ViBid: Linear Vision Transformer with Bidirectional Normalization

Jeonggeun Song[1,*] **Heung-Chang Lee**[1,*]

[1]AI Lab & Service, Kakao Enterprise, Seongnam-si, South Korea
[*]Equal Contributions

Abstract

The vision transformer has achieved state-of-the-art performance in various vision tasks; however, the memory consumption is larger than those of previous convolutional neural network based models because of $O(N^2)$ time and memory complexity of the general self-attention models. Many approaches aim to change the complexity to $O(N)$ to solve this problem; however, they stack deep convolutional layers to retain locality or complicate the architecture as seen in window attention, to compensate for the performance degradation. To solve these problems, we propose ViBid algorithm, which resolves the complexity problem of $O(N^2)$ by replacing Softmax with bidirectional normalization (BiNorm). In addition, it has a much simpler architecture than the existing transformer model with $O(N)$ complexity. Owing to our simple architecture, we were able to use larger resolutions for training, and we obtained a lighter and superior GPU throughput model with competitive performance. ViBid can be used with any transformer method that uses queries, keys, and values (QKV) because of BiNorm, and it is quite universal due to its simple architectural structure.

1 INTRODUCTION

Transformers have been used in various fields. Initially, they were mostly employed in natural language processing (NLP) [Dosovitskiy et al., 2020, Touvron et al., 2020, Wu et al., 2021, Srinivas et al., 2021, Heo et al., 2021, Graham et al., 2021, El-Nouby et al., 2021], but currently, transformers are used in many domains of vision [Dosovitskiy et al., 2020, Touvron et al., 2020, Jiang et al., 2021, Esser et al., 2021, Durall et al., 2021]. The transformer has achieved state-of-the-art performance on several benchmark datasets. In the early stages, the vision transformer splits the input into patch units and then learns the image features after securing the locality with the convolutional layer. In this process, general transformer models which have an $O(N^2)$ complexity, for the number of tokens N, prohibit the size of the model parameters from growing excessively by using deeper convolutional layers or decreasing the size of the input sent to the transformer.

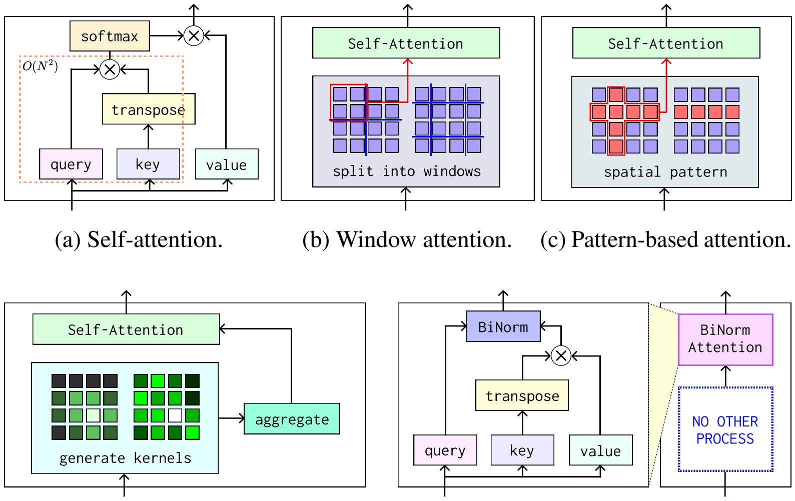
This is because the size of the model parameters increases as the square of the input size, that is, the size of the token, increases. Our proposed algorithm dramatically reduces the complexity of $O(N^2)$ to $O(N)$ by changing Softmax, which is the most commonly used function, to bidirectional normalization (BiNorm) and changing the multiplication order of the query, key, and value (QKV). This allows to stack the transformer module deeper and use a higher resolution as the input because the number of model parameters does not increase owing to the increased token size. Because L_2-normalization operates in distinct directions on the channel axis of Q and the spatial axis of K^TV, BiNorm is defined as bidirectional normalization.

One of the most essential aspects of the proposed method is that it has the simplest architecture among $O(N)$ complexity transformer algorithms. Contrastingly, existing methods for reducing complexity have resulted in performance degradation. To compensate, additional modules were added to the models, resulting in a complicated architecture, as seen in Figure 1(b), 1(c) and 1(d). However, ViBid is a linear transformer with $O(N)$ complexity and has an extremely simple architecture that does not require additional modules to supplement performance. Consequently, our suggested approach can be used with any transformer algorithm that has QKV and for any vision-related tasks.

The contributions of our algorithm can be summarized as follows.

- Our proposed algorithm facilitates the building of an efficient architecture, even when the token size(resolution) increases, by improving the complex-

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).



(a) Self-attention. (b) Window attention. (c) Pattern-based attention.

