COMPRESSION OF VISION TRANSFORMER BY REDUCTION OF KERNEL COMPLEXITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-attention and transformer architectures have become foundational components in modern deep learning. Recent efforts have integrated transformer blocks into compact neural architectures for computer vision, giving rise to various efficient vision transformers. In this work, we introduce Transformer with Kernel Complexity Reduction, or KCR-Transformer, a compact transformer block equipped with differentiable channel selection, guided by a novel and sharp theoretical generalization bound. To reduce the substantial computational cost of the MLP layers, the KCR-Transformer performs channel selection on the outputs of its self-attention layer. Furthermore, we provide a rigorous theoretical analysis establishing a tight generalization bound for networks equipped with KCR-Transformer blocks. Leveraging such strong theoretical results, the channel pruning by KCR-Transformer is conducted in a generalization-aware manner, ensuring that the resulting network retains a provably small generalization error. Our KCR-Transformer is compatible with many popular and compact transformer networks, such as ViT and Swin, and it reduces the FLOPs of the vision transformers while maintaining or even improving the prediction accuracy. In the experiments, we replace all the transformer blocks in the vision transformers with KCR-Transformer blocks, leading to KCR-Transformer networks with different backbones. The resulting KCR-Transformers achieve superior performance on various computer vision tasks, achieving even better performance than the original models with even less FLOPs and parameters. The code of the KCR-Transformer is available at https://anonymous.4open.science/status/KCR-Transformer.

1 INTRODUCTION

Vision transformers have demonstrated promising performance on a variety of computer vision tasks (Yuan et al., 2021; Dosovitskiy et al., 2021; Liu et al., 2021a; Zhu et al., 2021; Liang et al., 2021) and multi-modal learning tasks (Liu et al., 2023a; Singh et al., 2022). However, the superior performance of vision transformers comes at the cost of substantial computational overhead (Dosovitskiy et al., 2021; Touvron et al., 2021). To reduce the computational costs of the vision transformers, various model compression methods have been developed, including knowledge distillation (Zhao et al., 2023; Yang et al., 2023), quantization (Li et al., 2022a; Lin et al., 2022; Li et al., 2023; Liu et al., 2021b), neural architecture search (Gong et al., 2022; Su et al., 2022), and pruning (Chen et al., 2021b; Chavan et al., 2022; Zheng et al., 2022; Yu et al., 2022b;a; Rao et al., 2021; Kong et al., 2022; Wang et al., 2022; Bolya et al., 2023; Bonnaerens & Dambre, 2023; Kim et al., 2024). The pruning methods, which typically involve a pruning stage to remove redundant parameters and a fine-tuning stage to recover performance, have been shown to be particularly effective due to the substantial parameter redundancy in ViT models (Chen et al., 2021b; Chavan et al., 2022; Zheng et al., 2022; Yu et al., 2022b;a; Rao et al., 2021; Kong et al., 2022; Wang et al., 2022; Bolya et al., 2023; Bonnaerens & Dambre, 2023; Kim et al., 2024). Despite substantial progress, the conventional model compression methods primarily focus on identifying optimal compression strategies through direct performance–efficiency trade-offs, guided by empirical heuristics rather than principled theoretical foundations.

More recently, inspired by advances in the theoretical understanding of deep neural networks (DNNs) through the lens of kernel methods, such as the Neural Tangent Kernel (NTK) (Jacot et al., 2018), the kernel-based compression method has emerged as a principled alternative. The kernel-based methods aim to preserve the generalization capability of the compressed model by ensuring that the compressed model retains the training dynamics, convergence behavior, and inductive biases of the original network through spectral alignment of the NTK (Wei et al., 2023; Chen et al., 2021c; Mok et al., 2022; Wang et al., 2023; Rachwan et al., 2022). For instance, NTK-SAP (Wang et al., 2023) and Early-Lottery (Rachwan et al., 2022) leverage spectral preservation of the NTK during pruning to maintain the eigenspectrum, thereby preserving the generalization characteristics of the NTK of the original DNNs. Despite recent progress, a substantial gap persists between theory and practice in enhancing the generalization capability of compressed DNNs through NTK-based kernel learning. Existing theoretical frameworks, particularly those grounded in NTK analyses, are predominantly limited to over-parameterized DNNs (Cao & Gu, 2019; Arora et al., 2019; Ghorbani et al., 2021), rendering them unsuited for modern architectures such as vision transformers with finite-width and diverse network architectures. Moreover, the linearized nature of the NTK regime inherently fails to model the dynamically evolving kernel characteristic of realistic training dynamics, thereby limiting its applicability to compressed models utilizing large-scale training for compelling performance on real-world tasks (Nichani et al., 2022; Damian et al., 2022; Takakura & Suzuki, 2024). To address this challenge, we first provide a theoretical analysis that establishes tight generalization upper and lower bounds in Theorem 3.1. Both an upper and lower bound of the expected loss, referred to as KCR upper and lower bounds, are established based on the training loss and the kernel complexity (KC) of the kernel gram matrix computed over the training data. The tightness of the KCR upper and lower bounds is studied in Section 4.6. *In contrast to the current NTK-based methods with their feature learning capability limited by the linear region of NTK, the KC measures the complexity of the dynamically evolving kernel formed by the DNN during the training process, accommodating the rich feature learning capability of DNNs*. Since the training loss is usually optimized to a small value by training the DNN, the KCR upper and lower bounds in Theorem 3.1 can be tight and close to the expected loss if the KC is small. However, the computation of the KC involves the costly computation of the eigenvalues of a potentially large-scale gram matrix. To mitigate this issue, we introduce an approximate TNN through the efficient Nyström method (Kumar et al., 2012), and the KC is reduced by the reduction of the approximate TNN. The approximate TNN is computed efficiently as a regularization term to the regular cross-entropy training loss. Since the approximate TNN is separable, it can be optimized by the standard SGD-based optimization algorithms. Based on the reduction of the KC, we propose a novel vision transformer termed the Transformer with Kernel Complexity Reduction, or KCR-Transformer. The training of the KCR-Transformer involves a search stage and a retrain stage. The channel selection in the input/output features of the MLP in the transformer block of the KCR-Transformer is performed in the search stage to obtain a compressed network architecture with reduced computation costs. To guarantee the generalization capability of the compressed model, the compressed network is retrained with the approximate TNN as a regularization term to reduce the KC, leading to enhanced prediction accuracy. It is verified through extensive experiments that the reduction of the approximate TNN effectively reduces the KC.

Contributions. The contributions of this paper are presented as follows.

First, we present a compact transformer block termed Transformer with Kernel Complexity Reduction, or KCR-Transformer. By selecting the channels in the input of the MLP layers via channel pruning, KCR-Transformer blocks effectively reduce the computational costs of the vision transformer. KCR-Transformer blocks can be used to replace all the transformer blocks in many popular vision transformers, rendering compact KCR-Transformer networks with comparable or even better performance. The effectiveness of KCR-Transformer is evidenced by replacing all the transformer blocks with KCR-Transformer blocks in popular vision transformers, such as ViT (Dosovitskiy et al., 2021), and Swin (Liu et al., 2021a), rendering compact models with competitive performance. Experimental results show that KCR-Transformer not only reduces the parameter size and FLOPs but also outperforms the original models on tasks including image classification, object detection, instance segmentation, and visual question answering.

Second, we provide a theoretical analysis showing tight generalization upper and lower bounds for the KCR-Transformer network. With such strong theoretical results in Theorem 3.1, the channel pruning by KCR-Transformer is performed in a *generalization-aware* manner. That is, the channel pruning of KCR provably keeps a small generalization error bound for the DNN with KCR-Transformer blocks, effectively guaranteeing the generalization capability of the DNN after channel pruning. This goal is achieved through the reduction of the KC of the DNN with KCR-Transformer blocks. Since the KC involves the computation of the eigenvalues of a potentially large gram matrix, we introduce an approximate truncated nuclear norm (TNN) through the Nyström method (Kumar et al., 2012), which is computed efficiently as a regularization term to the regular cross-entropy training loss and separable, so that it can be optimized by the standard SGD-based optimization algorithms. The reduction of the approximate TNN effectively reduces the KC, leading to the superior prediction accuracy of the compressed vision transformers by KCR-Transformer. Furthermore, unlike existing NTK-based compression methods constrained by NTK's linear regime, our KC captures the complexity of the evolving DNN kernel, enabling its rich feature learning capacity.

We note that training KCR-Transformer networks with the KCR regularization is efficient and stable with respect to the regularization weight, as evidenced in Section D.5 of the appendix. This paper is organized as follows. The related works in efficient vision transformers and compression of vision transformers are discussed in Section 2. The formulation of KCR-Transformer with our theoretical results is detailed in Section 3. The effectiveness of KCR-Transformer is demonstrated in Section 4 for image classification, dense prediction tasks, and multi-modal learning tasks. Throughout this paper, we use $a \lesssim b$ to denote $a \le Cb$ if there exists such a positive constant C, and $a = \Theta(b)$ indicates that $a \lesssim b$ and $b \lesssim a$. $[n]$ denotes all the natural numbers between 1 and n inclusively.

2 Related Works

2.1 Efficient Vision Transformers and Compression of Vision Transformers

To mitigate the issue of substantial computational overhead of the vision transformers, sparse attention mechanisms have been introduced to reduce computational demands (Zhu et al., 2021; Yuan et al., 2021; Papa et al., 2024), while other efforts integrate convolutional operations into the transformer architecture (Cai et al., 2023; Mehta & Rastegari, 2022; Yuan et al., 2021; Bravo-Ortiz et al., 2024). Additional gains in efficiency have been realized through Neural Architecture Search (Chen et al., 2021a; Gong et al., 2022; Wei et al., 2024) and Knowledge Distillation (Graham et al., 2021; Radosavovic et al., 2020; Gong et al., 2022; Yang et al., 2024), which aim to maintain accuracy with reduced computational resources. To further compress vision transformers, pruning techniques have been extensively explored. Channel pruning aims to eliminate redundant attention heads and channels (Chen et al., 2021b; Chavan et al., 2022; Zheng et al., 2022; Xu et al., 2024; Ahmed et al., 2025). Block pruning reduces the depth and width of models by removing entire transformer blocks (Yu et al., 2022b;a; Liu et al., 2024a). Token pruning techniques improve efficiency by adaptively discarding, merging, or filtering less informative tokens (Rao et al., 2021; Kong et al., 2022; Bolya et al., 2023; Wang et al., 2022; Liu et al., 2024b; Mao et al., 2025).

2.2 Related Works about Kernel Methods for Deep Neural Networks (DNNs)

Kernel methods have offered a principled view for analyzing the training dynamics, generalization properties, and architectural components of DNNs. One of the most prominent lines of work centers on the neural tangent kernel (NTK) (Jacot et al., 2018). Subsequent studies have extended NTK theory to better capture realistic scenarios, including finite-width settings (Seleznova & Kutyniok, 2022), deep narrow networks (Lee et al., 2022), and the empirical evolution of the NTK during training (Fort et al., 2020). Following these, researchers have also studied the limitations of purely kernel-based theories (Woodworth et al., 2020; Barrett & Dherin, 2021). Recent works have examined kernel-based interpretations of feature learning and generalization, revealing how hierarchical or implicit kernel structures emerge within deep models (Montavon et al., 2011; Belkin et al., 2018; Xiao et al., 2020; Canatar & Pehlevan, 2022; Deng et al., 2022). Building on these theoretical foundations, recent efforts propose reproducing kernel Hilbert space (RKHS) representations and operator-theoretic formulations as a basis for deep learning (Hashimoto et al., 2023), and develop hierarchical kernels tailored for representation learning (Huang et al., 2023). Beyond theoretical analysis, the study of kernels has also inspired reinterpretations and enhancements of transformer architectures. Several studies formulate self-attention as a kernel operation (Song et al., 2021; Chen et al., 2023). Others leverage spectral or integral transforms grounded in kernel theory (Nguyen et al., 2022; 2023). Positional encoding has also benefited from this perspective, with kernelized relative embeddings proposed for improved sequence extrapolation (Chi et al., 2022). Efficient attention variant, Performer (Choromanski et al., 2021), exploits kernel approximations to achieve linear complexity while maintaining expressiveness. Additionally, kernel-based models have been used to improve calibration in transformers via sparse Gaussian processes (Chen & Li, 2023).

