Structured Initialization for Vision Transformers

Jianqiao Zheng* **Xueqian Li** **Hemanth Saratchandran** **Simon Lucey**

Australian Institute for Machine Learning
The University of Adelaide

Abstract

Convolutional Neural Networks (CNNs) inherently encode strong inductive biases, enabling effective generalization on small-scale datasets. In this paper, we propose integrating this inductive bias into ViTs, not through an architectural intervention but solely through initialization. The motivation here is to have a ViT that can enjoy strong CNN-like performance when data assets are small, but can still scale to ViT-like performance as the data expands. Our approach is motivated by our empirical results that random impulse filters can achieve commensurate performance to learned filters within a CNN. We improve upon current ViT initialization strategies, which typically rely on empirical heuristics such as using attention weights from pretrained models or focusing on the distribution of attention weights without enforcing structures. Empirical results demonstrate that our method significantly outperforms standard ViT initialization across numerous small and medium-scale benchmarks, including Food-101, CIFAR-10, CIFAR-100, STL-10, Flowers, and Pets, while maintaining comparative performance on large-scale datasets such as ImageNet-1K. Moreover, our initialization strategy can be easily integrated into various transformer-based architectures such as Swin Transformer and MLP-Mixer with consistent improvements in performance.

1 Introduction

Despite their success on large-scale training datasets, Vision Transformers (ViTs) often suffer a notable drop in performance when trained on small-scale datasets. This limitation is primarily attributed to their lack of architectural inductive biases, which are crucial for generalization with insufficient data. In contrast, Convolutional Neural Networks (CNNs) possess strong inductive biases that allow them to perform well even with limited training data.

To bridge this performance gap, several strategies have been proposed. These include self-supervised pretraining on large-scale datasets [8, 26], advanced data augmentation techniques [36, 6], and hybrid architectures that incorporate convolutional layers into Vision Transformers [32, 16, 35, 15, 7]. More recently, Zhang *et al*. [37] explored the use of pretrained weights to initialize ViTs. Building on this idea, subsequent works have shown that carefully designed network initialization strategies can enhance ViT performance on small-scale datasets without modifying the model architecture. In particular, Trockman and Kolter [28] introduced mimetic initialization that replicates the weight distribution of pretrained ViTs. Similarly, Xu *et al*. [33] proposed directly sampling weights from large pretrained models.

While effective, these methods come with three notable limitations: (1) they focus on replicating the distribution of pretrained attention weights rather than structuring attention maps; (2) they rely on access to pretrained models trained on large-scale data, which is often impractical in domain-specific scenarios; and (3) their effectiveness is often tied to specific model architectures.

*jianqiao.zheng@adelaide.edu.au. Code is available at `https://github.com/osiriszjq/structured_initialization`

