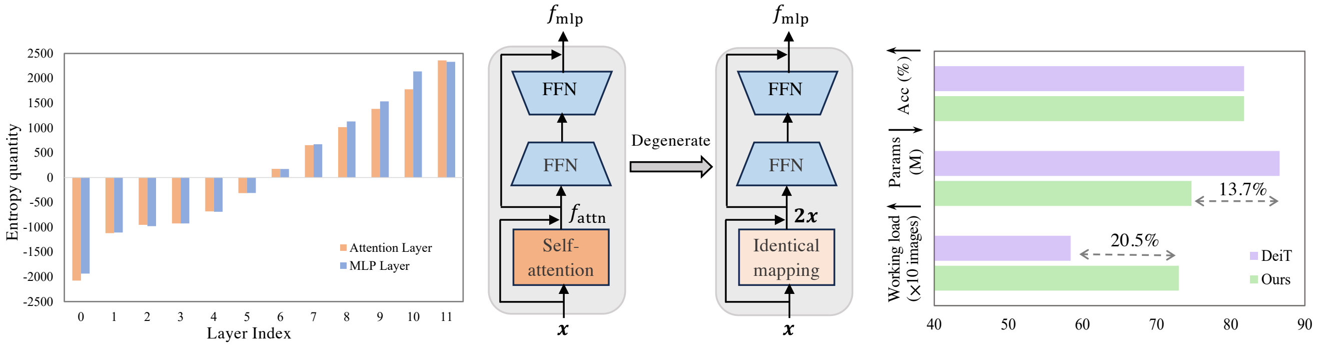
MLP Can Be A Good Transformer Learner

Sihao Lin[1]* Pumeng Lyu[2]* Dongrui Liu[2,3] Tao Tang[4] Xiaodan Liang[4,5,7]
Andy Song[1] Xiaojun Chang[6,7]†
[1]RMIT University [2]Shanghai AI Laboratory [3]Shanghai Jiao Tong University [4]Shenzhen Campus of Sun Yat-sen University
[5]DarkMatter AI Research [6]University of Technology Sydney [7]MBZUAI
{linsihao6,trent.tangtao,xdliang328}@gmail.com {lvpumeng,liudongrui}@pjlab.org.cn
andy.song@rmit.edu.au xiaojun.chang@uts.edu.au



(a) Entropy distribution (b) Integrating attention layer into MLP layer (c) Improving efficiency without performance drop

Figure 1. **Pruning the attention layer from the perspective of entropy.** (a) We use entropy to illustrate the information amount carried out by the attention layers and MLP layers (*i.e.* two FFN layers) in each transformer block of DeiT-B [33]. We observe that the entropy quantity of the bottom blocks is lower than that of the top blocks. We identify a pattern that, the attention layer with low entropy is accompanied by the MLP layers with the entropy quantity at the same level. (b) In the bottom blocks, MLP layers can elicit the information as much as that of the attention layers. On the other hand, they are under-exploited given the low entropy quantity compared to those MLP layers in the top blocks. We thus propose to integrate the uninformative attention layer into its subsequent MLP layer through proper optimization. (c) As a result, our method can reduce 13.7% parameters of DeiT-B and improve 20.5% working load in the same memory budget without performance degradation.

Abstract

Self-attention mechanism is the key of the Transformer but often criticized for its computation demands. Previous token pruning works motivate their methods from the view of computation redundancy but still need to load the full network and require same memory costs. This paper introduces a novel strategy that simplifies vision transformers and reduces computational load through the selective removal of non-essential attention layers, guided by entropy considerations. We identify that regarding the attention layer in bottom blocks, their subsequent MLP layers, i.e. two feed-forward layers, can elicit the same entropy quantity. Meanwhile, the accompanied MLPs are under-exploited since they exhibit smaller feature entropy compared to those MLPs in the top blocks. Therefore, we propose to integrate the uninformative attention layers into their subsequent counterparts by degenerating them into identical mapping, yielding only MLP in certain transformer blocks. Experimental results on ImageNet-1k show that the proposed method can remove 40% attention layer of DeiT-B, improving throughput and memory bound without performance compromise.

1. Introduction

Vision Transformer [9, 33] is becoming dominant for vision tasks [14, 15, 36]. It is believed that self-attention mechanism is the key component for its success, which models the dense similarity between two entries. Nonetheless, researchers have found that the attention layer is redundant [4, 23], *e.g.* attention maps across different heads [20] or stages [3] might be similar to each other. To this end, a broad array of works [5, 6, 10, 17, 25, 27, 32, 35, 37] propose to prune/merge the redundant tokens to reduce computation redundancy. Nonetheless, these methods still need to

*Equal contribution.
†Corresponding author.

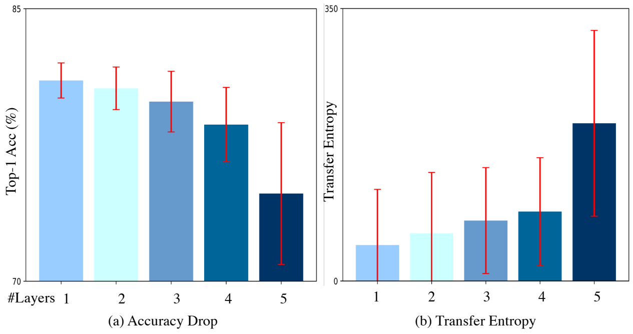
load the full network and consume the same memory costs as the original model.

To this end, this work aims to directly remove those uninformative attention layers to push the memory bound. We investigate this problem from the perspective of entropy, which measures the information quantity of a network. As a motivator, we visualize the entropy distribution of the attention layers, together with their subsequent MLP layers, of DeiT-B [33] as illustrated in Fig. 1 (a). Specifically, one can observe that in the bottom blocks, the entropy quantity of the attention layer is lower than that of the top blocks. In particular, we identify a pattern that, the attention layer with low entropy is accompanied by the MLP layers with the entropy quantity at the same level. Our finding brings a novel perspective to the inefficient attention layers. On one hand, since MLP layers in bottom blocks contain the entropy as same as the attention layers, they may elicit the same information. On the other hand, these MLPs are under-exploited and thus can be optimized to be as expressive as those MLPs in the top blocks. Therefore, a natural question is raised: Can we *integrate the uninformative attention layer into its subsequent MLPs*?