Kernel-Based Model Compression Methods. Building upon these insights, kernel-based methods, especially those centered on the NTK, provide a complementary theoretical framework for analyzing and guiding model compression. The NTK-Comp framework (Gu et al., 2022) investigates pruning in wide multilayer perceptrons under Gaussian input assumptions and introduces quantization techniques that preserve the NTK spectrum within linear layers. MLP-Fusion (Wei et al., 2023) advances LLM compression by clustering neurons to jointly approximate functional outputs and NTK similarity. NTK-based metrics have also enabled training-free architecture search (Chen et al., 2021c), and facilitated early-stage performance prediction in neural architecture search (Mok et al., 2022), though their predictive power may diminish in regimes dominated by highly non-linear dynamics. In addition, methods, such as NTK-SAP (Wang et al., 2023) and Early-Lottery (Rachwan et al., 2022), further highlight the importance of preserving NTK spectral properties during pruning, emphasizing spectral alignment as critical for maintaining stable training dynamics. Nonetheless, the core limitation of existing NTK-based compression methods lies in their dependence on static or "lazy" training regimes, limiting their applicability to models with dynamically evolving representations.

3 Formulation

In this section, we introduce the KCR-Transformer, a compact transformer block designed to reduce the computational overhead of vision transformers through differentiable channel pruning in the MLP layers. To guide this pruning in a theoretically grounded manner, we present a novel generalization bound based on the kernel complexity (KC) of the network, and introduce the training algorithm of the network with KCR-Transformer for minimizing the upper bound.

3.1 Channel Selection for Attention Outputs

The vision transformer blocks usually apply a series of MLP layers to the output of the multi-head self-attention, which incurs substantial computation costs. To improve the efficiency of the vision transformer block, we propose pruning the channels in the attention outputs, thereby reducing the computation cost of the MLP layers, leading to the compact KCR-Transformer. To this end, 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 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 vision transformers have the same input and output dimensions, 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 is then changed to $\tilde{D} = \sum_{i=1}^{D} g_i$.

3.2 Generalization Bound by Approximate Kernel Complexity Loss and Its Optimization

Given the training data $\{\mathbf{x}_i, y_i\}_{i=1}^n$ where $\mathbf{x}_i$ is the i-th input training feature, $\mathbf{y}_i \in \mathbb{R}^C$ is the corresponding one-hot vector as the class label of $\mathbf{x}_i$ and C is the number of classes. We denote the label matrix as $\mathbf{Y} \in \mathbb{R}^{n \times C}$ where the i-th row is $\mathbf{y}_i$ for all $i \in [n]$. Let $\mathbf{F} \in \mathbb{R}^{n \times d}$ be the features extracted on the entire training data set, where $g(\cdot) \in \mathbb{R}^d$ denotes the mapping function of a DNN, such as ViT (Dosovitskiy et al., 2021), and d is the output dimension of the DNN before the final softmax layer for classification. We remark that almost all the DNNs use a linear layer to generate the output of the network for discriminative learning tasks, so that the mapping function of a DNN can be formulated as $g(\cdot) = g(\mathcal{W}, \cdot) = F(\mathcal{W}_2, \cdot)\mathcal{W}_1$, where $\mathcal{W}_1 \in \mathbb{R}^{d \times C}$ contains the weights in the final linear layer of the DNN where d is the hidden dimension, $F(\mathcal{W}_2, \cdot) \in \mathbb{R}^d$ represents the feature extraction backbone of the network before the final linear layer, and $\mathcal{W}_2$ are the weights of the backbone F, with $\mathcal{W} = \{\mathcal{W}_1, \mathcal{W}_2\}$. The DNN is denoted as $\mathrm{NN}_{\mathcal{W}}(\cdot)$. It is noted that such a formulation does not impose any limitation on the feature backbone F so as to admit a broad class of DNNs with various architectures for real-world vision discriminative tasks. We define a positive definite kernel for the DNN as $K(\mathbf{x}, \mathbf{x}') = F(\mathbf{x})^\top F(\mathbf{x}')$, where $\mathcal{X}$ is the input domain of the DNN and $F(\mathcal{W}, \cdot)$ is abbreviated into $F(\cdot)$. Compared to existing NTK-based methods (Wang et al., 2023; Wei et al., 2023; Gu et al., 2022) where a static NTK is used as the kernel, our kernel K is dynamic with the learned feature backbone F during the training process of the DNN. Let $\mathbf{F}_i = F(\mathcal{W}_2, \mathbf{x}_i)$ be the learned representation for the i-th training data, and $\mathbf{F} \in \mathbb{R}^{n \times d}$ is the matrix of all the learned representations over the training data. Then the gram matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ of the kernel K over the training data is calculated by $\mathbf{K} = \mathbf{F}\mathbf{F}^\top \in \mathbb{R}^{n \times n}$, and the eigenvalues of $\mathbf{K}_n := \mathbf{K}/n$ are $\widehat{\lambda}_1 \ge \widehat{\lambda}_2 \ldots \ge \widehat{\lambda}_{r_0} \ge \widehat{\lambda}_{r_0+1} = \ldots = \widehat{\lambda}_n = 0$ with $r_0 = \min\{n, d\}$, since $\operatorname{rank}(\mathbf{K}) \le r_0$.

Suppose the input feature $\mathbf{x}$ and its class label vector $\mathbf{y}$ follow an unknown joint distribution P, then the expected risk of the DNN is defined as $L_{\mathcal{D}}(\mathrm{NN}_{\mathcal{W}}) = \mathbb{E}_{(\mathbf{x},\mathbf{y}) \in P}\left[\|\mathrm{NN}_{\mathcal{W}}(\mathbf{x}) - \mathbf{y}\|_2^2\right]$, which also represents the generalization error of the DNN. The following theorem, Theorem 3.1, based on the local complexity of the function class of the DNN feature extraction backbones and rooted in the well-established local Rademacher complexity literature (Bartlett et al., 2005; Koltchinskii, 2006; Mendelson, 2002), gives sharp upper and lower bounds for the generalization error of the DNN. Theorem 3.1 uses the kernel complexity (KC) of the dynamic kernel K over the training data as $\mathrm{KC}(\mathbf{K}) := \min\limits_{h \in [0, r_0]} \left(\frac{h}{n} + \sqrt{\frac{1}{n} \sum\limits_{i=h+1}^{r_0} \widehat{\lambda}_i}\right)$.

Theorem 3.1. *Let K be the dynamic kernel after a particular optimization epoch by GD or SGD. Then for every $x > 0$, with probability at least $1 - \exp(-x)$, we have*

$$\underbrace{\mathbb{E}_{P_n}\left[\|\mathrm{NN}_{\mathcal{W}}(\mathbf{x}) - \mathbf{y}\|_2^2\right] - \mathrm{KC}(\mathbf{K}) - \frac{x}{n}}_{\text{KCR Lower Bound}} \lesssim L_{\mathcal{D}}(\mathrm{NN}_{\mathcal{W}}) \lesssim \underbrace{\mathbb{E}_{P_n}\left[\|\mathrm{NN}_{\mathcal{W}}(\mathbf{x}) - \mathbf{y}\|_2^2\right] + \mathrm{KC}(\mathbf{K}) + \frac{x}{n}}_{\text{KCR Upper Bound}}, \tag{1}$$

where $\mathbb{E}_{P_n}\left[\|\mathrm{NN}_{\mathcal{W}}(\mathbf{x}) - \mathbf{y}\|_2^2\right] = 1/n \cdot \sum_{i=1}^n \|\mathrm{NN}_{\mathcal{W}}(\mathbf{x}_i) - \mathbf{y}_i\|_2^2$ is the empirical loss on the training data.

The proof of this theorem is deferred to Section A of the appendix. The empirical loss $\mathbb{E}_{P_n}\left[\|\mathrm{NN}_{\mathcal{W}}(\mathbf{x}) - \mathbf{y}\|_2^2\right]$ is usually optimized to a small value by training the network $\mathrm{NN}_{\mathcal{W}}$, so it can be observed from (1) that the KCR upper and lower bounds for $L_{\mathcal{D}}(\mathrm{NN}_{\mathcal{W}})$ can be tight and close to $L_{\mathcal{D}}(\mathrm{NN}_{\mathcal{W}})$ if $\mathrm{KC}(\mathbf{K})$ is small, guaranteeing the generalization capability of the DNN. To reduce the KC, we introduce the truncated nuclear norm (TNN) of $\mathbf{K}$ denoted as $\|\mathbf{K}\|_r := \sum_{i=r+1}^{r_0} \widehat{\lambda}_i$ where $r \in [0\colon r_0]$, with $r_0 = \gamma_0 \min\{n, d\}$. γ_0 is to be chosen by cross-validation. It can be verified that a smaller TNN $\|\mathbf{K}\|_r$ leads to a smaller KC, $\mathrm{KC}(\mathbf{K})$. Since the computation of the TNN involves the computation of the eigenvalues of the potentially large-scale gram matrix $\mathbf{K}$ over large-scale training data, we then describe below how to efficiently and effectively approximate the TNN. We first compute the approximate top-r_0 eigenvectors of $\mathbf{K}_n$, $\tilde{\mathbf{U}}^{(r_0)}$, by the Nyström method (Kumar et al., 2012). Here $\mathbf{A}^{(r)}$ denotes a submatrix of $\mathbf{A}$ formed by its top r columns.

Efficient Computation of the Top-r_0 Eigenvectors of K. To approximate the top-r_0 eigenvectors, $\mathbf{U}^{(r_0)} \in \mathbb{R}^{n \times r_0}$, of the gram matrix $\mathbf{K}$ using the Nyström method (Kumar et al., 2012), we first sample m landmark points from the training set, indexed by $\mathcal{I} \subset [n]$ with $|\mathcal{I}| = m \ll n$. Let $\mathbf{F}_{\mathcal{I}} \in \mathbb{R}^{m \times d}$ be the features corresponding to the landmark set. We define $\mathbf{C} = \mathbf{F}\mathbf{F}_{\mathcal{I}}^\top \in \mathbb{R}^{n \times m}$ as the cross-covariance matrix and $\mathcal{W} = \mathbf{F}_{\mathcal{I}}\mathbf{F}_{\mathcal{I}}^\top \in \mathbb{R}^{m \times m}$ as the gram matrix on the landmarks. Next, we compute the top-r_0 eigen-decomposition of $\mathcal{W}$ as $\mathcal{W} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$, where $\mathbf{Q} \in \mathbb{R}^{m \times r_0}$ contains the top-r_0 eigenvectors and $\mathbf{\Lambda} \in \mathbb{R}^{r_0 \times r_0}$ is the diagonal matrix of corresponding eigenvalues. The Nyström approximation of $\mathbf{K}$ is then given by $\tilde{\mathbf{K}} = \mathbf{C}\mathcal{W}^\dagger\mathbf{C}^\top$, and the approximate top-r_0 eigenvectors are computed as $\tilde{\mathbf{U}}^{(r_0)} = \mathbf{C}\mathbf{Q}\mathbf{\Lambda}^{-1/2} \in \mathbb{R}^{n \times r_0}$, which serves as an efficient approximation to $\mathbf{U}^{(r_0)}$ with significantly reduced computational cost.

