Visual Transformer with Differentiable Channel Selection: An Information Bottleneck Inspired Approach

Yancheng Wang [1] **Ping Li** [2] **Yingzhen Yang** [1]

Abstract

Self-attention and transformers have been widely used in deep learning. Recent efforts have been devoted to incorporating transformer blocks into different types of neural architectures, including those with convolutions, leading to various visual transformers for computer vision tasks. In this paper, we propose a novel and compact transformer block, Transformer with Differentiable Channel Selection, or DCS-Transformer. DCS-Transformer features channel selection in the computation of the attention weights and the input/output features of the MLP in the transformer block. Our DCS-Transformer is compatible with many popular and compact transformer networks, such as MobileViT and EfficientViT, and it reduces the FLOPs of the visual transformers while maintaining or even improving the prediction accuracy. In the experiments, we replace all the transformer blocks in MobileViT and EfficientViT with DCS-Transformer blocks, leading to DCS-Transformer networks with different backbones. The DCS-Transformer is motivated by reduction of Information Bottleneck, and a novel variational upper bound for the IB loss which can be optimized by SGD is derived and incorporated into the training loss of the network with DCS-Transformer. Extensive results on image classification and object detection evidence that DCS-Transformer renders compact and efficient visual transformers with comparable or much better prediction accuracy than the original visual transformers. The code of DCS-Transformer is available at https://github.com/Statistical-Deep-Learning/DCS-Transformer.

[1]School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85281, USA [2]VecML Inc., Bellevue, WA 98004, USA. Correspondence to: Yingzhen Yang <yingzhen.yang@asu.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Building upon the success of Transformer in natural language processing (Vaswani et al., 2017), visual transformers have demonstrated remarkable performance compared to state-of-the-art Convolution Neural Networks (CNNs) across a wide range of computer vision tasks, including image classification (Yuan et al., 2021; Dosovitskiy et al., 2021a), object detection (Liu et al., 2021; Zhu et al., 2021), image restoration (Liang et al., 2021), and semantic segmentation (Cai et al., 2023). However, the achievements of visual transformers are accompanied by heavy computational costs (Dosovitskiy et al., 2021a; Touvron et al., 2021), making their deployment impractical under resource-limited scenarios. The aforementioned limitations have spurred recent research endeavors aimed at developing efficient visual transformers. Several approaches have been explored, including incorporation of convolution layers into visual transformers and Neural Architecture Search (NAS), to design efficient transformer architectures, and pruning redundant weights in the transformer networks. In this paper, we study the problem of compressing visual transformers by channel selection.

Channel selection is an effective method for pruning large-weight convolutional neural networks (CNNs) (Han et al., 2015; Zhou et al., 2020; Sun et al., 2021). In addition to reducing the computation cost of network architecture, adaptively selecting informative channels is also shown to be beneficial for representation learning (He et al., 2021; Han et al., 2021). Models can extract and retain essential visual information while discarding noise or irrelevant features by focusing on the most relevant channels selected for different input images. Recent works in selecting informative channels for visual transformers (Zheng et al., 2022a; Chen et al., 2021c; Yu et al., 2022a) have already proven to be effective in compressing the size of the networks. However, existing works usually select fixed channels in the linear layers of transformer blocks (Yu et al., 2022b; Fang et al., 2023). In addition, they neglect the importance of selecting informative channels for calculating the affinity between

their visual tokens, which is the pivotal component in aggregating global information.

Motivation. A typical transformer block can be written as Output $=$ MLP $\big(\sigma(QK^\top) \times V\big)$ where $Q, K, V \in \mathbb{R}^{N\times D}$ denote the query, key, and value respectively with N being the number of tokens and D being the input channel number. $\sigma(\cdot)$ is an operator, such as Softmax, which generates the attention weights or affinity between the tokens. We refer to $W = \sigma(QK^\top) \in \mathbb{R}^{N\times N}$ as the affinity matrix between the tokens. MLP (multi-layer perceptron network) generates the output features of the transformer block. There are D channels in the input and output features of the MLP, and D is also the channel of the attention outputs. Due to the fact that the MLP accounts for a considerable amount of FLOPs in a transformer block, the size and FLOPs of a transformer block can be significantly reduced by reducing the channels of the attention outputs from D to a much smaller $\tilde{D}$. **Our goal is to prune the attention output channels while maintaining and even improving the prediction accuracy of the original transformer.** However, directly reducing the channel's attention outputs, even by carefully designed methods, would adversely affect the performance of the model. In this paper, we propose to maintain or even improve the prediction accuracy of a visual transformer with pruned attention outputs channels by computing a more informative affinity matrix W by selecting informative channels in the query Q and the key V. That is, only selected columns of Q, which correspond to the same selected rows of $K^\top$, are used to compute the affinity matrix $W = \sigma(QK^\top)$, which is referred to as channel selection for attention weights and illustrated in Figure 1. We note that the attention outputs, which are also the input features to the MLP, is $W \times V$, and every input feature to the MLP is an aggregation of the rows of the value V using the attention weights in W. As a result, pruning the channels of $W \times V$ amounts to pruning the channels of V in the weighted aggregation. If the affinity W is more informative, it is expected that a smaller number of features (rows) in V contribute to such weighted aggregation, and the adverse effect of channel selection on the prediction accuracy of the transformer network is limited. Importantly, with a very informative affinity W, every input feature of the MLP is obtained by aggregation of the most relevant features (rows) in V, which can even boost the performance of visual transformers after channel selection or pruning of the attention outputs.

The idea of channel selection for the attention weights can be viewed from the perspective of Information Bottleneck (IB). Let X be the input training features, $\tilde{X}$ be the learned features by the network, and Y be the ground truth training labels for a classification task. The principle of IB is maximizing the mutual information between $\tilde{X}$ and Y while minimizing the mutual information between $\tilde{X}$ and X. That is, IB encourages the network to learn features more correlated with the class labels while reducing their correlation with the input. Extensive empirical and theoretical works have evidenced that models respecting the IB principle enjoy compelling generalization. With channel selection for the attention weights, every feature in the attention outputs aggregates less features of the value V, so the attention outputs are less correlated with the training images so the IB principle is better adhered. This is reflected in Table 4 in Section 4.5, where a model for ablation study with channel selection for attention weights, DCS-Arch1 w/o IB Loss, enjoys less IB loss and higher top-1 accuracy than the vanilla transformer, MobileViT-S. It is noted that the model, DCS-Arch1 w/o IB Loss, only uses the regular cross-entropy loss in the retraining step, and a smaller IB loss indicates that the IB principle is better respected. In order to further decrease the IB loss, we propose an Information Bottleneck (IB) inspired channel selection for the attention weights $\mathbf{W}$ where the learned attention weights can be more informative by explicitly optimizing the IB loss for visual transformers. Our model termed "DCS-MobileViT-S" in Table 4 is the visual transformer with the IB loss optimized, so that more informative attention weights are learned featuring even smaller IB loss and even higher top-1 accuracy. Figure 7 and Figure 6 in the supplementary illustrate the visualization results by Grad-CAM and the attention weights produced by the proposed DCS-Transformer with channel selection in attention weights, evidencing that the channel selection inspired by the IB principle leads to more informative attention weights for learning semantic concepts.

1.1. Contributions

The contributions of this paper are presented as follows.

First, we present a novel and compact transformer block termed Transformer with Differentiable Channel Selection, or DCS-Transformer. Using our proposed channel selection in both the computation for attention weights and the features of the MLP, DCS-Transformer blocks automatically select channels in queries and keys to compute more informative attention weights inspired by the IB principle. DCS-Transformer blocks can be used to replace all the transformer blocks in many popular visual transformers, rendering compact visual transformers with comparable or even better performance. The effectiveness of DCS-Transformer is evidenced by replacing all the transformer blocks with DCS-Transformer blocks into popular visual transformers which are already compact, MobileViT (Mehta & Rastegari, 2022), EfficientViT (Cai et al., 2023), ViT-S/16 (Dosovitskiy et al., 2021a), and Swin-T (Liu et al., 2021), for image classification, object detection and instance segmentation tasks.

Second, to the best of our knowledge, we derive the first *separable* variational upper bound for the IB loss, which is $I(\tilde{X}) - I(\tilde{X}, Y)$ where $I(\cdot, \cdot)$ denotes mutual information. Such separable upper bound for IB can be directly incorporated into existing training loss of deep neural networks even beyond transformers, and optimized in an end-to-end manner by standard SGD. Experimental results demonstrate that the IB loss of the visual transformer can be reduced by optimizing the composite loss formed by our variational upper bound for the IB loss and the regular cross-entropy loss, and the transformer network trained with such variational upper bound exhibits stronger generalization. We remark that as shown in Table 4 of the supplementary, channel selection in either the attention weights or the attention output without optimizing the IB loss can already reduce the IB loss. By explicitly optimizing the IB loss using its variational upper bound, network with DCS-Transformer enjoys smaller IB loss, higher classification accuracy and less FLOPs.

This paper is organized as follows. The related works in efficient visual transformers and compression of visual transformers are discussed in Section 2. The formulation of DCS-Transformer is detailed in Section 3. The effectiveness of DCS-Transformer is demonstrated in Section 4 for image classification and object detection tasks, by replacing all the transformer blocks of MobileViT and EfficientViT with DCS-Transformer blocks. We conclude the paper in Section 5.