More concretely, in the context of entropy, we question whether the information carried out by the attention layer can be transplanted into the corresponding MLP layer, through proper optimization. As shown in Fig. 1 (b), the output feature of the attention layer is the input of the subsequent MLP layer. Given this fact, we propose a simple dilution learning technique that gradually degenerates the attention layer into identical mapping. Eventually, the resulting identical mapping together with the residual connection can be integrated into the subsequent MLPs, yielding only MLPs in certain Transformer blocks.

Another question is which attention layers should be selected for consequent manipulation. Probably it is natural to perform on the consecutive bottom blocks since they carry out less information. However, such a strategy neglects the potential interaction among different layers. For instance, we randomly mask N attention layers (N is from 1 to 5) of a pre-trained DeiT-B and repeat this process over 20 times. We therefore get the means (bars) and variances (red lines) of model performances (Fig. 2 (a)) and the corresponding transfer entropies (Fig. 2 (b)) when removing 1~5 layers. Model performance drops as transfer entropy increases in both mean and variance, indicating the importance of interaction among multiple layers.

To this end, we propose the E**n**tr**o**py-based **S**election Strat**e**gy, dubbed as **NOSE**, to identify the combination of different attention layers that cause minimum impact on the consequent performance. Specifically, we use transfer entropy to approximate the interaction between an ordered array of attention layers and the final output layer. Fig. 2 (b) shows that the transfer entropy has a great variation across different combinations.



(a) Accuracy Drop

(b) Transfer Entropy

Figure 2. **Interaction of multiple layers.** Both figures have the same x-axis (#Layer). We use the idea of transfer entropy to measure the interaction on multiple layers. Here, we randomly mask 1~5 attention layers of a pre-trained DeiT-B. We record the means (bars) and variances (red lines) of model performances in (a) and the corresponding transfer entropies in (b). It is clear that model performance drops as transfer entropy increases in both mean and variance. As a motivator, we aim to remove attention layers with fewer interactions (*i.e*. transfer entropy).

We validate the proposed method on three benchmarks ImageNet-1k [8], CIFAR-100 [16], and ADE20k [43]. The experimental result evidences that our framework can effectively discard uninformative attention layers and learn the robust feature without performance compromise. For instance, our method removes 40% attention layers of DeiT-B without performance drop in ImageNet-1k. To summarize, we claim the following contribution in this work:

- We propose a novel framework that transplants the knowledge of non-essential attention layers into their subsequent MLP layers.
- From the perspective of transfer entropy, we propose the Entropy-based Selection Strategy to identify the correlation between an ordered array of attention layers and the final output layer, which causes less or even no degradation to the network performance.
- We propose a simple yet effective dilution learning technique that degenerates attention layers into identical mapping layers. Eventually, the identical mapping together with the residual connection are taken as the input of the MLP layer, yielding only MLP in certain blocks.

2. Related Work

It is acknowledged that the concept of Transformer is proposed by Vaswani *et al*. [34] for natural language process. Lately, Dosovitskiy *et al*. [9] introduce the vision transformer for image recognition. Since its emergence, the *self-attention* efficiency regarding the quadratic complexity has engaged considerable interest from industrial and research community. Existing methods motivate this problem from two perspectives: token aggregation and token pruning.

Token aggregation. There are works that approximate the full attention with partial attention by leveraging the locality. SwimT [21] and FocalT [39] use local windows to extract the feature from neighbor tokens, resulting in the feature map of smaller resolution. MetaFormer [41] and PSViT [6] apply simple pooling operation among local tokens to reduce the length of the token array. Recently, ToMe [5] proposes to gradually combine two similar tokens by bipartite soft matching without needing to train.

Token pruning. Some work aim to dynamically prune the uninformative token during training. DynamicViT [27] uses a prediction module to measure the importance score for each token and progressively prunes the redundant tokens stage by stage. Rather, Patch Slimming [32] performs token pruning in a top-down manner. It identifies the valuable tokens in the last layer and in turn requires the previous layer to discriminate these tokens from the redundant one. EViT [17] simply identifies the attentive tokens given the similarity with the classification token. Then the inattentive tokens are fused into a supplement token. Evo-ViT [37] proposes the Fast-slow Token Evolution where valuable tokens and uninformative tokens are separately updated using different strategies. Recently, TPS [35] proposes the Join Pruning and Squeezing module that first identifies the reserved tokens and pruned tokens, which are fused into the reserved tokens according to their matching score.

Yet, existing token pruning methods discussed above require the same memory cost as the original model since they are compelled to the full network architecture and even additional modules. Our method can push the limit of memory bound since we combine attention layers with subsequent MLP layers and remove self-attention architectures.

3. Methods

We first briefly introduce the preliminaries of vision transformer in Sec. 3.1. We use entropy to quantify the information carried out by the attention layer (Sec. 3.2) and propose the selection strategy to identify which layers are supposed to be removed (Sec. 3.3). In Sec. 3.4, we present a simple network *dilution* recipe that gradually degenerates the attention layer into identical mapping.

