Scratching Visual Transformer's Back with Uniform Attention

Nam Hyeon-Woo[1*] Kim Yu-Ji[1]
Byeongho Heo[2] Dongyoon Han[2] Seong Joon Oh[3] Tae-Hyun Oh[1*]
[1]POSTECH, [2]NAVER AI Lab, [3]Tübingen University

Abstract

The favorable performance of Vision Transformers (ViTs) is often attributed to the multi-head self-attention (`MSA`*), which enables global interactions at each layer of a ViT model. Previous works acknowledge the property of long-range dependency for the effectiveness in* `MSA`*. In this work, we study the role of* `MSA` *in terms of the different axis, density. Our preliminary analyses suggest that the spatial interactions of learned attention maps are close to dense interactions rather than sparse ones. This is a curious phenomenon because dense attention maps are harder for the model to learn due to* `softmax`*. We interpret this opposite behavior against* `softmax` *as a strong preference for the ViT models to include dense interaction. We thus manually insert the dense uniform attention to each layer of the ViT models to supply the much-needed dense interactions. We call this method Context Broadcasting,* `CB`*. Our study demonstrates the inclusion of* `CB` *takes the role of dense attention and thereby reduces the degree of density in the original attention maps by complying* `softmax` *in* `MSA`*. We also show that, with negligible costs of* `CB` *(1 line in your model code and no additional parameters), both the capacity and generalizability of the ViT models are increased.*

1. Introduction

After the success of Transformers [58] in language domains, Dosovitskiy *et al*. [12] have extended to Vision Transformers (ViTs) that operate almost identically to the Transformers but for computer vision tasks. Recent studies [12, 56] have shown that ViTs achieve superior performance on image classification tasks. Further, the universal nature of ViTs' input has demonstrated its potential to multi-modal input extensions [2, 14, 26, 31].

The favorable performance is often attributed to the multi-head self-attention (`MSA`) in ViTs [12, 56, 59, 7, 50, 44], which facilitates long-range dependency[1]. Specifically, `MSA` is designed for long-range interactions of spatial information in all layers. This is a structurally contrasting feature with a large body of successful predecessors, convolutional neural networks (CNNs), which gradually increase the range of interactions by stacking many fixed and hard-coded local operations, *i.e*., convolutional layers. Raghu *et al*. [44] and Naseer *et al*. [39] have shown the effectiveness of the self-attention in ViTs for the global interactions of spatial information compared to CNNs.

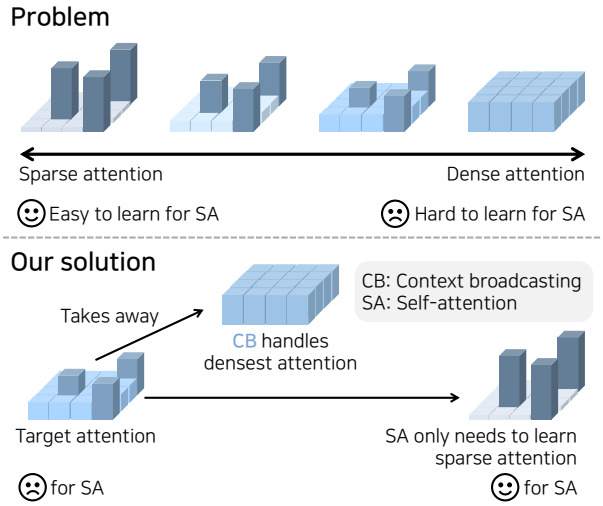


Figure 1. **Motivation of our work. Top:** dense attention is hard to learn with softmax, but self-attention tends to learn it more than sparse one. **Bottom:** we infuse dense attention explicitly, named `CB`, to split the responsibility of interactions; the burden of interactions of self-attention is reduced. Self-attention is now more likely to learn sparse interaction that is in favor of softmax.

Unlike previous works [44, 39] that focused on the effectiveness of *long-range dependency*, we study the role of *density* in spatial attention. "Long-range" can be either "sparse" or "dense". We examine whether the learned attention of ViTs is dense or sparse. Our preliminary analysis based on the entropy measure suggests that the learned

*This work was done during N. Hyeon-Woo's intern at NAVER AI Lab. Tae-Hyun Oh is in Department of Electrical Engineering and Grad. School of Artificial Intelligence, POSTECH, and joint affiliated with Institute for Convergence Research and Education in Advanced Technology, Yonsei University, Korea.

[1]Long-range dependency is described in the literature with various terminologies: non-local, global, large receptive fields, etc.

attention maps tend to be dense across all spatial locations. This is a curious phenomenon because denser attention maps are harder to learn by the softmax operation. Its gradients become larger (less stable) around denser attention maps. In other words, ViTs are trying hard to learn dense attention maps despite the difficulty of learning them through gradient descent.

While dense attention is unlikely to be learned via gradient descent, it is easy to implement it manually. We insert uniform attention explicitly, the densest form of attention, to confirm our observation of the effort of learning dense attention. We call our module Context Broadcasting (CB). The module adds the averaged token to every individual token at intermediate layers. We find that when CB is added to ViT, CB reduces the degree of density in attention maps in all layers preserving the long-range dependency. CB takes over the role of the dense global aggregation from self-attention, as illustrated in Fig. 1. CB also makes the overall optimization for a ViT model easier and improves its generalization.

CB brings consistent gains in the image classification task on ImageNet [47, 46, 4] and the semantic segmentation task on ADE20K [67, 68]. Overall, CB seems to help a ViT model divert its resources from learning dense attention maps to learning other informative signals. We also demonstrate that our module improves the Vision-Language Transformer, ViLT [31], on a multi-modal task, VQAv2 [16]. Such benefits come with only negligible costs. Only 1 line of code needs to be inserted in your `model.py`. No additional parameters are introduced; only a negligible number of operations are. Our contributions are as follows:

- Our observations of the dense interaction preference of ViTs but the learning difficulty from `softmax` (Sec. 3.1);
- A simple and effective modules, CB and CB_S, for infusing dense interactions (Sec. 3.2);
- Phenomena for CB to divert the capacity of MSA for sparse interactions (Sec. 3.3);

2. Related Work

Transformers. Since the seminal work of the Transformers [58], it has been the standard architecture in the natural language processing (NLP) domain. Dosovitskiy *et al*. [12] have pioneered the use of Transformers in the visual domain with their Vision Transformers (ViTs). The way of ViTs work is almost identical to the original Transformers, where ViTs tokenize non-overlapping patches of the input image and apply the Transformers architecture on top. The Transformers with multi-head self-attention (MSA) are especially appealing in computer vision because their non-convolutional neural architectures do not have conventional hard-coded operations, such as convolution and pooling with fixed local kernel sizes. Cordonnier *et al*. [9] and Ramachandran *et al*. [45] corroborate that the expressiveness of MSA even includes convolution.