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

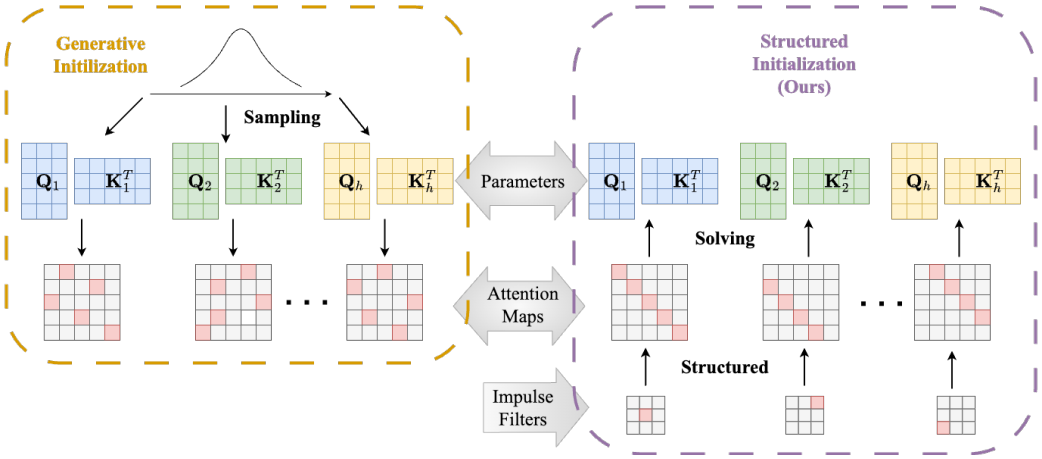


Figure 1: Illustration of conventional generative initialization and structured initialization (ours) strategies for the weights $\mathbf{Q}$ and $\mathbf{K}$ of the attention map in transformers. Conventional generative initialization involves sampling $\mathbf{Q}$ and $\mathbf{K}$ from certain distributions, such as Gaussian or Uniform, resulting in unstructured attention maps. In contrast, our structured initialization imposes constraints on the structure of the initial attention maps, specifically requiring them to be random impulse filters. The initialization of $\mathbf{Q}$ and $\mathbf{K}$ is computed based on this requirement. Note that in both attention maps and random impulse filters, the pink cells indicate ones, while the gray cells represent zeros.

To overcome the limitations of existing approaches, we propose a novel "structured initialization" strategy for ViTs that imposes convolutional structures on attention maps without requiring any pretrained models. Our approach is grounded in our theoretical insights, aligned with the findings of recent work [3], that randomly initialized depthwise convolution filters can match the performance of their trained counterparts in models such as ConvMixer [27] and ResNet [11]. Inspired by this, we develop an initialization strategy for ViTs based on random impulse convolution kernels, which impart locality and structure directly into the attention maps, as shown in Fig. 1. This structured initialization yields inductive biases characteristic of CNNs, enabling ViTs to generalize effectively on small-scale datasets, while preserving their adaptability for large-scale applications. Unlike prior methods that modify ViT architectures by integrating convolutional components, our method preserves the original transformer structure, making it broadly applicable across a variety of ViT variants.

To conclude, our paper makes the following contributions:

- We establish a conceptual link between the structural inductive bias of CNNs and initialization in ViTs, and provide a theoretical justification for using random convolution filters to initialize attention maps.
- To the best of our knowledge, we are the first to introduce the initialization strategy that explicitly structures attention maps in ViTs, embedding convolutional inductive biases without modifying model architecture, enabling compatibility across diverse ViT variants.
- We demonstrate state-of-the-art performance on small-scale and medium-scale datasets, including Food-101, CIFAR-10, CIFAR-100, STL-10, Flowers, and Pets, while maintaining competitive results on large-scale datasets such as ImageNet-1K, and achieving improved performance across various ViT architectures, such as Swin Transformers and MLP-Mixer.

2 Related Work

Introducing inductive bias of CNN to ViT through architecture. Many efforts have aimed to incorporate a convolutional inductive bias into ViTs through architectural modifications. [7] proposed to combine convolution and self-attention by mixing the convolutional self-attention layers. [21, 15]introduced hybrid models wherein the output of each layer is a summation of convolution and self-attention. [32] explored using convolution for token projections within self-attention, while [35] showed promising results by inserting a depthwise convolution before the self-attention map. [9] introduced gated positional self-attention to imply a soft convolution inductive bias. Although these

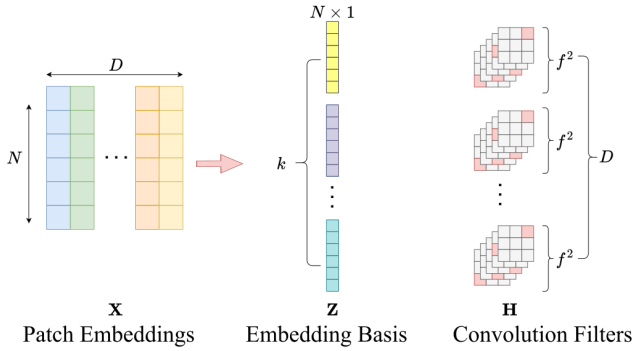


Figure 2: Illustration of why random spatial convolution filters are effective. Patch embeddings $\mathbf{X} \in \mathbb{R}^{N \times D}$ are typically rank-deficient and can be approximately decomposed to k basis. Meanwhile, a linear combination of f^2 linearly independent filters $\mathbf{h}$ can express any arbitrary filter in the filter space $\mathbb{R}^{f \times f}$. Based on these two observations, we derive the inequality $D \geq kf^2$ from Proposition 1.

techniques have been proven effective, they aim to introduce the inductive bias of convolution through architectural choices. Our approach, on the other hand, stands out by not requiring any modifications to the architecture, retaining the generalizability to be seamlessly applied to different settings.