3.1. Preliminary

Vision transformer (ViT) is first introduced by Dosovitskiy *et al.* [9] for image classification [8]. A ViT is composed of a patch embedding layer $\mathcal{P}$ and a stack of transformer blocks $\mathcal{A}$, following a task-specific head $\mathcal{G}$.

$$\begin{aligned} \text{ViT} &= \mathcal{G} \circ \mathcal{A} \circ \mathcal{P}, \\ \mathcal{A} &= A_l \circ \cdots \circ A_2 \circ A_1. \end{aligned} \tag{1}$$

Given a predefined patch size $h \times w$, the patch embedding layer encodes an image $I \in \mathbb{R}^{H \times W \times 3}$ into $P = H/h \times W/w$ patch tokens with dimension d. It is then prepended the classification token to form the image tokens, which are fed into the transformer blocks. Typically, a transformer block includes a self-attention layer Attn and a subsequent MLP layer (*i.e.* two feed-forward layers). Consider a transformer block in $\mathcal{A}$:

$$\begin{aligned} f_{\text{attn}} &= \text{Attn}(\boldsymbol{x}) + \boldsymbol{x}, \\ \text{Attn}(\boldsymbol{x}) &= \text{softmax}(\frac{Q \cdot K^{\top}}{\sqrt{d}}) \cdot V, \\ Q = W_Q(\boldsymbol{x}),\ K &= W_K(\boldsymbol{x}),\ V = W_V(\boldsymbol{x}). \end{aligned} \tag{2}$$

Here $\boldsymbol{x} = \{x_i\} \in \mathbb{R}^d$ is the input tokens for classification token $i = 0$ and patch tokens $1 \le i \le P$. W_Q, W_K and W_V are the linear projections that projects $\boldsymbol{x}$ to query Q, key K and value V of size $(P+1) \times d$. By convention, a residual connection is applied to the output of Attn, and the Layer Norm (LN) result [1] is fed into MLP, generating the output of this block:

$$f_{\text{mlp}} = \text{MLP}(\text{LN}(f_{\text{attn}})) + f_{\text{attn}}. \tag{3}$$

3.2. Entropy Quantification

By definition, entropy [11] can be used to measure the information quantity of a network. Accordingly, one can calculate the entropy of a certain layer given the probability of its feature:

$$H(F) = -\int p(f) \log p(f)\, df, f \in F. \tag{4}$$

Nonetheless, it is difficult to directly measure the probability distribution of a feature map: $p(f), f \in F$. Following [30, 31], we use the Gaussian distribution as the probability distribution of the intermediate feature in a layer. Therefore, the entropy of a certain layer is approximated as the mathematical expectation of $F \sim \mathcal{N}(\mu, \sigma^2)$:

$$\begin{aligned} H(F) &= -\mathbb{E}[\log \mathcal{N}(\mu, \sigma^2)] \\ &= -\mathbb{E}[\log[(2\pi\sigma^2)^{-1/2} \exp(-\frac{1}{2\sigma^2}(f-\mu)^2)]] \\ &= \log(\sigma) + \frac{1}{2}\log(2\pi) + \frac{1}{2}, \end{aligned} \tag{5}$$

where σ is the standard deviation of the feature set $f \in F$. Typically, a batch of images is passed into a vision transformer to obtain the feature set F of the attention layer and MLP layer (Eq. (2) & (3)), respectively. $H(F)$ is proportional to $\log(\sigma)$ plus two additional constants. Without loss of generality, the two constants are neglected in the following analysis. In practice, we apply Eq. (5) to each channel of the intermediate feature. Then, without considering constant terms, the entropy of each layer is proportional to the summation of logarithm of standard deviation of each feature channel:

$$H(F) \propto H_\sigma(F) = \sum_j \log[\phi(F^j)]. \tag{6}$$

Thus, $H_{\sigma}(F)$ is the value proportional to the entropy of a layer, either attention or MLP layer. $\phi(F^j)$ calculates the standard deviation of j^{th} channel of the feature set F.

3.3. Interaction among Multiple Attention Layers

The above discussion formulates the entropy of a single layer. Our goal is to remove an ordered array of attention layers that are less significant to the original architecture. As shown in Fig. 1 (a), it is plausible to remove the attention layers in the bottom blocks with relatively low entropy. However, such a strategy largely neglects the potential interaction across different layers, which is proved to be important in Fig. 2.

As a remedy, we resort to the *transfer entropy* (TE) [24, 29, 42] that measures the information amount of directed transfer between two layers. Given a target layer, transfer entropy compares the difference in entropy quantity in the presence and absence of the source layer.

$$TE = H(F_{\text{target}}) - H(F_{\text{target}}|\mathcal{A}\backslash\{\text{Attn}_{\text{source}}\}). \tag{7}$$

Here $H(F_{\text{target}})$ is the original entropy of the target layer defined in Eq. (6). We compute the entropy $H(F_{\text{target}}|\mathcal{A}\backslash\{\text{Attn}_{\text{source}}\})$ in the condition that source attention layer $\text{Attn}_{\text{source}}$ is masked out, *i.e.* set to identical mapping. Hence, the numeric value of TE can reflect the significance of the source layer over the target layer, measuring their correlation. We aim to identify the combination of multiple attention layers that have the minimum correlation with the final output layer of the network.

Therefore, we propose the **E**ntr**o**py-based **S**election Strat**e**gy, dubbed as NOSE, to select the attention layers with minimum transfer entropy to the final output layer. The proposed NOSE will measure the transfer entropy between the attention layers and the final output layer iteratively. At each round, NOSE traverses the candidate attention layers $\mathcal{C}$ and figures out the layer has a minimum transfer entropy using greedy search. This layer is appended to the state set $\mathcal{S}$, which will be detached from the candidate set and won't participate in the next loop. We then repeat the procedure by taking into account the previous state till the combination reaches a sufficient amount.

3.4. Integrating Attention Layer into MLP

Given the fact that MLP layer would take as input the output of the attention layer, our method degenerates the attention layers into identical mapping. Hence, the identical mapping and the associated residual connection, can be integrated into the subsequent MLP layer, yielding only MLP in the transformer block.