There have been attempts to understand the algorithmic behaviors of ViTs, including MSA, by contrasting them with CNNs [44, 9, 39, 40, 57]. Raghu *et al*. [44] empirically demonstrate early aggregation of global information and much larger effective receptive fields [36] over CNNs. Naseer *et al*. [39] show highly robust behaviors of ViTs against diverse nuisances, including occlusions, distributional shifts, adversarial and natural perturbations. Intriguingly, they attribute those advantageous properties to large and flexible receptive fields by MSA in ViTs and interactions therein. Similarly, there have been studies that attribute the effectiveness of MSA to global interaction in many visual scene understanding tasks [56, 7, 50, 44, 32, 3, 30]. Distinctively, we study the role of the density of the attention.

Attention module. The global context is essential to capture a holistic understanding of a visual scene [54, 43, 48, 59, 6], which aids visual recognition. To capture the global context, models need to be designed to have sufficiently large receptive fields to interact and aggregate all the local information. Prior arts have proposed to enhance the interaction range of CNNs by going deeper [49, 18, 27, 52] or by expanding the receptive fields [62, 10, 59, 33, 29]. Hu *et al*. [25] squeeze spatial dimensions by pooling to capture the global context. Cao *et al*. [6] notice that the attention map of the non-local block is similar regardless of query position and propose a global context block.

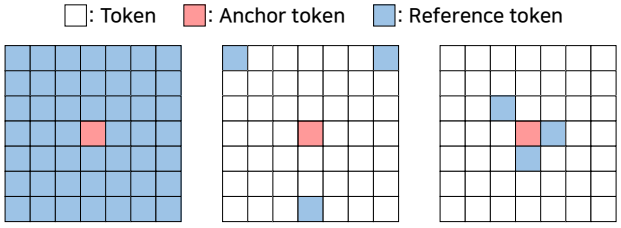
Our study focuses on the ViT architecture, which comprises concise layers and can serve as versatile usage, such as unified multi-modal Transformers [2, 14, 26, 31]. The receptive field of MSA in ViTs inherently covers the entire input space, which may facilitate the learning of global interactions and the modeling of global context more efficiently than CNNs [38]. However, the current global context modeling in ViTs may not be straightforward. Our research presents a few indications that while self-attention favors learning dense global interactions, it is challenging to achieve this due to `softmax`. To ascertain the benefits of dense global interaction, we explicitly inject it and observe an improvement of performance. Moreover, we observe the allocation of MSA capacities to better interactions.

3. Method

We first motivate *the need for dense interactions for the ViTs* in Sec. 3.1. Then, we propose a simple, lightweight module and a technique to inject explicitly dense interactions into ViTs in Sec. 3.2. Finally, we demonstrate how uniform attention affects to the ViT model in Sec. 3.3.

3.1. Motivation

The self-attention operations let ViTs conduct spatial interactions without limiting the spatial range in every layer.



(a) Dense global (b) Sparse global (c) Sparse local

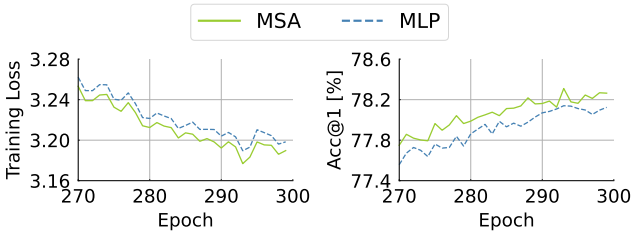
Figure 2. **Type of spatial interactions.** We categorize the spatial interactions of self-attention into three types. The anchor token interacts with reference tokens.

Long-range dependency, or global interactions, signifies connections that reach distant locations from the reference token. Density refers to the proportion of non-zero interactions across all tokens. Observe that "global" does not necessarily mean "dense" or "sparse" because an attention map can be sparsely global. We illustrate their difference in Fig. 2. The question of interest is the type of interaction that self-attention learns.

Before delving into the study of density, we examine which multi-head self-attention (MSA) or multi-layer perceptron (MLP) blocks further increase the capacity of the model. Our preliminary observation highlights the benefit of studying MSA. We then measure the layer-wise entropy of the attention to investigate the spatial interaction characteristics that ViTs prefer to learn.

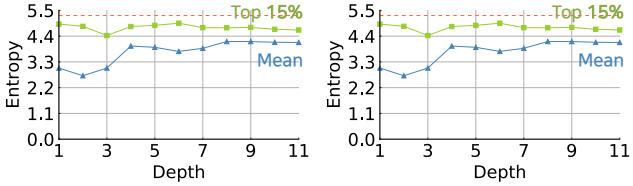
MSA vs. MLP. MSA and MLP in ViTs are responsible for spatial and channel interactions, respectively. We examine adding which block, either MSA or MLP, increases the performance of ViTs more. We train the eight-layer ViT on ImageNet-1K for 300 epochs with either an additional MSA or MLP layer inserted at the last layer. The additional number of parameters and FLOPs are nearly equal. In Fig. 3, we plot the training loss and top-1 accuracy. We observe that the additional MSA enables lower training loss and higher validation accuracy than the additional MLP. This suggests that, given a fixed budget in additional parameters and FLOPs, the ViT architecture seems to prefer to have extra spatial interactions rather than channel interactions. It leads us to investigate the spatial interactions of MSA.