(d) Kernel-based attention. (e) BiNorm-based attention. (Ours)

Figure 1: **Comparison with existing self-attention algorithms:** (a) **Self-attention:** The query and key generate $N \times N$ attention maps, resulting in an $O(N^2)$ operation. (b) **Window attention:** This method splits input images into several windows before computing self-attention. While it avoids the $O(N^2)$ problem of self-attention, it also slows down computation due to a fixed window size. (c) (d) **Efficient self-attention:** These methods choose a fixed number of tokens to generate attention maps. However, these procedures are heuristic and complicated, and some of them require operations not supported by common frameworks. (e) **Ours:** BiNorm-based self-attention is not as complicated and has an $O(N)$ complexity since it does not require any additional processing.

ity from $O(N^2)$ to $O(N)$.

- The sequence in which QKV is multiplied varies even within the same $O(N)$ model, and our proposed model has a simpler architectural structure than other models with the same $O(N)$ complexity; therefore, it is more effective for model parameters.
- When the resolution (token) must be large, most ViT designs tend to reduce the final input to the transformer by deeply stacking the early convolutional layer stage. However, because our algorithm is not burdened by large-sized inputs, it may be learned using a transformer without significantly reducing its size in the early convolutional layer stage.

2 RELATED WORKS

Vision transformers. Dosovitskiy et al. [Dosovitskiy et al., 2020] proposed a vision transformer (ViT), which demonstrated the use of transformer-based models for vision tasks. After the achievements of ViT, DeiT [Touvron et al., 2020] introduced data-efficient training strategies for vision transformers with detailed ablation studies. They solved the ViT data efficiency problem successfully, and most of the current transformer-based models follow their schemes.

In further research, various architectures based on transformer variants have been presented. Touvron et al. [Touvron et al., 2021b] proposed two simple types of modules. One is the class attention module, which is the additional SA layer used to extract class information. These layers help the model aggregate features from the last outputs. The other is the LayerScale modules. These are learnable parameters for scaling residual connections. This prevents larger models from being overfitted. A simple variant of the LayerScale was presented at ResMLP [Touvron et al., 2021a]. While MLP-based models are irrelevant to our model, we apply Affine modules to our model as scalers.

Liu et al. [Liu et al., 2021] proposed a shifting window and patch merging. This generates local attention using two types of windows: Normal windows and shifted windows. At the end of each stage, this method merges the patches to preserve large receptive fields without heavy computation. Swin Transformer is organized in a hierarchical structure. The Swin Transformer alters the image resolution as the layers deeper, similar to how CNN reduces the resolution of the input image as the layers deepen. Feature Pyramid Network (FPN) structure of object detection can be employed since they have varied scale information. The performance in object detection and segmentation tasks is invariably superior to ViT because it takes advantage of multi-scale information via the FPN structure.

Hybrid architectures. Various methods for integrating convolutional layers [Heo et al., 2021, Wang et al., 2021, Graham et al., 2021, El-Nouby et al., 2021, Xiao et al., 2021, Hassani et al., 2021] instead of searching for new spatial structures have been introduced. LeViT, designed by Graham et al. [Graham et al., 2021], applies multi-stage networks to transformers using SA with convolution and pooling layers. Xiao et al. [Xiao et al., 2021] found that replacing linear patch embedding layers with convolutions helps transformers better capture low-level features. This is very similar to the stemming stage of existing CNN networks. El-Nouby et al. introduced local patch interactions in XCiT [El-Nouby et al., 2021]. With two depthwise convolutions [Chollet, 2017] added after XCA, XCiT achieved better performance. Our models are generally inspired by the intrinsic optimization strategies that XCiT introduced, while we present our own SA method.

Efficient self-attention. Instead of architectural strategies, several approaches have been proposed to solve the $O(N^2)$ problem of the self-attention (SA) mechanism. They are classified into several categories: those that use their own spatial patterns [Ho et al., 2019, Child et al., 2019, Sukhbaatar et al., 2019], linear approximation by sampling important tokens [Kitaev et al., 2020, Xiong et al., 2021], various low-rank factorization methods [Choromanski et al., 2020, Shen et al., 2021, Wang et al., 2020], and local attention [Liu et al., 2021].

However, these approaches have issues beyond complexity. The pattern and sampling methods are difficult to implement, and the GPU efficiency is low because a dynamic graph has to be created each time. The low-rank factorization method has the disadvantage of being a human heuristic, in which a person must empirically decide the kernel function. Local attention has a complex architecture, and the Swin transformer [Liu et al., 2021], which is used as an example, has the disadvantage of not having a simple architecture, such as using it as input through split windows.

These approaches, however, have issues beyond the complexity. The pattern and sampling methods were not easy to implement, and gpu efficiency was low because a dynamic graph had to be created each time. The low-rank factorization method has the disadvantage of being human heuristic, in which a person has to empirically decide the kernel function. The local attention has a complex architecture, and the Swin Transformer [Liu et al., 2021], which is used as an example, has the disadvantage of not being a simple architecture, such as using it as input through split windows.