Diluting the attention output. Following [13, 22], an attention layer is decoupled to the original architecture and a sparse mask. The Eq. (2) is reformulated as:

$$f_{\text{attn}} = M \odot \text{Attn}(\boldsymbol{x}) + \boldsymbol{x},\ M \in \mathbb{R}^{(P+1)\times d}, \tag{8}$$

where $\odot$ is element-wise multiplication. The sparse mask M is usually subject to some constraints, *e.g.* L_0 norm [13, 22], and is used to regularize the sparsity of the attention output. In our case, M is initialized as 1 and is manually decayed till 0 along the training process. We showcase in the experiments that the implementation of M is robust to different choices. Once the sparse mask is decayed to 0, the output f_{attn} of the attention layer becomes the residual connection.

Algorithm 1 Training Procedure of Our Method

```
Input  a ViT, state set $\mathcal{S}$, candidate set $\mathcal{C}$, amount of selecting layers $N$, training set $[\mathcal{I},\mathcal{Y}]$, decay function $D$, sparse mask $M$, training iterations $T$, loss function $\mathcal{L}$.
Output  simplified ViT with $N$ attention layer get removed.

$\mathcal{S} \leftarrow \emptyset$, $\mathcal{C} \leftarrow \{\text{Attn}_1, \text{Attn}_2, ..., \text{Attn}_l\}$, $M \leftarrow 1$
Identify the combination of attention layers:
for $n = 0, 1, 2, ..., N$ do
    Traverse the attention layer in candidate set $\mathcal{C}$:
    $\arg\min_i(TE(\mathcal{S} \cup \{\text{Attn}_i\}, \mathcal{G}))$    ◃ greedy search
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\text{Attn}_i\}$, $\mathcal{C} \leftarrow \mathcal{C}\backslash\{\text{Attn}_i\}$ ◃ update state
end for
Diluting the attention layers:
for $t = 0, 1, 2, ..., T$ do
    Fit a batch of data $[I,Y]$ sampled from $[\mathcal{I},\mathcal{Y}]$:
    minimize $\mathcal{L}(\text{ViT}(I), Y)$ ◃ apply Eq. (9) on $\mathcal{S}$
    $M \leftarrow D(M)$    ◃ decay sparse mask
end for
```

Feature compensation. As the sparse mask is decayed, it continuously vanishes the gradient of the attention layer. Hence, the backward gradient of the degenerated output will be smaller than the original one, which incurs training instability. To this end, we propose the feature compensation, which adaptively compensates the gradient loss brought by sparse mask:

$$\begin{aligned} f_{\text{attn}} &= M \odot \text{Attn}(\boldsymbol{x}) + (1 - M) \odot \boldsymbol{x} + \boldsymbol{x} \\ &= M \odot \text{Attn}(\boldsymbol{x}) + (2 - M) \odot \boldsymbol{x}. \end{aligned} \tag{9}$$

Here, we introduce a new term $(1 - M) \odot \boldsymbol{x}$ compared to Eq. (8). It will correspondingly compensate the loss of attention output $\text{Attn}(\boldsymbol{x})$ following the pace of M. Eventually, the attention layer is degenerated to an identical mapping, resulting in the output $2\boldsymbol{x}$. As a result, the attention layer is integrated into the subsequent MLP layer and is no longer required in the inference stage. We summarize our pipeline in Algorithm 1.

4. Experiment

4.1. Baseline Setting

Benchmark. CIFAR-100 [16] is an image classification benchmark with 100 semantic categories. The training set

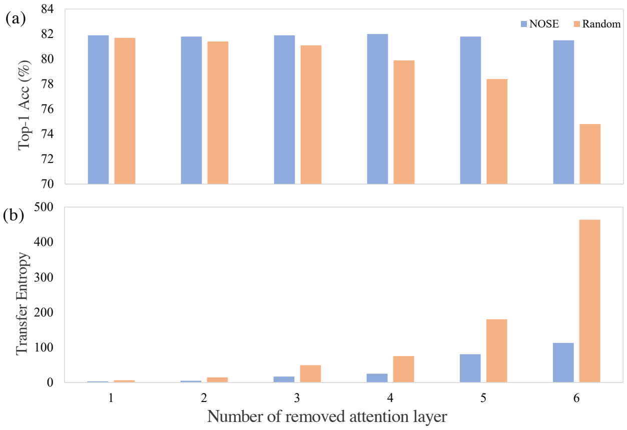


Figure 3. **NOSE *vs*. Random selection.** (a) NOSE consistently outperforms the random selection on ImageNet-1k. (b) This is because NOSE can identify the attention layers with less interaction with the final output layer, which is reflected by transfer entropy.

and validation set have 50,000 and 10,000 samples, respectively. ImageNet-1k [8] is a challenging classification dataset with 1,000 categories. It has more than 1**M** training samples and 50,000 validation samples. ADE20K [43] is a semantic segmentation dataset with 150 classes, which has 20,000 training samples and 2,000 validation samples.

Evaluation protocol. We first assess our method on ImageNet-1k. Furthermore, we verify the proposed method on ADE20k for dense classification. To evaluate the feature richness learned by our method, we perform the transfer learning on CIFAR-100 using the weights pre-trained on ImageNet-1k.

Implementation Details. Given the popularity and influence, we adopt the DeiT [33] as the implementation of the ViT, which has the standard softmax-attention layer. Throughout the paper, we perform experiments on the network architecture DeiT-B. We adopt the training recipe of DeiT on ImageNet-1k and CIFAR-100. For ImageNet-1k, we decay the sparse mask M from 1 to 0 for 300 epochs. We follow the experimental setting of TinyMIM [28] on ADE20k. We provide more details and additional experiments on other backbones in Sec. 10 of the appendix.

4.2. Main Result

Validating the entropy-guided selection strategy. We inspect the effectiveness of the proposed selection strategy NOSE. We compare the method against the random selection scheme where the same amount of attention layers are sampled randomly. Due to the high space complexity, we sample three times from the feasible combinations C_N^n.