Initializating ViTs from pretrained weights. The exploration of applying inductive bias through initialization within a transformer is limited to date. [37] posited that the benefit of pretrained models in ViTs can be interpreted as a more effective strategy for initialization. [28, 29] recently investigated the empirical distributions of self-attention weights, learned from large-scale datasets, and proposed a mimetic initialization strategy. While this approach lies between structured and generative initialization, it relies on the pretraining results of large models. [23, 24, 33, 14] directly sampled weights from pretrained large-scale models as initialization for smaller models. While effective, the sampled weights must follow the distribution of these pretrained weights. A key difference in our approach is that our method does not require offline knowledge of pretrained models. Instead, our initialized structure is derived from a theoretical analysis of convolution layers.

Convolution as attention. Since their introduction [30, 8], the relationship between transformers and CNNs has been a topic of immense interest. [1] studied the structural similarities between attention and convolution, bridging them into a unified framework. Building on this, [5] demonstrated that self-attention layers can express any convolutional layers through a careful theoretical construction. While these studies highlighted the functional equivalence between self-attention in ViTs and convolutional spatial mixing in CNNs, they did not delve into how the inductive bias of ViTs could be adapted through this theoretical connection. In contrast, our work offers a theoretical insight: a random convolutional impulse filter can be effectively approximated by softmax self-attention.

3 Why Random Impulse Filters Work?

It is well established that both ConvMixer and ViTs use alternating blocks of spatial and channel mixing, where ViTs replace the spatial convolutions in ConvNets with attention mechanisms. A fundamental difference, except for the receptive field, lies in their parameterization: spatial convolutions with hundreds of channels typically use distinct kernels for each channel, whereas self-attention relies on a shared mechanism with only a limited number (~ 10) of attention heads. By introducing a key observation that input embeddings are often rank-deficient, we demonstrate that as long as the spatial kernels or attention patterns sufficiently span the kernel space, the uniqueness or repetition of individual kernels becomes less critical. This insight, supported by a recent empirical observation [3], offers a new perspective on the relationship between ViTs and CNNs, motivating the use of impulse-based structures to embed convolutional inductive biases into the attention map initialization.

In recent work [3], Cazenavette *et al*. demonstrated a remarkable performance of randomly initialized convolution filters in ConvMixer and ResNet when solely learning the channel mixing parameters. However, they failed to offer any insights into the underlying reasons. In this section, we provide a theoretical analysis of how solely learning channel mixing can be sufficient for achieving reasonably

good performance. Our theoretical findings are significant as they establish a conceptual link between the architecture of ConvMixer and the initialization of ViT, offering a deeper understanding of desired properties for spatial mixing matrices. Without losing generality, we have omitted activations (*e.g.*, GeLU, ReLU, *etc.*), bias, batch normalization, and skip connections in our equations for clarity.

Remark 1 *Let us define the patch embeddings or intermediate layer outputs in ConvMixer as* $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_D]$*, where* D *is the number of channels and* N *is the number of pixels in the vectorized patch* $\mathbf{x} \in \mathbb{R}^N$*. An interesting observation is the rank (stable rank, defined as* $\sum \sigma^2 \,/\, \sigma^2_{max}$*) of* $\mathbf{X}$ *is consistently much smaller than the minimum dimension* $min(N, D)$ *of* $\mathbf{X}$*, indicating a significant amount of redundancy in patch embeddings or intermediate layer outputs in deep networks. This rank deficiency is common in various deep neural networks [10], especially in ViTs [20, 18, 12].*

