Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet

Li Yuan[1]*, Yunpeng Chen[2], Tao Wang[1]*, Weihao Yu[1], Yujun Shi[1],
Francis EH Tay[1], Jiashi Feng[1], Shuicheng Yan[2]

[1]National University of Singapore [2] YITU Technology
yuanli@u.nus.edu, yunpeng.chen@yitu-inc.com, shuicheng.yan@gmail.com

Abstract

Transformers, which are popular for language modeling, have been explored for solving vision tasks recently, e.g., *the Vision Transformers (ViT) for image classification. The ViT model splits each image into a sequence of tokens with fixed length and then applies multiple Transformer layers to model their global relation for classification. However, ViT achieves inferior performance compared with CNNs when trained from scratch on a midsize dataset* (e.g., *ImageNet). We find it is because: 1) the simple tokenization of input images fails to model the important local structure* (e.g., *edges, lines) among neighboring pixels, leading to its low training sample efficiency; 2) the redundant attention backbone design of ViT leads to limited feature richness in fixed computation budgets and limited training samples.*

To overcome such limitations, we propose a new Tokens-To-Token Vision Transformers (T2T-ViT), which introduces 1) a layer-wise Tokens-to-Token (T2T) transformation to progressively structurize the image to tokens by recursively aggregating neighboring Tokens into one Token (Tokens-to-Token), such that local structure presented by surrounding tokens can be modeled and tokens length can be reduced; 2) an efficient backbone with a deep-narrow structure for vision transformers motivated by CNN architecture design after extensive study. Notably, T2T-ViT reduces the parameter counts and MACs of vanilla ViT by 200%, while achieving more than 2.5% improvement when trained from scratch on ImageNet. It also outperforms ResNets and achieves comparable performance with MobileNets by directly training on ImageNet. For example, T2T-ViT with ResNet50 comparable size can achieve 80.7% accuracy on ImageNet. [1]

1. Introduction

Self-attention models for language modeling like Transformers [37] have been recently applied to vision tasks, including image classification [5, 12, 43], object detection [3, 59] and image processing like denoising, super-resolution and deraining [4]. Among them, the Vision Transformer (ViT) [12] is the first fully-transformer model that can be directly applied for image classification. In particular, ViT splits each image into 14×14 or 16×16 patches (*a.k.a.*, tokens) with fixed length; then following the practice of transformer for language modeling, ViT applies transformer layers to model global relation among these tokens for input classification.

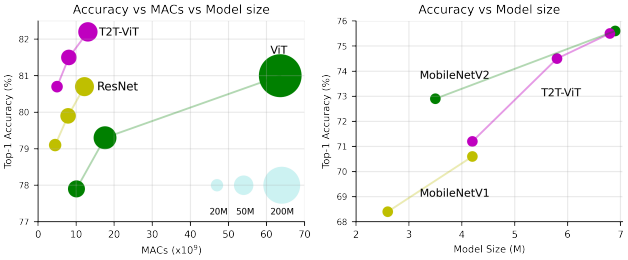


Figure 1. The comparison between T2T-ViT with ViT, ResNet and MobileNet when trained from scratch on ImageNet. Left: the performance curve of MACs vs. top-1 accuracy. Right: the performance curve of model size vs. top-1 accuracy.

Though ViT proves full-transformer architecture is promising for vision tasks, its performance is still inferior to similar-sized CNN counterparts (*e.g.*, ResNets), when trained from scratch on a midsize dataset (*e.g.*, ImageNet). We hypothesize that such performance gap roots in two main limitations of ViT: 1) the straightforward tokenization of input images by hard split makes ViT unable to model the image local structure like edges and lines, thus it requires significantly more training samples (like JFT-300M for pre-training) than CNNs for achieving similar performance; 2) the backbone of ViT is not well-designed as CNNs for vision tasks where the redundancy in the attention backbone design of ViT leads to limited feature richness and difficulties in model training.

To verify our hypothesises, we conduct a pilot study to investigate the difference in the learned features of ViT-L/16 [12] and ResNet50 [15] through visualization in Fig 2. We observe that features of ResNet capture the desired local

*Work done during an internship at Yitu Tech.
[1]Code: https://github.com/yitu-opensource/T2T-ViT

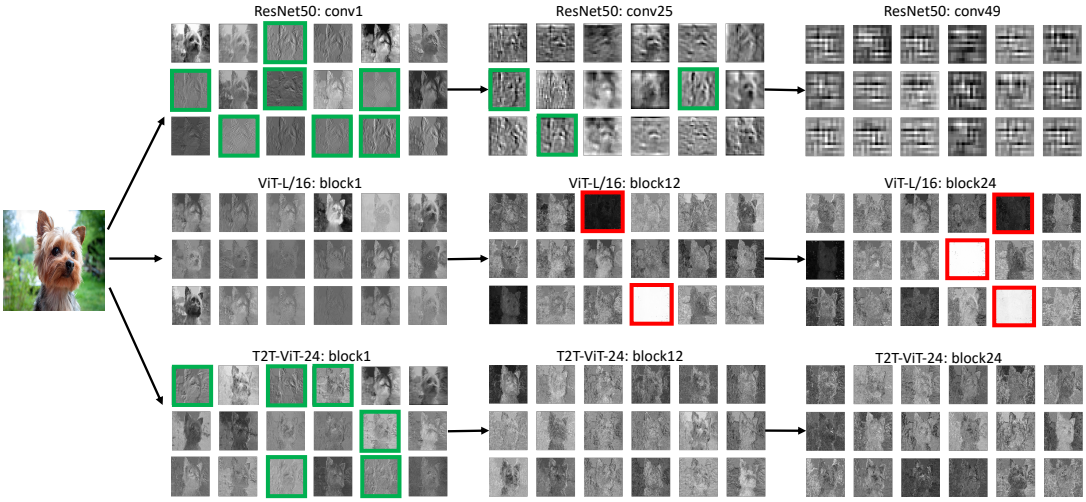


Figure 2. Feature visualization of ResNet50, ViT-L/16 [12] and our proposed T2T-ViT-24 trained on ImageNet. The green boxes highlight learned low-level structure features such as edges and lines. The red boxes highlight invalid feature maps with zero or too large values. Note the feature maps visualized here for ViT and T2T-ViT are not attention maps, but image features reshaped from tokens. For better visualization, we scale input image to size 1024×1024.

structure(edges, lines, textures, etc.) progressively from the bottom layer (conv1) to the middle layer (conv25). However, the features of ViT are quite different: the structure information is poorly modeled while the global relations (*e.g.*, the whole dog) are captured by all the attention blocks. This observation verifies our hypothesis that the vanilla ViT ignores the local structures when directly split images to tokens with a fixed length. Besides, we find many channels in ViT have zero value (highlighted in red in Fig 2), implying the backbone of ViT is not efficient as ResNets and presents limited feature richness when training samples are not enough.