The result on ImageNet-1k demonstrated the effectiveness of the proposed NOSE. As illustrated in Fig. 3 (b), our method proved to identify the combination of attention layers with lower transfer entropy compared to random selection. As a result, the proposed NOSE would cause less or even no degradation to the performance. Specifically, in Fig. 3 (a), when a few attention layers (*e.g.* 1 or 2) are removed, the random selection scheme would not affect the vision transformer too much. On the other hand, when increasing the amount of removed attention layers, the network would suffer from the random selection scheme since it is likely to select the inappropriate combination, declining the classification result. For instance, when randomly selecting 4 out of 12 attention layers, the network performance would deteriorate from 81.8% to 79.9%. In contrast, the proposed NOSE can properly identify the attention layers with less transfer entropy and consequently preserve the performance. We can observe that the proposed NOSE is able to remove 5 out of 12 attention layers of DeiT-B without performance compromise. Additionally, when half of the attention layers are removed, our method slightly declines the performance by 0.3% while random selection would lead to a drastic drop of 7%, demonstrating the effectiveness of the proposed NOSE. We also implement the First-N baseline which removes the first N consecutive attention layers in Tab. 9 of the appendix.

Comparison on ImageNet-1k. We compare our method with other works on ImageNet-1k. As illustrated on Tab. 1, our method showcases competitive performance compared to token pruning method. For instance, our method exceeds TPS by 0.4% and EViT by 0.5% regarding Top-1 Acc.

The issue of memory bound remains untouched in current token pruning methods [35, 37], yet it is important for compact devices with limited memory budget. Without bells and whistles, we record the maximum amount of input images during inference till the model fills up 10GB budget of the GPU memory. Since our method unloads the attention layer, it has a considerable reduction in model size. For instance, removing 40% attention layers can lead to a reduction of 13.7% regarding the network parameters, as shown in Tab. 1. Consequently, our model consumes less memory and eliminates the issue of memory bound, improving more than 20% working load compared to other methods. We test the throughput in a V100 (32GB) GPU with batch size 128. When removing 50% attention layers, our method improves the throughput by a margin of 36.5% (408 *vs*. 299). In addition, our method can combine with the unsupervised token merging method (*e.g.* [5]) seamlessly and further improve the throughput by 69.6% (507 *vs*. 299) while maintaining a competitive performance. Given these results, our method can boost both the throughput and memory bound, bringing the best of two worlds.

Transfer learning on CIFAR-100. We assess the transferable ability of the learned feature from ImageNet-1k to CIFAR-100. The experiments are conducted in two protocols: 1) *Fine-tuning:* The backbone is initialized with the pre-trained weights from ImageNet-1k and updated through end-to-end training. 2) *Linear probing:* The learned feature

Table 1. Compare to other methods on ImageNet-1k. We report the performance, throughput, and memory bound. * means using training. † means a more aggressive configuration.

Method	Top-1 (%)	Top-5 (%)	FLOPs (G)	Throughput (images/s)	Params (M)	Memory bound (images/10GB)
Deit-B [33]	81.8	95.6	17.6	299	86.6	606
DynamicViT [27]	81.3	-	11.5	464	89.5	606
Evo-ViT [37]	81.3	-	11.7	474	87.3	608
EViT [17]	81.3	95.3	11.5	458	86.6	608
TPS [35]	81.4	-	11.5	468	89.5	606
ToMe [5]	80.6	-	11.5	462	86.6	606
ToMe* [5]	81.4	-	11.5	462	86.6	606
DiffRate [7]	81.5	-	11.5	465	86.6	606
Ours(40%)	**81.8**	**95.6**	15.0	390	74.7 (↓13.7%)	730 (↑20.5%)
Ours(40%)+ToMe	81.6	95.4	11.4	478	74.7 (↓13.7%)	730 (↑20.5%)
Ours(40%)+ToMe†	81.4	95.3	**10.9**	**507**	74.7 (↓13.7%)	730 (↑20.5%)
Ours(50%)	81.5	**95.6**	14.5	408	**72.4** (↓16.4%)	**732**(↑20.8%)
Ours(50%)+ToMe	81.3	95.4	11.9	462	**72.4** (↓16.4%)	**732**(↑20.8%)

Table 2. Transfer learning on CIFAR-100.

Method	Fine-tuning	Linear probing
Deit-B [33]	**90.5**	80.6
Evo-ViT [37]	90.1	79.1
EViT [17]	90.0	80.2
TPS [35]	90.1	76.5
Ours(40%)	**90.3**	**81.3**
Ours(50%)	90.2	80.6

from ImageNet-1k is frozen and only a linear classifier (*i.e.* a full-connected layer plus a softmax layer) is trained.

As illustrated in Tab. 2, for the setting of fine-tuning, our method slightly outperforms other comparison methods and is close to the original DeiT-B. In particular, when it comes to linear probing, the proposed method can exceed other methods by a clear margin. For instance, when removing 50% attention layer, our model surpasses TPS [35] and Evo-ViT [17] by 4.1% and 1.5%, respectively. This is because token pruning methods implicitly encode the dataset bias in order to discriminate the useful tokens from the redundant ones. Thus, their learned representations exhibit less generalization ability to unseen datasets.

Result on ADE20k. We generalize the proposed framework to the task of dense prediction at ADE20k, which is rarely explored by previous work [17, 35, 37]. By default, the model is trained for 160k iterations, and the sparse mask M is decayed at every single iteration. As illustrated in Tab. 3, when training from scratch, our method reduces 40% attention layer while maintaining the performance, exhibiting its application in real-world scenarios. In addition, our method consistently outperforms other comparison methods [17, 35] in terms of the mIoU metric. A possible reason is that token pruning method explicitly drops the uninformative tokens and thus loses the global context, which is crucial for dense classification tasks. Even though TPS and EViT would re-utilize these pruned tokens, they might undermine the global dependency between tokens.

When the model is pre-trained on ImageNet-1k. All the model's performance is greatly boosted. In this case, our model with 40% attention layer removed shows a gap (~ 0.7%) compared to the baseline. We conjecture that baseline model can learn a better global dependency from ImageNet-1k. Again, our method consistently outperforms other methods.

Table 3. Results on ADE20k. * means using 2× training iterations.