2. Related Works

2.1. Efficient Visual Transformers

Recently, visual transformer models have emerged as a popular alternative to convolutional neural networks (CNNs) in computer vision tasks, such as image classification (Dosovitskiy et al., 2021b; Liu et al., 2021), object detection (Carion et al., 2020; Zhu et al., 2021), and image restoration (Liang et al., 2021; Wang et al., 2022a). Albeit the great performance of visual transformers, they usually suffer from high computation costs due to the quadratic complexity of point-wise attention modules as well as a large number of Multi-Layer Perception (MLP) encoding layers. To reduce the computation cost of visual transformer models, many recent works (Zhu et al., 2021; Yuan et al., 2021) have been proposed using different methodologies, mainly by employing sparse mechanisms to design efficient attention modules and optimizing the network architecture of the transformer models. In order to design compact network architecture of transformer models, another series of works (Cai et al., 2023; Mehta & Rastegari, 2022; Yuan et al., 2021) introduce convolutions into the network architecture of visual transformers. For instance, MobileViT (Mehta & Rastegari, 2022) introduces a hybrid architecture that combines lightweight MobileNet convolution blocks (MBConv) and MHSA modules. MobileViT places the convolution blocks at early stages in its architecture to extract low-level features while placing MHSA in late stages to achieve global representation learning. In addition, several works leverage Neural Architecture Search (NAS) (Chen et al., 2021b; Gong et al., 2022) to design efficient visual transformers. Other works also attempt to enhance the performance of efficient visual transformers by incorporating knowledge distillation into their training (Graham et al., 2021; Radosavovic et al., 2020; Gong et al., 2022).

2.2. Compressing Visual Transformers

Recent studies have also investigated compressing existing visual transformers to reduce their computation cost. Current compression methods for visual transformers usually fall into three categories: (1) Channel Pruning, which prune redundant heads and channels in ViT Blocks (Chen et al., 2021c; Chavan et al., 2022; Zheng et al., 2022a). (2) Block Pruning, which drops redundant transformer blocks in transformer networks (Yu et al., 2022b;a). (3) Token Pruning, which only keeps informative tokens as the inputs for transformer blocks (Rao et al., 2021; Kong et al., 2022a; Bolya et al., 2023; Wang et al., 2022b).

In this paper, we focus on channel selection for compression of visual transformers, and we propose to prune the channels of the MLP features or the attention outputs in all the transformer blocks in a visual transformer. In order to achieve comparable or even better prediction accuracy than the original visual transformer, channel selection is also performed in the computation of attention weights inspired by the IB principle so as to generate more informative attention weights, so that the adverse effect of channel reduction in the MLP features is compensated.

2.3. Related Works about Information Bottleneck

(Saxe et al., 2019) first discusses existing information bottleneck theories of deep learning. By building the connection of the compression phase of training in existing information bottleneck theories with the neural nonlinearity, they prove that the compression phase of training is not related to the excellent generalization performance of deep networks. (Lai et al., 2021) proposes to learn probabilistic maps in a spatial attention module that reduces the mutual information between the masked representation and the input while increasing the mutual information between the masked representation and the task label. (Zhou et al., 2022) proves that self-attention can be written as an iterative optimization step of the Information Bottleneck objective. Next, they show that self-attention can promote the robustness of neural networks through improved mid-level

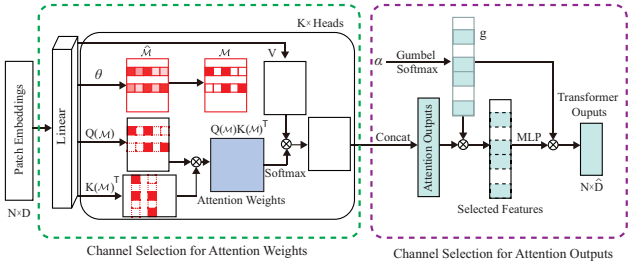


Figure 1: DCS-Transformer

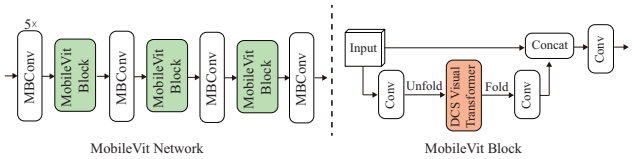


Figure 2: Architecture of DCS-MobileViT

representations. They further propose a family of fully attentional networks (FANs) that take advantage of such merits of self-attention.

In contrast with most existing works that model the IB principle implicitly, our work directly optimizes the IB loss by adding its variational upper bound to the training loss of a neural network and optimizing the joint loss by standard SGD algorithms. In this manner, any neural network designed for classification tasks can enjoy potential improvement by the IB principle through the separable variational upper bound for the IB loss.

3. Formulation

In this section, we present two types of channel selection in our DCS-Transformer, which are (1) channel selection for attention weights that renders more informative attention weights or affinity between tokens; (2) channel selection for attention outputs which prunes the channels of the MLP features so as to reduce the FLOPs of the transformer block. We then present the novel variational upper bound for the IB loss, and introduce the training algorithm of the network with DCS-Transformer and the variational upper bound.

3.1. Channel Selection for Attention Weights

Given the input patch embeddings $x \in \mathbb{R}^{N\times D}$, where N is the number of visual tokens and D is the token dimension, visual transformer blocks first obtain the query $Q \in \mathbb{R}^{N\times D}$, key $K \in \mathbb{R}^{N\times D}$, and value $V \in \mathbb{R}^{N\times D}$ for the multi-head self-attention module with linear layers. The attention weights are then calculated by the dot-product between Q and K. Let x_i denote the i-th row of x. The (i,j)-th entry of the attention weight matrix $QK^\top$ is the correlation between the feature x_i and x_j. We aim to select informative channels for the computation of the attention weights. To achieve this goal, we maintain a binary decision mask $\mathcal{M} \in \{0,1\}^{N\times D}$, where its (i,d)-th element $\mathcal{M}_{id} = 1$ indicates the d-th channel of feature x_i is selected. As a result, the attention weights or the affinity matrix W is computed by $W = \sigma\left((Q\odot\mathcal{M})(K\odot\mathcal{M})^\top\right)$, where σ is the Softmax operator on each row of the input features and $\odot$ indicates elementwise product. To optimize the discrete binary decision mask with gradient descent, we adopt simplified binary Gumbel-Softmax (Verelst & Tuytelaars, 2020) to relax $\mathcal{M} \in \{0,1\}^{N\times D}$ into its approximation in the continuous domain $\widehat{\mathcal{M}} \in (0,1)^{N\times D}$. The approximated soft decision mask can be computed by $\widehat{\mathcal{M}}_{id} = \sigma\left(\frac{\theta_{id}+\epsilon_{id}^{(1)}-\epsilon_{id}^{(2)}}{\tau}\right)$, where $\widehat{\mathcal{M}}_{id}$ is the (i,d)-th element of $\widehat{\mathcal{M}}$. $\epsilon_{id}^{(1)}$ and $\epsilon_{id}^{(2)}$ are Gumbel noise for the approximation. τ is the temperature, and $\sigma(\cdot)$ is the Sigmoid function. $\theta \in \mathbb{R}^{N\times D}$ is the sampling parameter. In this work, we obtain θ by applying a linear layer on x so that the soft decision mask is dependent on the input features of the transformer block. Inspired by the straight-through estimator (Verelst & Tuytelaars, 2020; Bengio et al., 2013), we directly set $\mathcal{M} = \widehat{\mathcal{M}}$ in the backward pass. In the forward pass, the binary decision mask is computed by $\mathcal{M}_{id} = 1$ if $\widehat{\mathcal{M}}_{id} > 0.5$, and 0 otherwise. During inference, the Gumbel noise $\epsilon_{id}^{(1)}$ and $\epsilon_{id}^{(2)}$ are set to 0. Using the Gumbel-Softmax above, the informative channels for the attention weights computation are selected in a differentiable manner.

3.2. Channel Selection for Attention Outputs

After applying the multi-head self-attention on the input patch embedding x, we obtain the attention outputs $z \in \mathbb{R}^{N\times D}$. Our DCS-transformer block then applies MLP layers to the attention outputs. The MLP layers in visual transformer blocks are usually computationally expensive. To improve the efficiency of the DCS-transformer, we propose to prune the channels in the attention outputs so that the computation cost of MLP layers can be reduced. Similar to the channel selection for attention weights, we maintain a decision mask $g_i \in \{0,1\}^D$, where $g_i = 1$ indicates that the i-th channel is selected, and 0 otherwise. Thus, the informative channels can be selected by multiplying g by each row of the attention output. To optimize the binary decision mask with gradient descent, we replace g with Gumbel Softmax weights in the continuous domain, which is computed by $\widehat{g}_i = \sigma\left(\frac{\alpha_i+\epsilon_i^{(1)}-\epsilon_i^{(2)}}{\tau}\right)$, where $\epsilon_i^{(1)}$ and $\epsilon_i^{(2)}$ are Gumbel noise. τ is the temperature. $\alpha \in \mathbb{R}^D$ is the sampling parameter. We define α as the architecture parameters of the DCS-Transformer block that can be optimized by gradient descent during the differentiable search process. By gradually decreasing the temperature

τ in the search process, α_i will be optimized such that g_i will approach 1 or 0. Note that since the MLP layers in visual transformers have the same input and output dimensions, so we multiply the decision mask g with both the input and output features of the MLP layers. After the search is finished, we apply the gather operation on the attention outputs from the selected channels. The dimension of the input and output features of the MLP layers are then changed to $\tilde{D} = \sum_{i=1}^{D} g_i$. The architecture and the two types of channel selection are illustrated in Figure 1. Our DCS-Transformer block is compatible with most visual transformers, and Figure 2 illustrates the architecture of DCS-MobileViT, which is obtained by replacing each transformer block in MobileViT with a DCS-Transformer block. In a similar manner, we can have DCS-X, where X stands for a visual transformer. For example, the details about DCS-EfficientViT are introduced in Appendix A.1.