These properties motivate us to design a new full-transformer vision model to overcome the above limitations: 1) Instead of the naive tokenization used in ViT [12], we propose a progressive tokenization module to aggregate neighboring *Tokens* to one *Token* (named Tokens-to-Token module), which can model the local structure information from surrounding tokens and reduce the length of tokens iteratively. Specifically, in each Token-to-Token (T2T) step, the tokens output by a Transformer layer are reconstructed as an image (*re-structurization*) which is then split into tokens with overlapping (*soft split*) and finally the surrounding tokens are aggregated together by flattening the split patches. Thus the local structure from surrounding patches is embedded into the tokens to be input into the next Transformer layer. By conducting T2T iteratively, the local structure is aggregated into tokens and the length of tokens can be reduced by the aggregation process. 2) To find an efficient backbone for vision transformers, we explore to borrow some useful architecture from CNNs to design transformer layers to improve the feature richness, and we find that "deep-narrow" architecture design with fewer channels but more layers in ViT brings much better performance with comparable model size and MACs(Multi-Adds). Specifically, we extensively investigate Wide-ResNets (shallow-wide vs deep-narrow structure) [50], DenseNet (dense connection) [21], ResneXt structure [44], Ghost operation [14, 57] and channel attention [20], and we find that among these CNN-based architecture designs, deep-narrow structure [50] is the most efficient and effective architecture to ViT by reducing the model size and MACs significantly with nearly no degradation in performance, which also indicates that the architecture engineering of CNNs can benefit the backbone design of vision transformers.

Based on the T2T module and deep-narrow backbone architecture, we develop the Tokens-to-Token Vision Transformers (T2T-ViT), which significantly boosts the performance when trained from scratch on ImageNet (Fig 1), and is more lightweight than the vanilla ViT. As shown in Fig 1, our T2T-ViT with 21.5M parameters and 5.2G MACs can achieve 80.7% top-1 accuracy on ImageNet, much higher than the accuracy of ViT [12] with 48.6M parameters and 10.1G MACs (78.1%). This result is also higher than the popular CNNs of similar size, like ResNet50 with 25.5M parameters (76%-79%). Besides, we also design lite variant of T2T-ViT by simply adopting fewer layers, which achieves comparable results with MobileNetV1 [17] (Fig 1).

To sum up, our contributions are three-fold:

- For the first time, we show that by carefully designing transformers architecture (T2T module and efficient backbone), it can outperform CNNs at different complexities on the standard midsize dataset (i.e. Im-

ageNet) without pre-training on JFT-300M.

- We develop a novel progressive tokenization for Vision Transformers and demonstrate its advantage over the simple tokenization approach by ViT, and propose a T2T module that can encode the important local structure for each token.

- We show the architecture engineering of CNNs can benefit the backbone design of ViT to improve the feature richness and reduce redundancy. Through extensive experiments, we find deep-narrow architecture design works best for Vision Transformers.

2. Related Work

Transformers in Vision Transformers [37] are the models that entirely rely on the self-attention mechanism to draw global dependencies between input and output, which has dominated in natural language modelling [10, 30, 2, 46, 29, 23]. A Transformer layer mostly consists of a multi-head self-attention layer (MSA) and an MLP block. Layer-norm (LN) is applied before each layer and residual connections in both the self-attention layer and MLP block. Recent works have explored to apply Transformers to various vision tasks: image classification [5, 12], object detection [3, 59, 56, 8, 34], segmentation [4, 40], image enhancement [4, 45], image generation [27], video processing [58, 51], and 3D point cloud processing [54]. Among these works, the Vision Transformer (ViT) proves that a pure Transformer architecture can attain state-of-the-art performance on image classification. However, ViT heavily relies on large-scale datasets such as ImageNet-21k and JFT-300M (which is not publically available) for model pre-training, requiring huge computation resources. In contrast, our proposed T2T-ViT is more efficient and can be trained on ImageNet without using those large-scale datasets. A recent concurrent work DeiT [36] applies Knowledge Distillation [16, 48] to improve the original ViT by adding a KD token along with the class token, which is orthogonal to our work, as our T2T-ViT focuses on the architecture design.

Self-attention in CNN Self-attention mechanism has been widely applied to CNN in vision task [38, 55, 19, 47, 20, 39, 1, 6, 18, 31, 42, 13]. Among these works, the SE block [20] applies attention to channel dimensions and non-local networks [39] are designed for capturing long-range dependencies via global attention. Compared with most of the work exploring the global attention on images [1, 42, 13, 39], some works [18, 31] also explore the self-attention in a local patch to reduce the memory and computation cost. More recently, SAN [53] explores both the pairwise and patchwise self-attention for image recognition, where the patchwise self-attention is a generalization of convolution. In this work, we also replace the T2T module with multiple convolution layers in the experiments and find that the convolution layers can not perform better than our designed T2T module.