We let $\mathbf{U}_r = \tilde{\mathbf{U}}^{(r)}$, then the sum of the top-r eigenvalues of $\mathbf{K}_n$ is approximated by $\operatorname{tr}\left({\mathbf{U}_r}^\top \mathbf{K}_n \mathbf{U}_r\right)$. Since $\operatorname{tr}(\mathbf{K}_n) = \sum_{i=1}^n K(\mathbf{x}_i, \mathbf{x}_i) = \sum_{i=1}^n \widehat{\lambda}_i$, and $\operatorname{tr}\left({\mathbf{U}_r}^\top \mathbf{K}_n \mathbf{U}_r\right) = \sum\limits_{i=1}^n (\sum\limits_{s=1}^r \sum\limits_{k=1}^n [\mathbf{U}_r]_{si}^\top [\mathbf{K}_n]_{ik} [\mathbf{U}_r]_{ks})$, we can approximate the $\|\mathbf{K}\|_r$ by $\overline{\|\mathbf{K}\|_r} = \operatorname{tr}(\mathbf{K}_n) - \operatorname{tr}\left({\mathbf{U}_r}^\top \mathbf{K}_n \mathbf{U}_r\right)$ which is separable. In particular, $\overline{\|\mathbf{K}\|_r} = \sum_{i=1}^n K(\mathbf{x}_i, \mathbf{x}_i) -$

$\sum\limits_{i=1}^{n} \left(\sum\limits_{s=1}^{r} \sum\limits_{k=1}^{n} \left[\mathbf{U}_r\right]_{si}^{\top} \left[\mathbf{K}_n\right]_{ik} \left[\mathbf{U}_r\right]_{ks} \right)$. We remark that $\overline{\left\|\mathbf{K}\right\|_r}$ is ready to be optimized by standard SGD algorithms because it is separable and expressed as the summation of losses on individual training data points. The training loss of a neural network with KCR-Transformer blocks has $\text{KCR}(\mathcal{W}) = \overline{\left\|\mathbf{K}\right\|_r}$, the approximate TNN, as a regularization term. The following functions are needed for minibatch-based training with SGD, with the subscript j indicating the corresponding loss on the j-th batch $\mathcal{B}_j$:

$$\text{KCR}_j(\mathcal{W}) = \tfrac{1}{\left|\mathcal{B}_j\right|} \sum_{i=1}^{\left|\mathcal{B}_j\right|} \left(K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{s=1}^{r} \sum_{k=1}^{n} \left[\mathbf{U}_r\right]_{si}^{\top} \left[\mathbf{K}_n\right]_{ik} \left[\mathbf{U}_r\right]_{ks} \right), \text{CE}_j^{(t)}(\mathcal{W}) = \tfrac{1}{\left|\mathcal{B}_j\right|} \textstyle\sum_{i=1}^{\left|\mathcal{B}_j\right|} H(X_i(\mathcal{W}), Y_i),$$

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

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.

3.3 Search and Retraining Processes

To obtain a compact vision transformer network with KCR-Transformer, or the KCR-Transformer network, we optimize both the accuracy of the network and the inference cost (FLOPs) of the network. We follow the standard techniques in neural architecture search (Tan et al., 2019), where the attention output channels are pruned in the search process, and the pruned network is retrained in the retraining process. The search process involves 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 described in Section 3.1. We estimate the FLOPs of the MLP after the channel selection on the attention outputs following the KCR-Transformer. The FLOPs related to a single Transformer block is $\texttt{cost}_j = l_j \cdot \left(2\tilde{D}^2 + \tilde{D}\right)$, where j indexes the KCR-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 l_j denotes the number of layers in the MLP of the j-th KCR-Transformer block. The inference cost objective of the network architecture is computed by $\texttt{cost} = \sum_{j=1}^{M} \texttt{cost}_j$, where M is the number of transformer blocks. The overall loss function for search on each batch $\mathcal{B}_j$ at epoch t is formulated by $\mathcal{L}_{\text{search},j}^{(t)}(\mathcal{W}, \alpha) = \mathcal{L}_{\text{train},j}^{(t)}(\mathcal{W}) + \lambda \cdot \log \texttt{cost}(\alpha)$, where α is the architecture parameter. λ controls the magnitude of the cost term. In the search phase, the search loss is optimized to perform the two types of channel selection for all the KCR-Transformer blocks. After the search process, we use the selected channels for the attention outputs in a searched network and then perform retraining on the searched network using the training loss (2). Algorithm 1 in Section B of the appendix describes the search and retraining processes.

4 Experimental Results

We present the implementation details of our experiments in Section 4.1. We evaluate the performance of KCR-Transformers for image classification on the ImageNet-1k dataset in Section 4.2. In Section 4.3, we study the effectiveness of using KCR-Transformer as the feature extraction backbone for semantic segmentation and object detection. In Section 4.4, we study the effectiveness of the KCR-Transformer in reducing the KC of the networks. In Section 4.5, we examine the performance of KCR-Transformer as a vision encoder in vision-language models for the visual question answering tasks. In Section 4.6, we study the expected loss computed and the approximated KCR upper/lower bounds of the KCR-Transformers. Additional experiment results are presented in Section D of the appendix. In Section D.1, we evaluate the transferability of KCR-Transformer on downstream benchmarks. In Section D.2, we assess KCR-Transformer for self-supervised pretraining. In Section D.3, we describe the implementation details for instance segmentation. In Section D.4, we present the detailed object detection results. In Section D.5, we compare the training efficiency of KCR-Transformers with their corresponding baseline vision transformers and study the sensitivity to hyperparameters γ_0, η, and m. In Section D.6, we analyze the effect of varying λ on the compression of the vision transformers by KCR. In Section D.7, we compare the inference time of KCR-Transformers with the competing baseline models. We present Grad-CAM visualizations results of KCR-Transformers in Section D.8.

4.1 Experiment Settings

During the architecture search phase, we randomly sample a subset of 100 ImageNet classes (Russakovsky et al., 2015) for training. The network is optimized using the AdamW optimizer with a cosine learning rate schedule, where the initial learning rate of 0.001 is gradually annealed to 0.0001 over 200 epochs. In each epoch, 70% of the training samples are used for updating the model weights, while the remaining 30% are dedicated to optimizing the architecture parameters of the KCR-Transformer blocks. The temperature τ is initialized at 4.5 and decayed by a factor of 0.95 per epoch. Empirical results indicate that setting $t_{\text{warm}} = 90$ and $\eta = 1$ yields the best performance across all KCR model variants. During the retraining phase, the searched network is trained on the training set of ImageNet-1K using the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We set $t_{\text{train}} = 300$ as the total number of training epochs, and $t_{\text{warm}} = 90$ for all the experiments. In addition, we search for the optimal values of feature rank r. Let $r = \lceil \gamma_0 \min(n,d) \rceil$, where γ_0 is the rank ratio. We select the value by performing 5-fold cross-validation on 20% of the training data. The value of γ_0 is selected from 0.1 to 0.5 with a step size of 0.05, and $\gamma_0 = 0.2$ is found to be the optimal. Throughout all the experiments, m is set to 50000. λ and η are set to 0.2. As shown in Section D.5 of the appendix, the performance of the KCR-Transformer is stable with respect to different choices of the hyperparameters γ_0, η, m. The hyperparameter λ controls the size of the searched architecture. Section D.6 shows the performance of KCR-Transformers compressed to variant sizes by setting λ to different values. To improve the training efficiency, we compute $\tilde{\mathbf{U}}^{(r_0)}$, that is the approximation of $\mathbf{U}^{(r_0)}$, for every 30 epochs. Additional settings are presented in Section C of the appendix.

4.2 Image Classification

We adopt ViT-S (Dosovitskiy et al., 2021), ViT-B (Dosovitskiy et al., 2021), Swin-T (Liu et al., 2021a), and Swin-B (Liu et al., 2021a) as backbone models. Each transformer block in these architectures is substituted with a KCR-Transformer block. For comparison, we evaluate the performance of KCR-Transformers against vision transformers compressed by the state-of-the-art kernel-based model compression method, NTK-SAP (Wang et al., 2023), and the state-of-the-art pruning method DeepCompress (Ahmed et al., 2025). Table 1 shows that KCR-Transformers consistently exhibit reduced computational costs and improved performance compared to the baseline vision transformers. For instance, KCR-Swin-B achieves a 1.1% gain in top-1 accuracy while reducing the computational cost by 2.8 GFLOPs, highlighting the effectiveness of KCR in enhancing both efficiency and predictive performance. Notably, KCR-EfficientViT-B1 achieves the lowest computational cost among all evaluated models, requiring only 0.44 GFLOPs while even outperforming its uncompressed counterpart by 1.0% and all baselines in the first part of Table 1 (T2T-SPViT) in top-1 accuracy, demonstrating the effectiveness of KCR-Transformers for resource-constrained applications. Moreover, Table 9 in the appendix shows that the performance and efficiency improvements by the KCR-Transformer are achieved with only marginal increases ($< 7.8\%$) in the training time. Table 14 shows that KCR-Transformers exhibit faster inference speed than models compressed by competing baselines.

Model	# Params	FLOPs	Top-1
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., 2023b)	12.4 M	0.5 G	77.1
TinyViT (Wu et al., 2022)	5.4 M	1.3 G	79.1
EfficientFormer (Li et al., 2022b)	12.3 M	1.3 G	79.2
VTC-LFC (Wang et al., 2022)	5.0 M	1.3 G	78.0
SPViT (Kong et al., 2022)	4.9 M	1.2 G	77.8
EfficientViT-B1 (Cai et al., 2023)	9.1 M	0.52 G	79.4
MLP-Fusion-EfficientViT-B1 (Wei et al., 2023)	7.9 M	0.48 G	79.1
NTK-SAP-EfficientViT-B1 (Wang et al., 2023)	8.0 M	0.49 G	79.4
DeepCompress-EfficientViT-B1 (Ahmed et al., 2025)	7.9 M	0.46 G	79.2
KCR-EfficientViT-B1	7.8 M	0.44 G	**80.4**
ViT-S (Dosovitskiy et al., 2021)	22.1 M	4.3 G	81.2
MLP-Fusion-ViT-S (Wei et al., 2023)	19.8 M	4.0 G	81.0
NTK-SAP-ViT-S (Wang et al., 2023)	20.3 M	3.9 G	80.9
DeepCompress ViT-S (Ahmed et al., 2025)	20.0 M	3.9 G	81.1
KCR-ViT-S (Ours)	19.8 M	3.8 G	**82.2**
ViT-B (Dosovitskiy et al., 2021)	86.5 M	17.6 G	83.7
MLP-Fusion-ViT-B (Wei et al., 2023)	70.2 M	15.3 G	83.5
NTK-SAP-ViT-B (Wang et al., 2023)	71.8 M	15.6 G	83.5
DeepCompress ViT-B (Ahmed et al., 2025)	70.5 M	15.1 G	83.6
KCR-ViT-B (Ours)	69.5 M	14.5 G	**84.6**
Swin-T (Liu et al., 2021a)	29.0 M	4.5 G	81.3
MLP-Fusion-Swin-T (Wei et al., 2023)	24.8 M	4.1 G	81.0
NTK-SAP-Swin-T (Wang et al., 2023)	25.5 M	4.2 G	81.2
DeepCompress Swin-T (Ahmed et al., 2025)	24.8 M	4.1 G	81.1
KCR-Swin-T (Ours)	24.6 M	3.9 G	**82.4**
Swin-B (Liu et al., 2021a)	88.0 M	15.4 G	83.5
MLP-Fusion-Swin-B (Wei et al., 2023)	70.8 M	13.3 G	83.2
NTK-SAP-Swin-B (Wang et al., 2023)	72.6 M	13.2 G	83.2
DeepCompress Swin-B (Ahmed et al., 2025)	71.5 M	13.0 G	83.1
KCR-Swin-B (Ours)	70.2 M	12.6 G	**84.7**

Table 1: Performance comparison on ImageNet-1k.