3 METHOD

For the SA algorithm of the transformer [Vaswani et al., 2017], the query (Q) and key (K) are multiplied first to compute every pairwise relation of the tokens. The multiplication has time and memory complexity that is quadratic to the number of tokens. If the matrix multiplication of K and V is computed first, the computational resource of the SA is reduced to $O(N)$. However, the Softmax function must be applied to key-query interactions to generate a probability distribution for the attention mechanism. Softmax is a nonlinear operation; therefore, it must be removed from the SA to change the order of matrix multiplication.

To determine the effect of removing Softmax, we experimented with the ImageNet1k classification task for the Softmax-free ViT models. It implied that these models did not employ probabilistic approaches to SA. Interestingly, it was found that removing Softmax had no effect on the performance of the ViT models, as hypothesized. However, if Softmax was eliminated, the models were trained slowly and unstably during the early epochs of training. Furthermore, when additional experiments were conducted for other architectural optimizations, such as convolutional modules, the performance decreased. We proposed BiNorm method and integrated it with several existing architectural strategies to address these issues. This section explains how our proposed strategy generates stable Softmax-free SA models while avoiding quadratic complexities.

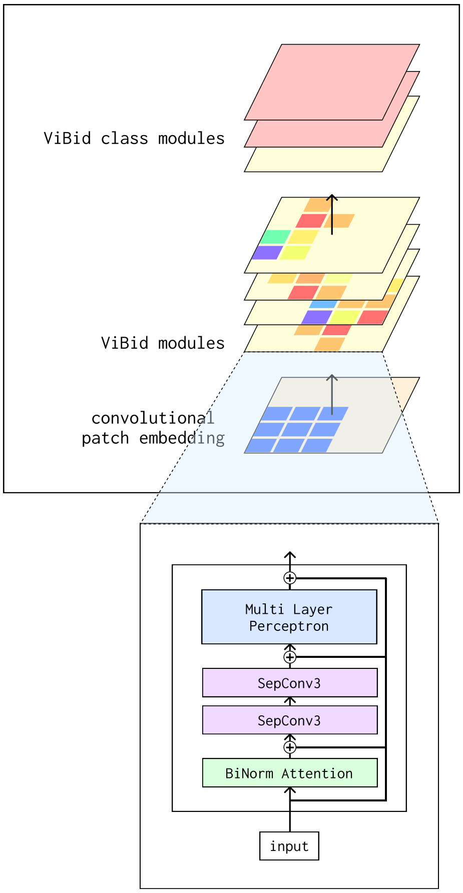
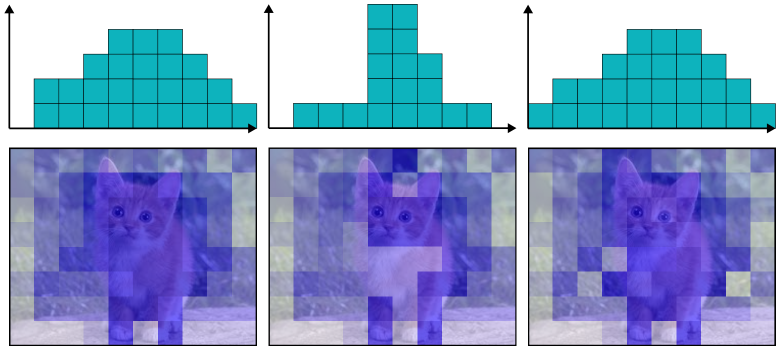


Figure 2: **ViBid model.** Our proposed ViBid model consists of a BiNorm attention module, two 3×3 separable convolution layers, and a feedforward layer. Note that LayerNorm and Affine layers are omitted for simplicity.

3.1 BINORM

The pixel-to-pixel relationship which can be obtained in basic SA methods as Softmax(QK^T) is calculated as a dot product. Softmax smooths the range of output vectors from 0 to 1 for obtaining an attention map, as shown in Figure 2(a). When Softmax is removed from the QKV, its output has a normal distribution from the initialization. However, in the case of a normal distribution, it has a range of [-inf, inf]. This makes the initial value more biased for a specific pixel



(a) Normal SA. (b) Softmax-free SA. (c) BiNorm-based SA.

Figure 3: **Attention maps in the early epochs.** (a) **Normal SA.** (b) **Softmax-free SA.** If the softmax is eliminated without replacement, it is heavily biased to specific patches. (c) **BiNorm-based SA.** BiNorm makes the output vectors unit-sized to debias the attention maps.

without Softmax, as shown in Figure 2(b).