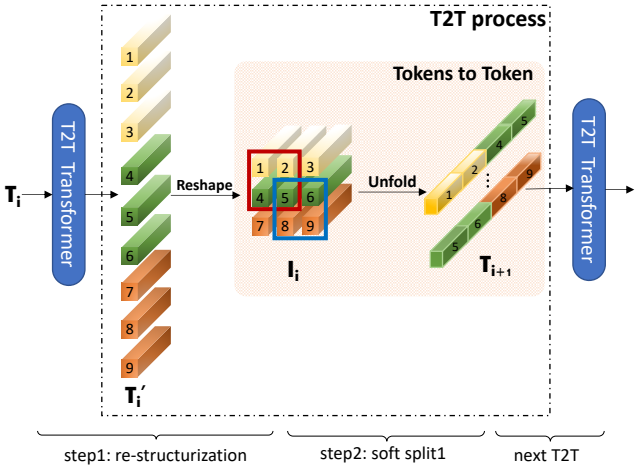


Figure 3. The tokens T_i are re-structurized as image I_i after a transformation and reshaping, then I_i is split with overlapping to tokens T_{i+1} again. Tokens-to-Token in T2T process: the four tokens (1,2,4,5) of the input I_i are concatenated to form one token in T_{i+1}. The T2T Transformer can be a normal Transformer layer [37] or efficient transformer like Performer layer [34] in the case of limited GPU memory.

3. Tokens-to-Token ViT

To overcome the limitations of simple tokenization and the inefficient backbone of ViT, we propose Tokens-to-Token Vision Transformers (T2T-ViT) which can progressively tokenize the image to tokens and has an efficient backbone. So T2T-ViT consists of two main components (Fig 4): 1) A layer-wise "Tokens-to-Token module" (T2T module) to model the local structure information of images and reduce the length of tokens progressively; 2) An efficient "T2T-ViT backbone" to draw the global attention relations on the tokens from T2T module. We adopt a deep-narrow structure for the backbone to reduce redundancy and improve the feature richness after exploring several CNN-based architecture designs. We now explain these components one by one.

3.1. Tokens-to-Token: Progressive Tokenization

The Token-to-Token (T2T) module aims to overcome the limitation of simple tokenization in ViT, which can progressively structurize image to tokens and model the local structure information. And the length of tokens can be reduced iteratively. Each T2T process has two steps: *Re-structurization* and *Soft Split (SS)* (Fig 3).

Re-structurization As shown in Fig 3, give a sequence of tokens T from last layer, here T is the "word" tokens from last layer. The T will be transformed by the self-attention block (the T2T Transformer as shown in Fig 3):

$$T' = \text{MLP}(\text{MSA}(T)), \tag{1}$$

where MSA denotes the multihead self-attention operation with layer normalization and "MLP" is the multilayer perceptron with layer normalization in the standard Transformer [12]. Then the tokens will be reshaped as image on spatial dimension,

$$I = \text{Reshape}(T'), \tag{2}$$

where "Reshape" re-organizes tokens $T' \in \mathbb{R}^{l \times c}$ to $I \in \mathbb{R}^{h \times w \times c}$, where l is the length of T', h, w, c are height, width and channel respectively, and $l = h \times w$.

Soft Split As shown in Fig 3, after obtaining the re-structurized image I, we apply the soft split on it to model the local structure information and reduce the length of tokens. Specifically, to avoid information loss in generating tokens from the re-structurization image, we split the image into patches with overlapping. As such, each patch is correlated with surrounding patches to establish prior knowledge that there should be stronger correlations between surrounding tokens. The tokens in each split patch are concatenated as one token (Tokens-to-Token, Fig 3), thus the local information can be aggregating from surrounding pixels and patches.

When conducting the soft split, the size of each patch is $k \times k$ with s overlapping and p padding on the image, where $k - s$ is similar to the stride in convolution operation. So for the reconstructed image $I \in \mathbb{R}^{h \times w \times c}$, the length of output tokens T_o after soft split is

$$l_o = \left\lfloor \frac{h + 2p - k}{k - s} + 1 \right\rfloor \times \left\lfloor \frac{w + 2p - k}{k - s} + 1 \right\rfloor. \tag{3}$$

Each split patch has size $k \times k \times c$. We flatten all patches in spatial dimensions to tokens $T_o \in \mathbb{R}^{l_o \times ck^2}$. After the soft split, the output tokens are fed for the next T2T process.

T2T module By conducting the above Re-structurization and Soft Split iteratively, the T2T module can progressively reduce the length of tokens and transform the spatial structure of the image. The iterative process in T2T module can be formulated as:

$$\begin{aligned} T_i' &= \text{MLP}(\text{MSA}(T_i), \\ I_i &= \text{Reshape}(T_i'), \\ T_{i+1} &= \text{SS}(I_i), \qquad i = 1...(n-1) \end{aligned} \tag{4}$$

For the input image I_0, we apply a soft split at first to split the image to tokens: $T_1 = \text{SS}(I_0)$. After the final iteration, the output tokens T_f of T2T module has fixed length, thus the backbone of T2T-ViT can model the global relations on T_f.

Additionally, as the length of tokens in T2T module is large than the normal case (14×14 or 16×16) in ViT, the MACs and memory usage are huge. To overcome the limitations, in our T2T module, we set the channel dimension of T2T layer small (32 or 64) to reduce MACs, and optionally adopt an efficient Transformer such as Performer [7] layer to reduce memory usage in the case of limited GPU memory. We provide an ablation study on the difference between adopting the standard Transformer layer and Performer layer in our experiments.

3.2. T2T-ViT Backbone

As many channels in the backbone of vanilla ViT are invalid (Fig 2), we plan to find an efficient backbone for our T2T-ViT to reduce the redundancy and improve the feature richness. Thus we explore different architecture designs for ViT and borrow some designs from CNNs to improve the backbone efficiency and enhance the richness of the learned features. As each transformer layer has skip connection as ResNets, a straightforward idea is to apply dense connection as DenseNet [21] to increase the connectivity and feature richness, or apply Wide-ResNets or ResNeXt structure to change the channel dimension and head number in the backbone of ViT. To give a comprehensive comparison, we explore five architecture designs from CNNs to ViT:

1. Dense connection as DenseNet [21];
2. Deep-narrow vs shallow-wide structure as discussed in Wide-ResNets [50];
3. Channel attention as Squeeze-an-Excitation (SE) Networks [20];
4. More split heads in multi-head attention layer as ResNeXt [44];
5. Ghost operations as GhostNet [14].