3.3. Variational Upper Bound for the IB Loss

Given the training data $\{X_i, y_i\}_{i=1}^n$ where x_i is the i-the input training feature and y_i is the corresponding class label, we first specify how to compute the IB loss, $\text{IB}(\mathcal{W}) = I(\tilde{X}(\mathcal{W}), X) - I(\tilde{X}(\mathcal{W}), Y)$, where $\mathcal{W}$ is the weights of a neural network, X is a random variable representing the input feature, which takes values in $\{X_i\}_{i=1}^n$, $\tilde{X}(\mathcal{W})$ is a random variable representing the learned feature which takes values in $\left\{\tilde{X}_i(\mathcal{W})\right\}_{i=1}^n$ with $\tilde{X}_i(\mathcal{W})$ being the i-th learned feature by the network. Y is a random variable representing the class label, which takes values in $\{y_i\}_{i=1}^n$. After performing K-means clustering on $\left\{\tilde{X}_i(\mathcal{W})\right\}_{i=1}^n$ and $\{X_i\}_{i=1}^n$, we have the clusters $\{\mathcal{C}_a\}_{a=1}^A$ and $\{\mathcal{C}_b\}_{b=1}^B$ for the learned features and the input features respectively. Here we set $A = B = C$ where C is the number of classes. We also abbreviate $\tilde{X}(\mathcal{W})$ as $\tilde{X}$ for simplicity of the notations. Then we define the probability that $\tilde{X}_i$ belongs to cluster $\mathcal{C}_a$ as $\Pr\left[\tilde{X} \in a\right] = \frac{1}{n}\sum_{i=1}^{n} \phi(\tilde{X}_i, a)$ with $\phi(\tilde{X}_i, a) = \frac{\exp\left(-\left\|\tilde{X}_i - \mathcal{C}_a\right\|_2^2\right)}{\sum_{a=1}^{A}\exp\left(-\left\|\tilde{X}_i - \mathcal{C}_a\right\|_2^2\right)}$. Similarly, we define the probability that X_i belongs to cluster $\mathcal{C}_b$ as $\Pr[X \in b] = \frac{1}{n}\sum_{i=1}^{n} \phi(X_i, b)$. Moreover, we have the joint probabilities $\Pr\left[\tilde{X} \in a, X \in b\right] = \frac{1}{n}\sum_{i=1}^{n} \phi(\tilde{X}_i, a)\phi(X_i, b)$ and $\Pr\left[\tilde{X} \in a, Y = y\right] = \frac{1}{n}\sum_{i=1}^{n} \phi(\tilde{X}_i, a)\mathbb{1}_{\{y_i = y\}}$ where $\mathbb{1}_{\{\}}$ is an indicator function. As a result, we can compute the mutual information $I(\tilde{X}, X)$ and $I(\tilde{X}, Y)$ by

$$I(\tilde{X}, X) = \sum_{a=1}^{A}\sum_{b=1}^{B} \Pr\left[\tilde{X} \in a, X \in b\right] \log \frac{\Pr\left[\tilde{X} \in a, X \in b\right]}{\Pr\left[\tilde{X} \in a\right]\Pr\left[X \in b\right]},$$

$$I(\tilde{X}, Y) = \sum_{a=1}^{A}\sum_{y=1}^{C} \Pr\left[\tilde{X} \in a, Y = y\right] \log \frac{\Pr\left[\tilde{X} \in a, Y = y\right]}{\Pr\left[\tilde{X} \in a\right]\Pr\left[Y = y\right]},$$

and then compute the IB loss $\text{IB}(\mathcal{W})$. Given a variational distribution $Q(\tilde{X} \in a | Y = y)$ for $y \in \{1, \dots C\}$ and $a \in \{1, \dots A\}$, the following theorem gives a variational upper bound, $\text{IBB}(\mathcal{W})$, for the IB loss $\text{IB}(\mathcal{W})$.

Theorem 3.1.

$$\text{IB}(\mathcal{W}) \leq \text{IBB}(\mathcal{W}), \tag{1}$$

where

$$\text{IBB}(\mathcal{W}) := \frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{A}\sum_{b=1}^{B} \phi(\tilde{X}_i, a)\phi(X_i, b)\log\phi(X_i, b) - \frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{A}\sum_{y=1}^{C} \phi(\tilde{X}_i, a)\mathbb{1}_{\{y_i = y\}}\log Q(\tilde{X} \in a | Y = y).$$

The proof of this theorem follows by applying Lemma D.1 and Lemma D.2 in Section D of the supplementary. We remark that $\text{IBB}(\mathcal{W})$ is ready to be optimized by standard SGD algorithms because it is separable and expressed as the summation of losses on individual training points. Algorithm 1 describes the training process of a neural network with DCS-Transformer blocks where $\text{IBB}(\mathcal{W})$ is a term in the training loss. In order to compute $\text{IBB}(\mathcal{W})$ before a new epoch starts, we need to update the variational distribution $Q^{(t)}$ at the end of the previous epoch. The following functions are needed for minibatch-based training with SGD, with the subscript j indicating the corresponding loss on the j-th batch $\mathcal{B}_j$:

$$\text{IBB}_j^{(t)}(\mathcal{W}) = \frac{1}{|\mathcal{B}_j|}\left(\sum_{\substack{i \in [|\mathcal{B}_j|],\\ a \in [A],\\ b \in [B]}} \phi(\tilde{X}_i(\mathcal{W}), a)\phi(X_i, b)\log\phi(X_i, b) - \sum_{\substack{i \in [|\mathcal{B}_j|],\\ a \in [A],\\ y \in [C]}} \phi(\tilde{X}_i(\mathcal{W}), a)\mathbb{1}_{\{y_i = y\}}\log Q^{(t-1)}(\tilde{X} \in a | Y = y)\right),$$

$$\mathcal{L}_{\text{train},j}^{(t)}(\mathcal{W}) = \text{CE}_j^{(t)} + \eta\text{IBB}_j^{(t)}(\mathcal{W}), \tag{2}$$

$$\text{CE}_j^{(t)} = \frac{1}{|\mathcal{B}_j|}\sum_{i=1}^{|\mathcal{B}_j|} H(X_i(\mathcal{W}), Y_i).$$

Here $\text{CE}_j^{(t)}$ is the cross-entropy loss on batch $\mathcal{B}_j$ at epoch t. $H(,)$ is the cross-entropy function. η is the balance factor for the loss of information bottleneck.

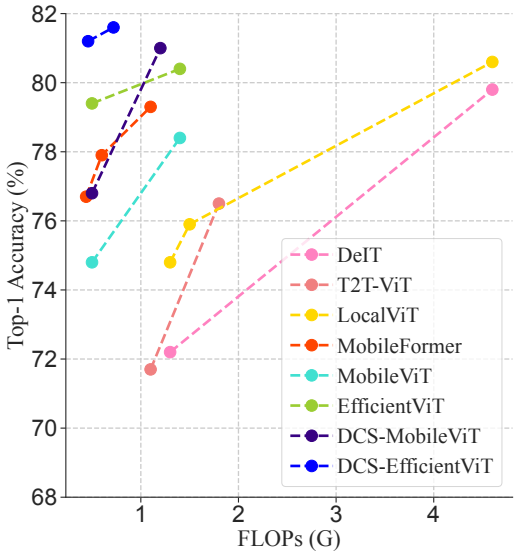


Figure 3: Top-1 accuracy vs FLOPs (G) on ImageNet-1k validation set.

Model	# Params	FLOPs	Top-1
T2T (Yuan et al., 2021)	4.3 M	1.1 G	71.7
DeIT (Touvron et al., 2021)	5.7 M	1.2 G	72.2
CrossViT (Chen et al., 2021a)	6.9 M	1.6 G	73.4
MobileViT-XS (Mehta & Rastegari, 2022)	2.3 M	0.7 G	74.8
DCS-MobileViT-XS (Ours)	2.0 M	0.5 G	**76.8**
DeIT (Touvron et al., 2021)	10 M	2.2 G	76.6
T2T (Yuan et al., 2021)	6.9 M	1.8 G	76.5
PiT (Heo et al., 2021)	10.6 M	1.4 G	78.1
Mobile-Former (Chen et al., 2021e)	9.4 M	0.2 G	76.7
EViT (Liu et al., 2023)	12.4 M	0.5 G	77.1
TinyViT (Wu et al., 2022)	5.4 M	1.3 G	79.1
DeIT (Touvron et al., 2021)	22 M	4.6 G	79.8
ToMe (Bolya et al., 2023)	22 M	2.7 G	79.4
EfficientFormer (Li et al., 2022)	12.3 M	1.3 G	79.2
MobileViT-S (Mehta & Rastegari, 2022)	5.6 M	1.4 G	78.4
VTC-LFC (Wang et al., 2022b)	5.0 M	1.3 G	78.0
SPViT (Kong et al., 2022a)	4.9 M	1.2 G	77.8
ToMe (Bolya et al., 2023)	5.6 M	1.2 G	77.3
DCS-MobileViT-S (Ours)	4.8 M	1.2 G	**81.0**
EfficientViT-B1 [r224] (Cai et al., 2023)	9.1 M	0.52 G	79.4
EfficientViT-B1 [r288] (Cai et al., 2023)	9.1 M	0.86 G	80.4
EViT (Liu et al., 2023)	8.8 M	0.29 G	74.3
VTC-LFC (Wang et al., 2022b)	8.7 M	0.76 G	79.3
SPViT (Kong et al., 2022a)	8.3 M	0.71 G	79.0
ToMe (Bolya et al., 2023)	9.1 M	0.47 G	78.8
DCS-EfficientViT-B1 [r224] (Ours)	8.2 M	0.46 G	**80.8**
DCS-EfficientViT-B1 [r288] (Ours)	8.2 M	0.72 G	**81.6**
ViT-S/16 (Dosovitskiy et al., 2021a)	22.1 M	4.3 G	81.2
DCS-ViT-S/16 (Ours)	20.2 M	3.9 G	**82.0**
Swin-T (Liu et al., 2021)	29.0 M	4.5 G	81.3
DCS-Swin-T (Ours)	26.1 M	4.0 G	**82.0**

Table 1: Comparisons with baseline methods on ImageNet-1k validation set.