4.3 Object Detection and Semantic Segmentation

We incorporate the ImageNet pre-trained KCR-Swin-B into the Cascade Mask R-CNN framework (Cai & Vasconcelos, 2021) for object detection. All models are evaluated on the MS-COCO dataset (Lin et al., 2014). The implementation details for the experiments are presented in Section D.4 of the appendix. It is observed in Table 2 that compressing the Swin backbones with KCR consistently improves both box-level and mask-level detection performance within the Cascade Mask R-CNN framework. For example, KCR-Swin-B achieves a box mAP of 52.5% and a mask mAP of 45.6%, with improvements of 0.6% and 0.6% over the standard Swin-B baseline. Results with AP at IoU thresholds of 50 and 75 are deferred to Table 8 in Section D.4 of the appendix.

In addition, we evaluate the performance of KCR for segmentation on the ADE20K (Zhou et al., 2019) using UperNet (Xiao et al., 2018) with our KCR-Swin-B as the feature extraction backbone. We include Swin-B (Liu et al., 2021a) and SETR (Zheng et al., 2021) as baselines for comparisons. We follow the training and evaluation protocol in (Liu et al., 2021a), where both our model and the baselines are trained on the training split and evaluated on the validation split of the dataset. More implementation details for the experiments are presented in Section D.3. As shown in Table 3, UperNet equipped with our KCR-Swin-B backbone achieves the highest validation mIoU of 52.4%, surpassing UperNet with Swin-B by 0.8% and SETR with ViT-L by 2.1%. These performance gains further highlight the effectiveness of the low-rank regularization and kernel complexity reduction introduced by KCR in enhancing the representational capacity of the vision backbone.

Framework	Feature Backbone	mAP^{box}	mAP^{m}
Mask R-CNN	Swin-B	51.9	45.0
Mask R-CNN	NTK-SAP-Swin-B	51.5	44.6
Mask R-CNN	KCR-Swin-B (Ours)	**52.5**	**45.6**

Table 2: Object Detection Results on COCO.

Framework	Feature Backbone	Val mIoU
SETR	ViT-L	50.3
UperNet	Swin-B	51.6
UperNet	NTK-SAP-Swin-B	51.3
UperNet	KCR-Swin-B (Ours)	**52.4**

Table 3: Segmentation Results on ADE20K.

4.4 Ablation Study on the Effects of KCR-Transformer in Reducing the KC

Model	# Params	FLOPs	Top-1	KC
EfficientViT-B1 (Cai et al., 2023)	9.1 M	0.52 G	79.4	4.08
MLP-Fusion-EfficientViT-B1 (Wei et al., 2023)	7.9 M	0.48 G	79.1	3.28
NTK-SAP-EfficientViT-B1 (Ahmed et al., 2025)	8.0 M	0.49 G	79.4	3.17
DeepCompress-EfficientViT-B1 (Cai et al., 2023)	7.9 M	0.46 G	79.2	3.68
KCR-EfficientViT-B1 (Cai et al., 2023)	7.8 M	0.44 G	**80.4**	**0.62**
ViT-S (Dosovitskiy et al., 2021)	22.1 M	4.3 G	81.2	4.12
MLP-Fusion-ViT-S (Wei et al., 2023)	19.8 M	4.0 G	81.0	3.88
NTK-SAP-ViT-S (Wang et al., 2023)	20.3 M	3.9 G	80.9	3.25
DeepCompress ViT-S (Ahmed et al., 2025)	20.0 M	3.9 G	81.1	3.97
KCR-ViT-S (Ours)	19.8 M	3.8 G	**82.2**	**0.65**
ViT-B (Dosovitskiy et al., 2021)	86.5 M	17.6 G	83.7	4.35
MLP-Fusion-ViT-B (Wei et al., 2023)	70.2 M	15.3 G	83.5	3.62
NTK-SAP-ViT-B (Wang et al., 2023)	71.8 M	15.6 G	83.5	3.58
DeepCompress ViT-B (Ahmed et al., 2025)	70.5 M	15.1 G	83.6	3.85
KCR-ViT-B (Ours)	69.5 M	14.5 G	**84.6**	**0.52**
Swin-T (Liu et al., 2021a)	29.0 M	4.5 G	81.3	3.42
MLP-Fusion-Swin-T (Wei et al., 2023)	24.8 M	4.1 G	81.0	3.10
NTK-SAP-Swin-T (Wang et al., 2023)	25.5 M	4.2 G	81.2	1.86
DeepCompress Swin-T (Ahmed et al., 2025)	24.8 M	4.1 G	81.1	2.42
KCR-Swin-T (Ours)	24.6 M	3.9 G	**82.4**	**0.44**
Swin-B (Liu et al., 2021a)	88.0 M	15.4 G	83.5	3.21
MLP-Fusion-Swin-B (Wei et al., 2023)	70.8 M	13.3 G	83.4	2.60
NTK-SAP-Swin-B (Wang et al., 2023)	72.6 M	13.2 G	83.2	2.42
DeepCompress Swin-B (Ahmed et al., 2025)	71.5 M	13.0 G	83.4	2.68
KCR-Swin-B (Ours)	70.2 M	12.6 G	**84.7**	**0.41**

Table 4: Ablation Study on the Effects of KCR-Transformer in Reducing the KC.

We conduct an ablation study to examine the effectiveness of KCR-Transformer in reducing the KC across four widely used vision transformer architectures, including ViT-S, ViT-B, Swin-T, and Swin-B. For each model, we compare the vanilla version against its KCR-Transformer counterpart. As shown in Table 4, the KCR-enhanced models consistently exhibit a substantial reduction in KC while simultaneously achieving lower parameter counts, reduced FLOPs, and higher top-1 classification accuracy on ImageNet-1K. For instance, KCR-ViT-S reduces the KC from 4.12 to 0.65 and improves the top-1 accuracy from 81.2% to 82.2% with fewer FLOPs and parameters. These improvements demonstrate that our approximate TNN regularization effectively reduces the KC of the vision backbones, leading to improved generalization capability. In contrast, the existing kernel-based compression method NTK-SAP (Wang et al., 2023) is limited by the conventional NTK limit, so it does not improve the top-1 accuracy of the pruned models, while its KC is slightly smaller than that of the original model as a result of pruning. MLP-Fusion-ViT-B (Wei et al., 2023) and DeepCompress ViT-B (Ahmed et al., 2025) also renders compressed models with worse top-1 accuracy and slightly reduced KC compared to the original model, due to the compression effect.

4.5 Visual Question Answering (VQA) with KCR-Transformer as the Vision Encoder in Vision-Language Models (VLMs)

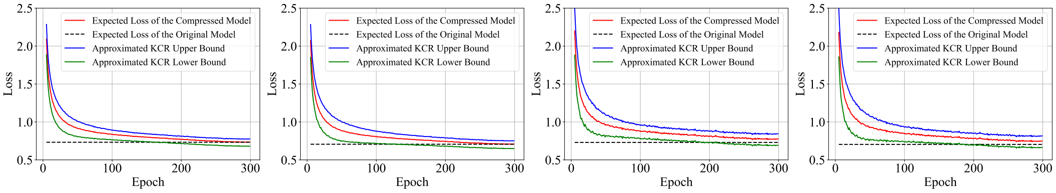
We further evaluate the KCR-Transformer in the visual question answering (VQA) task by employing it as the vision encoder within the vision-language model (VLM) framework, FLAVA (Singh et al., 2022). The evaluation is performed on widely used VQA datasets VQAv2 (Goyal et al., 2017) and SNLI-VE (Xie et al., 2019). We replace the vision encoder, ViT-B, in the VLM framework, FLAVA, with our KCR-Transformer during the vision encoder pre-training process. For VQAv2, we follow the settings in FLAVA (Singh et al., 2022) and fine-tune the model using a cross-entropy loss over the top frequent answer candidates. For SNLI-VE, we also follow the settings in FLAVA (Singh et al., 2022) and fine-tune the model using a standard three-way classification loss to distinguish entailment, contradiction, and neutrality. The test-dev VQA score for VQAv2 and the test accuracy for SNLI-VE are reported following (Singh et al., 2022). It is observed in Table 5 that FLAVA with KCR-ViT-B consistently outperforms the original FLAVA with ViT-B and the baseline model with ViT-B compressed by NTK-SAP across both benchmarks while maintaining a lower computational cost. For example, FLAVA with KCR-ViT-B achieves a VQAv2 score of 73.26%, outperforming FLAVA with ViT-B by 0.77% and FLAVA with NTK-SAP-ViT-B by 1.18%.

Model	# Params	FLOPs	VQAv2	SNLI-VE
FLAVA (ViT-B) (Dosovitskiy et al., 2021)	86.5 M	17.6 G	72.49	78.89
FLAVA (NTK-SAP-ViT-B) (Wang et al., 2023)	71.8 M	15.6 G	72.08	78.16
FLAVA (KCR-ViT-B)	69.5 M	14.5 G	**73.26**	**79.35**

Table 5: Performance for VQA on VQAv2 (Goyal et al., 2017) and SNLI-VE (Xie et al., 2019).

4.6 Study on the Tightness of the KCR Upper and Lower Bounds

Figure 1 illustrates the expected loss computed over the training and validation sets of the ImageNet-1K and the approximated KCR upper/lower bounds computed at different training epochs for ViT-S, ViT-B, Swin-T, and Swin-B trained on the ImageNet-1K dataset. We note that the approximated KCR upper/lower bounds are the KCR upper/lower bounds with the KC replaced by the approximate KC, A-KC, which is defined as $\text{A-KC}(\mathbf{K}) \coloneqq \min\limits_{h \in [0, r_0]} \left(\frac{h}{n} + \sqrt{\frac{1}{n} \overline{\|\mathbf{K}\|}_h} \right)$. Since the approximate TNN $\overline{\|\mathbf{K}\|}_h$ is expected to be close to the TNN $\|\mathbf{K}\|_h$ for each $h \in [0 \colon r_0]$, the approximate KC A-KC$(\mathbf{K})$ is also expected to be close to the KC, KC$(\mathbf{K})$. It can be observed that the approximated KCR upper/lower bounds are tightly correlated to the expected loss, revealing the tightness of the upper/lower bounds for the generalization error of the DNNs with KCR-Transformer blocks.



(a) Swin-B (b) ViT-B (c) Swin-T (d) ViT-S

Figure 1: Illustration of the expected loss and the approximated KCR upper/lower bounds over different training epochs for ViT-S, ViT-B, Swin-T, and Swin-B compressed by KCR.

5 Conclusion

In this paper, we propose KCR-Transformer, a novel and generalization-aware transformer block equipped with differentiable channel selection for the MLP layers in vision transformers. Guided by a novel and sharp theoretical generalization bound derived from the kernel complexity (KC) of the network, KCR-Transformer enables channel pruning in a theoretically grounded and principled manner. Our method is compatible with a wide range of vision transformer architectures and can be seamlessly integrated to replace standard transformer blocks. Extensive experiments across image classification, object detection, and semantic segmentation demonstrate that KCR-Transformer consistently achieves superior performance with fewer FLOPs and parameters, validating its effectiveness for building efficient vision transformers.