Which type of spatial interactions does MSA learn? Here, we examine the types of spatial interactions that are particularly preferred by MSA. Knowing the type of interactions will guide us on how we could improve attention performance. While previous studies [44, 39] have focused on the effectiveness of long-range dependency in MSA, we focus on the density in MSA. We measure the dispersion of attention according to the depth through the lens of entropy. Low entropy indicates that the attention is sparse, whereas high entropy suggests that the attention is dense. Entropy provides a more objective view rather than relative and subjective measures such as visualization [37, 17].



(a) Training loss (b) Top-1 accuracy

Figure 3. **Impact on the capacity of the ViT model with a single extra block.** Training loss and top-1 accuracy (y-axis) versus epochs (x-axis) of 8-depth ViT with additional MSA and MLP blocks. The decrease in training loss and the increase in validation accuracy implies an increase in the model capacity.



(a) ViT-S (b) ViT-B

Figure 4. **Entropy analysis.** We use pre-trained ViTs to measure layer-wise entropy. We plot the average and 15^{th} percentile of entropy values. The red dot line stands for the maximum entropy upper bound.

Figure 4 shows the trends of the average and 15^{th} percentile entropy values across the heads and tokens for each MSA layer in ViT-S/-B [12, 56]. We observe that attention maps tend to have greater entropy values as high as 4.4 on average, towards the maximal entropy value, $-\sum \frac{1}{N} \log \frac{1}{N} \approx 5.3$, where N is the number of tokens and 197. The top 15% of entropy values are much close to the maximal entropy value corresponding to uniform attention. It is remarkable that a majority of the attention in ViTs has such high entropy values; it suggests that MSA tends to learn the dense interactions.

Steepest gradient around the uniform attention. The extreme form of dense interactions is the uniform distribution. To examine the difficulty of finding the uniform distribution for the self-attention in MSA, we delve into the characteristics of the softmax function. In a nutshell, we show that the gradient magnitude is the largest around the inputs inducing a uniform output. We further formalize this intuition below. The self-attention consists of the row-wise softmax operation $\mathbf{A} = \sigma(\lambda \mathbf{S}) \in \mathbb{R}^{N \times N}$ where $\mathbf{S} \in \mathbb{R}^{N \times N}$ is the collection of dot products of queries and keys, possibly with a scale factor $\lambda > 0$. For simplicity, we consider the softmax over a single row: $\mathbf{a} = \sigma(\lambda \mathbf{s}) \in \mathbb{R}^N$. The gradient of $\mathbf{a}$ with respect to the input $\mathbf{s}$ is $\mathbf{J}_{jk} := \partial \mathbf{a}_j / \partial \mathbf{s}_k = \lambda(\mathbb{1}_{j=k} \mathbf{a}_j - \mathbf{a}_j \mathbf{a}_k)$ for $1 \leq j, k \leq N$. We measure the magnitude of the gradient $\mathbf{J} \in \mathbb{R}^{N \times N}$ using the nuclear norm $||\mathbf{J}||_* = \sum_{i=1}^{N} \nu_i$ where $\{\nu_i\}$ are the singular values of $\mathbf{J}$. Note that $\mathbf{J}$ is a real, symmetric, and pos-

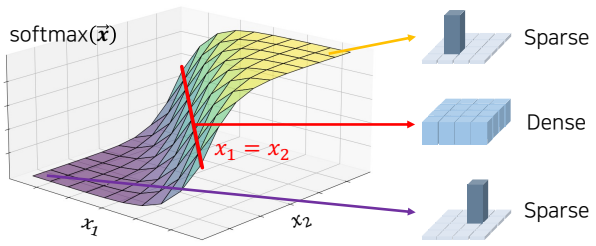


Figure 5. **Gradient around uniform attention.** Softmax operation has high gradients around uniform attention ($x_1 = x_2$).

itive semi-definite matrix. Thus, the nuclear norm coincides with the sum of its eigenvalues, which in turn is the trace: $||\mathbf{J}||_* = \sum_j \lambda(\mathbf{a}_j - \mathbf{a}_j^2)$. With respect to the constraint that $\sum_j \mathbf{a}_j = 1$ and $\mathbf{a}_j \geq 0$ for all j, the nuclear norm $||\mathbf{J}||_*$ is maximal when $\mathbf{a}_j = 1/N$ for every j. Figure 5 describes the `softmax` function in 2D input. This shows that uniform attention with softmax can be easily broken by a single gradient step, meaning it is the most unstable type of attention to learn, in the optimization point of view.

Conclusion. We have examined the density of the interactions in the `MSA` layers. We found that further spatial connections benefit ViT models more than further channel-wise interactions. `MSA` layers tend to learn dense interactions with higher entropies. ViT's preference for dense interactions is striking, given the difficulty of learning dense interactions: the gradient for the `MSA` layer is steeper with denser attention maps. This implies that dense attention maps are hard to learn but seem vital to ViTs.

3.2. Explicitly Broadcasting the Context

We observe the curious phenomena: `MSA` learns dense interaction, though it is unstable in terms of the gradient. We decide to inject uniform attention because (1) uniform attention is the densest attention and is unstable in terms of gradient view, but (2) humans can supply uniform attention easily, and (3) uniform attention requires no additional parameters and small computation costs. We do this through the broadcasting context with the `CB` module.

Context Broadcasting (`CB`). Given a sequence of N tokens $\mathbf{X} \in \mathbb{R}^{N \times d}$, our `CB` module supplies the averaged token back onto the tokens as follows:

$$\texttt{CB}(\mathbf{x}_i) = \frac{\mathbf{x}_i + \frac{1}{N}\sum_{j=1}^{N} \mathbf{x}_j}{2} \quad \text{for every token } i, \qquad (1)$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the i^{th} token in $\mathbf{X}$. Figure 6a illustrates our `CB` module. The `CB` module is placed at the end of `MLP` block (See Fig. 6b). Our analysis in Sec. 4.1 shows that the insertion of `CB` increases the performance of ViTs regardless of its position. As we shall see, the performance increase is most significant when it is inserted after the `MLP` block.

Computational efficiency. The `CB` module is implemented with 1 line of code in deep learning frameworks like `PyTorch` [41], `Tensorflow` [1], and `JAX` [5]:

```
X = 0.5 * X + 0.5 * X.mean(dim=1, keepdim=True)
```

It does not increase the number of parameters and incurs negligible additional operations for inference and training.