Method	mIoU (%)	mAcc (%)	aAcc (%)
	From scratch		
Deit-B [33]	24.4	32.3	71.0
EViT [17]	24.0	32.2	70.5
TPS [35]	23.5	31.7	70.5
Our (40%)	**24.6**	**32.5**	**71.0**
Ours (50%)	23.9	31.9	70.6
Deit-B* [33]	26.2	33.4	72.3
EViT* [17]	25.7	**33.5**	71.9
TPS* [35]	25.1	33.1	71.6
Ours* (40%)	**26.1**	33.4	**72.3**
Ours* (50%)	25.6	33.3	71.9
	Pre-trained on ImageNet-1k		
Deit-B [33]	47.0	57.5	82.6
EViT [17]	45.5	55.9	81.9
TPS [35]	45.3	55.1	81.9
Ours (40%)	**46.2**	**56.5**	**82.2**
Ours (50%)	45.6	55.2	82.0
Deit-B* [33]	48.2	58.4	83.1
EViT* [17]	46.7	57.1	82.4
TPS* [35]	46.4	56.9	82.1
Ours* (40%)	**47.5**	**57.7**	**82.7**
Ours* (50%)	46.7	57.3	82.2

Visualization of NOSE. We visualize the trajectory of the proposed NOSE where it identifies 5 out of 12 attention layers for elimination in Fig. 4 on ImageNet-1k. Each row represents the transfer entropy of the attention layers (*i.e.* source layer) related to the final output layer of the vision transformer (*i.e.* target layer). The greedy search is applied to select the attention layer, denoted by the red dashes box, with minimum transfer entropy at each step. In the next step, the selected layer of the previous step is denoted by a gray dotted box and suspended to the state set. NOSE repeats to calculate the transfer entropy for each candidate layer by taking into account the previously selected layers.

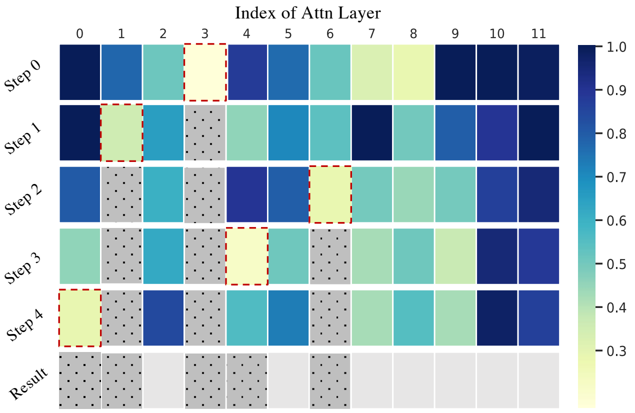


Figure 4. **Visualization of the proposed NOSE.** For each step, a row visualizes the transfer entropy, normed to [0,1], of each attention layer associated with the final output layer. We use greedy search to select the one with minimum transfer entropy, denoted by the red dashed box, *e.g.*, layer 3 is selected at step 0. The selected layer is denoted by a gray dotted box and is suspended to a state set. In the next step, NOSE repeats this procedure on the rest attention layers considering the previous state. Finally, the attention layer indexed by [0,1,3,4,6] will be integrated into their subsequent MLP layers.

It is counterfactual that, though layer 0 has the least entropy (Fig. 1 (a)), NOSE would select layer 3 at the first step. This is because NOSE would consider the interaction between layers rather than treat them separately. In the early stage, NOSE would select the layers that are not consecutive to each other. We conjecture that two consecutive attention layers would result in a complex interaction towards the final output layer. Thus, in the beginning, NOSE tends to select the interval layers. As for the layers at the top blocks, they also have a complex interaction with the output layer since they often learn the high-level semantics that are significant to the output layer. Finally, the attention layers indexed by [0,1,3,4,6] are identified as the combination that has the least interaction with the output layer.

4.3. Ablation Study and Sensitivity Analysis

Sensitivity of the sparse mask M. In Eq. (9), we introduce the sparse mask M that is used to dilute the attention layer. For ImageNet-1k, we adopt the linear decay from 1 to 0 with 300 epochs in the main experiments. For ADE20k, it is decayed linearly at each iteration. Here we investigate the robustness of M on ImageNet-1k using different step sizes and implementations, where a quantity of 40% attention layers are selected to be integrated into the MLPs. As shown in Tab. 4, when the decay epoch is set to 300, both cosine and linear function would result in the same performance, implying the robustness of our methods. Not surprisingly, reducing the decay epochs, *i.e.* increasing the decay step size, will provoke a slight performance drop. In contrast, decreasing the step size will stabilize the training process and lead to a minor improvement.

Table 4. Ablation study on sparse mask.

Function	Decay Epoch	Top-1 Acc. (%)	Top-5 Acc. (%)
Linear	200	81.4	95.6
	300	81.8	95.6
	400	82.0	95.7
Cosine	200	81.5	95.5
	300	81.8	95.6
	400	81.9	95.6

Table 5. Ablation on feature compensation.

Remove ratio	Feat. compensation	Top-1 Acc. (%)	Top-5 Acc. (%)
40%	✗	81.4	95.6
	✓	81.8	95.6
50%	✗	80.9	95.4
	✓	81.5	95.6

Ablating the feature compensation. Eq. (8) naively diminishes the attention layer by applying the sparse mask. In the end, only the residual connection will be forwarded to the subsequent MLP layers. However, since the backward gradient of the attention layer becomes smaller as long as the sparse mask is decayed, it would incur instability for training. As a remedy, we introduce the feature compensation in Eq. (9). Finally, the attention layer is degenerated to an identical mapping. We conduct the ablation study to validate the effectiveness of the proposed feature compensation. The result is shown in Tab. 5. We observe that feature compensation can consistently improve the consequent performance on ImageNet-1k. The more attention layers are removed, the more benefit it can bring.