The details of these structure designs in ViT are given in Appendix A.1. We conduct extensive experiments on the structures transferring in Experiment 4.2. We empirically find that 1) by adopting a deep-narrow structure that simply decreases channel dimensions to reduce the redundancy in channels and increase layer depth to improve feature richness in ViT, it can reduce both the model size and MACs but improve performance; 2) the channel attention as SE block also improves ViT but is less effective than using the deep-narrow structure.

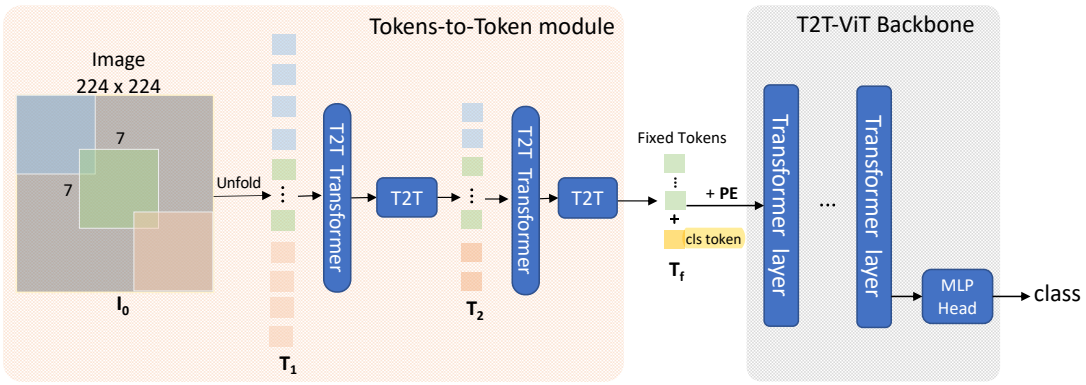


Figure 4. The overall network architecture of T2T-ViT. In the T2T module, the input image is first soft split as patches, and then unfolded as a sequence of tokens T_0. The length of tokens are reduced progressively in the T2T module (we use two iteration here and output T_f). Then the T2T-ViT backbone takes the fixed tokens as input and output the predictions. The two T2T blocks are the same as Fig 3 and PE is Position Embedding.

Table 1. The structure details of T2T-ViT, where T2T-ViT-14, T2T-ViT-19, T2T-ViT-24 have comparable model size with ResNet50, ResNet101 and ResNet152. T2T-ViT-7, T2T-ViT-10 and T2T-ViT-12 have comparable model size with MobileNetV1 [17] and MobileNetV2 [32]. For T2T Transformer layer, we adopt Transformer layer for T2T-ViT$_t$-14 and Performer layer for T2T-ViT-14 in the case of limited GPU memory. For ViT, 'S' means Small, 'B' is Base and 'L' is large, here 'ViT-S/16' is a variant from the original ViT-B/16 [12].

Models	Tokens-to-Token module				T2T-ViT backbone			Model size	
	T2T Transformer	Depth	Hidden dim	MLP size	Depth	Hidden dim	MLP size	Params (M)	MACs (G)
ViT-S/16 [12]	-	-	-	-	8	786	2358	48.6	10.1
ViT-B/16 [12]	-	-	-	-	12	786	3072	86.8	17.6
ViT-L/16 [12]	-	-	-	-	24	1024	4096	304.3	63.6
T2T-ViT$_t$-14	Transformer	2	64	64	14	384	1152	21.5	5.2
T2T-ViT$_t$-19	Transformer	2	64	64	19	448	1344	39.0	8.4
T2T-ViT$_t$-24	Transformer	2	64	64	24	512	1536	64.1	13.2
T2T-ViT-14	Performer	2	64	64	14	384	1152	21.4	4.8
T2T-ViT-19	Performer	2	64	64	19	448	1344	39.0	8.0
T2T-ViT-24	Performer	2	64	64	24	512	1536	63.9	12.6
T2T-ViT-7	Performer	2	64	64	8	256	512	4.2	0.9
T2T-ViT-10	Performer	2	64	64	10	256	512	5.6	1.2
T2T-ViT-12	Performer	2	64	64	12	256	512	6.8	1.4

Based on the findings of the above studies on transferring from CNN-based architecture designs to ViT, we design a deep-narrow architecture for our T2T-ViT backbone. Specifically, it has small channel number and hidden dimension d but more layers b. For the tokens with fixed length T_f from the last layer of T2T module, we concatenate a class token to it and then add Sinusoidal Position Embedding (PE) to it, which is same as ViT to do classification:

$$\begin{aligned} T_{f_0} &= [t_{cls}; T_f] + E, & E &\in \mathbb{R}^{(l+1)\times d} \\ T_{f_i} &= \mathrm{MLP}(\mathrm{MSA}(T_{f_i-1})), & i &= 1...b \\ y &= \mathrm{fc}(\mathrm{LN}(T_{f_b})), \end{aligned} \tag{5}$$

where E is Sinusoidal Position Embedding, LN is layer normalization, fc is one fully-connected layer for classification and y is the output prediction.

3.3. T2T-ViT Architecture

The T2T-ViT has two parts: the Tokens-to-Token (T2T) module and the T2T-ViT backbone (Fig 4). There are various possible design choices for the T2T module. Here, we set $n = 2$ as shown in Fig 4, which means there is $n+1 = 3$ soft split and $n = 2$ re-structurization in T2T module. The patch size set for the three soft split is $P = [7, 3, 3]$, and the overlapping set is $S = [3, 1, 1]$, which reduces size of the

input image from 224×224 to 14×14 according to Eqn 3.

The T2T-ViT backbone takes the tokens with fixed length from the T2T module as input, same as ViT; but it has a deep-narrow architecture design with smaller hidden dimensions (256-512) and MLP size (512-1536) than ViT. For example, the T2T-ViT-14 has 14 transformer layers in the T2T-ViT backbone with 384 hidden dimensions, while the ViT-B/16 with 12 transformer layers and 768 hidden dimensions is 3x larger than T2T-ViT-14 in parameters and MACs.