Algorithm 1 Training Algorithm with the Variational Upper Bound for IB by SGD

Initialize the weights of the network by $\mathcal{W} = \mathcal{W}(0)$ through random initialization
for $t \leftarrow 1$ to t_{search} **do**
  for $j \leftarrow 1$ to J **do**
    Perform gradient descent with batch $\mathcal{B}_j$ using the loss $\mathcal{L}^{(t)}_{\text{search},j}(\mathcal{W}, \alpha)$ defined Section 3.4.
  end for
end for
for $t \leftarrow 1$ to t_{train} **do**
  for $j \leftarrow 1$ to J **do**
    Update $\phi(\tilde{X}_i, a)$ for all the clusters $a \in \{1, \ldots, A\}$ and $i \in \{1, \ldots, n\}$.
    if $t > t_{\text{warm}}$ **then**
      Perform gradient descent with batch $\mathcal{B}_j$ using the loss $\mathcal{L}^{(t)}_{\text{train},j}(\mathcal{W})$ by Eq. (2).
    else
      Perform gradient descent with batch $\mathcal{B}_j$ using the loss $\text{CE}^{(t)}_j$ by Eq. (2).
    end if
  end for
  Compute $Q^{(t)}(\tilde{X} \in a | Y = y)$ by Eq. (7) in the supplementary, perform K-means clustering on $\{\tilde{X}_i\}_{i=1}^n$ and update the clusters $\{\mathcal{C}_a\}_{a=1}^A$.
end for
return The trained weights $\mathcal{W}$ of the network

3.4. Optimization in the Search Process

To obtain a compact visual transformer network with DCS-Transformer, we need to optimize both the accuracy of the network and the inference cost (FLOPs) of the network. Therefore, the differentiable inference cost of the network needs to be estimated and optimized during the search phase. It is worthwhile to mention that we follow the standard techniques in neural architecture search (Tan et al., 2019) in the searching process, including channel selection by Gumbel-Softmax and entropy minimization for architecture search. We optimize the FLOPs of the operations whose computation cost is decided by the channel selection for attention outputs in Section 3.2. For DCS-MobileViT, we estimate the FLOPs of the MLP after the channel selection on the attention outputs and the FLOPs of the convolution block following the DCS-Transformer. The estimation of the FLOPs related to a single MobileViT block is $\texttt{cost}_j = 2 \cdot \left(2\tilde{D}^2 + \tilde{D}\right) + (1 + \tilde{D}) \cdot HWC$, where j indexes the DCS-Transformer block. $2\tilde{D}^2 + \tilde{D}$ is the FLOPs of a layer of the MLP after the channel selection on the attention outputs, and there are two layers in the MLP. $(1 + \sum_{i=1}^{D} g_i) \cdot HWC$ is the FLOPs of the convolution block following the DCS-Transformer. C is the number of filters of the convolution block. H and W are the height and width of the input features of the DCS-Transformer block. As a result, we can calculate the inference cost objective of the network architecture by $\texttt{cost} = \sum_{j=1}^{M} \texttt{cost}_j$, where M is the number of transformer blocks. For DCS-EfficientViT, we have $\texttt{cost}_j = (2\tilde{D}^2 + \tilde{D}) + (1 + \tilde{D}) \cdot HWC$ which is detailed in Section A.1 of the supplementary.

To supervise the search process, we designed a loss

function incorporating cost-based regularization to enable multi-objective optimization. The overall loss function for search on each batch $\mathcal{B}_j$ at epoch t is formulated by $\mathcal{L}^{(t)}_{\text{search},j}(\mathcal{W}, \alpha) = \text{CE}^{(t)}_j + \lambda \cdot \log \texttt{cost}(\alpha)$, $\text{CE}^{(t)}_j = \frac{1}{|\mathcal{B}_j|}\sum_{i=1}^{|\mathcal{B}_j|} H(X_i(\mathcal{W}), Y_i)$, where $\mathcal{W}$ denotes the weights in the supernet. α is the architecture parameters. λ is the hyper-parameters that control the magnitude of the cost term, which is selected by cross-validation. In the search phase, the search loss is optimized to perform the two types of channel selection for all the DCS-Transformer blocks. After the search phase, we use the selected channels for both attention weights and attention outputs in a searched network and then perform retraining on the searched network.

4. Experimental Results

In this section, we first evaluate the performance of DCS-Transformers on the ImageNet-1k dataset for image classification, and show that both models render better performance than state-of-the-art networks in Section 4.1 with more compact models. In Section 4.2 and Section 4.3, we show that using DCS-MobileViT and DCS-EfficientViT as the feature extraction backbones achieve better mAP with lower FLOPs than the competing baselines for semantic segmentation and object detection.

4.1. Image Classification

Implementation details about Search/Retraining. We use MobileViT-S (Mehta & Rastegari, 2022), MobileViT-XS (Mehta & Rastegari, 2022), and EfficientViT-B (Cai et al., 2023) as the backbones for our experiments in ImageNet classification. We replace all the transformer blocks in the backbones with DCS-Transformer blocks. In the search phase, we sample 100 classes from ImageNet (Russakovsky et al., 2015) as the training data. A cosine learning rate schedule is used in the AdamW optimizer with parameters β_1 and β_2 set to 0.9 and 0.999 respectively. The learning rate is initialized as 0.001 and then annealed to 0.0001 in 200 epochs. In each epoch, we use 70% of the training data to optimize the network weights and 30% to optimize the architecture parameters in all the DCS-Transformer blocks. We set the initial value of the temperature τ to 4.5 and decrease it by a factor of 0.95 every epoch.

After the search is finished, we sample the searched architecture from the supernet and perform retraining. We defer more Details about retraining, EfficientViT with DCS-Transformer, and tuning hyper-parameters by cross-validation to Section A.2 of the supplementary. We select the values of t_{warm} and η in Algorithm 1 by cross-validation as described in Section A.2 of the supplementary. The validation results suggest that $t_{\text{warm}} = 90$ and $\eta = 50$ work the best for all our DCS models.

Methods	mAP^{box}	AP^b_{50}	AP^b_{75}	mAP^m	AP^m_{50}	AP^m_{75}
EViT (Liu et al., 2023)	32.8	54.4	34.5	31.0	51.2	32.2
EfficientViT-B1 (Cai et al., 2023)	33.5	55.4	34.8	31.9	52.3	32.7
DCS-EfficientViT-B1	**34.8**	**56.3**	**35.3**	**33.2**	**53.1**	**33.3**

Table 2: Instance Segmentation Results on COCO.

Tuning Hyper-Parameters by Cross-Validation. To decide the best balancing factor λ for the overall search loss, η for the overall training loss, and the number of warm-up epochs t_{warm}, we perform 5-fold cross-validation on 10% of the training data. The value of λ is selected from $\{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0, 5\}$. The value of η is selected from $\{0.1, 0.5, 1, 5, 10, 50, 100\}$. The value of t_{warm} is selected from $\{0.1t_{\text{train}}, 0.2t_{\text{train}}, 0.3t_{\text{train}}, 0.4t_{\text{train}}, 0.5t_{\text{train}}, 0.6t_{\text{train}}\}$, where $t_{\text{train}} = 300$ is the total number of training epochs. We select the values of η, λ, and t_{warm} that lead to the smallest validation loss. It is reveled that $t_{\text{warm}} = 90$ is chosen for all the three visual transformers in our experiments.

Results. It can be observed from Table 1 that models with DCS-Transformer always enjoy less FLOPs than its original visual transformer and better accuracy. For example, DCS-EfficientViT has an accuracy improvement of almost 1% while the FLOPs is reduced from 0.54G to 0.46G compared to the original EfficientViT. In addition, DCS-MobileViT-S has an accuracy improvement of 1.5% compared to the original MobileViT-S, while its FLOPs is reduced from 1.4G to 1.2G. To study how DCS impacts the attention mechanism, we show the Grad-CAM visualization examples in Figure 7 of the supplementary. In addition, we also visualize the attention weights computed on several examples in Figure 6 of the supplementary. More experiments on ImageNet are deferred to Section B of the supplementary.

4.2. Instance Segmentation

In this section, we evaluate the performance of DCS for the task of instance segmentation on the COCO (Lin et al., 2014) dataset. We adopt Mask R-CNN (He et al., 2017) with Feature Pyramid Network (FPN) as the segmentation head on top of the feature backbone of DCS-EfficientViT-B1. We include EfficientViT-B1 (Cai et al., 2023) and EViT (Liu et al., 2023) as baselines for comparisons. We train our model and the baselines on the train split of the dataset and report the performance on the validation split following (Chen et al., 2019). Both our method and the baselines are trained for 12 epochs following the settings of $1\times$ schedule in (Chen et al., 2019). We use AdamW as the optimizer in the training following (Liu et al., 2023). The initial learning rate is set to 0.001 and decays with a cosine learning rate schedule. We measure and report the mean

bounding box Average Precision (mAP^b) and mean mask Average Precision (mAP^b) as well as bounding box Average Precision (AP^b) and mask Average Precision (AP^b) under IoU thresholds of 0.5 and 0.75. The results are shown in Table 2. It can be observed that DCS-EfficientViT-B1 consistently improves the performance of segmentation across various thresholds.