Let us define a 2D convolution filter as $\mathbf{h} \in \mathbb{R}^{f \times f}$. In general, this kernel can be represented as a circulant matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$, such that $\mathbf{h} * \mathbf{x} = \mathbf{H}\mathbf{x}$, where $*$ denotes convolution operator. The relation between the convolutional matrix and convolution filters is explained in detail in the appendix. A ConvMixer block $\mathbf{T}^{\text{Conv}} : \mathbb{R}^{N \times D} \to \mathbb{R}^{N \times D}$ is composed of a spatial mixing layer $\mathbf{T}^{\text{Conv}}_S : \mathbb{R}^{N \times D} \to \mathbb{R}^{N \times D}$ and a channel mixing layer $\mathbf{T}^{\text{Conv}}_C : \mathbb{R}^{N \times D} \to \mathbb{R}^{N \times D}$, where $\mathbf{T}^{\text{Conv}}_S$ is defined by a sequence of convolution filters $\mathbf{H} = [\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_D] \in \mathbb{R}^{D \times N \times N}$, $\mathbf{H}_i \in \mathbb{R}^{N \times N}$, and $\mathbf{T}^{\text{Conv}}_C$ is defined by a weight matrix $\mathbf{W} \in \mathbb{R}^{D \times D}$. With input $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_D] \in \mathbb{R}^{N \times D}$, $\mathbf{T}^{\text{Conv}}$ is

$$\mathbf{T}^{\text{Conv}}_S(\mathbf{X}; \mathbf{H}) = [\mathbf{H}_1\mathbf{x}_1, \mathbf{H}_2\mathbf{x}_2, \ldots, \mathbf{H}_D\mathbf{x}_D], \tag{1}$$

$$\mathbf{T}^{\text{Conv}}_C(\mathbf{X}; \mathbf{W}) = \mathbf{X}\mathbf{W}, \tag{2}$$

$$\mathbf{T}^{\text{Conv}}(\mathbf{X}) = \mathbf{T}^{\text{Conv}}_C(\mathbf{T}^{\text{Conv}}_S(\mathbf{X}; \mathbf{H}); \mathbf{W}) = [\mathbf{H}_1\mathbf{x}_1, \mathbf{H}_2\mathbf{x}_2, \ldots, \mathbf{H}_D\mathbf{x}_D]\mathbf{W}. \tag{3}$$

Definition 1 ***(M–k Spanned Set):*** *Let* $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\} \subset \mathbb{R}^d$ *be a finite set of vectors. We say that* $\mathcal{V}$ *is* M*–*k spanned *(on* W*) if there exists a partition of* $\mathcal{V}$ *into at least* k *non-overlapping subsets:*

$$\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \ldots \cup \mathcal{V}_k, \tag{4}$$

such that there exist a M*–dimensional subspace* $W \subset \mathbb{R}^d$ *in the span of each subset* $\mathcal{V}_i$

$$W \subseteq \textit{Span}(\mathcal{V}_i) \quad \forall i = 1, 2, \ldots, k. \tag{5}$$

Proposition 1 *A ConvMixer block* $\mathbf{T}$ *consists of a spatial mixing layer* $\mathbf{T}_S(\,\cdot\,; \mathbf{H})$ *with convolution filters* $\mathbf{H}$ *and a channel mixing layer* $\mathbf{T}_C$*.* $\mathbf{T}'$ *is another ConvMixer block composed of* $\mathbf{T}'_S(\,\cdot\,; \mathbf{H}')$ *and* $\mathbf{T}'_C$*. Let* k *be the rank of input* $\mathbf{X}$*. If* $\mathbf{H}$ *is* M*–*k *spanned on* W*, then for any set of filters* $\mathbf{H}' \subset W$ *and any* $\mathbf{T}'_C$*, there always exists a* $\mathbf{T}_C$ *such that* $\mathbf{T}(\mathbf{X}) = \mathbf{T}'(\mathbf{X})$*.*

For simplicity, we include the full proof in the appendix. Note that since H are convolution matrices, their span lies in $\mathbb{R}^{f \times f}$ instead of $\mathbb{R}^{N \times N}$, where f is the kernel size. In practice, the fact that $D \geq kf^2$ indicates that randomly initialized spatial convolution kernels are f^2*–*k spanned and satisfy Proposition 1, as illustrated in Fig. 2. Consequently, any trained results of $\mathbf{T}'_C$ and $\mathbf{T}'_S$ can be achieved by solely training the $\mathbf{T}_C$, while keeping a fixed spatial mixing layer $\mathbf{T}_S$. Hence, the following corollaries can be obtained, and Corollary 1 explains the phenomenon found in [3], mentioned at the beginning of the section. Corollary 2 inspires our proposed structure initialization for ViTs. The related experimental evidence of these corollaries and our proposition is given in the appendix.