References

Sabbir Ahmed, Abdullah Al Arafat, Deniz Najafi, Akhlak Mahmood, Mamshad Nayeem Rizve, Mohaiminul Al Nahian, Ranyang Zhou, Shaahin Angizi, and Adnan Siraj Rakin. Deepcompress-vit: Rethinking model compression to enhance efficiency of vision transformers at the edge. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 30147–30156. Computer Vision Foundation / IEEE, 2025.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning (ICML)*, volume 97 of *Machine Learning Research*, pp. 322–332. PMLR, 2019.

David Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021.

Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *Ann. Statist.*, 33(4):1497–1537, 08 2005.

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pp. 541–549. PMLR, 2018.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *ICLR*, 2023.

Maxim Bonnaerens and Joni Dambre. Learned thresholds token merging and pruning for vision transformers. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Mario Alejandro Bravo-Ortiz, Esteban Mercado-Ruiz, Juan Pablo Villa-Pulgarin, Carlos Angel Hormaza-Cardona, Sebastian Quiñones-Arredondo, Harold Brayan Arteaga-Arteaga, Simon Orozco-Arias, Oscar Cardona-Morales, and Reinel Tabares-Soto. Cvtstego-net: A convolutional vision transformer architecture for spatial image steganalysis. *Journal of Information Security and Applications*, 81:103695, 2024.

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

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: high quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(5):1483–1498, 2021.

Abdulkadir Canatar and Cengiz Pehlevan. A kernel analysis of feature learning in deep neural networks. In *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1–8. IEEE, 2022.

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 10835–10845, 2019.

Arnav Chavan, Zhiqiang Shen, Zhuang Liu, Zechun Liu, Kwang-Ting Cheng, and Eric P Xing. 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.

Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12270–12280, 2021a.

Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 34:19974–19988, 2021b.

Wenlong Chen and Yingzhen Li. Calibrating transformers via sparse gaussian processes. *arXiv preprint arXiv:2303.02444*, 2023.

Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four GPU hours: A theoretically inspired perspective. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021c.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 9620–9629. IEEE, 2021d.

Yingyi Chen, Qinghua Tao, Francesco Tonin, and Johan Suykens. Primal-attention: Self-attention through asymmetric kernel svd in primal representation. *Advances in Neural Information Processing Systems*, 36:65088–65101, 2023.

Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. *arXiv preprint arXiv:2108.05895*, 2021e.

Ta-Chung Chi, Ting-Han Fan, Peter J Ramadge, and Alexander Rudnicky. Kerple: Kernelized relative positional embedding for length extrapolation. *Advances in Neural Information Processing Systems*, 35:8386–8399, 2022.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Alexandru Damian, Jason D. Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In Po-Ling Loh and Maxim Raginsky (eds.), *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pp. 5413–5452. PMLR, 2022.

Zhijie Deng, Jiaxin Shi, and Jun Zhu. Neuralef: Deconstructing kernels by deep neural networks. In *International Conference on Machine Learning*, pp. 4976–4992. PMLR, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Stanislav Fort, Leo Scherlis, Surya Ganguli, and Jascha Sohl-Dickstein. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5850–5861, 2020.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029 – 1054, 2021.

Spyros Gidaris, Andrei Bursuc, Oriane Siméoni, Antonín Vobecký, Nikos Komodakis, Matthieu Cord, and Patrick Pérez. MOCA: self-supervised representation learning by predicting masked online codebook assignments. *Trans. Mach. Learn. Res.*, 2024, 2024.

Chengyue Gong, Dilin Wang, Meng Li, Xinlei Chen, Zhicheng Yan, Yuandong Tian, qiang liu, and Vikas Chandra. NASVit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In *International Conference on Learning Representations*, 2022.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 6325–6334. IEEE Computer Society, 2017.

Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12259–12269, 2021.

Lingyu Gu, Yongqi Du, Yuan Zhang, Di Xie, Shiliang Pu, Robert C. Qiu, and Zhenyu Liao. ”lossless” compression of deep neural networks: A high-dimensional neural tangent kernel approach. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Yuka Hashimoto, Masahiro Ikeda, and Hachem Kadri. Deep learning with kernels through rkhm and the perron-frobenius operator. *Advances in Neural Information Processing Systems*, 36:50677–50696, 2023.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Wentao Huang, Houbao Lu, and Haizhang Zhang. Hierarchical kernels in deep kernel learning. *Journal of Machine Learning Research*, 24(391):1–30, 2023.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8580–8589, 2018.

Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1383–1392, 2024.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350 of *Lecture Notes in Computer Science*, pp. 491–507. Springer, 2020.

Vladimir Koltchinskii. Local rademacher complexities and oracle inequalities in risk minimization. *Ann. Statist.*, 34(6):2593–2656, 12 2006.

Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, 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, 2022.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, pp. 554–561. IEEE Computer Society, 2013.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the nyström method. *J. Mach. Learn. Res.*, 13:981–1006, 2012.

Jongmin Lee, Joo Young Choi, Ernest K Ryu, and Albert No. Neural tangent kernel analysis of deep narrow neural networks. In *International Conference on Machine Learning*, pp. 12282–12351. PMLR, 2022.

Yanjing Li, Sheng Xu, Baochang Zhang, Xianbin Cao, Peng Gao, and Guodong Guo. Q-vit: Accurate and fully quantized low-bit vision transformer. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:34451–34463, 2022a.

Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022b.

Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 17227–17236, 2023.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. *ICCV*, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1173–1179. ijcai.org, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *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*, 2023a.

Ji Liu, Dehua Tang, Yuanxian Huang, Li Zhang, Xiaocheng Zeng, Dong Li, Mingjie Lu, Jinzhang Peng, Yu Wang, Fan Jiang, Lu Tian, and Ashish Sirasao. UPDP: A unified progressive depth pruner for CNN and vision transformer. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 13891–13899. AAAI Press, 2024a.

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*, pp. 14420–14430, 2023b.

Yifei Liu, Mathias Gehrig, Nico Messikommer, Marco Cannici, and Davide Scaramuzza. Revisiting token pruning for object detection and instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2658–2668, 2024b.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021a.

Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 28092–28103, 2021b.

Junzhu Mao, Yang Shen, Jinyang Guo, Yazhou Yao, Xiansheng Hua, and Hengtao Shen. Prune and merge: Efficient token compression for vision transformer with spatial information preserved. *IEEE Transactions on Multimedia*, 2025.

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

Shahar Mendelson. Geometric parameters of kernel machines. In Jyrki Kivinen and Robert H. Sloan (eds.), *Conference on Computational Learning Theory*, volume 2375 of *Lecture Notes in Computer Science*, pp. 29–43. Springer, 2002.

Jisoo Mok, Byunggook Na, Ji-Hoon Kim, Dongyoon Han, and Sungroh Yoon. Demystifying the neural tangent kernel from a practical perspective: Can it be trusted for neural architecture search without training? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11851–11860. IEEE, 2022.

Grégoire Montavon, Mikio L Braun, and Klaus-Robert Müller. Kernel analysis of deep networks. *Journal of Machine Learning Research*, 12(9), 2011.

Tan Nguyen, Minh Pham, Tam Nguyen, Khai Nguyen, Stanley Osher, and Nhat Ho. Fourierformer: Transformer meets generalized fourier integral theorem. *Advances in Neural Information Processing Systems*, 35:29319–29335, 2022.

Tan Minh Nguyen, Tam Minh Nguyen, Nhat Ho, Andrea L. Bertozzi, Richard G. Baraniuk, and Stanley J. Osher. A primal-dual framework for transformers and neural networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Eshaan Nichani, Yu Bai, and Jason D. Lee. Identifying good directions to escape the NTK regime and efficiently learn low-degree plus sparse polynomials. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008*, pp. 722–729. IEEE Computer Society, 2008.

Lorenzo Papa, Paolo Russo, Irene Amerini, and Luping Zhou. A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pp. 3498–3505. IEEE Computer Society, 2012.

John Rachwan, Daniel Zügner, Bertrand Charpentier, Simon Geisler, Morgane Ayle, and Stephan Günnemann. Winning the lottery ahead of time: Efficient early network pruning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18293–18309. PMLR, 2022.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *arXiv preprint arXiv:2106.02034*, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Mariia Seleznova and Gitta Kutyniok. Analyzing finite neural networks: Can we trust neural tangent kernel theory? In *Mathematical and Scientific Machine Learning*, pp. 868–895. PMLR, 2022.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 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.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A foundational language and vision alignment model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 15617–15629. IEEE, 2022.

Kyungwoo Song, Yohan Jung, Dongjun Kim, and Il-Chul Moon. Implicit kernel attention. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 9713–9721. AAAI Press, 2021.

Xiu Su, Shan You, Jiyang Xie, Mingkai Zheng, Fei Wang, Chen Qian, Changshui Zhang, Xiao-Gang Wang, and Chang Xu. Vitas: Vision transformer architecture search. In *European Conference on Computer Vision (ECCV)*, volume 13681 of *Lecture Notes in Computer Science*, pp. 139–157. Springer, 2022.

Shokichi Takakura and Taiji Suzuki. Mean-field analysis on two-layer neural networks from a kernel perspective. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. 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.

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*, pp. 10347–10357. PMLR, 2021.

Yite Wang, Dawei Li, and Ruoyu Sun. NTK-SAP: improving neural network pruning by aligning training dynamics. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Zhenyu Wang, Hao Luo, Pichao Wang, Feng Ding, Fan Wang, and Hao Li. Vtc-lfc: Vision transformer compression with low-frequency components. *Advances in Neural Information Processing Systems*, 35:13974–13988, 2022.

Tianxin Wei, Zeming Guo, Yifan Chen, and Jingrui He. Ntk-approximating MLP fusion for efficient language model fine-tuning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36821–36838. PMLR, 2023.

Zimian Wei, Peijie Dong, Zheng Hui, Anggeng Li, Lujun Li, Menglong Lu, Hengyue Pan, and Dongsheng Li. Auto-prox: Training-free vision transformer architecture search via automatic proxy discovery. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 15814–15822. AAAI Press, 2024.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673. PMLR, 2020.

Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. 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.

Lechao Xiao, Jeffrey Pennington, and Samuel Schoenholz. Disentangling trainability and generalization in deep neural networks. In *International Conference on Machine Learning*, pp. 10462–10472. PMLR, 2020.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, volume 11209 of *Lecture Notes in Computer Science*, pp. 432–448. Springer, 2018.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *CoRR*, abs/1901.06706, 2019.

Kaixin Xu, Zhe Wang, Chunyun Chen, Xue Geng, Jie Lin, Xulei Yang, Min Wu, Xiaoli Li, and Weisi Lin. Lpvit: Low-power semi-structured pruning for vision transformers. In *European Conference on Computer Vision*, pp. 269–287. Springer, 2024.

Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. From knowledge distillation to self-knowledge distillation: A unified approach with normalized loss and customized soft labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17185–17194, 2023.

Zhendong Yang, Zhe Li, Ailing Zeng, Zexian Li, Chun Yuan, and Yu Li. Vitkd: Feature-based knowledge distillation for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1379–1388, 2024.

Fang Yu, Kun Huang, Meng Wang, Yuan Cheng, Wei Chu, and Li Cui. Width & depth pruning for vision transformers. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 3143–3151. AAAI Press, 2022a.

Shixing Yu, Tianlong Chen, Jiayi Shen, Huan Yuan, Jianchao Tan, Sen Yang, Ji Liu, and Zhangyang Wang. Unified visual transformer compression. *ICLR*, 2022b.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021.

Borui Zhao, Renjie Song, and Jiajun Liang. Cumulative spatial knowledge distillation for vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 6146–6155, 2023.