Comparison of entropy of the MLP layers. Our work proposes that the attention layers with low entropy quantity can be integrated into their subsequent MLP layers. Consequently, the MLP layers are expected to be more expressive in order to compensate for the reduction of attention layers. We investigate this property by comparing the entropy quantity of MLP layers at the pruned index (Sec. 4.2). Specifically, our model removes the 6 out of 12 attention layers indexed by [0,1,3,4,6,9]. We measure the entropy of the corresponding MLP layers in ImageNet-1k. The result in Fig. 6 shows that the MLP layers of our method can surpass the original DeiT-B in terms of the entropy metric by a large margin, evidencing that they are more informative.

Removal rates. We investigate the removal rates on DeiT-B in Tab. 10 of the appendix.

5. A Look at Feature Expressivity

In Tab. 1 and Tab. 2, our method and the comparison methods exhibit comparable performance on ImageNet-1k as well as the CIFAR-100 in the setting of fine-tuning.

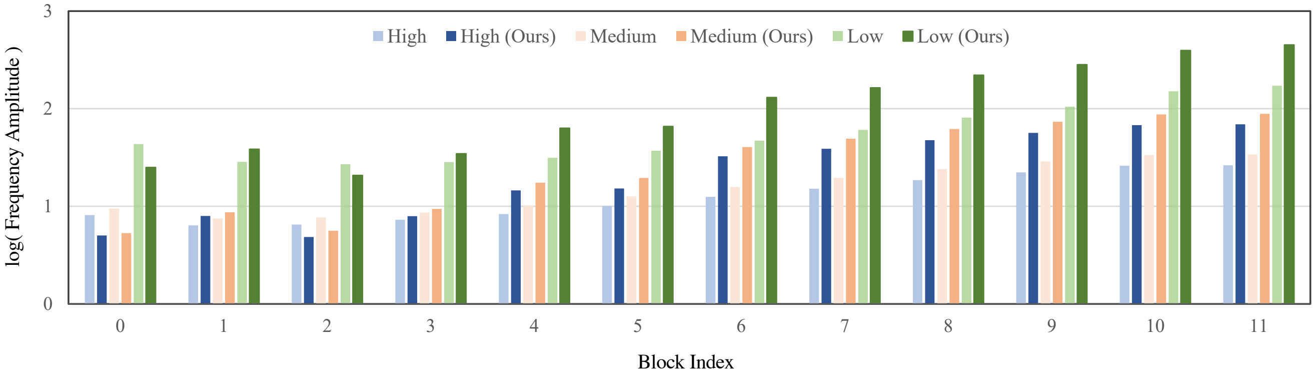


Figure 5. **Visualization of feature frequency.** We analyze feature expressivity from the frequency perspective. We apply Discrete Fourier Transform to the output feature of each block, where frequency domain is divided into low, medium, and high components. From blocks 3 to 11, our model encodes more significant high-frequency components compared to DeiT-B, implying superior feature power [2, 12].

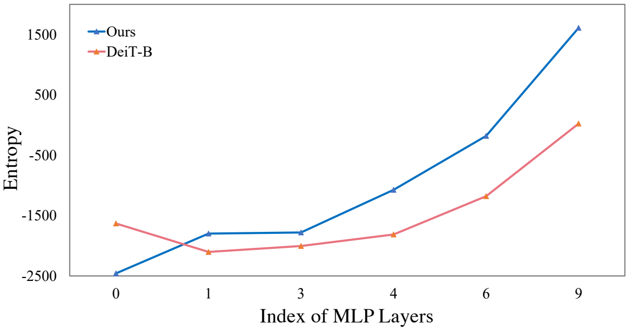


Figure 6. **Entropy of MLP layer at the pruned index.** Compared to the original DeiT-B, the MLP layers of our method can lead to a high entropy quantity at the pruned index [0,1,3,4,6,9].

Nonetheless, when it turns to linear probing on CIFAR-100, the comparison methods lag behind our method by a substantial margin. Though removing 40% attention layers, our method can even surpass the original DeiT-B by 0.7% (81.3% *vs*. 80.6%). Given this finding, we are interested in analyzing the feature space learned by our method.

To this end, we aim to analyze the representation power of the DNN from a frequency perspective [12, 18, 26, 38]. Specifically, we apply the Discrete Fourier Transform (DFT) to the output feature of each transformer block on ImageNet-1k. The frequency domain is divided into low [0,0.3π), medium [0.3π,0.7π) and high [0.7π,π] components. Fig. 5 shows that the DNN trained by the proposed method encodes more significant high-frequency components in top blocks, *i.e*., the high-frequency component's strength of the proposed method is greater than that of the DeiT-B from block 3 to 11. Although previous studies indicate that high-frequency components are more difficult and slower to be encoded by DNNs [19, 26, 38, 44], the proposed method enforces the DNN to encode more high-frequency components. Furthermore, high-frequency components are useful for generalization ability [2, 12]. Combining these previous findings and experimental observations, we may explain the effectiveness of the proposed method. *I.e.*, reducing 40% parameters and encoding more significant high-frequency components does not lead to performance degradation.

6. Conclusion

This work aims to remove the attention layers from the perspective of entropy. In particular, we propose the entropy-guided selection strategy (NOSE) to measure the interaction among multiple layers, which identifies the combination of attention layers that has the least influence on the model outputs. Then, we gradually degenerate those attention layers into identical mapping using a dilution learning technique, yielding only MLP in those transformer blocks. We demonstrate the effectiveness of our method on ImageNet-1k, ADE20k, and CIFAR-100 by comparing it to current state-of-the-art strategies. Our method reduces the network parameters as well as memory requirements. Therefore, it is able to increase the working load, which remains untouched by previous token pruning methods. Combined with the unsupervised token merging method, it strikingly boosts the throughput of the vision transformer. We also discuss the learned features of our model through DFT. The result shows that compared to the original DeiT-B, our model's feature map has a significant amplitude in the high-frequency components, implying superior feature power.