To fairly compare with the common hand-designed CNNs, we design T2T-ViT models to have comparable size with ResNets and MobileNets. Specifically, we design three models: T2T-ViT-14, T2T-ViT-19 and T2T-ViT-24 of comparable parameters with ResNet50, ResNet101 and ResNet152 respectively. To compare with small models like MobileNets, we design three lite models: T2T-ViT-7, T2T-ViT-10 and T2T-ViT-12 with comparable model size with MibileNetV1 and MibileNetV2. The three lite TiT-ViT have no special designs or tricks like efficient convolution [26] but simply reduce the layer depth, hidden dimension, and MLP ratio. The network details are summarized in Table 1.

4. Experiments

We conduct the following experiments with T2T-ViT for image classification on ImageNet: (a) We validate that the T2T-ViT can outperform ViT with fewer parameters and MACs, and it can perform better than ResNet of comparable size when trained from scratch on ImageNet (Sec. 4.1); (b) Among the five T2T-ViT backbone architecture designs with inspiration from CNNs, the deep-narrow structure perform the best, and SE block and Ghost operations can further improve the performance of T2T-ViT (Sec. 4.2); (c) We conduct ablation study to demonstrate the effects of the T2T module and the deep-narrow architecture design of T2T-ViT (Sec. 4.3).

4.1. T2T-ViT on ImageNet

All the experiments are conducted on ImageNet dataset [9], with around 1.3 million images in the training set and 50k images in the validation set. We use batch size 256 or 512 with 8 NVIDIA GPUs for training. We adopt Pytorch [28] library and Pytorch image models library (timm) [41] to implement our models and conduct all the experiments. For fair comparisons, we implement the same training scheme for the CNN models, the ViT, and our T2T-ViT. Throughout the experiments on ImageNet, we set image size as 224×224, and adopt some common data augmentation methods such as mixup [52] and cutmix [11, 49] for both CNN and ViT&T2T-ViT model training, because the ViT models need more training data to reach a reasonable performance. We train these models for 310 epochs, using AdamW [25] as the optimizer and cosine learning rate decay [24]. We also use both Transformer layer and Performer layer in T2T module for our models, resulting in T2T-ViT$_t$-14/19/24 (Transformer) and T2T-ViT-14/19/24 (Performer).

Table 2. Comparison between T2T-ViT and ViT by training from scratch on ImageNet.

Models	Top1-Acc (%)	Params (M)	MACs (G)
ViT-S/16 [12]	78.1	48.6	10.1
T2T-ViT$_t$-14	**80.7**	**21.5**	**5.2**
T2T-ViT$_t$-19	**81.4**	**39.0**	**8.4**
ViT-B/16 [12]	79.7	86.4	17.6
ViT-L/16 [12]	81.1	304.3	63.6
T2T-ViT$_t$-24	**82.2**	**64.1**	**13.2**

T2T-ViT vs ViT We first compare the performance of T2T-ViT and ViT on ImageNet. The experimental results are given in Tab 2. Our proposed T2T-ViT is much smaller than ViT in terms of the number of parameters and MACs, yet giving higher performances. For example, the small ViT model ViT-S/16 with 48.6M and 10.1G MACs has 78.1% top-1 accuracy when trained from scratch on ImageNet, while our T2T-ViT$_t$-14 with only 44% parameter and 51% MACs achieves 80.7% accuracy. Comparing ViT-S/16 with T2T-ViT$_t$-19, the two models have similar parameters and MACs, and our T2T-ViT$_t$-19 is even smaller than ViT-S/16, but it achieves more than 3.0% improvement. If we compare the T2T-ViT$_t$-24 with ViT-L/16, our T2T-ViT$_t$-24 can reduce the parameters and MACs around 500% but achieve more than 1.0% improvement on ImageNet.

T2T-ViT vs ResNet For fair comparisons, we set up three T2T-ViT models that have a similar model size and MACs with ResNet50, ResNet101 and ResNet152. The experimental results are given in Tab 3. The proposed T2T-ViT achieves 1%-2.5% superior to ResNets with similar model size and MACs. For example, compared with ResNet50 of 25.5M parameters and 4.3G MACs, our T2T-ViT-14 and T2T-ViT$_t$-14 have 21.5M parameters and 4-5G MACs and obtain 80.6% and 80.7% accuracy on ImageNet.

T2T-ViT vs MobileNets The T2T-ViT-7 and T2T-ViT-10 have similar model size with MobileNetV1 [17] and MobileNetV2 [32], but achieve comparable performance with MobileNetV1 (Table 4). But we also note that the MACs of our T2T-ViT are still slightly larger than MobileNets because of the dense operations in Transformers. However, there is no special operation or tricks like efficient convolu-

Table 3. Comparison between our T2T-ViT with ResNets on ImageNet. T2T-ViT$_t$-14: using Transformer in T2T module. T2T-ViT-14: using Performer in T2T module. * means we train the model with our training scheme for fair comparisons.

Models	Top1-Acc (%)	Params (M)	MACs (G)
ResNet50 [15]	76.2	25.5	4.3
ResNet50*	79.1	25.5	4.3
T2T-ViT-14	**80.6**	21.4	4.8
T2T-ViT$_t$-14	**80.7**	21.5	5.2
ResNet101 [15]	77.4	44.6	7.9
ResNet101*	79.9	44.6	7.9
T2T-ViT-19	**81.2**	39.0	8.0
T2T-ViT$_t$-19	**81.4**	39.0	8.4
ResNet152 [15]	78.3	60.2	11.6
ResNet152*	80.8	60.2	11.6
T2T-ViT-24	**81.8**	63.9	12.6
T2T-ViT$_t$-24	**82.2**	64.1	13.2

Table 4. Comparison between our T2T-ViT-7,10 with MobileNets.