Chuanyang Zheng, Kai Zhang, Zhi Yang, Wenming Tan, Jun Xiao, Ye Ren, Shiliang Pu, et al. Savit: Structure-aware vision transformer pruning via collaborative optimization. *Advances in Neural Information Processing Systems*, 35:9010–9023, 2022.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 6881–6890. Computer Vision Foundation / IEEE, 2021.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *Int. J. Comput. Vis.*, 127(3):302–321, 2019.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ICLR*, 2021.

A PROOF OF THEOREM 3.1

Proof of Theorem 3.1. We consider the dynamic kernel $K(\mathbf{x}, \mathbf{x}') = F(\mathbf{x})^\top F(\mathbf{x}')$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ defined in Section 3.2. Then it follows from (Bartlett et al., 2005, Theorem 3.3) that for every $K > 1$,

$$\begin{aligned} L_{\mathcal{D}}(\mathrm{NN}_{\mathcal{W}}) &\le \frac{K}{K-1}\mathbb{E}_{P_n}\left[\|\mathrm{NN}_{\mathcal{W}}(\mathbf{x}) - \mathbf{y}\|_2^2\right] + \Theta(r^*) + \Theta\left(\frac{x}{n}\right), \\ L_{\mathcal{D}}(\mathrm{NN}_{\mathcal{W}}) &\ge \frac{K}{K+1}\mathbb{E}_{P_n}\left[\|\mathrm{NN}_{\mathcal{W}}(\mathbf{x}) - \mathbf{y}\|_2^2\right] - \Theta(r^*) - \Theta\left(\frac{x}{n}\right), \end{aligned} \tag{3}$$

and each inequality in (3) holds with probability at least $1 - \exp(-x)$. K is a positive definite kernel and the network $\mathrm{NN}_{\mathcal{W}} \in \mathcal{H}_K$ where $\mathcal{H}_K$ denotes the Reproducing Kernel Hilbert Space

(RKHS) associated with K, and we recall that $\mathrm{KC}(\mathbf{K}) = \min_{h \in [0, r_0]} \left(\frac{h}{n} + \sqrt{\frac{1}{n} \sum_{i=h+1}^{r_0} \widehat{\lambda}_i} \right)$ is the kernel complexity. It then follows from (Bartlett et al., 2005, Corollary 6.7) and (3) that

$$
\begin{aligned}
L_{\mathcal{D}}(\mathrm{NN}_{\mathcal{W}}) &\le \frac{K}{K-1} \mathbb{E}_{P_n} \left[\|\mathrm{NN}_{\mathcal{W}}(\mathbf{x}) - \mathbf{y}\|_2^2 \right] + \Theta(\mathrm{KC}(\mathbf{K})) + \Theta\left(\frac{x}{n}\right), \\
L_{\mathcal{D}}(\mathrm{NN}_{\mathcal{W}}) &\ge \frac{K}{K+1} \mathbb{E}_{P_n} \left[\|\mathrm{NN}_{\mathcal{W}}(\mathbf{x}) - \mathbf{y}\|_2^2 \right] - \Theta(\mathrm{KC}(\mathbf{K})) - \Theta\left(\frac{x}{n}\right).
\end{aligned}
\tag{4}
$$

Combining (3) and (4) proves (1).

□

B Algorithms

Algorithm 1 describes the search and the retraining process of the KCR-Transformer.

Algorithm 1 Training Algorithm with the Approximate Truncated Nuclear Norm by SGD

Require: Training dataset $\mathcal{D}$, number of search epochs t_{search}, number of training epochs t_{train}, warm-up epochs t_{warm}, batch size J, learning rates $\eta_{\mathcal{W}}$ and η_α

Ensure: Trained weights $\mathcal{W}$ of the network

```
Split $\mathcal{D}$ into $\mathcal{D}_{\mathcal{W}}$ and $\mathcal{D}_\alpha$, where $70\%$ of samples ($\mathcal{D}_{\mathcal{W}}$) are used to update model weights $\mathcal{W}$, and $30\%$ of samples ($\mathcal{D}_\alpha$) are used to optimize architecture parameters $\alpha$.
Initialize the network weights $\mathcal{W} = \mathcal{W}(0)$ by random initialization.
for $t \leftarrow 1$ to $t_{\text{search}}$ do
    for $j \leftarrow 1$ to $J$ do
        Sample mini-batches $\mathcal{B}_j^{\mathcal{W}} \subset \mathcal{D}_{\mathcal{W}}$, $\mathcal{B}_j^{\alpha} \subset \mathcal{D}_\alpha$.
        Perform gradient descent on $\mathcal{B}_j^{\alpha}$ to update $\alpha$ by $\alpha \leftarrow \alpha - \eta_\alpha \nabla_\alpha \mathcal{L}_{\text{search},j}^{(t)}(\mathcal{W}, \alpha)$
        Perform gradient descent on $\mathcal{B}_j^{\mathcal{W}}$ to update $\mathcal{W}$ by $\mathcal{W} \leftarrow \mathcal{W} - \eta_{\mathcal{W}} \nabla_{\mathcal{W}} \mathcal{L}_{\text{search},j}^{(t)}(\mathcal{W}, \alpha)$
    end for
end for
for $t \leftarrow 1$ to $t_{\text{train}}$ do
    if $t \bmod 30 = 0$ then
        Update $\tilde{\mathbf{U}}^{(r_0)}$, the approximation of $\mathbf{U}^{(r_0)}$.
    end if
    for $j \leftarrow 1$ to $J$ do
        Sample mini-batch $\mathcal{B}_j \subset \mathcal{D}_{\mathcal{W}}$.
        if $t > t_{\text{warm}}$ then
            Perform gradient descent on $\mathcal{B}_j^{\mathcal{W}}$ to update $\mathcal{W}$ by $\mathcal{W} \leftarrow \mathcal{W} - \eta_{\mathcal{W}} \nabla_{\mathcal{W}} \mathcal{L}_{\text{train},j}^{(t)}(\mathcal{W})$
        else
            Perform gradient descent on $\mathcal{B}_j^{\mathcal{W}}$ to update $\mathcal{W}$ by $\mathcal{W} \leftarrow \mathcal{W} - \eta_{\mathcal{W}} \nabla_{\mathcal{W}} \mathrm{CE}_j^{(t)}(\mathcal{W})$
        end if
    end for
end for
return $\mathcal{W}$
```

C Implementation Details

All experiments are conducted on 40GB NVIDIA A100 GPUs with an effective batch size of 512. Following standard practice (Cai et al., 2023), we apply widely adopted data augmentation techniques during training, including random scaling, random horizontal flipping, and random cropping. The weight decay is set to 0.01. The learning rate is linearly increased from 0.0002 to 0.002 over the first five epochs, then gradually annealed back to 0.0002 using a cosine decay schedule over the remaining epochs. Inference is performed using the exponential moving average (EMA) of model weights.

D Additional Experiment Results

D.1 Transfer Learning Capability of the KCR-Transformer

We evaluate the transfer learning capability of KCR-ViT-B in comparison to the baseline ViT-B using three widely adopted benchmarks, including Oxford Flowers-102 (Nilsback & Zisserman, 2008), Oxford-IIIT Pet (Parkhi et al., 2012), and Stanford Cars (Krause et al., 2013). Following the established transfer learning protocol in (Kolesnikov et al., 2020), both models are first pre-trained on ImageNet and subsequently fine-tuned on the respective training sets of the target datasets for 50 epochs using the Adam optimizer, with a fixed learning rate of 1×10^{-5}. The experimental results are presented in Table 6. It is observed that KCR-ViT-B consistently outperforms ViT-B and NTK-SAP-ViT-B across all three datasets, while requiring fewer FLOPs and model parameters. For example, KCR-ViT-B achieves a top-1 accuracy of 93.8%, outperforming ViT-B and NTK-SAP-ViT-B by 1.1% and 1.3%, respectively, on the Cars dataset. The improvements demonstrate the superior transferability of the features learned by the KCR compressed model across diverse visual domains.

Model	# Params	FLOPs	ImageNet	Flowers	Pet	Cars
ViT-B (Dosovitskiy et al., 2021)	86.5 M	17.6 G	83.7	97.8	96.0	92.7
NTK-SAP-ViT-B (Wang et al., 2023)	71.8 M	15.6 G	83.5	97.4	95.6	92.5
KCR-ViT-B	69.5 M	14.5 G	**84.6**	**98.2**	**96.6**	**93.8**

Table 6: Top-1 classification accuracy comparison for transfer learning on the Oxford Flowers-102 (Nilsback & Zisserman, 2008), Oxford-IIIT Pet (Parkhi et al., 2012), and Stanford Cars (Krause et al., 2013) datasets.

D.2 Self-Supervised Learning with KCR-Transformers

We further evaluate the effectiveness of KCR-Transformers in the self-supervised learning (SSL) setting using both MoCoV3 (Chen et al., 2021d) and MOCA (Gidaris et al., 2024). In our experiments, the ViT-B model pre-trained with each SSL method serves as the baseline. We pre-train ViT-B, NTK-SAP-ViT-B, and our KCR-ViT-B strictly following the training settings described in the respective papers (Chen et al., 2021d; Gidaris et al., 2024) on the ImageNet1K without using training labels. The pre-trained models are then subsequently fine-tuned with class labels for downstream classification on ImageNet1K following the fine-tuning protocols in (Chen et al., 2021d; Gidaris et al., 2024). As shown in Table 7, KCR-ViT-B consistently achieves superior performance over ViT-B and NTK-SAP-ViT-B across both SSL pipelines, while requiring fewer FLOPs and parameters. For example, under the MOCA (Gidaris et al., 2024) pretraining pipeline, KCR-ViT-B achieves a top-1 accuracy of 84.4%, outperforming ViT-B and NTK-SAP-ViT-B by 1.0% and 1.2%, respectively, with reduced computational cost and parameter size. The improvements highlight the effectiveness of the KCR compression method in enhancing the efficiency and accuracy of the models pre-trained by SSL methods.

Network	SSL Method	# Params	FLOPs	Top-1
ViT-B (Dosovitskiy et al., 2021)	MoCoV3 (Chen et al., 2021d)	86.5 M	17.6 G	83.2
NTK-SAP-ViT-B (Wang et al., 2023)		71.8 M	15.6 G	82.9
KCR-ViT-B (Ours)		69.5 M	14.5 G	**84.1**
ViT-B (Dosovitskiy et al., 2021)	MOCA (Gidaris et al., 2024)	86.5 M	17.6 G	83.4
NTK-SAP-ViT-B (Wang et al., 2023)		71.8 M	15.6 G	83.2
KCR-ViT-B (Ours)		69.5 M	14.5 G	**84.4**

Table 7: Top-1 classification accuracy comparison for models pre-trained with the self-supervised learning (SSL) methods MoCoV3 (Chen et al., 2021d) and MOCA (Gidaris et al., 2024) on ImageNet1K.

D.3 Implementation Details for Instance Segmentation

ADE20K has 25000 images in total, with 20000 for training, 2000 for validation, and another 3000 for testing. We adopt UperNet (Xiao et al., 2018) as the segmentation framework with our KCR-

Swin-B as the feature extraction backbone. We follow the training and evaluation protocol in (Liu et al., 2021a), where both our model and the baselines are trained on the training split and evaluated on the validation split of the dataset. All models are optimized using AdamW for a total of 160000 iterations with a batch size of 16, an initial learning rate of 6×10^{-5}, and a weight decay of 0.01. The learning rate follows a linear decay schedule after a warm-up phase of 1500 iterations. To enhance generalization, we employ data augmentation techniques including random horizontal flipping, random rescaling with a ratio range of $[0.5, 2.0]$, and random photometric distortions. Stochastic depth regularization is applied with a drop rate of 0.2. For inference, we use multi-scale testing with scale factors varying from 0.5 to 1.75.