Considering the previous insight, we conclude that the primary priority of the Softmax function is not to construct a probability distribution. Its sole purpose is to limit the output range of the function. Consequently, we determined that the Softmax function is not essential in ViT. We propose BiNorm, which is a combination of two L_2-normalizations, applied bidirectionally to the spatial dimension of K^TV and channel dimension of Q. It can make the SA have a complexity of $O(N)$ with a few lines of modification. Let $\mathbf{x} \in \mathbb{R}^{b \times N \times d}$ be the input image, where b is the batch size, N is the number of tokens, and d is the number of channels. Then, BiNorm-based SA is defined as:

$$Q = W_Q\mathbf{x}, K = W_K\mathbf{x}, V = W_V\mathbf{x} \tag{1}$$

$$(Q \in \mathbb{R}^{N \times d_{q,k}}, K \in \mathbb{R}^{N \times d_{q,k}}, V \in \mathbb{R}^{N \times d_v}) \tag{2}$$

$$\text{SA}(\mathbf{x}) = \text{BiNorm}(Q, K^TV) \tag{3}$$

$$(Q \in \mathbb{R}^{N \times d_{q,k}}, K^TV \in \mathbb{R}^{d_{q,k} \times d_v}) \tag{4}$$

$$\text{BiNorm}(A, B) = [L_2(A)_{\text{dim}=2}]^T L_2(B)_{\text{dim}=1} \tag{5}$$

for arbitrary matrices $A \in \mathbb{R}^{b \times N \times d}$ and $B \in \mathbb{R}^{b \times N \times d}$. BiNorm consists of two simple L_2- normalizations that apply to the channel dimension of Q and the spatial dimension of K^TV.

The output vectors of BiNorm-based attention are limited to unit size. All vectors have the same weight during the attention mechanism. Therefore, because Q and K are calculated as a unit vector by L_2-normalization, BiNorm generates a smoothed attention map that differs from Softmax-free, as shown in Figure 2(c). Mathematically, it is a cosine similarity matrix of Q and K^TV that generates clearer relations. Empirically, the ViT models with BiNorm converged faster than those without BiNorm. Additionally, the performance of BiNorm-based models did not decrease when other architectural optimizations were added to the models.

3.2 COMPARISON OF COMPUTATIONAL COMPLEXITY

The original SA has a complexity of $O(N^2)$ when computing QK^T.

$$Q = W_Q\mathbf{x}, K = W_K\mathbf{x}, V = W_V\mathbf{x} \tag{6}$$

$$(Q \in \mathbb{R}^{N \times d_{q,k}}, K \in \mathbb{R}^{N \times d_{q,k}}, V \in \mathbb{R}^{N \times d_v}) \tag{7}$$

$$\text{SA}(\mathbf{x}) = \frac{\text{Softmax}(QK^T)}{\sqrt{d}}V \tag{8}$$

$$(QK^T \in \mathbb{R}^{N \times N}, V \in \mathbb{R}^{N \times d_v}) \tag{9}$$

If the order of matrix multiplication is changed to sequential order, using BiNorm, both K^TV and $Q(K^TV)$ have a complexity of $O(N)$ (see Equation 1 for details). For vision tasks, the number of tokens is proportional to the resolution and reciprocal of the patch size. For example, if the height and width of an input image are scaled to 2×, the original SA requires $16\times$ computational resources. This is not efficient for cases require high-resolution inputs, such as the compound scaling method at EfficientNet. [Tan and Le, 2019]

In previous studies, various methods have been proposed to make self-attention $O(N)$ complexity (see Section 2 for further information). Most of these reduce the tokens that generate attention maps by utilizing local functions, learnable kernel functions, or human-designed patterns. However, they have several limitations. Primarily, they rely heavily on human heuristics. When the entire workflow is altered to some degree, new heuristics are required for the entire workflow. Further, they frequently require specialized operations that are not generally supported. This implies that they may be difficult to optimize for different tasks, frameworks, and devices. Finally, they confuse the overall flow. Many machine-learning devices have been designed for dense operations. A complicated computational graph may result in redundancy and memory leakage when it is used to operate on them.

BiNorm-based algorithms can reduce the complexity from $O(N^2)$ to $O(N)$ by modifying a few lines of the code. As

Table 1: **Design of ViBid models.** The architectural parameters contain the depth of model, the output dimension of each model d, the size of embedding h, and the number of heads.

Model	depth	d	h	#heads
ViBid-U	12	192	96	4
ViBid-T	24	192	96	4
ViBid-S	12	384	128	8
ViBid-M	24	384	128	8
ViBid-B	24	512	128	8

depicted in Figure 1, the computational graph of BiNorm-based attention is not complicated compared with the original SA. Because of its simple structure, our module is much more efficient at GPU than most other SA algorithms. We should discuss the numerical analysis of the computational efficiency in Section 4. Our proposed method consumes the least GPU memory and has the highest GPU throughput on a similar scale of FLOPs and the size of the model parameters.