Models	Top1-Acc (%)	Params (M)	MACs (G)
MobileNetV1 1.0x*	70.8	4.2	0.6
T2T-ViT-7	**71.2**	4.2	0.9
MobileNetV2 1.0x*	72.8	3.5	0.3
MobileNetV2 1.4x*	75.6	6.9	0.6
T2T-ViT-10	**74.1**	5.8	1.2
T2T-ViT-12	**75.5**	6.8	1.4

tion [26, 32] in the current T2T-ViT-7, T2T-ViT-10 and T2T-ViT-12, and we only reduce the model size by reducing the hidden dimension, MLP ratio and depth of layers, which indicates that T2T-ViT also is a promising lite model. Overall, the experimental results show that our T2T-ViT can achieve superior performance when it has mid-size as ResNets, and also obtains reasonable results when it has a small model size as MobileNets.

4.2. From CNN to ViT

To explore an efficient backbone for our T2T-ViT, we conduct experiments to apply DenseNet structure (dense connection), Wide-ResNet structure (wide or narrow channel dimensions), SE block (channel attention), ResNeXt structure (more heads in multihead attention), and Ghost operation from CNN to ViT. The details of the architecture design are given in Appendix A.1. From the experimental results on "CNN to ViT" in Table 5, we can find that both SE (ViT-SE) and Deep-Narrow structure (ViT-DN) benefit the ViT but the most effective structure is deep-narrow structure, which decreases the model size and MACs nearly 2x but bring 0.9% improvement on the baseline model ViT-S/16.

To further improve the proposed T2T-ViT, we also apply these CNN-based structures from CNN to our T2T-ViT. To study the effects of different designs and architectures, we conduct experiments on ImageNet on both the CNN and ViT&T2T-VIT with the same train scheme. We take ResNet50 as the baseline of CNN, ViT-S/16 as the baseline of ViT, and T2T-ViT-14 as the baseline of T2T-ViT. All experimental results are given in Table 5, and corresponding results on CNN and ViT&T2T-ViT have same colors. We can summarize the effects of each CNN-based structures as:

Deep-narrow structure benefit ViT: The model ViT-DN (Deep-Narrow) and ViT-SW (Shallow-Wide) in Table 5 are two opposite design in channel dimensions and layer depth, where ViT-DN has 384 hidden dimensions and 16 layers and ViT-SW has 1024 hidden dimensions and 4 layers. Compared with the baseline model ViT-S/16 with 768 hidden dimensions and 8 layers, shallow-wide model ViT-SW has 8.2% decrease in performance while ViT-DN with only half of model size and MACs achieve 0.9% increase. These results validate our hypothesis that vanilla ViT with shallow-wide structure is redundant in channel dimensions and limited feature richness with shallow layers.

Dense connection hurts the performance of both ViT and T2T-ViT: Compared with the ResNet50, DenseNet201 has smaller parameters and comparable MACs, while it is higher than the original ResNet50. However, the dense connection can hurt the performance of ViT-Dense and T2T-ViT-Dense (the dark blue rows in Tab 5).

SE block improves both ViT and T2T-ViT: From the red rows in Tab 5, we can find that the SENets, ViT-SE and T2T-ViT-SE are higher than the corresponding baseline. The SE module can improve the performance on both CNN and ViT, which means that applying attention to channels benefits both CNN and ViT models.

ResNeXt structure has few effects on ViT and T2T-ViT: ResNeXts adopt multi-head on ResNets, while Transformers are also multi-head attention structure. When we adopt more heads like 32, we can find that it has few effects on performance (the red rows in Tab 5). However, adopting a large number of heads makes the GPU memory large, so it is not needed to adopt more head numbers in ViT and T2T-ViT.

Table 5. Transfer some common designs in CNN to ViT&T2T-ViT, including: DenseNet, Wide-ResNet, SE module, ResNeXt, Ghost operation. The same color means the correspond transferring. All models are trained from scratch on ImageNet. (* means we reproduce the model with our training scheme for fair comparisons.)

Model Type	Models	Top1-Acc (%)	Params (M)	MACs (G)	Depth	Hidden_dim
Traditional CNN	AlexNet [22]	56.6	61.1	0.77	-	-
	VGG11 [33]	69.1	132.8	7.7	11	-
	Inception v3 [35]	77.4	27.2	5.7	-	-
Skip-connection CNN	ResNet50 [15]	76.2	25.6	4.3	50	-
	ResNet50* (Baseline)	79.1	25.6	4.3	50	-
	Wide-ResNet18x1.5*	78.0 (-1.1)	26.0	4.1	18	-
	DenseNet201*	77.5 (-1.6)	20.1	4.4	201	-
	SENet50*	80.3 (+1.2)	28.1	4.9	50	-
	ResNeXt50*	79.9 (+0.8)	25.0	4.3	50	-
	ResNet50-Ghost*	76.2 (-2.9)	19.9	3.2	50	-
CNN to ViT	ViT-S/16 (Baseline)	78.1	48.6	10.1	8	768
	ViT-DN	79.0 (+0.9)	24.5	5.5	16	384
	ViT-SW	69.9 (-8.2)	47.9	9.9	4	1024
	ViT-Dense	76.8 (-1.3)	46.7	9.7	19	128-736
	ViT-SE	**78.4** (+0.3)	49.2	10.2	8	768
	ViT-ResNeXt	78.0 (-0.1)	48.6	10.1	8	768
	ViT-Ghost	73.7 (-4.4)	32.1	6.9	8	768
CNN to T2T-ViT	T2T-ViT-14 (Baseline)	80.6	21.4	4.8	14	384
	T2T-ViT-Wide	77.7 (-2.9)	25.0	5.4	14	768
	T2T-ViT-Dense	79.1 (-1.5)	23.6	5.1	19	128-584
	T2T-ViT-SE	**80.7** (+0.1)	21.7	4.8	14	384
	T2T-ViT-ResNeXt	80.6 (+0.0)	21.5	4.5	14	384
	T2T-ViT-Ghost	79.5 (-1.1)	16.2	3.7	14	384

Ghost can further compress model and reduce MACs of T2T-ViT: Compare the three experiment results of Ghost operation (the magenta row in Tab 5), the accuracy decrease 2.9% in ResNet50, 1.1% on T2T-ViT, and 4.4% on ViT. So the Ghost operation can further reduce the parameters and MACs of T2T-ViT with smaller performance degradation than ResNet. But for the original T2T, it would cause more decrease than ResNet.