4.3. Object Detection

Implementation details. We integrate ImageNet pre-trained DCS-MobileViT-XS, DCS-MobileViT-S, and DCS-EfficientViT with the single-shot object detection backbone SSDLite (Sandler et al., 2018). The models are evaluated on the MS-COCO dataset (Lin et al., 2014) that contains 117k training and 5k validation images. We fine-tune all pre-trained DCS-Transformer networks with the object detection framework at an input resolution of 320×320. All models are trained with AdamW for 200 epochs following the same settings in MobileViT (Mehta & Rastegari, 2022). With a cosine learning rate scheduler, the learning rate is decayed from 0.0009 to $1.6e^{-6}$. We use smooth ℓ^1 for the object localization objective and cross-entropy losses for classification. The performance is evaluated on the validation set using mAP@IoU of 0.50:0.05:0.95.

Results. We conduct a comparative analysis of our models against alternative lightweight feature backbones that are integrated with the SSDLite object detection framework. The results are shown in Table 3 of the supplementary, which show that object detection performance is greatly improved by replacing the feature backbone with models with DCS-Transformer blocks. For instance, the mAP is improved by 1% after replacing MobilViT-S with DCS-MobilViT-S while saving 0.3G FLOPs. Furthermore, with the same FLOPs SSDLite as MobilNetV3, SSDLite with DCS-EfficientViT achieves performance improvement of 7%.

Feature backbone	# Params.	FLOPs	mAP
MobileNetv3 (Howard et al., 2019)	4.9 M	1.4 G	22.0
MobileNetv2 (Sandler et al., 2018)	4.3 M	1.6 G	22.1
MobileNetv1 (Howard et al., 2017)	5.1 M	2.6 G	22.2
MixNet (Tan & Le, 2019)	4.5 M	2.2 G	22.3
MNASNet (Tan et al., 2019)	4.9 M	1.7 G	23.0
YoloV5-N (640×640) (Redmon & Farhadi, 2017)	1.9 M	4.5 G	28.0
Vidt (Song et al., 2022)	7.0 M	6.7 G	28.7
MobileViT-XS	2.7 M	1.7 G	24.8
DCS-MobileViT-XS(Ours)	2.4 M	1.5 G	**25.8**
MobileViT-S	5.7 M	2.4 G	27.7
DCS-MobileViT-S(Ours)	4.7 M	2.1 G	**28.7**
EfficientViT	9.9 M	1.5 G	28.4
DCS-EfficientViT(Ours)	9.0 M	1.4 G	**29.0**

Table 3: Object detection performance with SSDLite.

4.4. Comparison to Weight Pruning Methods

We compare DCS-Transformer to the current state-of-the-art weight pruning methods for visual transformers in Section B.3 of the supplementary.

4.5. Ablation Study

Model	# Params	FLOPs	Top-1	IB Loss
MobileViT-S	5.6 M	1.6 G	78.4	-0.00432
DCS-MobileViT-S (w/o IB Loss)	4.8 M	1.2 G	79.9	-0.00475
DCS-MobileViT-S	4.8 M	1.2 G	**81.0**	**-0.05122**
MobileViT-XS	2.3 M	0.71 G	74.8	-0.00419
DCS-MobileViT-XS (w/o IB Loss)	2.0 M	0.54 G	75.7	-0.00466
DCS-MobileViT-XS	2.0 M	0.54 G	**76.8**	**-0.04918**
EfficientViT (r224)	9.1 M	0.52 G	79.4	-0.00451
DCS-EfficientViT (r224) (w/o IB Loss)	8.2 M	0.46 G	80.0	-0.00507
DCS-EfficientViT (r224)	8.2 M	0.46 G	**80.8**	**-0.06102**

Table 4: Ablation Study of DCS MobileViT-S. The models followed by "(w/o IB Loss)" are trained by regular cross-entropy without IB loss.

Study on the effects of DCS in reducing IB loss. We study how DCS helps in reducing the IB loss with three visual transformers, including MobilVit-S, MobileViT-XS, and EfficientViT (r224). We compare the performance of the vanilla visual transformers, their ablation models with channel selection but trained without IB Loss, and the corresponding DCS-Tranformer models. The results are shown in Table 4. It is observed from the results that while optimizing the regular cross-entropy loss can decrease the IB loss so that the IB principle, learning features more strongly correlated with class labels while decreasing their correlation with the input, is adhered, the IB loss can be decreased further to a considerable extent by optimizing the IB loss explicitly. In particular, our DCS-Transformer models improve the vanilla visual transformers and their ablation study models by a large margin in terms of both IB loss and top-1 accuracy.

Study on IB loss and its effect on generalization, and more visualization results are in Section C of the supplementary. Figure 8 illustrates the test loss and the IB loss during the training. It can be observed that the IB loss for all the models decreases starting from the 90-th epoch after the warm-up stage, and test loss of a model with IB loss optimized drops faster than that of the vanilla model. Figure 7 shows the heatmaps of different models generated by Grad-CAM. It can be observed that our DCS models focused more on the object for classification in the input images. The attention weights computed for some specific queries in the transformer blocks in both MobilViT-S and DCS-MobileViT-S are illustrated in Figure 6 of the supplementary.

5. Conclusion

In this paper, we propose a novel transformer block, Transformer with Differentiable Channel Selection, or DCS-Transformer. Channel selection for attention weights and channel selection for attention outputs are performed in a DCS-Transformer block. The training of a network with DCS-Transformer enjoys a reduction of IB loss with a novel and separable variational upper bound for the IB loss, rendering compact visual transformers with reduced FLOPs while enjoying comparable or even better prediction accuracy. We demonstrate the effectiveness of DCS-Transformer by replacing all the transformer blocks in several popular visual transformers with DCS-Transformer blocks. Extensive experiments on various tasks demonstrate the effectiveness of DCS-Transformer.

Acknowledgments

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-07-00. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security. This work is also partially supported by the 2023 Mayo Clinic and Arizona State University Alliance for Health Care Collaborative Research Seed Grant Program.

Impact Statement

This paper presents work whose goal is to advance the field of Deep Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bengio, Y., Léonard, N., and Courville, A. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *arXiv preprint arXiv:1308.3432*, 2013.

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., Feichtenhofer, C., and Hoffman, J. Token merging: Your vit but faster. *ICLR*, 2023.

Cai, H., Gan, C., and Han, S. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pp. 213–229. Springer, 2020.

Chavan, A., Shen, Z., Liu, Z., Liu, Z., Cheng, K.-T., and Xing, E. P. Vision transformer slimming: Multi-dimension searching in continuous optimization space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4931–4941, 2022.

Chen, C.-F., Fan, Q., and Panda, R. CrossViT: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021a.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Chen, M., Peng, H., Fu, J., and Ling, H. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12270–12280, 2021b.

Chen, T., Cheng, Y., Gan, Z., Yuan, L., Zhang, L., and Wang, Z. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 34:19974–19988, 2021c.

Chen, T., Cheng, Y., Gan, Z., Yuan, L., Zhang, L., and Wang, Z. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 34:19974–19988, 2021d.

Chen, Y., Dai, X., Chen, D., Liu, M., Dong, X., Yuan, L., and Liu, Z. Mobile-former: Bridging mobilenet and transformer. *arXiv preprint arXiv:2108.05895*, 2021e.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021a.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021b.

Fang, G., Ma, X., Song, M., Mi, M. B., and Wang, X. Depgraph: Towards any structural pruning. *CVPR*, 2023.

Gong, C., Wang, D., Li, M., Chen, X., Yan, Z., Tian, Y., qiang liu, and Chandra, V. NASVit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In *International Conference on Learning Representations*, 2022.

Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., and Douze, M. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12259–12269, 2021.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pp. 1135–1143, 2015.

Han, Y., Huang, G., Song, S., Yang, L., Wang, H., and Wang, Y. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7436–7456, 2021.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

He, W., Wu, M., Liang, M., and Lam, S.-K. Cap: Context-aware pruning for semantic segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 960–969, 2021.

Heo, B., Yun, S., Han, D., Chun, S., Choe, J., and Oh, S. J. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019.

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

Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pp. 620–640. Springer, 2022a.

Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European Conference on Computer Vision*, pp. 620–640. Springer, 2022b.

Lai, Q., Li, Y., Zeng, A., Liu, M., Sun, H., and Xu, Q. Information bottleneck approach to spatial attention learning. In Zhou, Z. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 779–785. ijcai.org, 2021.

Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., Wang, Y., and Ren, J. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., and Timofte, R. Swinir: Image restoration using swin transformer. *ICCV*, 2021.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Liu, X., Peng, H., Zheng, N., Yang, Y., Hu, H., and Yuan, Y. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14420–14430, 2023.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

Mehta, S. and Rastegari, M. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *ICLR*, 2022.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., and Hsieh, C.-J. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *arXiv preprint arXiv:2106.02034*, 2021.

Redmon, J. and Farhadi, A. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

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

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124020, 2019.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Song, H., Sun, D., Chun, S., Jampani, V., Han, D., Heo, B., Kim, W., and Yang, M.-H. Vidt: An efficient and effective fully transformer-based object detector. *ICLR*, 2022.

Sun, W., Jiang, Y., Zhang, S., and Liu, Y. Cascade pruning: Towards class imbalance in convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9860–9869, 2021.

Tan, M. and Le, Q. V. Mixconv: Mixed depthwise convolutional kernels. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, 2019.