3.3 VIBID MODEL

As shown in Figure 2, the input images are passed through convolutional patch embedding layers and divided into 16×16 patches. The convolutional patch embedding layers outperform the linear patch embedding layers in terms of the model performance. ViBid module consists of BiNorm-based SA, two 3×3 separable convolution layers, and a feedforward module. Unlike other models, BiNorm-based attention adopts a bottleneck design. The embedding size h is smaller than the output channel size d. In our experiments, a smaller embedding size prevented overfitting of the model. Locality is used as a weak inductive bias by convolutional layers that use relatively fewer resources. We adopted the class attention layers proposed by Touvron [Touvron et al., 2021b] but used BiNorm-based class attention layers differently from the existing ones. Table 1 presents the model design used in our experiments.

4 EXPERIMENTS

4.1 IMAGE CLASSIFICATION

Implementation details. For the image classification task, we evaluate our models using the ImageNet1k [Deng et al., 2009] dataset which spans 1000 semantic classes. It contains 1,281k images for training and 50k images for validation. There is no additional labeled or unlabeled dataset used. We train our model for 400 epochs with the AdanW optimizer [Loshchilov et al., 2017]. Following the linear scaling rule [You et al., 2017], the learning rate is scaled by $b/512$ for batch size b. It warms up linearly for the first 5 epochs before decaying using a cosine schedule. LayerNorm [Ba et al., 2016] and Affine [Touvron et al., 2021a] are used in each residual block to improve generalization. As strong regularization, our proposed method utilize RandAugment, stochastic depth [Huang et al., 2016], and CutMix [Yun et al., 2019] for data-efficient training. The size of each model affects the amplitude of RandAugment [Cubuk et al., 2020] and the probability of dropping residual connections. To improve the training enough for the larger models, stronger regularization is required. We do not employ distilled knowledge from a pre-trained instructor model to boost performance. All training procedures are performed on 32 NVIDIA A100 GPUs.

Table 2: **The results of fine-tune at higher resolutions.** Our models show the fastest GPU throughput and the lowest peak memory in comparison to the other models which accomplish similar performance. Note that XCiT [El-Nouby et al., 2021] models use 224×224 resolution, since they use smaller patch size.

Model	Top-1 Acc.	FLOPs (G)	Res.	GPU Thr. (img/s)
EfficientNet-B7	84.3	37.0	600	53.6
XCiT-S24/8	83.9	36.0	224	106.5
XCiT-M24/8	83.7	63.9	224	69.5
DeiT-B	83.1	49.4	384	87.8
Swin-B	84.5	47.0	384	86.5
ViBid-M	83.8	20.5	384	171.2
ViBid-B	84.5	35.1	384	114.9
ViBid-B	84.7	62.4	512	66.6
ViBid-B	84.8	140.4	768	28.6

Comparison with the concurrent models. In Table 3, we compare our models with existing transformer-based and CNN models. Our models achieved higher performance than the other models at a similar scale of FLOPs and parameters (refer to Figure 4). Our models can perform well with fewer computational resources and lower capacities, even they does not utilize the architectural optimizations for vision like local self-attention or multi-scale structure. As a metric of performance, we measured the GPU throughput of each model. In particular, at a resolution of 768×768, the GPU throughput of our models surpassed the CNN models as well as the other transformer-based models. While our proposed method computes global spatial relations, our models show superior performance at various resolutions compared with CNN models that utilize local relations.

Fine-tune at higher resolution. Instead of training the models from scratch, we fine-tuned ViBid-M and ViBid-B at a higher resolution for 10 epochs. We report the results of fine-tuning at resolutions of 384, 512, and 768. The batch sizes for each training session were set to 1024, 512, and 256. Owing to the benefit of lower memory consumption, our

Table 3: **Comparison with the concurrent models.** The image classification results include the top-1 accuracy, param size, FLOPs, and GPU throughput of various models on ImageNet1k. Our models show competitive results for top-1 accuracy, and show the fastest GPU throughput among models which achieve similar performance.

