with different configurations. A larger average standard deviation gap means that MAE training objective alleviates more over-smoothing issue.