`CB` with dimension scaling. Although we focus on the simplest form, we propose another variant of dense interaction $\texttt{CB}_\texttt{S}$ by introducing a minimal number of parameters. The proposed `CB` injects the dense interaction into all channel dimensions, but some channel dimensions of a token would require dense interaction, whereas others would not. We then introduce weights to scale the channels, $\mathbf{\Lambda} \in \mathbb{R}^d$, to infuse uniform attention selectively for each dimension as follows: $\texttt{CB}_\texttt{S}(\mathbf{x}_i) = \mathbf{x}_i + \mathbf{\Lambda} \odot \left(\frac{1}{N}\sum_{j=1}^{N} \mathbf{x}_j\right)$ where $\odot$ is the element-wise product. $\texttt{CB}_\texttt{S}$ introduces few parameters: 0.02% additional parameters for ViT-S.

Module	# Params [M]	Acc@1 [%]
ViT-S	22	79.9
(A)	21	80.1
(B)	22	80.3
(C)	29	80.6
`CB` (ours)	22	**80.8**
$\texttt{CB}_\texttt{S}$ (ours)	22	80.4

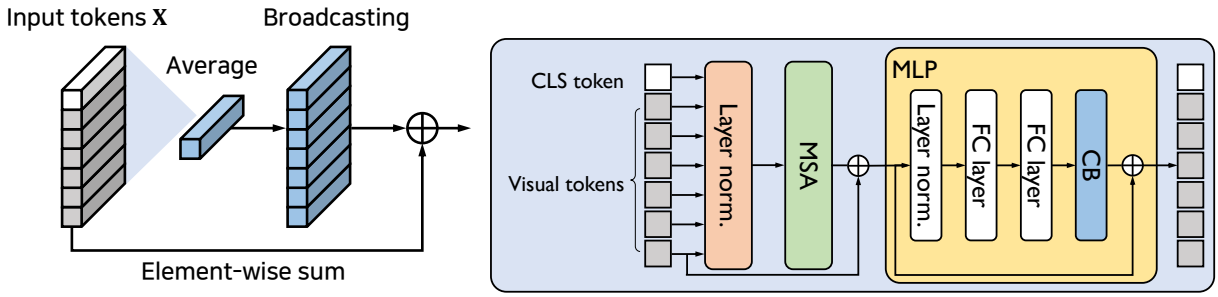
Table 1. **Injection of uniform attention to ViT-S.** We inject uniform attention to `MSA` **(A, B, C)** or `MLP` (`CB`, $\texttt{CB}_\texttt{S}$). ViT benefits from uniform attention.

3.3. How Does Uniform Attention Affect ViT?

In the following experiments, we delve into the effect of uniform attention. We train ViTs on ImageNet-1K during 300 epochs following the DeiT setting [56].

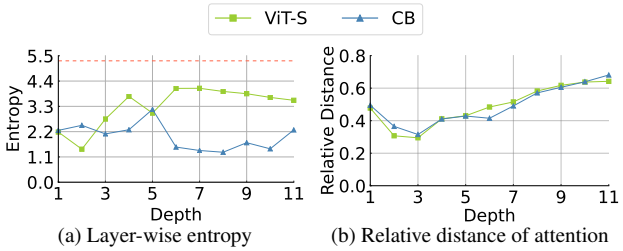
Does uniform attention help? To examine the effectiveness of the uniform attention, we inject the uniform attention in several ways to ViT-S as follows: **(A)** We replace one of the multi-head self-attention heads to be `CB` which reduces the number of parameters corresponding to the replaced head, **(B)** adjust the number of parameters of **(A)** to be comparable to the original ViT, **(C)** append `CB` to `MSA` as an extra head which increases the number of parameters, and infuse `CB` and $\texttt{CB}_\texttt{S}$. Table 1 shows the top-1 accuracy at ImageNet-1K. In **(A)**, **(B)**, **(C)**, `CB`, and $\texttt{CB}_\texttt{S}$ improve the accuracy consistently. The result explicitly tells us the broad benefits of injecting dense interactions into ViTs.

Attention entropy according to the depth. We have observed in Sec. 3.1 that the entropy of learned attention in ViT models tends to be high. From that, we have hypothesized that ViTs may benefit from an explicit injection of uniform attention. We examine now whether our `CB` module lowers the burden of the self-attention to learn dense interactions. We compare the entropy of the attention maps between ViT models with and without our `CB` module. Figure 7a shows layer-wise entropy values on ViT-S with and



(a) Our proposed CB module (b) Our vision transformer

Figure 6. **Context Broadcasting (CB) module.** (a) Our CB module broadcasts the context to each token. (b) The CB module is inserted at the end of the MLP block of the Vision Transformer (ViT) architectures. ViTs have other possible positions for our module, but we analyze that inserting at the end of MLP outperforms others.



(a) Layer-wise entropy (b) Relative distance of attention

Figure 7. **Attention entropy and relative distance.** We visualize the averaged entropies of the class token and the relative distance of spatial interactions across the layers. CB changes the spatial interactions of attention and reserves the long-range dependency.

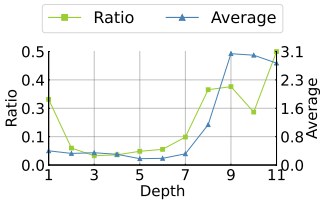


Figure 8. **Analysis of dimension scaling.** We plot values of the ratio and average of scaling weights across the layers. The high ratio and average indicate the preference for dense interactions.

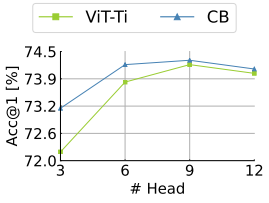


Figure 9. **Accuracy vs. # heads.** CB is effective with fewer heads where the lack of abundant spatial interactions may happen.

without our CB module. The insertion of CB lowers the entropy values significantly, especially in deeper layers. It seems that CB relaxes the representational burden for the MSA block and lets MSA focus on sparse interactions.