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

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

Verelst, T. and Tuytelaars, T. Dynamic convolutions: Exploiting spatial sparsity for faster inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2320–2329, 2020.

Wang, Y., Xu, N., Chen, C., and Yang, Y. Adaptive cross-layer attention for image restoration. *arXiv preprint arXiv:2203.03619*, 2022a.

Wang, Z., Luo, H., Wang, P., Ding, F., Wang, F., and Li, H. Vtc-lfc: Vision transformer compression with low-frequency components. *Advances in Neural Information Processing Systems*, 35:13974–13988, 2022b.

Wu, K., Zhang, J., Peng, H., Liu, M., Xiao, B., Fu, J., and Yuan, L. Tinyvit: Fast pretraining distillation for small vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXI*, pp. 68–85. Springer, 2022.

Yu, F., Huang, K., Wang, M., Cheng, Y., Chu, W., and Cui, L. Width & depth pruning for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3143–3151, 2022a.

Yu, S., Chen, T., Shen, J., Yuan, H., Tan, J., Yang, S., Liu, J., and Wang, Z. Unified visual transformer compression. *ICLR*, 2022b.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F. E., Feng, J., and Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

Zheng, C., Zhang, K., Yang, Z., Tan, W., Xiao, J., Ren, Y., Pu, S., et al. Savit: Structure-aware vision transformer pruning via collaborative optimization. *Advances in Neural Information Processing Systems*, 35:9010–9023, 2022a.

Zheng, C., Zhang, K., Yang, Z., Tan, W., Xiao, J., Ren, Y., Pu, S., et al. Savit: Structure-aware vision transformer pruning via collaborative optimization. *Advances in Neural Information Processing Systems*, 35:9010–9023, 2022b.

Zhou, D., Yu, Z., Xie, E., Xiao, C., Anandkumar, A., Feng, J., and Alvarez, J. M. Understanding the robustness in vision transformers. In *International Conference on Machine Learning*, pp. 27378–27394. PMLR, 2022.

Zhou, Z., Yao, H., Guo, X., and Xu, Y. Rethinking the value of network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5867–5875, 2020.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *ICLR*, 2021.

A. Implementation Details

A.1. FLOPs Computation of DCS-EfficientViT

We introduce the details about the FLOPs cost of DCS-EfficientViT. For DCS-EfficientViT, we have $\texttt{cost}_j = (2\tilde{D}^2 + \tilde{D}) + (1 + \tilde{D}) \cdot HWC$. The first term, $2\tilde{D}^2 + \tilde{D}$, is the FLOPs of the linear layer at the end of the Lightweight Multi-Scale Attention block, $(1 + \tilde{D}) \cdot HWC$ is the FLOPs of the first convolution layer in the MBConv block after the lightweight Multi-Scale Attention (lightweight MSA) block. $\tilde{D}$ denotes the compressed input dimension of the linear layer at the end of the lightweight MSA block. Note that $\tilde{D}$ is compressed from concatenated features of $3 \cdot D$ dimensions, where D is the output dimension of the ReLU Global Attention and 3 is the number of branches in Lightweight Multi-Scale Attention block in EfficientViT.

A.2. Settings for Searching and Retraining

More Details about Retraining and Searching with DCS-Transformer. In the retraining phase, we train the searched network on ImageNet-1K. We also use AdamW optimizer and set β_1 and β_2 to 0.9 and 0.999. The retraining phase takes 300 epochs, whereas the first 90 epochs are the warm-up stage that only optimizes the cross-entropy loss. After the warm-up stage, we optimize the overall training loss function that minimizes the cross-entropy loss and the IB loss. All the training is performed on 4 NVIDIA V100 GPUs with an effective batch size of 512 images. Following previous works (Cai et al., 2023), we adopt the widely used and standard data augmentation techniques, including random scaling, random horizontal flip, and random cropping in training. The weight decay is set to 0.01. The learning rate is increased from 0.0002 to 0.002 linearly in the first 5 epochs and then annealed to 0.0002 using a cosine learning rate schedule in the following 300 epochs. The inference is performed on the exponential moving average of model weights. Different from works such as LeViT (Graham et al., 2021) and NASVIT (Gong et al., 2022), knowledge distillation is not used in the training of DCS models. Our results are compared exclusively with baseline models trained without knowledge distillation.

For transformer blocks in EfficientViT, we apply our method of channel selection for attention weights, illustrated in the green box in Figure 1, on the queries and keys for the ReLU Global Attention in every branch of the lightweight MSA module before applying ReLu and dot-product on them. We also apply our channel selection method for attention outputs to the features concatenated from different scale branches and prune the number of channels in the following linear layer.

B. Additional Experiments

B.1. DCS Transformers Trained with Knowledge Distillation

We provide additional results of visual transformers with DCS-Transformer blocks, which are trained by knowledge distillation. Existing efficient visual transformers, such as NASViT (Gong et al., 2022) and LeViT (Graham et al., 2021), incorporate knowledge distillation to improve their performance. They employ the outputs of large models as supervision during their own training. In this section, we adopt knowledge distillation in the retraining process of our models. To ensure fair comparisons, we follow the settings in NASViT and employ EfficientNet-B5 as the teacher model for knowledge distillation. The results are shown in Table 5, where KD denotes models trained with knowledge distillation. Furthermore, to compare with previous SOTA methods NASViT-A4 and NASViT-A5, we also evaluate DCS-EfficientViT on an input resolution of 288×288, which is denoted by $r288$ in the table. It is shown in the results that DCS-EfficientViT trained with both input resolutions outperforms NASViT in terms of higher accuracy while enjoying lower FLOPs. For example, with input resolution of 288×288, DCS-EfficientViT has a lower FLOPs (0.72G compared to 0.76G) while its accuracy is higher (83.1% compared to 81.8%).

Figure 4 illustrates the histogram of the entropy of the rows of the affinity matrices over randomly chosen 100 validation images of ImageNet-1k for all the four heads in the last transformer block in MobileViT-S and DCS-MobileViT-S. It can be observed that more rows of the affinity matrices for DCS-Transformer have lower entropy, indicating that the affinity matrices in the DCS-Transformer blocks are more informative.

B.2. Study on the Attention Weights learned by DCS

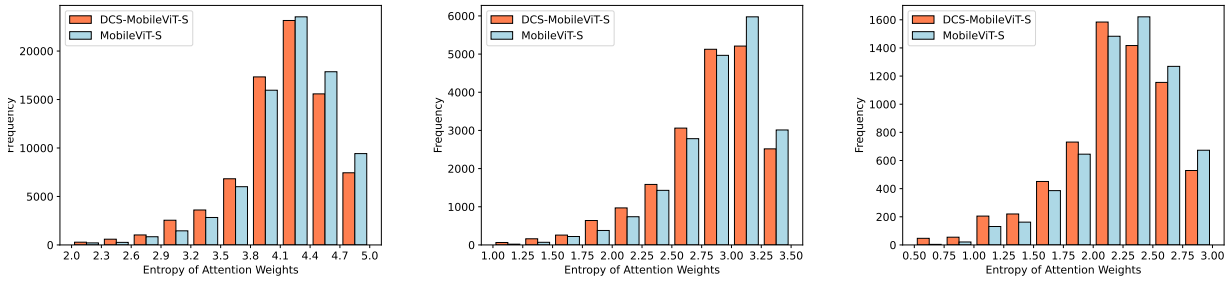
As explained in the introduction and Section 3.1, the channel selection for attention weights is expected to generate a more informative affinity matrix W. It is noted that every row of the affinity matrix $\mathbf{W}$ is a normalized nonnegative vector with all elements summing to 1, which represents the affinity scores of a certain query to each key. Figure 5 illustrates

Model	# Params	FLOPs	Top-1
LeViT-128 (KD) (Graham et al., 2021)	9.2 M	0.41 G	78.6
LeViT-192 (KD) (Graham et al., 2021)	11.0 M	0.67 G	80.0
LeViT-256 (KD) (Graham et al., 2021)	19.0 M	1.12 G	81.6
NASViT-A2 (r224, KD) (Gong et al., 2022)	16.5 M	0.42 G	80.5
NASViT-A3 (r256, KD) (Gong et al., 2022)	16.5 M	0.53 G	81.0
NASViT-A4 (r288, KD) (Gong et al., 2022)	16.5 M	0.59 G	81.4
NASViT-A5 (r288, KD) (Gong et al., 2022)	16.5 M	0.76 G	81.8
DCS-EfficientViT-B1 [r224] (Ours)	8.2 M	0.46 G	80.8
DCS-EfficientViT-B1 [r288] (Ours)	8.2 M	0.72 G	81.6
DCS-EfficientViT-B1 [r224, KD] (Ours)	8.2 M	0.46 G	**82.1**
DCS-EfficientViT-B1 [r288, KD] (Ours)	8.2 M	0.72 G	**83.1**

Table 5: Comprehensive Comparisons on ImageNet-1k validation set. KD denotes models trained with knowledge distillation.

Block	Block 1	Block 2	Block 3
N	196	49	16
D	36	48	60
r	34.2, 34.5, 34.7, 35.2	42.1, 42.7, 43.0, 43.2	46.9, 47.7, 50.0, 51.3

Table 6: The rank $r = \mathrm{rank}((Q \odot \mathcal{M})(K \odot \mathcal{M})^\top)$ for all the attention heads for each DCS-MobileViT-S block.