7. Acknowledgement

This work was supported in part by the National Science and Technology Major Project under Grant No. 2020AAA0109704, Guangdong Outstanding Youth Fund (Grant No. 2021B1515020061), and Australian Research Council (ARC) Discovery Program under Grant No. DP240100181. The authors thank Yuetian Weng (@MonashU) for discussions.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[2] Jiawang Bai, Li Yuan, Shu-Tao Xia, Shuicheng Yan, Zhifeng Li, and Wei Liu. Improving vision transformers by revisiting high-frequency components. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022. 8

[3] Srinadh Bhojanapalli, Ayan Chakrabarti, Andreas Veit, Michal Lukasik, Himanshu Jain, Frederick Liu, Yin-Wen Chang, and Sanjiv Kumar. Leveraging redundancy in attention with reuse transformers. *arXiv preprint arXiv:2110.06821*, 2021. 1

[4] Yuchen Bian, Jiaji Huang, Xingyu Cai, Jiahong Yuan, and Kenneth Church. On attention redundancy: A comprehensive study. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 930–945, 2021. 1

[5] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *Proceedings of ICLR*, 2023. 1, 3, 5, 6, 2

[6] Boyu Chen, Peixia Li, Baopu Li, Chuming Li, Lei Bai, Chen Lin, Ming Sun, Junjie Yan, and Wanli Ouyang. Psvit: Better vision transformer via token pooling and attention sharing. *arXiv preprint arXiv:2108.03428*, 2021. 1, 3

[7] Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. Diffrate: Differentiable compression rate for efficient vision transformers. *arXiv preprint arXiv:2305.17997*, 2023. 6, 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 3, 5

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 3

[10] Matthew Dutson, Yin Li, and Mohit Gupta. Eventful transformers: Leveraging temporal redundancy in vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16911–16923, 2023. 1

[11] Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. Towards a deep and unified understanding of deep neural models in nlp. In *International conference on machine learning*, pages 2454–2463. PMLR, 2019. 3

[12] Jintao Guo, Na Wang, Lei Qi, and Yinghuan Shi. Aloft: A lightweight mlp-like architecture with dynamic low-frequency transform for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24132–24141, 2023. 8

[13] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *International Conference on Machine Learning*, pages 3854–3863, 2020. 4

[14] Mingfei Han, Yali Wang, Zhihui Li, Lina Yao, Xiaojun Chang, and Yu Qiao. Html: Hybrid temporal-scale multimodal learning framework for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13414–13423, 2023. 1

[15] Mingfei Han, Linjie Yang, Xiaojun Chang, and Heng Wang. Shot2story20k: A new benchmark for comprehensive understanding of multi-shot videos. *arXiv e-prints*, pages arXiv–2312, 2023. 1

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 4

[17] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *Proceedings of ICLR*, 2022. 1, 3, 6, 2

[18] Shiqi Lin, Zhizheng Zhang, Zhipeng Huang, Yan Lu, Cuiling Lan, Peng Chu, Quanzeng You, Jiang Wang, Zicheng Liu, Amey Parulkar, et al. Deep frequency filtering for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11797–11807, 2023. 8

[19] Dongrui Liu, Huiqi Deng, Xu Cheng, Qihan Ren, Kangrui Wang, and Quanshi Zhang. Towards the difficulty for a deep neural network to learn concepts of different complexities. In *NeurIPS*, 2023. 8

[20] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14430, 2023. 1, 2

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 3

[22] Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. In *Proceedings of ICLR*, 2018. 4

[23] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019. 1

[24] Harikrishnan NB, Aditi Kathpalia, and Nithin Nagaraj. Causality preserving chaotic transformation and classification using neurochaos learning. *Advances in Neural Information Processing Systems*, 35:2046–2058, 2022. 4

[25] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. Ia-red 2: Interpretability-aware redundancy reduction for vision transformers. *Advances in Neural Information Processing Systems*, 34:24898–24911, 2021. 1

[26] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks.

In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 8

[27] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021. 1, 3, 6

[28] Sucheng Ren, Fangyun Wei, Zheng Zhang, and Han Hu. Tinymim: An empirical study of distilling mim pre-trained models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3687–3697, 2023. 5

[29] Thomas Schreiber. Measuring information transfer. *Physical review letters*, 85(2):461, 2000. 4

[30] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020. 3

[31] Zhenhong Sun, Ce Ge, Junyan Wang, Ming Lin, Hesen Chen, Hao Li, and Xiuyu Sun. Entropy-driven mixed-precision quantization for deep network design. *Advances in Neural Information Processing Systems*, 35:21508–21520, 2022. 3

[32] Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12165–12174, 2022. 1, 3

[33] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 1, 2, 5, 6

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[35] Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2092–2101, 2023. 1, 3, 5, 6, 2

[36] Yuetian Weng, Mingfei Han, Haoyu He, Mingjie Li, Lina Yao, Xiaojun Chang, and Bohan Zhuang. Mask propagation for efficient video semantic segmentation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 1

[37] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2964–2972, 2022. 1, 3, 5, 6, 2

[38] Zhiqin John Xu. Understanding training and generalization in deep learning by fourier analysis. *arXiv preprint arXiv:1808.04295*, 2018. 8

[39] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. *Advances in Neural Information Processing Systems*, 34:30008–30022, 2021. 3

[40] Xingyi Yang, Daquan Zhou, Songhua Liu, Jingwen Ye, and Xinchao Wang. Deep model reassembly. *Advances in neural information processing systems*, 35:25739–25753, 2022. 1

[41] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022. 3

[42] Jun Zhang, Wen Yao, Xiaoqian Chen, and Ling Feng. Transferable post-hoc calibration on pretrained transformers in noisy text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13940–13948, 2023. 4

[43] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 5

[44] Huilin Zhou, Hao Zhang, Huiqi Deng, Dongrui Liu, Wen Shen, Shih-Han Chan, and Quanshi Zhang. Concept-level explanation for the generalization of a dnn. *arXiv preprint arXiv:2302.13091*, 2023. 8