Relative distance according to the depth. We compute the relative distance of spatial interactions to see whether CB affects the range of spatial interactions. We define the distance as follows: $\texttt{dist} = \mathbb{E}_{i\neq j,\{i,j\}\in[1,N]}(a_{ij}||\mathbf{p}_i - \mathbf{p}_j||_1)$, where N is the number of spatial tokens, a_{ij} is the weight of attention between i-th and j-th tokens, and $\mathbf{p}_i$ is the normalized coordinate of i-th token. We exclude the case of self-interaction to analyze interactions of other tokens. As shown in Fig. 7b, ViT-S and CB have a similar tendency. Injecting the dense global interactions into ViT does not hurt the range of interactions.

Analysis on dimension scaling. We analyze the magnitude of scaling weights $\lambda \in \mathbf{\Lambda}$ in $\texttt{CB}_\texttt{S}$ to identify the trend of the need for uniform attention according to depth. We measure the ratio of the quantile of 90% and 10%, $|\lambda_{0.1}|/|\lambda_{0.9}|$. The ratio tells us how much high and low values of scaling weights are similar. We also compute the average of scaling weights according to depth. The average is related to the importance of uniform attention. As shown in Fig. 8, the ratio and average increase along with the depth. This indicates upper layers prefer dense interactions more than lower ones. The result coincides with the above observation of entropy analysis, as shown in Fig. 7a.

Model	No module	CB†	CB
ViT-Ti	72.2	73.2	**73.4**
ViT-S	79.9	80.5	**80.8**
ViT-B	81.8	82.0	**82.1**

Table 2. **ImageNet-1K performance of CB.** We denote CB† as CB applied to all layers.

Deeper Layers Need More Dense Interaction. As shown in Fig. 7a and Fig. 8, we observe that ViTs prefer dense interactions in the deeper layers. We compare infusing CB to all layers and upper layers. We denote the insertion of all layers as CB†. As shown in Table 2, CB achieves 1.2%p, 0.9%p, and 0.3%p higher accuracy than the vanilla ViT-Ti/-S/-B, respectively. CB also increases the top-1 accuracy further by 0.2%p, 0.3%p, and 0.1%p compared to ViT-Ti/-S/-B with CB†. Inserting CB in deeper layers improves the performance further; thus, deeper layers benefit more from the dense interactions.

Accuracy according to the number of heads. MSA can model abundant spatial interactions between tokens as the number of attention heads increases. To examine the relationship between the number of heads and spatial interac-

Extra resources	SE	CB
Extra parameters	Yes	**No**
Computation costs	High	**Low**
Implementation difficulty	Easy	**1 line**

(a) Extra resources.

Module	Acc@1 [%]
ViT-S	79.9
+ SE [25]	80.3
+ CB	**80.8**

(b) ImageNet-1K acc.

Table 3. **Comparison with SE and CB.** (a) Comparison in terms of the use of parameters, computation costs, and implementation difficulty. (b) Comparison of ImageNet-1K performance. Our CB contributes more to ViT-S compared with SE.

tions in MSA, we train ViT-Ti with and without CB by adjusting the number of heads of MSA. As shown in Fig. 9, the accuracy gap increases as the number of heads decreases. Our proposed module is, therefore, more effective in a lower number of heads rather than the large number of heads.

Comparison against SE. The SE module [25] shares a certain similarity to CB: both are modular attachments to neural network architecture. However, SE is designed to model the channel inter-dependency by exploiting pooling to construct a channel descriptor, two FC layers, and a sigmoid function. See the comparison between CB and SE in Table 3a. Finally, we compare the performance of the models with SE and CB. As shown in Table 3b, CB and SE improve the accuracy by 0.9%p and 0.4%p, respectively. Both modules improve the performance of ViT models, but the improvement is greater for CB.

Conclusion. We observe that the global dense interaction enhances the performance of ViTs and diverts the role of MSA to sparse interaction without reducing the distance of interaction. It validates that the injection of useful interactions helps MSA focus on other interactions. We believe exploring other sophisticated explicit interactions will further benefit MSA. In Sec. 4, we present the results of typical experiments based on our simple module.

4. Experiments

In Sec. 4.1, we experiment with which location we put our module in. In Secs. 4.2-4.4, we evaluate our modules on image classification, semantic segmentation, and object detection tasks. Sec. 4.5 provides the visualization of attention maps from ViT-S fine-tuned on segmentation task. In Sec. 4.6, we show results on the robustness benchmarks, including occlusion and adversarial attack. In Secs. 4.7 and 4.8, we evaluate our module on the vision-language Transformer for the Visual Question Answering task and on other architectures.

4.1. Where to Insert CB in a ViT

We study the best location for CB with respect to the main blocks for ViT architectures: MSA and MLP. We train ViT-S with our module positioned on MLP, MSA, and both and validate on ImageNet-1K. For simplicity, we infuse CB into

Module	Position MLP	Position MSA	FLOPs [G]	Acc@1 [%]
ViT-S	✗	✗	4.6	79.9
CB	✓	✗	4.6	**80.5**
	✗	✓	4.6	80.1
	✓	✓	4.6	80.1

(a) Position of CB to MLP and MSA.

Module	Position Front	Position Mid	Position End	FLOPs [G]	Acc@1 [%]
ViT-S	✗	✗	✗	4.6	79.9
CB	✓	✗	✗	4.6	79.9
	✗	✓	✗	4.6	**80.5**
	✗	✗	✓	4.6	**80.5**

(b) Position of CB in MLP.

Table 4. **Experiments with the position of CB.** (a) ImageNet-1K performance when CB is inserted to either MLP and MSA. (b) ImageNet-1K performance when CB is placed at Front, Mid, or End in an MLP block.

all layers. Note that this setting is different from the experiment in Table 1. We place CB to the main block without complex adjustments. As shown in Table 4a, CB improves the performance regardless of blocks but achieves higher accuracy by 0.4%p in an MLP block than either in an MSA block or both. It is notable, though, that adding CB increases the performance regardless of the location. We have chosen MLP as the default location of our CB module for the rest of the paper. This means that the self-attention and uniform attentions conduct their operation in MSA and MLP alternately. The alternation pattern considering the responsibility of modules can be found in prior work [3, 42].