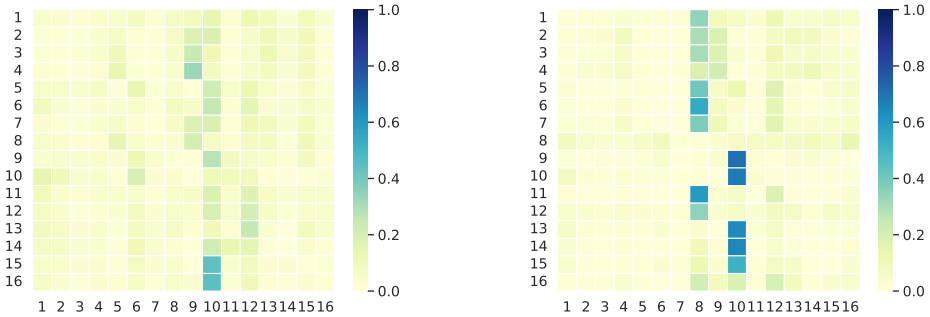
(a) Block 1 (b) Block 2 (c) Block 3

Figure 4: Comparisons on entropy for attention weights of query tokens from all attention heads for each of the three DCS-Transformer blocks in MobileVit-S and DCS-MobileVit-S. The results are accumulated from 100 randomly sampled images from the validation set of ImageNet-1k.

the affinity matrix W of the first head in the last transformer block of MobileViT-S and DCS-MobileViT-S trained on the ImageNet-1k dataset, using a particular image sampled from the training data as the input. More spiked patterns are observed for DCS-MobileViT-S, indicating that fewer keys contribute to the feature aggregation process for computing the attention outputs. As a result, channel selection on the attention outputs has less adverse effect on the performance of the resultant model. We also adopt entropy as a quantitative measure for the information in the affinity matrix W. We show the average entropy of all the rows of the matrix $\mathbf{W}$ for the first head in the last transformer block of MobileViT-S and DCS-MobileViT-S in Figure 5, and a lower average entropy of 1.84 compared to 2.50 indicates that the affinity matrix in a DCS-Transformer block is more informative than its counterpart. We also compare such average entropy of the affinity matrix over all the validation images of ImageNet-1k for each the four heads in the last transformer block in MobileViT-S and DCS-MobileViT-S, evidencing that DCS-Transformer renders more informative affinity: Head 1, Baseline 2.27 vs. DCS 2.09; Head 2, Baseline 2.18 vs. DCS 2.05; Head 3, Baseline 2.33 vs. DCS 2.15; Head 4, Baseline 2.24 vs. DCS 2.08.

B.3. Comparisons with Weight Pruning Methods for Visual Transformers

We compare DCS-Transformer to the current state-of-the-art weight pruning methods for visual transformers. The results are shown in Table. DCS-Transformer is compared to S^2ViTE (Chen et al., 2021d), SPViT (Kong et al., 2022b), and



(a) Attention weights of MobileViT-S. The average entropy of attention weights for all queries is 2.50.

(b) Attention weights of DCS-MobileViT-S. The average entropy of attention weights for all queries is 1.84.

Figure 5: Comparisons between attention weights of MobileViT-S and DCS-MobileViT-S. The attention weights are from the first head in the last transformer block of MobileViT-S and DCS-MobileViT-S.

Figure 6: Visualization of attention weights. The query patch positions for all examples are marked with red bounding boxes in the first row. We visualize the attention weights of the first head in the first transformer block in MobileViT-S (second row) and DCS-MobileViT-S (third row). The original input image in the shape of 256×256 corresponds to 16×16 non-overlapped patches for self-attention, where each patch corresponds to 16×16 pixels in the input image. We visualize the attention weights between the query patch and all 16×16 patches.

SAViT (Zheng et al., 2022b) on EfficientViT-B1 (r224). In the implementation of S^2ViTE, SPViT, and SAViT, the pruning

Model	# Params	FLOPs	Inference Time (ms/image)	Top-1
EfficientViT-B1 (r224) [4]	9.1 M	0.52 G	2.654	79.4
S^2ViTE-EfficientViT-B1 (r224) (Chen et al., 2021d)	8.2 M	0.47 G	2.438	79.0
SPViT-EfficientViT-B1 (r224) (Kong et al., 2022b)	9.2 M	0.49 G	2.451	79.3
SAViT-EfficientViT-B1 (r224) (Zheng et al., 2022b)	8.4 M	0.47 G	2.435	79.2
DCS-EfficientViT-B1 (r224)	8.2 M	0.46 G	2.427	80.8

Table 7: Comparisons with weight pruning methods on EfficientViT-B1 (r224).

is performed on the ImageNet training data. After obtaining the pruned networks, we fine-tune them using the same setting as in (Cai et al., 2023). It can be observed from the results in Table 7 that DCS-EfficientViT-B1 outperforms all pruned models by at least 1.5% in top-1 accuracy with even fewer parameters and FLOPs.

B.4. Attention Weights Visualization

To further study how DCS impacts the attention mechanism, we visualize the attention weights computed for some specific queries in the transformer blocks in both MobilViT-S and DCS-MobileViT-S. The visualization heatmaps are shown in Figure 6.

C. Detailed Ablation Study

C.1. Study on Effects of Channel Selection in DCS

To demonstrate the strength of the channel selection in our DCS, we introduce an ablation model named DCS-QK-Mask where different binary decision masks are used in Q and K when computing the attention weights W, that is, $W = \sigma\left((Q \odot \mathcal{M}_1)(K \odot \mathcal{M}_2)^\top\right)$ where $\mathcal{M}_1$ and $\mathcal{M}_2$ are learned separately in the search phase. As shown in Table 8, the DCS-MobileViT still outperforms DCS-QK-Mask using MobileViT as the backbone, and this is because different decision masks introduce unnecessary complexity in the self-attention module.

Model	# Params	FLOPs	Top-1	IB Loss
MobileViT-S	5.6 M	1.6 G	78.4	-0.00432
DCS-Arch1 (w/o IB Loss)	5.6 M	1.6 G	79.4	-0.00459
DCS-Arch2 (w/o IB Loss)	4.8 M	1.2 G	78.2	-0.00446
DCS-QK-Mask (w/o IB Loss)	4.8 M	1.2 G	79.7	-0.00462
DCS-MobileViT-S (w/o IB Loss)	4.8 M	1.2 G	79.9	-0.00475
DCS-MobileViT-S	4.8 M	1.2 G	**81.0**	-0.05122

Table 8: Study on the channel selection mechanism in DCS MobileViT-S. The models followed by "(w/o IB Loss)" are trained by regular cross-entropy without IB loss.

In order to demonstrate the effects of the two types of channel selection, channel selection for attention weights and channel selection for attention outputs described in Section 3.1 and Section 3.2 respectively, we design two ablation DCS-Transformer modules, which are DCS-Arch1 and DCS-Arch2. DCS-Arch1 only has channel selection for attention weights, and DCS-Arch2 only has channel selection for attention outputs. We replace all the transformer blocks in MobileViT-S with one of these two ablation DCS-Transformer modules, respectively, and report their performance on the ImageNet-1k dataset in Table 8. It can be observed that DCS-Arch2 reduces the FLOPs of the model by pruning the channels of the MLP features at a cost of loss in accuracy. In contrast, the proposed channel selection for attention weights significantly boosts the accuracy of DCS-MobileViT-S compared to that of the original MobileViT-S by 2.6% while enjoying lower FLOPs.

C.2. Study on the effects of DCS in Reducing IB loss

In this section, we study how DCS helps in reducing the IB loss with three visual transformers, including MobilVit-S, MobileViT-XS, and EfficientViT (r224). We compare the performance of the vanilla visual transformers, their ablation models with channel selection but trained without IB Loss, and the corresponding DCS-Tranformer models. The results

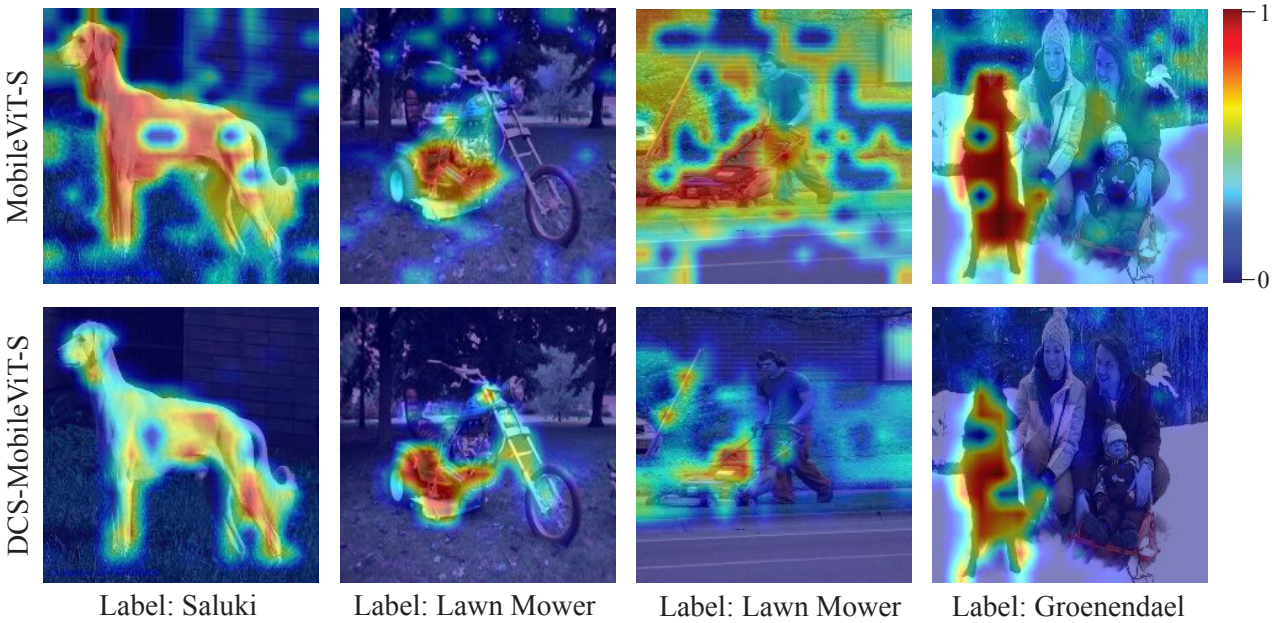


Figure 7: Grad-CAM visualization results. The figures in the first row are the visualization results of MobileViT-S, and the figures in the second row are the visualization results of DCS-MobileViT-S.

are shown in Table 4. It is observed from the results that while optimizing the regular cross-entropy loss can decrease the IB loss so that the IB principle, learning features more strongly correlated with class labels while decreasing their correlation with the input, is adhered, the IB loss can be decreased further to a considerable extent by optimizing the IB loss explicitly. In particular, our DCS-Transformer models improve the vanilla visual transformers and their ablation study models by a large margin in terms of both IB loss and top-1 accuracy.