Besides, for all the five structures, the T2T-ViT performs better than VIT, which further validates the superior of our proposed T2T-ViT. And we also wish this study of transferring CNN structure to ViT can motivate the network design of Transformers in vision tasks.

4.3. Ablation study

To further identify the effects of T2T module and deep-narrow structure, we conduct an ablation study on the proposed T2T-ViT.

T2T module To verify the effects of the proposed T2T module, we compare the experimental results on three different models: T2T-ViT-14, T2T-ViT-14$_{wo\ T2T}$, and T2T-ViT$_t$-14, where T2T-ViT-14$_{wo\ T2T}$ has the same T2T-ViT backbone but without T2T module. We can find that with similar model size and MACs, the T2T module can improve the model performance by 1.7%-1.8% on ImageNet.

Table 6. Ablation study results on T2T module, Deep-Narrow(DN) structure.

Ablation type	Models	Top1-Acc (%)	Params (M)	MACs (G)
T2T module	T2T-ViT-14$_{wo\ T2T}$	78.9	21.1	4.3
	T2T-ViT-14	80.6 (+1.7)	21.4	4.8
	T2T-ViT$_t$-14	80.7 (+1.8)	21.5	5.2
	T2T-ViT$_c$-14	79.8 (+0.9)	21.3	4.4
DN Structure	T2T-ViT-14	80.6	21.4	4.8
	T2T-ViT-d768-4	78.0 (-2.6)	25.0	5.4

As the soft split in T2T module is similar to convolution operation without convolution filters, we also replace the T2T module with 3 convolution layers with kernel size (7,3,3), stride size (4,2,2) respectively. Such a model with

convolution layers as T2T module is T2T-ViT$_c$-14. From the Tab 6, we can find that the T2T-ViT$_c$-14 is worse than T2T-ViT-14 and T2T-ViT$_t$-14 by 0.5%-1.0% on ImageNet. We also noted that the T2T-ViT$_c$-14 is still higher than T2T-ViT-14$_{wo\ T2T}$, as the convolution layers in the early stage can also model the structure information. But our designed T2T module is better than the convolution layers as the T2T module can both model the global relations and structure information of images.

Deep-narrow structure The deep-narrow structure has fewer hidden dimensions but more layers. We adopt such a structure rather than the shallow-wide structure in the original ViT. We compare the T2T-ViT-14 and T2T-ViT-d768-4 to verify the effects of deep-narrow structure, where T2T-ViT-d768-4 is a shallow-wide structure with the hidden dimension of 768 and 4 layers and it has a similar model size and MACs with T2T-ViT-14. From Table 6, we can find that after change our deep-narrow to shallow-wide structure, the T2T-ViT-d768-4 has 2.9% decrease in top-1 accuracy, validating that deep-narrow structure is crucial for T2T-ViT.

5. Conclusion

In this work, we propose a new T2T-ViT model that can be trained from scratch on ImageNet and achieve comparable or even better performance than CNNs. T2T-ViT effectively models the structure information of images and enhances feature richness, which overcomes the limitations of ViT. It introduces the novel tokens-to-token (T2T) process to progressively tokenize images to tokens and structurally aggregate tokens. On the other hand, we investigate various architecture design choices from CNNs for improving T2T-ViT performance, and we empirically find that the deep-narrow architecture can perform better than shallow-wide structure. Our T2T-ViT achieves superior performance than ResNets and comparable performance with MobileNetV1 with a similar model size when trained from scratch on ImageNet. It paves the way for further developing transformer-based models for vision tasks.

References

[1] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le. Attention augmented convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3286–3295, 2019.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

[4] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020.

[5] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020.

[6] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng. Aˆ 2-nets: Double attention networks. In *Advances in neural information processing systems*, pages 352–361, 2018.

[7] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[8] Z. Dai, B. Cai, Y. Lin, and J. Chen. Up-detr: Unsupervised pre-training for object detection with transformers. *arXiv preprint arXiv:2011.09094*, 2020.

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

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

[11] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

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

[13] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[14] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1580–1589, 2020.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[18] H. Hu, Z. Zhang, Z. Xie, and S. Lin. Local relation networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3464–3473, 2019.

[19] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *Advances in neural information processing systems*, 31:9401–9411, 2018.

[20] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[24] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[25] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[27] N. Parmar, A. Vaswani, J. Uszkoreit, Ł. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[29] M. E. Peters, M. Neumann, R. L. Logan IV, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*, 2019.

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

[31] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] Z. Sun, S. Cao, Y. Yang, and K. Kitani. Rethinking transformer-based set prediction for object detection. *arXiv preprint arXiv:2011.10881*, 2020.

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

[36] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.

[38] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017.

[39] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[40] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020.

[41] R. Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[42] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[43] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, M. Tomizuka, K. Keutzer, and P. Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020.

[44] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[45] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5791–5800, 2020.

[46] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

[47] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[48] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020.

[49] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. 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.

[50] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[51] Y. Zeng, J. Fu, and H. Chao. Learning joint spatial-temporal transformations for video inpainting. In *European Conference on Computer Vision*, pages 528–543. Springer, 2020.

[52] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[53] H. Zhao, J. Jia, and V. Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10076–10085, 2020.

[54] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun. Point transformer. *arXiv preprint arXiv:2012.09164*, 2020.

[55] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018.

[56] M. Zheng, P. Gao, X. Wang, H. Li, and H. Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020.

[57] D. Zhou, X. Jin, Q. Hou, K. Wang, J. Yang, and J. Feng. Neural epitome search for architecture-agnostic network compression. In *International Conference on Learning Representations*, 2019.

[58] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018.