Now, we study the best position of the CB module *within* an MLP block, which consists of two fully-connected (FC) layers and the Gaussian Error Linear Unit (GELU) nonlinear activation function [20]. Omitting the activation function for simplicity, we have three possible positions for CB: $<$ Front $>$ $-$FClayer$-$ $<$ Mid $>$ $-$FClayer$-$ $<$ End $>$. We train ViT-S with CB located at Front, Mid, and End, and validate on ImageNet-1K. Table 4b shows the performance; Mid and End increase accuracy by 0.6%p compared to the vanilla ViT-S. Mid demands four times larger computation costs than End because an MLP layer expands its channel dimensions four times rather than Front and End. We conclude that inserting CB at End of MLP tends to produce the best results overall.

Why is the improvement of the rear position larger than others? We conjecture that the gradient signal propagates to all parameters when CB is located at the End of MLP compared to being at the other places. For simplicity, we assume a single layer composed of the MSA and MLP blocks. If

Architecture	# Params [M]	FLOPs [G]	Acc@1 [%]	Acc@5 [%]	IN-V2 [%]	IN-ReaL [%]
ViT-Ti	5.7	1.3	72.2	91.1	59.9	80.1
+ CB	5.7	1.3	73.4	**91.9**	61.3	81.0
+ CB_S	5.7	1.3	**73.5**	**91.9**	**61.4**	**81.2**
ViT-S	22.0	4.6	79.9	95.0	68.1	85.7
+ CB	22.0	4.6	**80.8**	**95.4**	**69.3**	**86.2**
+ CB_S	22.0	4.6	80.4	95.1	68.7	85.9
ViT-B[2]	86.6	17.6	81.8	95.6	70.5	86.7
+ CB	86.6	17.6	**82.1**	95.7	**71.1**	**86.9**
+ CB_S	86.6	17.6	**82.1**	**95.8**	**71.1**	**86.9**

Table 5. **ImageNet-1K performance.** We train vision transformer architectures [12, 56] with CB and CB_S and evaluate the accuracy on ImageNet-1K [11], ImageNet-V2 [46], and ImageNet-ReaL [4]. **Bold** is the best number at each row. Our module improves all the metrics incurring negligible extra computational costs.

CB is located at End, the preceding weights in the MSA and MLP block are updated by the gradient signals by uniform attention. If CB is located at Front, the subsequent weights in the corresponding MLP block cannot receive the gradient signals during training.

Why is the improvement of Mid and End similar? There is no non-linear function (e.g., GELU) between Mid and End positions. Since uniform attention is the addition of a globally averaged token, the output is identical wherever CB is located at Mid and End. Therefore, the accuracy of both positions is similar.

As a further study, we compare infusing CB to all layers or upper layers. CB to upper layers achieves higher top-1 accuracy compared to CB to all layers in ViT-Ti/-S/-B.[3] Inserting CB in deeper layers improves the performance further; thus, deeper layers benefit more from the dense interactions.

4.2. Image Classification

We train ViTs [12] with our CB module on the ImageNet-1k training set and report accuracy on the validation set. We adopt strong regularizations following the DeiT [56]. We apply the random resized crop, random horizontal flip, Mixup [64], CutMix [63], random erasing [66], repeated augmentations [24], label-smoothing [51], and stochastic depth [28]. We use AdamW [35] with betas of (0.9, 0.999), a learning rate of $10^{-3}\cdot(\text{batch size})/1024$, and a weight decay of 0.05. The one-cycle cosine scheduling is used to decay the learning rate during the total epochs of 300. We implement based on PyTorch [41] and timm [60] on 8 V100 GPUs. We use torchprofile library to count the number of FLOPs. More details and additional experiments can be found in Appendix.

ViT-Ti/-S/-B [12] with our modules trained on ImageNet-1K are further validated on ImageNet-V2 [46] and ImageNet-Real [4]. Table 5 shows our modules CB and CB_S improve both precision and robustness of a model. CB does not add extra parameters, and CB_S increases only a few parameters; our modules demand negligible computation costs yet are effective for image classification. The results signify our observations about the preference and learning difficulty of dense global attention and injecting dense attention explicitly are all valid.

[3]The experiment can be found in Appendix.

Backbone	# Params [M]	mIoU [%]	
		40K	160K
ViT-Ti	34.1	35.5	38.9
+ CB		**36.5**	39.0
+ CB_S		36.1	**39.8**
ViT-S	53.5	41.5	43.3
+ CB		**41.9**	**43.9**
+ CB_S		41.6	43.1
ViT-B	127.0	44.3	45.0
+ CB		**45.1**	**45.6**
+ CB_S		44.6	45.3

Table 6. **ADE20K performance.** All models are based on UperNet [61]. Ours significantly improves the performance, and this is presumably because our module supplements global attention more to ViTs (like the atrous convolution [8]).

4.3. Semantic Segmentation

We validate our method for semantic segmentation on the ADE20K dataset [67, 68] consisting of 20K training and 5K validation images. For a fair comparison, we follow the protocol of XCiT [13] and Swin Transformer [34]. We adopt UperNet [61] and train for 40K iterations or 160K for longer training. Hyperparameters are the same as XCiT: the batch size of 16, AdamW with betas of (0.9, 0.999), the learning rate of $6\cdot10^{-5}$, weight decay of 0.01, and polynomial learning rate scheduling. We set the head dimension as 192, 384, and 512 for ViT-Ti/-S/-B, respectively. Table 6 shows the results with 40K and 160K training settings.

We observe that ViT-Ti/-S/-B with CB increase mIoU by 1.0, 0.4, and 0.8 for 40K iterations and 0.1, 0.6, and 0.6

[3]We increase the warm-up epochs for learning stability in ViT-B.

Model	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ViT-Ti	34.8	57.4	36.5	32.5	54.3	33.7
+ CB	**35.1**	**57.9**	**36.8**	**32.8**	**54.4**	**34.2**

Table 7. **COCO object detection and instance segmentation performance.** We finetune ViT-Ti on the COCO dataset for 12 epochs (1× schedule).

for 160K iterations, respectively. Similarly, CB_S improves mIoU except for 160K iterations in ViT-S. Infusing the context shows improvement in semantic segmentation; the performance improvement of ViT-B is not marginal, especially. The result would be related to the prior work [8, 65], which introduces the global context by atrous convolution and pyramid module. CB not only performs the dense interactions across tokens, which the original self-attention is hard to learn, but also supplies the global context.