Model	Top-1 Acc.	Params (M)	FLOPs (G)	GPU Throughput (img/s, res=224)	GPU Throughput (img/s, res=768)
RegNetY-800MF [Radosavovic et al., 2020]	76.3	6	0.8	1642.2	145.0
RegNetY-1.6G [Radosavovic et al., 2020]	78.0	11	1.6	932.0	103.1
DeiT-Ti [Touvron et al., 2020]	72.2	5	1.3	2390.3	70.4
ViBid-U	76.3	6	1.0	1177.7	163.8
ViBid-T	78.8	10	1.9	650.3	94.4
ResNet-50 [He et al., 2016]	75.3	26	3.8	1097.0	104.7
RegNetY-4G [Radosavovic et al., 2020]	80.0	21	4.0	837.2	33.9
DeiT-S [Touvron et al., 2020]	79.8	22	4.6	892.5	31.7
Swin-T [Liu et al., 2021]	81.3	29	4.5	729.0	-
XCiT-S12/16 [El-Nouby et al., 2021]	82.0	26	4.8	678.4	52.8
CoAtNet-0 [Dai et al., 2021]	81.6	25	4.2	-	-
PVTv2-B2 [Wang et al., 2022]	82.0	25	4.0	-	-
MViTv2-T [Li et al., 2022]	82.3	24	4.7	-	-
ViBid-S	82.0	21	3.7	832.2	61.6
ResNet-101 [He et al., 2016]	76.4	47	7.6	657.1	63.1
RegNetY-8G [Radosavovic et al., 2020]	81.7	39	8.0	477.5	27.8
Swin-S [Liu et al., 2021]	83.0	50	8.7	409.1	-
XCiT-S24/16 [El-Nouby et al., 2021]	82.6	48	9.1	369.3	31.4
ViBid-M	82.8	37	7.0	465.8	38.6
RegNetY-16G [Radosavovic et al., 2020]	82.9	84	16.0	317.3	16.3
DeiT-B [Touvron et al., 2020]	81.8	86	17.5	303.4	13.1
Swin-B [Liu et al., 2021]	83.5	88	15.4	274.6	21.3
XCiT-M24/16 [El-Nouby et al., 2021]	82.9	84	16.2	249.0	21.5
ViBid-B	83.3	64	11.9	330.8	28.6

models can be trained faster by utilizing a large batch size, whereas the capable computational resources are limited.

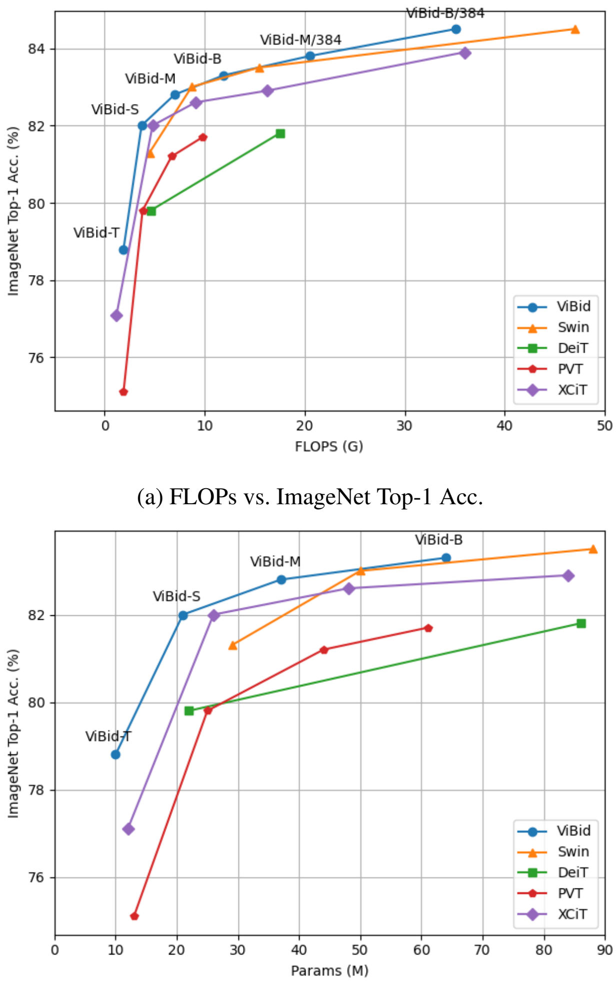
Our models showed higher performance than DeiT-B and Swin-B at the same resolution. In addition, the GPU throughput was not reduced much at a higher resolution than that of the other models. Moreover, our models allocated a much smaller amount of memory. DeiT-B trained at a resolution of 384×384 possesses 20% more memory than ViBid-B trained at a resolution of 512×512, even if DeiT-B uses half the number of tokens that ViBid-B uses. As the resolution increases, fine-tuning offers a boost without increasing the model capacity. This implies that the model learns high-resolution features without additional parameters.

4.2 ABLATION STUDY

The effect of the convolutional layers. We conducted an experiment to compare the performance of BiNorm with a pure architecture by subtracting the depth-wise convolutional layer, which is Local Patch Interactions (LPI) proposed at XCiT [El-Nouby et al., 2021]. We compared ViBid-S(w/o conv) and DeiT-S [Touvron et al., 2020] in the ImageNet1k dataset because our architecture is essentially the same as DeiT, except for BiNorm. In conclusion, ViBid-S(w/o Conv) performed better than DeiT-S, which has an accuracy of 79.8%, with 80.3%. Undoubtedly, adding LPI results in better performance. However, the role of BiNorm is proven to be accurately represented because it is better than DeiT and higher than XCiT when LPI is present.