D.4 Implementation Details and Additional Results for Object Detection

We incorporate the ImageNet pre-trained KCR-Swin-T and KCR-Swin-B into the Cascade Mask R-CNN framework (Cai & Vasconcelos, 2021) for object detection. All models are evaluated on the MS-COCO dataset (Lin et al., 2014), which consists of 117000 training images and 5000 validation images. We follow the training configuration of (Liu et al., 2021a), where each input image is resized such that the shorter side falls within $[480, 800]$ pixels while the longer side does not exceed 1333 pixels. Training is performed using the AdamW optimizer with an initial learning rate of 0.0001, a weight decay of 0.05, and a batch size of 16, for a total of 36 epochs following the 3× schedule. In line with (Cai & Vasconcelos, 2021), we report standard COCO metrics, including the box-level mean Average Precision (mAP^{box}) and mask-level mean Average Precision (mAP^{m}), as well as AP at IoU thresholds of 50 and 75.

Detection Framework	Feature Backbone	mAP^{box}	AP^{b}_{50}	AP^{b}_{75}	mAP^{m}	AP^{m}_{50}	AP^{m}_{75}
Mask R-CNN	Swin-T	50.5	69.3	54.9	43.7	66.6	47.1
Mask R-CNN	Swin-B	51.9	70.9	56.5	45.0	68.4	48.7
Mask R-CNN	KCR-Swin-T (Ours)	50.9	69.7	55.3	44.0	67.1	47.6
Mask R-CNN	KCR-Swin-B (Ours)	52.5	71.4	56.8	45.6	68.9	49.1

Table 8: Detailed Object Detection Results on COCO.

It is observed in Table 8 that compressing the Swin backbones with KCR consistently improves both box-level and mask-level detection performance within the Cascade Mask R-CNN framework. For example, KCR-Swin-T achieves a box mAP of 50.9% and a mask mAP of 44.0%, with improvements of 0.4% and 0.3% over the standard Swin-T baseline. Similarly, KCR-Swin-B achieves the highest box mAP of 52.5% and mask mAP of 45.6%, surpassing the Swin-B baseline by 0.6% and 0.6%, respectively. These results demonstrate that KCR compression effectively enhances the feature expressiveness of vision backbones for object detection, with even less computational costs. The consistent improvements across multiple IoU thresholds further validate the robustness and generalization capability of vision backbones compressed by KCR.

D.5 Training Efficiency and Sensitivity of Hyperparameters γ_0, η, m

In this section, we first compare the training time of the KCR-Transformers with the corresponding baseline vision transformers without the KCR regularization. The comparison is performed on one NVIDIA A100 40GB GPU with full-precision, and the average training time per epoch is reported. It is observed from Table 9 that the KCR regularization only marginally increases the training time ($< 7.8\%$) compared to the corresponding baseline vision transformers, while greatly increasing the top-1 classification accuracy and reducing the model size of FLOPs. For example, KCR-Swin-B achieves a 1.2% top-1 accuracy improvement while reducing the number of parameters from 88.0M to 70.2M and FLOPs from 15.4G to 12.6G, with a negligible training overhead of 1.6 minutes per epoch, which is only 4.18% of the original training time of Swin-B. In addition, we also report the overall search time and re-training time in comparison with the training time of the baseline model. It is observed that the search phase only brings marginal training overhead, costing less than 7.2% of the baseline model's training time.

We further study the sensitivity of KCR-Transformer to the key hyperparameters γ_0, η, and m by conducting experiments on the Swin-B backbone with different values of γ_0, η, and m. When

Model	# Params	FLOPs	Top-1	Training Efficiency (Minutes/Epoch)	Search Time (Hours)	Re-Training Time (Hours)
ViT-S (Dosovitskiy et al., 2021)	22.1 M	4.3 G	81.2	16.8	-	84.0
KCR-ViT-S (Ours)	19.8 M	3.8 G	**82.2**	18.1	6.03	88.5
ViT-B (Dosovitskiy et al., 2021)	86.5 M	17.6 G	83.7	33.5	-	167.5
KCR-ViT-B (Ours)	69.5 M	14.5 G	**84.6**	35.2	11.7	173.4
Swin-T (Liu et al., 2021a)	29.0 M	4.5 G	81.3	20.8	-	104.0
KCR-Swin-T (Ours)	24.6 M	3.9 G	**82.4**	22.1	7.3	108.5
Swin-B (Liu et al., 2021a)	88.0 M	15.4 G	83.5	38.3	-	191.5
KCR-Swin-B (Ours)	70.2 M	12.6 G	**84.7**	39.9	13.3	197.1

Table 9: Training time comparisons with baseline methods on the training set of ImageNet-1K.

performing the study on one of the hyperparameters, the remaining hyperparameters are set to their corresponding optimal values. The results summarized in Tables 10, 11, and 12. Across all settings, the top-1 accuracy remains stable within an error range of 0.3%, indicating that KCR-Transformer is robust to the choice of these hyperparameters.

γ_0	0.1	0.2	0.3	0.4	0.5
Top-1 (%)	84.6	84.7	84.5	84.7	84.6

Table 10: Sensitivity analysis of KCR-Swin-B to the choices of γ_0 on the ImageNet-1K dataset.

η	0.1	0.2	0.3	0.4	0.5
Top-1 (%)	84.6	84.7	84.7	84.6	84.5

Table 11: Sensitivity analysis of KCR-Swin-B to the choices of η on the ImageNet-1K dataset.

m	25000	50000	75000	100000	200000
Top-1 (%)	84.4	84.7	84.6	84.7	84.7

Table 12: Sensitivity analysis of KCR-Swin-B to the choices of m on the ImageNet-1K dataset.

D.6 KCR-Transformers with Different Compression Ratios

In this section, we investigate the performance of KCR-Transformer under different compression ratios by varying the hyperparameter λ in the search loss. Larger values of λ impose stronger penalization on the computational cost, thereby yielding more aggressively compressed architectures. We evaluate four representative settings, $\lambda \in \{0.2, 0.4, 0.6, 0.8\}$, using the Swin-B backbone on the ImageNet-1K dataset. In addition, we compare KCR-Swin-B with Swin-B compressed by NTK-SAP (Wang et al., 2023) of similar sizes. As shown in Table 13, KCR-Transformer consistently outperforms NTK-SAP across all compression levels. Notably, for each comparable model size, KCR-Swin-B achieves significantly higher top-1 accuracy while maintaining lower parameter count and FLOPs. For example, at $\lambda = 0.2$, KCR-Swin-B achieves 84.7% top-1 accuracy with only 12.6 G FLOPs, outperforming NTK-SAP-Swin-B (82.8%, 10.2 G FLOPs) by 1.9% in top-1 accuracy. This trend persists even under more aggressive compression settings, such as $\lambda = 0.8$, where KCR-Swin-B achieves a top-1 accuracy of 82.9%, despite having fewer parameters and FLOPs than its NTK-SAP counterpart.

D.7 Inference Time Comparison

In this section, we further compare the inference latency of the KCR-Transformers against their corresponding baseline vision transformers without the KCR regularization. All measurements are conducted on two NVIDIA A100 40GB GPUs using FP16 precision with a batch size of 128. As shown in Table 14, KCR-Transformer variants consistently achieve lower inference latency compared to their counterparts, while simultaneously improving top-1 classification accuracy and reducing both the parameter count and computational cost (FLOPs). For instance, KCR-Swin-B achieves a 1.2% gain in top-1 accuracy over Swin-B, while reducing the number of parameters from 88.0M

Model	λ	# Params (M)	FLOPs (G)	Top-1 (%)
Swin-B	-	88.0	15.4	83.5
NTK-SAP-Swin-B (Wang et al., 2023)	-	72.6	13.2	83.2
KCR-Swin-B	0.2	70.2	12.6	**84.7**
NTK-SAP-Swin-B (Wang et al., 2023)	-	57.9	10.2	82.8
KCR-Swin-B	0.4	55.4	9.7	**84.2**
NTK-SAP-Swin-B (Wang et al., 2023)	-	49.3	8.4	82.2
KCR-Swin-B	0.6	46.7	7.8	**83.4**
NTK-SAP-Swin-B (Wang et al., 2023)	-	40.2	6.3	81.5
KCR-Swin-B	0.8	39.5	6.0	**82.9**

Table 13: Performance of KCR-Swin-B under different compression ratios. Increasing λ enforces stronger compression, leading to smaller parameter sizes and FLOPs.

Table 14: Comparisons with baseline methods on ImageNet-1K validation set (inference time measured on 2× NVIDIA A100 40GB GPU, in the precision of FP16, with a batch size = 128).

Model	# Params	FLOPs	Top-1	Inference Time (ms/img)
EfficientViT-B1 (Cai et al., 2023)	9.1 M	0.52 G	79.4	0.312
MLP-Fusion-EfficientViT-B1 (Wei et al., 2023)	7.9 M	0.48 G	79.1	0.298
NTK-SAP-EfficientViT-B1 (Ahmed et al., 2025)	8.0 M	0.49 G	79.4	0.304
DeepCompress-EfficientViT-B1 (Cai et al., 2023)	7.9 M	0.46 G	79.2	0.285
KCR-EfficientViT-B1 (Cai et al., 2023)	7.8 M	0.44 G	**80.4**	**0.271**
ViT-S (Dosovitskiy et al., 2021)	22.1 M	4.3 G	81.2	0.642
MLP-Fusion-ViT-S (Wei et al., 2023)	19.8 M	4.0 G	81.0	0.596
NTK-SAP-ViT-S (Wang et al., 2023)	20.3 M	3.9 G	80.9	0.583
DeepCompress ViT-S (Ahmed et al., 2025)	20.0 M	3.9 G	81.1	0.574
KCR-ViT-S (Ours)	19.8 M	3.8 G	**82.2**	**0.561**
ViT-B (Dosovitskiy et al., 2021)	86.5 M	17.6 G	83.7	1.594
MLP-Fusion-ViT-B (Wei et al., 2023)	70.2 M	15.3 G	83.5	1.442
NTK-SAP-ViT-B (Wang et al., 2023)	71.8 M	15.6 G	83.5	1.467
DeepCompress ViT-B (Ahmed et al., 2025)	70.5 M	15.1 G	83.6	1.426
KCR-ViT-B (Ours)	69.5 M	14.5 G	**84.6**	**1.378**
Swin-T (Liu et al., 2021a)	29.0 M	4.5 G	81.3	0.673
MLP-Fusion-Swin-T (Wei et al., 2023)	24.8 M	4.1 G	81.0	0.631
NTK-SAP-Swin-T (Wang et al., 2023)	25.5 M	4.2 G	81.2	0.644
DeepCompress Swin-T (Ahmed et al., 2025)	24.8 M	4.1 G	81.1	0.622
KCR-Swin-T (Ours)	24.6 M	3.9 G	**82.4**	**0.598**
Swin-B (Liu et al., 2021a)	88.0 M	15.4 G	83.5	1.522
MLP-Fusion-Swin-B (Wei et al., 2023)	70.8 M	13.3 G	83.2	1.384
NTK-SAP-Swin-B (Wang et al., 2023)	72.6 M	13.2 G	83.2	1.369
DeepCompress Swin-B (Ahmed et al., 2025)	71.5 M	13.0 G	83.1	1.357
KCR-Swin-B (Ours)	70.2 M	12.6 G	**84.7**	**1.312**

to 70.2M, the FLOPs from 15.4G to 12.6G, and the inference time from 1.522 ms/image to 1.312 ms/image. These results underscore the practical deployment advantages of KCR regularization, which enhances accuracy and efficiency without sacrificing runtime performance.