4.4. Object detection

We fine-tune the pre-trained ViT-Ti on the COCO dataset and evaluate the performance of object detection and instance segmentation in Table 7. COCO consists of 118K training and 5K validation images with 80 categories. We follow the protocol of XCiT [13] and Swin Transformer [34]. We adopt Mask R-CNN with FPN and train models for 12 epochs (1× schedule) using AdamW with learning rate $10^{-4}\cdot\frac{\text{batch size}}{16}$ and weight decay 0.05. We do not apply CB to a block where features are feed-forwarded to FPN. Ours consistently improves performance. In object detection, CB improves 0.3, 0.5, and 0.3 in AP^b, AP^b_{50}, and AP^b_{75}, respectively. In instance segmentation, CB improves 0.3, 0.1, and 0.5 in AP^m, AP^m_{50}, and AP^m_{75}, respectively.

4.5. Segmentation Attention Visualization

We visualize the attention maps to understand how CB changes the interactions of MSA rather than entropies. We use the pre-trained ViT-S on ADE20K to extract the attention maps. The visualized attentions are extracted from the last layers before Feature Pyramid Network (FPN). See Fig. 10 for a comparison. We apply the same thresholding and min-max normalization in visualization for a fair comparison. ViT-S without CB needs dense aggregations more than ViT-S with CB. The visualization also validates that CB takes over the dense aggregations from the original self-attention. This implies that CB splits the burden of self-attention.

4.6. Evaluating Model Robustness

We evaluate the robustness of CB and CB_S with respect to center occlusion (Occ), ImageNet-A [21], and an adversarial attack [15]. For Occ, we zero-mask the center 112×112 patches of every validation image. ImageNet-A is the collection of challenging test images that an ensemble of ResNet50s has failed to recognize. We employ the fast sign gradient method (FGSM [15]) for the adversarial attack. Table 8 shows the results of the robustness benchmark. CB_S increases by 0.7, 0.1, and 0.6 of Occ, ImageNet-A, and FGSM, respectively. CB does 1.0, 2.2, and 5.1, respectively.

Architecture	Occ [%]	ImageNet-A [%]	FGSM [%]
ViT-S	73.0	19.0	27.2
+ CB_S	73.7	19.1	27.8
+ CB	**74.0**	**21.2**	**32.3**

Table 8. **Robustness evaluation.** We evaluate ViT-S with CB and CB_S on center occlusion (Occ), ImageNet-A, and fast sign gradient method (FGSM) attack. Ours shows improved robustness across the board against ViT-S.

Noise Type	ViT-S	CB
Nothing	43.3	43.9
Shot Noise	40.22 ± 0.15	41.09 ± 0.09
Gaussian Noise (sigma=5.0)	42.55 ± 0.08	43.44 ± 0.08
Gaussian Noise (sigma=10.0)	40.22 ± 0.07	41.07 ± 0.06
Gaussian Blur (sigma=1.0)	42.29	43.26
Gaussian Blur (sigma=2.0)	40.83	41.44

Table 9. **Robustness evaluation on ADE20K with input perturbations.** We evaluate ViT-S with and without CB on shot noise, Gaussian noise, and Gaussian blur. The performance gap of perturbations between ViT-S and CB is larger than the one of nothing. It shows that CB improves robustness. We run the experiments on random noise five times and report a mean with a confidence interval of 95%.

We also evaluate the robustness on ADE20K using input perturbations [19], *e.g.*, shot noise, Gaussian noise, and Gaussian blur. We run the experiments by five times on random noise and report the mean and confidence interval of 95%. Table 9 shows the performance of mIoU. The performance gap of ViT-S with and without CB increases from 0.6 up to 0.97. This shows that our 1 line of code can improve the ViT models' robustness against input perturbation in the semantic segmentation task.

4.7. Vision-Language Transformer

Transformer becomes the standard architecture for multi-modal learning because of the succinct structure. For example, Transformer employs modality-specific linear projection [31, 26, 23, 2]. We evaluate our module on the Vision-Language Transformer, ViLT [31]. We fine-tune the pre-trained ViLT on VQAv2 [16] using the official code. Table 10 shows the results of the performance. We first reproduce

Architecture	Acc [%]
ViLT [31]	71.28
+ CB_S (Image)	71.44
+ CB_S (Text)	**71.46**
+ CB_S (Both)	71.42

Table 10. **Vision language transformer results on VQAv2.** We fine-tune ViLT with CB_S on tokens of image, text, and both.

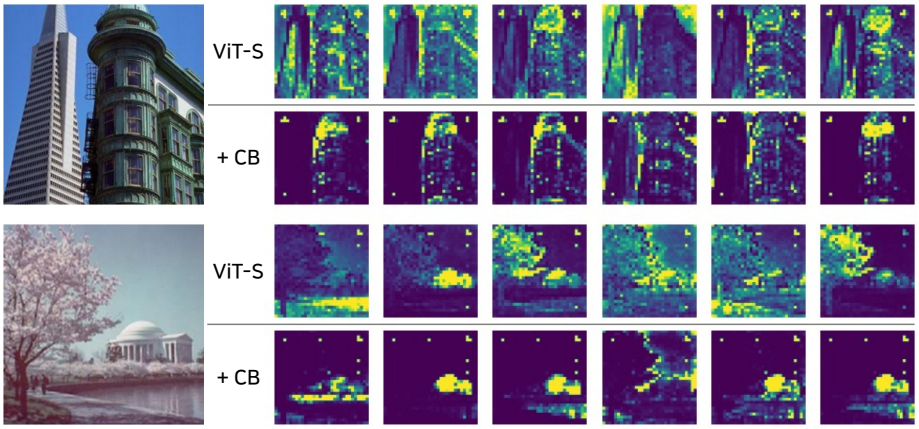


Figure 10. **Visualization of attention maps.** Using ViT-S fine-tuned on ADE20K, we visualize the attention maps of the last layers of heads. The first row of each image corresponds with ViT-S, and the second row does ViT-S with CB. We can observe that CB reduces the dense aggregation of self-attention. By infusing uniform attention, MSA aggregates more informative signals, such as objects.