[59] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A. Appendix

A.1. Details of transfer from CNN structure to ViT

We attempt to transfer the dense connection as DenseNets, wide or narrow channel dimensions as Wide-ResNet, channel attention as SE module, more heads as ResNeXt structure, and Ghost operation to ViT to validate the effects of CNN-based structure on ViT. ON the other hand, we also attempt to transfer these structure designs to our T2T-ViT. To simplify the designs, we only take ViT-S-8 and T2T-ViT-14 as examples and transfer the following designs strategies:

From ResNet-Wide to ViT&T2T-ViT Wide-ResNets are designed by decreasing layer depth and increasing width of ResNets, and such a design can improve model performance [50]. We thus design a ViT with deep-narrow backbone (ViT-DN) and Shallow-Wide backbone (ViT-SW), where ViT-DN has hidden dimensions 384 and 16 transformer layers and ViT-SW has hidden dimension 1024 and 4 layers.

From DenseNet to ViT&T2T-ViT Densely Connected Convolution Networks (DenseNets) [21] connect each convolutional layer with every other layer rather than only create short paths from early to later layer like ResNets, which can improve the information flow between layers in the network. As ViT use skip-connection as ResNets, a natural transfer is to apply the dense connection to ViT&T2T-ViT as ViT-Dense&T2T-ViT-Dense. Similar to DenseNet, if each block in ViT-Dense&T2T-ViT-Dense has L Transformer layers, there are $L(L+1)/2$ connections in this block and l-th layer has l input from the early layers. Specifically, we set the hidden dimension of the first layer in ViT-Dense&T2T-ViT-Dense as 128 and it increases 64 channels (''growth rate" as DenseNets) in each layer after concatenating with the early layers channels. The ViT-Dense&T2T-ViT-Dense has 4 blocks as [4,6,6,4] and transition layers can compress the channels after each block to improve model compactness. Such a design can make the ViT-Dense&T2T-ViT-Dense are deeper than ViT&T2T-ViT with a similar number of parameters and MACs.

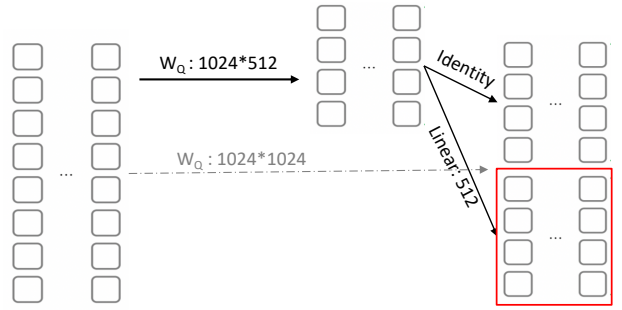
From SENet to ViT&T2T-ViT Squeeze-an-Excitation (SE) Networks [20] apply the SE module in channel dimension, which can learn the inter-dependency between channels and bring improvement in performance on ResNets. The SE module is extremely simple and useful in CNN, so we transfer such modules to ViT&T2T-ViT. In ResNets, the SE module is applied after each bottleneck structure, thus we add the SE module in the channels after multi-head attention computation, and create ViT-SE&T2T-ViT-SE. The SE module in ViT&T2T-ViT can not only simply learn the inter-dependency between channels but also learn the local attention on the spatial dimension, as in the patch embedding, the spatial information in each patch will be squeezed to channel dimension.

From ResNeXt to ViT&T2T-ViT ResNeXt is constructed by splitting the channels with multiple paths and then concatenate a set of transformations on each split path, which is similar to the split-transform-merge strategy in Inception models [44]. In each split path, only 4 channels are transformed and then concatenated with other paths. Such a strategy is the same as the multi-heads attention design by splitting the channel dimensions into multiple heads. The size of the set of transformations in ResNeXt is exactly the number of heads, which is always 32 in ResNeXt. So for ViT&T2T-ViT, we can simply add the number of heads from 8 to 32 as ViT-ResNeXt&T2T-ViT-ResNeXt to validate the effects of such aggregated transformations in ViT and T2T-ViT.

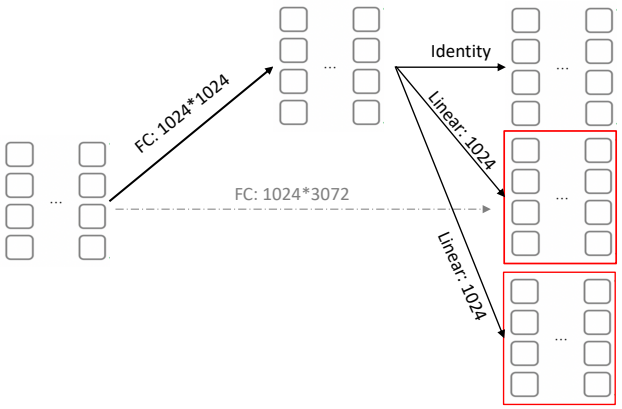
From Ghost-CNN to ViT&T2T-ViT GhostNets [14, 57] propose Ghost operation to generate more feature with cheap operations, which is a simple but effective method as the feature maps in ResNets always has redundant channels. The ViT models have more redundant channels and invalid channels than ResNets (Fig. 2. So we can transfer

the ghost operations from CNN to ViT by applying such operations on both attention blocks and feed-forward blocks. As shown in Fig. 5, the ghost operation can be simply applied to ViT structure. Different with T2T-ViT-Dense and T2T-ViT-SE with comparable model size with T2T-ViT-14, the ghost operation can reduce the number of parameters and MACs of models, so the T2T-ViT-Ghost only has 80% parameters and of T2T-ViT-14.

For fair comparisons, the above variants of T2T-ViT are designed with comparable size with T2T-ViT-14 and ResNet50 except for T2T-ViT-Ghost. It is noted that our design of each transferring is not the only choice, and we wish the transfers can motivate the model designs of Transformers in vision tasks.



(a) Ghost operation on attention block of ViT&T2T-ViT



(b) Ghost operation on feed-forward block of ViT&T2T-ViT

Figure 5. Ghost operation to reduce the hidden dimensions: (a) on the attention block (take the Query matrix W_Q as example). (b) on the feed-forward module. The dash line is the original operation and the solid lines are our ghost operation.