Comparison of the activation functions. As mentioned in Sec. 3, we compared Softmax-free ViT [Dosovitskiy et al., 2020], the original ViT, and ViBid-S without LPI to observe the main role of Softmax function. The results are in Table 4. In our tests, Softmax-free ViT models perform a little worse than the original ViT models. Although the divergence does not appear to be as large as we had anticipated, Softmax-free models did not converge well in the early epochs. (Refer to our supplementary materials.)

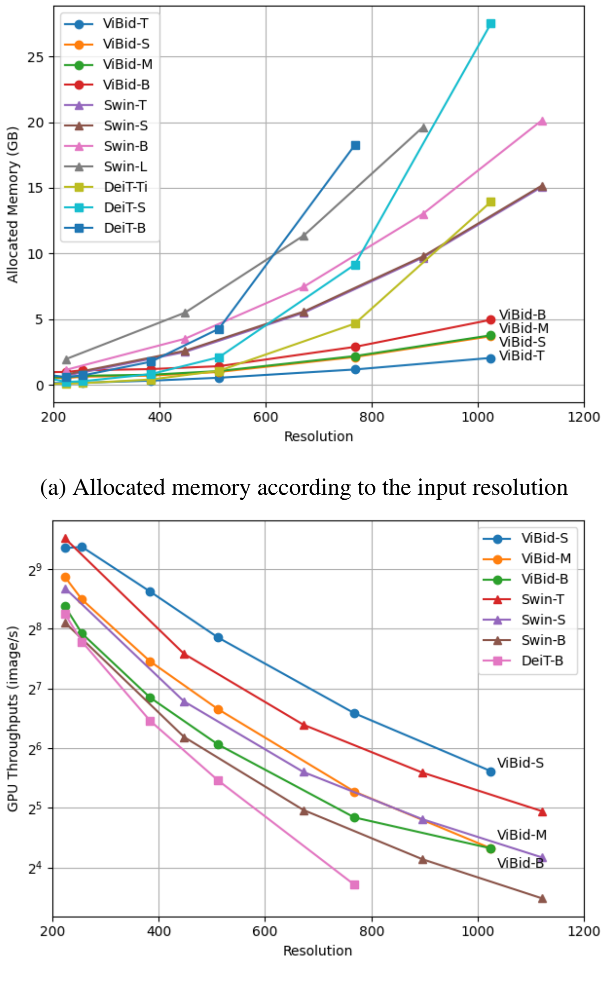
We trained the ViBid-S model using the same techniques as BiNorm to confirm its advantages. In comparison to Softmax-free and Softmax models, our models perform better. Empirically, those findings support both our theory regarding the primary function of Softmax in the original self-attention and the necessity of BiNorm as a replacement for Softmax.



(a) FLOPs vs. ImageNet Top-1 Acc.

(b) Params vs. ImageNet Top-1 Acc.

Figure 4: **Comparison with the transformer-based vision models.** Our models show superior performance at most regime of FLOPs and param size. More details containing the comparison with CNN models are in Table 3.



(a) Allocated memory according to the input resolution

(b) GPU throughput according to the input resolution

Figure 5: **Comparison of consumption of computational resources at high resolution.** (a) The results of the peak GPU memory measured on different resolutions. Our models require significantly less memory than the other models at overall resolutions. (b) The results of the GPU throughput measured on varying resolutions. GPU throughput axis is $\log_2$-scaled. Note that the GPU throughput of ViBid models is more slowly decreased as the resolution increases.

4.3 MEASURING COMPUTATIONAL EFFICIENCY

As denoted in Section 3, BiNorm-based self-attention can be a useful solution for transformer-based vision models when high resolution features are required. To perform quantitative analysis, we report the required computational resources on various resolutions for different vision models and our proposed models. (See Figure 5.) All measurements were performed on a NVIDIA V100 GPU with batch size $b = 32$.

Memory efficiency. In terms of capability for both training and inference, memory efficiency is one of the most important factors. As shown in Figure 5a, the models based on BiNorm consumed much less memory for larger resolutions compared to the other models, which are based on the original self-attention or the local attention. It demonstrates our BiNorm-based self-attention scheme works on high-resolution inputs more efficiently, even compared to the local attention algorithms such as Swin [Liu et al., 2021]. Our model can process up to a $4\times$ batch size compared with the other models showing similar performance. Another advantage of the proposed methods is that our models can be easily scaled up without concerning the growth of memory usage. As depicted in Figure 5a, the allocated memory of our models does not increase much whereas the size of the model grows up. It allows the training at a large scale with reasonable computational resources.