Architecture	# Params [M]	FLOPS [G]	Acc@1 [%]
PiT-B [22]	73.8	12.4	82.0
+ CB	73.8	12.4	**82.6**
Mixer-S/16 [53]	18.5	3.8	74.3
+ CB	18.5	3.8	**74.9**

Table 11. **ImageNet-1K performance on other architectures.** Ours also improves the performance in other models of Transformer and MLP.

the baseline and reach the reported number (71.26). Our module is applied to the image, text, and both tokens, and in all cases, it improves the accuracy by 0.16, 0.18, and 0.14, respectively, compared with the baseline accuracy.

4.8. Other Architectures

We evaluate CB on PiT [22] and Mixer [53]. PiT-B is the variant of the original Vision Transformer introducing spatial dimension reduction. Mixer is pioneering work of the feed-forward architectures [53, 55], mainly consisting of FC layers. The structure of feed-forward architecture follows ViT except for MSA. Spatial interactions of feed-forward are done through transposing visual data followed by an FC layer. We insert our module at MLP in PiT and Mixer [53]. For a fair comparison, we reproduce the baseline Mixer-S/16 with the DeiT training regime [56] and train ours with the same one. Our module increases the performance of those architectures, as shown in Table 11.

5. Conclusion

We look closer at the spatial interactions in ViTs, especially in terms of density. We have been motivated by the preliminary exploration and observations that suggest ViT models prefer dense interactions. We also show that, at least from the optimization point of view, uniform attention is perhaps the most challenging attention for softmax-based attention to learn. The preference and optimization difficulty of learning dense interactions are not aligned. It leads us to introduce further dense interactions manually by a simple module: Context Broadcasting (CB). Inserted at intermediate layers of ViT models, CB adds the averaged token to tokens. Additionally, we propose a dimension scaling version of CB, called CB_S, to infuse the dense interactions selectively. It turns out that our simple module improves the ViT performances across various benchmarks, including image classification, semantic segmentation, and visual-language tasks. CB only takes 1 line of code, a few more FLOPs, and zero parameters using this module. We hope that our module will further improve your ViT models and that our observations provide insights for modeling the token interactions of ViTs.

Our work introduces the simplest form of dense interaction that complement self-attention. One may propose a more sophisticated and effective module that makes self-attention focus on the crucial interactions that should be only dealt with by self-attention, intractable otherwise. We believe that this would be an exciting research direction.

Acknowledgment N. Hyeon-Woo, K. Y.-J. and T.-H. Oh were partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00124, Development of Artificial Intelligence Technology for Self-Improving Competency-Aware Learning Capabilities; No.2022-0-00290, Visual Intelligence for Space-Time Understanding and Generation based on Multi-layered Visual Common Sense; No. 2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network).

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016. 4

[2] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *NeurIPS*, 2021. 1, 2, 8

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 2, 6

[4] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020. 2, 7

[5] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. 4

[6] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Global context networks. *IEEE TPAMI*, to appear. 2

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017. 7, 8

[9] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *ICLR*, 2020. 2

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 7

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 3, 7

[13] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Herve Jegou. XCit: Cross-covariance image transformers. In *NeurIPS*, 2021. 7, 8

[14] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *CVPR*, 2022. 1, 2

[15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2014. 8

[16] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, 2017. 2, 8

[17] Haoyu He, Jing Liu, Zizheng Pan, Jianfei Cai, Jing Zhang, Dacheng Tao, and Bohan Zhuang. Pruning self-attentions into convolutional layers in single path. *arXiv preprint arXiv:2111.11802*, 2021. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[19] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 8

[20] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 6

[21] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 8

[22] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *ICCV*, 2021. 9

[23] Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*, 2020. 8

[24] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *CVPR*, 2020. 7

[25] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 2, 6

[26] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *ICCV*, 2021. 1, 2, 8

[27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2

[28] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 7

[29] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019. 2

[30] Ji-Yeon Kim, Hyun-Bin Oh, Dahun Kim, and Tae-Hyun Oh. Mindvps: Minimal model for depth-aware video panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023. 2

[31] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021. 1, 2, 8

[32] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 2

[33] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 2

[34] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 7, 8

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 7

[36] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *NeurIPS*, 2016. 2

[37] Xu Ma, Huan Wang, Can Qin, Kunpeng Li, Xingchen Zhao, Jie Fu, and Yun Fu. A close look at spatial modeling: From attention to convolution. *arXiv preprint arXiv:2212.12552*, 2022. 3

[38] Yuxin Mao, Jing Zhang, Zhexiong Wan, Yuchao Dai, Aixuan Li, Yunqiu Lv, Xinyu Tian, Deng-Ping Fan, and Nick Barnes. Transformer transforms salient object detection and camouflaged object detection. *arXiv preprint arXiv:2104.10127*, 2021. 2

[39] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. In *NeurIPS*, 2021. 1, 2, 3

[40] Namuk Park and Songkuk Kim. How do vision transformers work? In *ICLR*, 2022. 2

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 4, 7

[42] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multimodal fusion transformer for end-to-end autonomous driving. In *CVPR*, 2021. 6

[43] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *ICCV*, 2007. 2

[44] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *NeurIPS*, 2021. 1, 2, 3

[45] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *NeurIPS*, 32, 2019. 2

[46] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019. 2, 7

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 2

[48] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1):2–23, 2009. 2

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014. 2

[50] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *CVPR*, 2021. 1, 2

[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 7

[52] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 2

[53] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, 2021. 9

[54] Antonio Torralba. Contextual priming for object detection. *IJCV*, 53(2):169–191, 2003. 2

[55] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021. 9

[56] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1, 2, 3, 4, 7, 9

[57] Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L Griffiths. Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*, 2021. 2

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2

[59] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 1, 2

[60] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 7

[61] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 7

[62] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2

[63] 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 *ICCV*, 2019. 7

[64] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 7

[65] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 8

[66] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020. 7

[67] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 7

[68] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127(3):302–321, 2019. 2, 7