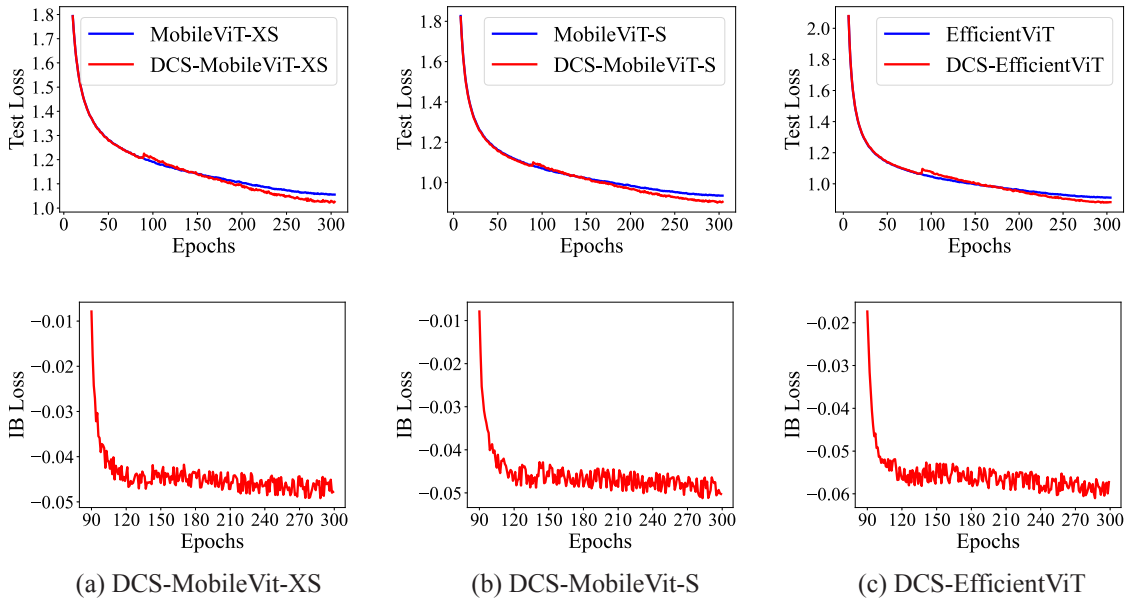


Figure 8: Comparison: attention weights of MobileViT-S vs. DCS-MobileViT-S. The attention weights are from the first head in the last transformer block of MobileViT-S and DCS-MobileViT-S.

C.3. Study on IB Loss and Its Effect on Generalization

Figure 8 illustrates the test loss and the IB loss during the training for DCS-MobileVit-XS, DCS-MobileVit-S, and DCS-EfficientViT. It can be observed that the IB loss for all the models decreases starting from the 90-th epoch after the warm-up stage, and test loss of a model with IB loss optimized drops faster than that of the vanilla model.

C.4. Visualization Results

In this section, we first apply the Grad-CAM (Selvaraju et al., 2017) visualization tool to study which parts in the input images are responsible for the predictions of the baseline models and models. Figure 7 shows the heatmaps of different models generated by Grad-CAM. It can be observed that our DCS models focused more on the object for classification in the input images. In contrast, the activation maps generated by the baseline models also have nonnegligible values in the background.

To study how DCS impacts the attention mechanism, we visualize the attention weights computed for some specific queries in the transformer blocks in both MobilViT-S and DCS-MobileViT-S in Figure 6 of the supplementary.

D. Proof of Theorem 3.1

Lemma D.1.

$$I(\tilde{X}, X) \leq \frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\phi(\tilde{X}_i, a)\phi(X_i, b)\log\phi(X_i, b) - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{b=1}^{B}\phi(X_i, b)\log\phi(X_j, b) \tag{3}$$

Proof. By the log sum inequality, we have

$$\begin{aligned}
I(\tilde{X}, X) &= \sum_{a=1}^{A}\sum_{b=1}^{B}\Pr\left[\tilde{X} \in a, X \in b\right]\log\frac{\Pr\left[\tilde{X} \in a, X \in b\right]}{\Pr\left[\tilde{X} \in a\right]\Pr\left[X \in b\right]} \\
&\leq \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\phi(\tilde{X}_i, a)\phi(X_i, b)\left(\log\left(\phi(\tilde{X}_i, a)\phi(X_i, b)\right) - \log\left(\phi(\tilde{X}_i, a)\phi(X_j, b)\right)\right) \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\phi(\tilde{X}_i, a)\phi(X_i, b)\log\phi(X_i, b) - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\phi(\tilde{X}_i, a)\phi(X_i, b)\log\phi(X_j, b) \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\phi(\tilde{X}_i, a)\phi(X_i, b)\log\phi(X_i, b) - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{a=1}^{A}\sum_{b=1}^{B}\phi(\tilde{X}_i, a)\phi(X_i, b)\log\phi(X_j, b).
\end{aligned} \tag{4}$$

□

Lemma D.2.

$$I(\tilde{X}, Y) \geq \frac{1}{n}\sum_{a=1}^{A}\sum_{y=1}^{C}\sum_{i=1}^{n}\phi(\tilde{X}_i, a)\mathbb{1}_{\{y_i=y\}}\log Q(\tilde{X} \in a|Y = y) \tag{5}$$

Proof. Let $Q(\tilde{X}|Y)$ be a variational distribution. We have

$$
\begin{aligned}
& I(\tilde{X}, Y) \\
&= \sum_{a=1}^{A} \sum_{y=1}^{C} \Pr\left[\tilde{X} \in a, Y = y\right] \log \frac{\Pr\left[\tilde{X} \in a, Y = y\right]}{\Pr\left[\tilde{X} \in a\right] \Pr\left[Y = y\right]} \\
&= \sum_{a=1}^{A} \sum_{y=1}^{C} \Pr\left[\tilde{X} \in a, Y = y\right] \log \frac{\Pr\left[\tilde{X} \in a | Y = y\right] Q(\tilde{X} \in a | Y = y)}{\Pr\left[\tilde{X} \in a\right] Q(\tilde{X} \in a | Y = y)} \\
&\geq \sum_{a=1}^{A} \sum_{y=1}^{C} \Pr\left[\tilde{X} \in a, Y = y\right] \log \frac{\Pr\left[\tilde{X} \in a | Y = y\right]}{Q(\tilde{X} \in a | Y = y)} + \sum_{a=1}^{A} \sum_{y=1}^{C} \Pr\left[\tilde{X} \in a, Y = y\right] \log \frac{Q(\tilde{X} \in a | Y = y)}{\Pr\left[\tilde{X} \in a\right]} \\
&= \mathrm{KL}\left(P(\tilde{X}|Y) \middle\| Q(\tilde{X}|Y)\right) + \sum_{a=1}^{A} \sum_{y=1}^{C} \Pr\left[\tilde{X} \in a, Y = y\right] \log \frac{Q(\tilde{X} \in a | Y = y)}{\Pr\left[\tilde{X} \in a\right]} \\
&\geq \sum_{a=1}^{A} \sum_{y=1}^{C} \Pr\left[\tilde{X} \in a, Y = y\right] \log \frac{Q(\tilde{X} \in a | Y = y)}{\Pr\left[\tilde{X} \in a\right]} \\
&= \sum_{a=1}^{A} \sum_{y=1}^{C} \Pr\left[\tilde{X} \in a, Y = y\right] \log Q(\tilde{X} \in a | Y = y) + H\left(P(\tilde{X})\right) \\
&\geq \sum_{a=1}^{A} \sum_{y=1}^{C} \Pr\left[\tilde{X} \in a, Y = y\right] \log Q(\tilde{X} \in a | Y = y) \\
&\geq \frac{1}{n} \sum_{a=1}^{A} \sum_{y=1}^{C} \sum_{i=1}^{n} \phi(\tilde{X}_i, a) \mathbb{1}_{\{y_i = y\}} \log Q(\tilde{X} \in a | Y = y). \qquad (6)
\end{aligned}
$$

□

Computation of $Q^{(t)}(\tilde{\mathbf{X}}|Y)$. The variational distribution $Q^{(t)}(\tilde{\mathbf{X}}|Y)$ can be computed by

$$
\begin{aligned}
Q^{(t)}(\tilde{X} \in a | Y = y) &= \Pr\left[\tilde{X} \in a | Y = y\right] \\
&= \frac{\sum_{i=1}^{n} \phi(\tilde{X}_i, a) \mathbb{1}_{\{y_i = y\}}}{\sum_{i=1}^{n} \mathbb{1}_{\{y_i = y\}}}. \qquad (7)
\end{aligned}
$$