Table 4: **Ablation study about the effect of Softmax.** For a fair comparison, we implemented and trained ViT models again. Note that ViT-B without Softmax performs the matrix multiplications sequentially for self-attention.

Model	Complexity	Activation Function	Top-1 Acc.
ViT-B	$O(N)$	None	78.6
ViT-B	$O(N^2)$	Softmax	78.8
ViBid-S (w/o conv)	$O(N)$	None	79.1
ViBid-S (w/o conv)	$O(N)$	BiNorm	**80.3**

Table 5: **Comparison of the linear attention algorithms at ImageNet1k top-1 accuracy.** To re-implement the LinFormer and Efficient Attention algorithms to compare ImageNet1k top-1 accuracy, we adopted ViT-S design for them. ViBid-S which LPI is absent acheives higher performance than any other algorithms.

Model	GPU Thr.	Top-1 Acc.
ViBid-S (w/o conv)	**547.3**	**80.3**
LinFormer	403.9	75.7
Efficient Attention	416.1	76.3

GPU throughput. In Figure 5b, the GPU throughput of transformer-based models is reported at various resolutions. As shown, our model is faster than other models showing similar performance. In addition, the GPU throughput of our model decreases slowly compared to other models as input resolution increases. It is because our proposed algorithm does not require additional kernel optimizations of the frameworks as well as it has $O(N)$ complexity.

Comparison with the existing linear attention algorithms. To compare to various the linear self-attention algorithms, which is introduced in Section 2, we implemented blending of ViT design and the linear self-attention algorithms [Wang et al., 2020, Qin et al., 2022, Shen et al., 2021]. Since all the other algorithms do not use the additional layers, we removed the convolutional layers from our models to perform experiment on the equal conditions. Our proposed method achieves the best GPU throughput and memory consumption for every input resolution, and outperforms the other algorithms for ImageNet1k classification task. We added experimental results in Table 5-6 for details.

Table 6: **Computational efficiency of the linear attention algorithms.** Note that all networks are implemented on the same architecture design. ViBid-M shows the highest GPU Throughputs, and consumes the lowest GPU memory. All measurement is performed on 1 NVIDIA V100 GPU.

Res.	ViBid-M (w/o conv)	Cos-Former	Lin-Former	Efficient Attention
224	**547.3**	384.9	403.9	416.1
384	**209.9**	147.8	151.7	186.3
512	**119.3**	84.7	85.9	114.5
1024	**24.2**	18.4	20.1	23.9

(a) GPU Throughput (img/s, batch size $b = 32$)

Res.	ViBid-M (w/o conv)	Cos-Former	Lin-Former	Efficient Attention
224	**0.38**	0.41	0.42	0.38
384	**0.65**	0.69	0.70	0.65
512	**1.05**	1.08	1.11	1.05
1024	**3.77**	3.81	3.91	3.77

(b) GPU Memory Allocation (GB)

5 DISCUSSION AND CONCLUSION

We propose the simplest transformer architecture to improve the time and memory complexity of any transformer from $O(N^2)$ to $O(N)$. The complexity of a general SA transformer algorithm is $O(N^2)$, and it scales linearly with the input resolution (token size). However, by multiplying BiNorm by K^TV and then Q, the proposed algorithm is designed to be $O(N)$, allowing the model parameters to be considerably lowered, and a large resolution can be achieved. Furthermore, previous transformer techniques had complex architectures to compensate for the reduced performance with window attention, kernel-based attention, and pattern-based attention; however, ViBid employing BiNorm does not have one. It performs similarly to the previous algorithm. We expect our proposed algorithm to be universally applied to all transformer algorithms with QKV because it can be used with very small code modifications.

Limitation. Our proposed algorithm can be used for all general vision tasks, such as image classification, object detection, and segmentation. However, its structures need to be more optimized for each task, so we did not introduce the performance of our models for the object detection or the segmentation tasks. In the future, we intend to perform experiments using generally used architectural optimizations, like multi-scale structures, or compound scaling, to show the SoTA-level performance at those tasks.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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.

Ricard Durall, Stanislav Frolov, Andreas Dengel, and Janis Keuper. Combining transformer generators with convolutional discriminators. *arXiv preprint arXiv:2105.10189*, 2021.

Alaaeldin El-Nouby, 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. *arXiv preprint arXiv:2106.09681*, 2021.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021.

Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. *arXiv preprint arXiv:2104.01136*, 2021.

Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. *arXiv preprint arXiv:2103.16302*, 2021.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, 2021.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 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. *arXiv preprint arXiv:2103.14030*, 2021.

Ilya Loshchilov et al. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. *arXiv preprint arXiv:2202.08791*, 2022.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020.

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3531–3539, 2021.

Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16519–16529, 2021.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. *arXiv preprint arXiv:1905.07799*, 2019.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021a.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. *arXiv preprint arXiv:2102.03902*, 2021.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.