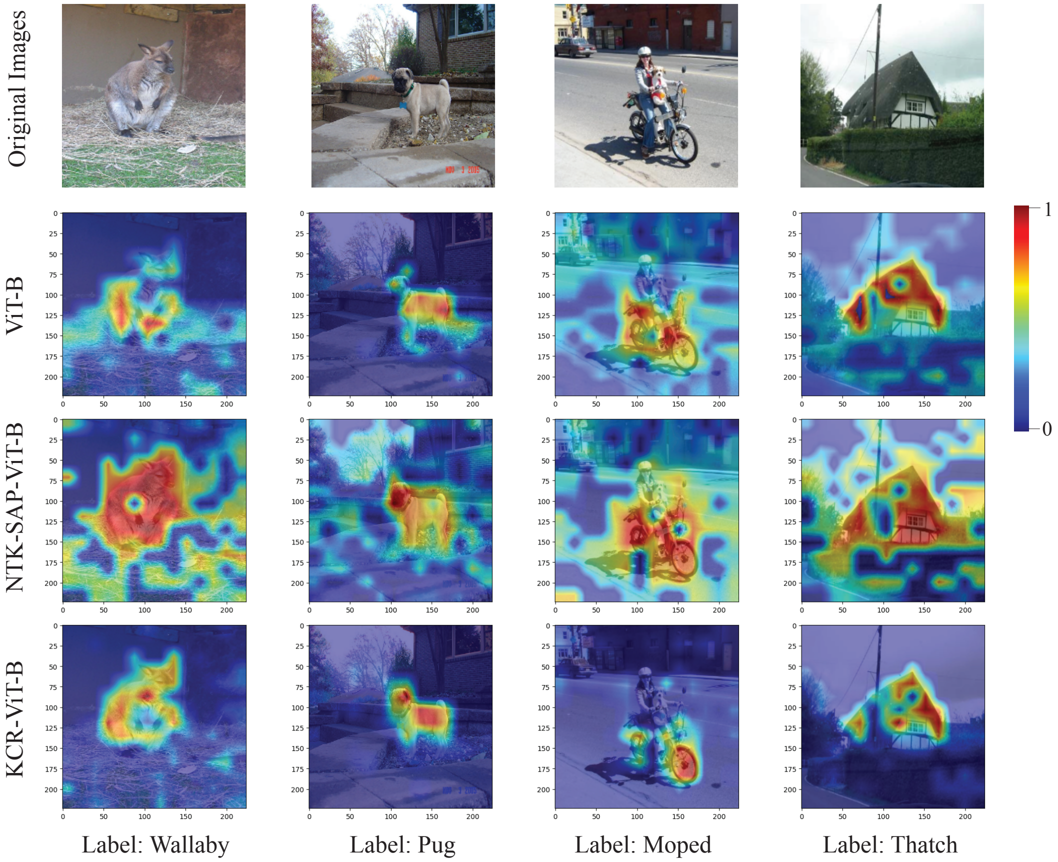


Figure 2: Grad-CAM visualization across four ImageNet classes, including wallaby, pug, moped, and thatch. The first row illustrates the original images. The second, third, and fourth row illustrates the Grad-CAM heatmaps from ViT-B (Dosovitskiy et al., 2021), NTK-SAP-ViT-B (Wang et al., 2023), and KCR-ViT-B, respectively.

D.8 Visualization Results

To qualitatively assess the discriminative capacity and spatial focus of different transformer variants, we apply the Grad-CAM technique (Selvaraju et al., 2017) to visualize the class-specific activation regions in the input images. We compute the Grad-CAM heatmaps for images from four representative ImageNet classes, including wallaby, pug, moped, and thatch, comparing the baseline ViT-B, NTK-SAP-ViT-B, and our KCR-ViT-B. Figure 2 illustrates that the proposed KCR-ViT-B consistently generates activation maps that are more spatially focused on the salient regions of the target object, while suppressing irrelevant background signals. In contrast, the attention maps of the ViT-B and NTK-SAP-ViT-B tend to exhibit higher activation in non-discriminative areas, such as sky, grass, or surrounding clutter, which may introduce unnecessary noise into the prediction process. The above observation suggests that the proposed KC regularization not only facilitates compression and computational efficiency but also enhances the model's representation learning capability for the classification task by guiding it to attend more selectively to task-relevant features.

D.9 Study on the Impact of KC

To assess the influence of KC on the performance of the KCR-Transformer, we perform an ablation study in which models have identical FLOPs and parameter sizes but different KC values. By varying the balancing weight η of the KCR regularization term in the training loss, the resulting models achieve different KC levels while maintaining matched model sizes. It is observed in Table 15 that reducing KC up to a certain range improves classification accuracy, whereas overly aggressive KC reduction eventually leads to a slight degradation in performance.

Table 15: Imapct of KC on the Top-1 Accuracy. The study is performed on KCR-Swin-B. The first row in the table denotes the baseline uncompressed Swin-B model.

η	KC	Top-1 (%)
-	3.21	83.5
0.01	1.75	83.9
0.05	0.87	84.3
0.2	0.52	**84.7**
0.4	0.52	84.6
0.6	0.49	84.5
0.8	0.47	84.4
1.0	0.45	84.5
1.5	0.43	84.5
2.0	0.41	84.4
2.5	0.40	84.3

D.10 Ablation Study on the KCR Regularization

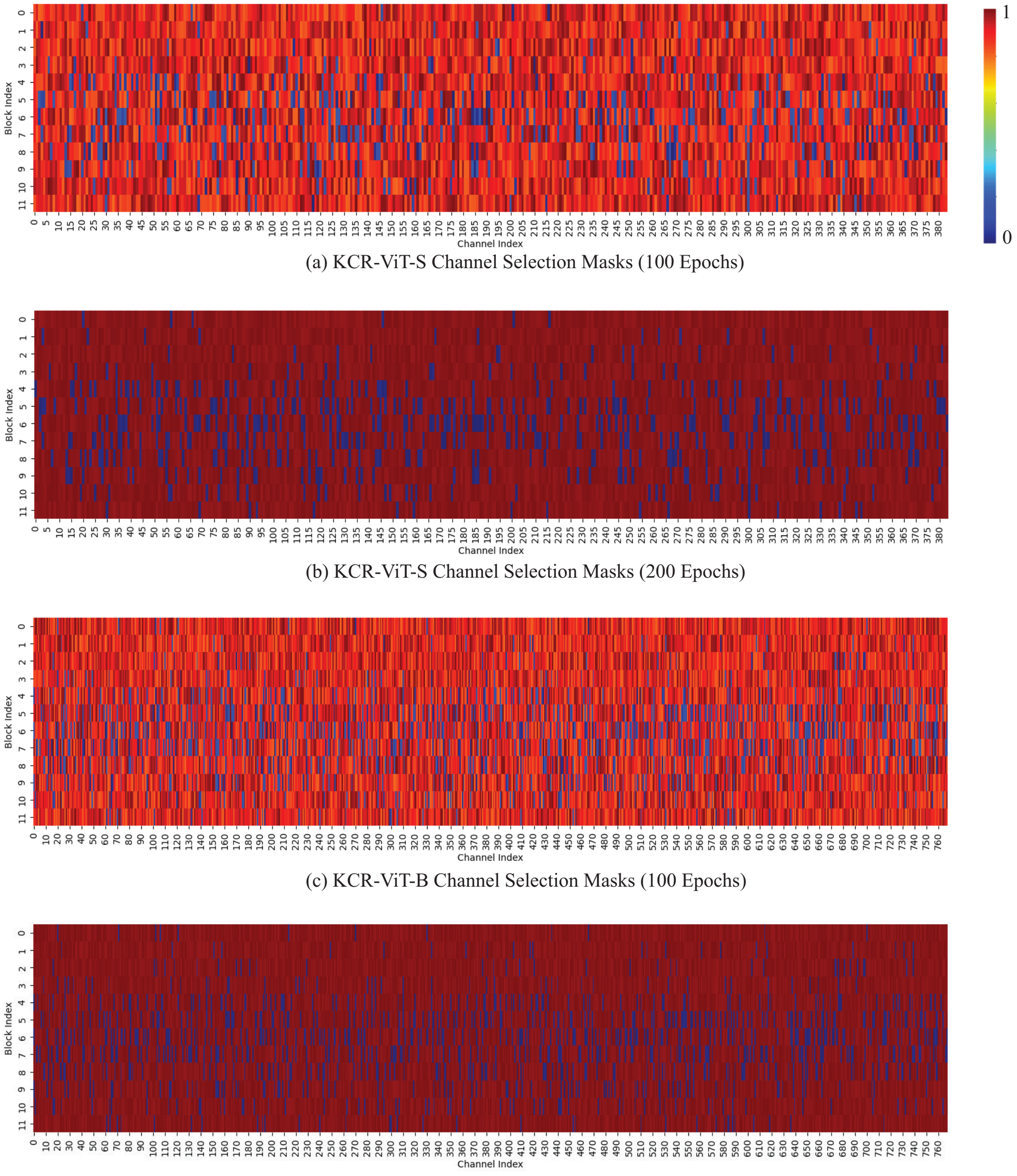
To study the effect of the two-stage training procedure independently from the contribution of the kernel-based regularization, we performed an additional ablation study in which the KCR regularization is disabled by setting $\eta = 0$ while keeping the same two-stage training pipeline for channel selection. This baseline isolates the impact of the KCR term itself. It is observed in Table 16 that the two-stage training alone does not lead to an improvement in model performance. In contrast, incorporating the KCR regularization leads to consistent performance improvements across different architectures, which demonstrates that the improvement primarily arises from the kernel-based regularization, which is aligned with the theoretical analysis in Theorem 3.1. In particular, reducing the KC through the KCR regularization effectively lowers the upper bound for the generalization error, leading to better generalization capability and classification accuracy.

Table 16: Ablation Study on the Impact of KCR Regularization.

Models	Top-1 (%)
ViT-B	83.7
KCR-ViT-B without KCR Regularization	83.3
KCR-ViT-B	**84.6**
Swin-B	83.5
KCR-Swin-B without KCR Regularization	83.1
KCR-Swin-B	**84.7**

D.11 Visualization of Channel Selection Masks

To further improve the interpretability of the channel selection mechanism by the Gumbel-Softmax operation, we illustrate the learned channel-selection masks for KCR-ViT-S and KCR-ViT-B in Figure 3. Each row in the heatmap corresponds to a layer, and each column corresponds to a channel of that layer. As the Gumbel-Softmax temperature decrease in the search phase, the learned masks evolve toward near-binary channel-selection masks, with most probabilities pushed close to either



(a) KCR-ViT-S Channel Selection Masks (100 Epochs)

(b) KCR-ViT-S Channel Selection Masks (200 Epochs)

(c) KCR-ViT-B Channel Selection Masks (100 Epochs)

(d) KCR-ViT-B Channel Selection Masks (200 Epochs)

Figure 3: Visualization of learned Gumbel–Softmax channel-selection masks for KCR-ViT-S and KCR-ViT-B. Figures (a) and (b) illustrate the channel selection masks for ViT-S after 100 and 200 epochs in the search phase for KCR-ViT-S. Figures (c) and (d) illustrate the channel selection masks for ViT-S after 100 and 200 epochs in the search phase for KCR-ViT-B. The entire search phase takes 200 epcohs.

0 or 1 at the end of the search phase. Furthermore, we observe that the shallow layers retain a larger proportion of channels, whereas the middle layers undergo a more aggressive reduction in the number of channels preserved.