Corollary 1 *Randomly initialized spatial convolution filters perform as well as trained spatial convolution filters since the* f^2*–*k *spanned condition in Proposition 1 is satisfied.*

Corollary 2 *Random impulse spatial convolution filters perform as well as trained spatial convolution filters since the* f^2*–*k *spanned condition in Proposition 1 is satisfied.*

Corollary 3 *Spatial convolution filters with all ones (referred to as "box" filters) will not perform well since they are* 1*–*k *spanned and can produce only averaging values.*

4 Structured Initialization for Attention Map

4.1 Expected Initialized Attention Map Structure

ConvMixer and ViT share most of the components in their architectures. The gap in their performance on small-scale datasets stems from their architectural choices regarding the spatial mixing matrix.

Although depthwise convolution (ConvMixer) and multi-head self-attention (ViT) may appear distinct at first glance, their underlying goal remains the same: to identify spatial patterns indicated by the spatial mixing matrix. As defined in Sec. 3, similar to the spatial mixing step in ConvMixer defined in Eq. (1), the spatial mixing step of multi-head attention can be expressed as

$$\begin{aligned}\mathbf{T}_S^{\text{ViT}}(\mathbf{X};\mathbf{M}) = [&\mathbf{M}_1\mathbf{x}_1,\ldots,\mathbf{M}_1\mathbf{x}_d,\mathbf{M}_2\mathbf{x}_{d+1},\ldots,\mathbf{M}_2\mathbf{x}_{2d},\\ &\ldots,\mathbf{M}_h\mathbf{x}_{(h-1)d+1},\ldots,\mathbf{M}_h\mathbf{x}_{h*d}],\end{aligned} \tag{6}$$

where d represents the feature dimension in each head, typically set to D/h, with h being the number of heads, and the matrices $\mathbf{M}_i$ for multi-head self-attention can be expressed as follows:

$$\mathbf{M}_i \ = \text{softmax}(\mathbf{X}\mathbf{Q}_i\ \mathbf{K}_i^T\mathbf{X}^T), \tag{7}$$

where $\mathbf{Q}_i$, $\mathbf{K}_i \in \mathbb{R}^{D\times d}$ are attention weight matrices.

It is worth noting that in Eq. (1), the spatial matrices $\mathbf{H}$ are in convolutional structure, resulting in a span of $\mathbb{R}^{f\times f}$ instead of $\mathbb{R}^{N\times N}$, despite each $\mathbf{H}_i$ having a size of $N\times N$. This structural constraint ensures that CNNs focus on local features but struggle to capture long-range dependencies. In contrast, the span of spatial matrices $\mathbf{M}$ in Eq. (6) is $\mathbb{R}^{N\times N}$, allowing for greater learning capacity without these limitations. However, a randomly initialized $\mathbf{Q}$ and $\mathbf{K}$ contain no structural information, resulting in random matrices as depicted in the bottom left of Fig. 1.

Leveraging this insight, we propose to initialize the attention map for each head in ViT to a convolutional structure as denoted in the bottom right of Fig. 1. Our initialization strategy preserves both the advantage of locality and the capacity to learn long-range information. For clarity and brevity, the following discussions will focus only on one head of multi-head self-attention. Therefore, from Eq. (6) and Eq. (1), our structured initialization strategy can be represented as

$$\mathbf{T}_S^{\text{ViT}}(\mathbf{X};\mathbf{M}) \xleftarrow{\text{init}} \mathbf{T}_S^{\text{Conv}}(\mathbf{X};\mathbf{M}) \ \Rightarrow\ \mathbf{M}_{\text{init}} = \text{softmax}(\mathbf{X}\mathbf{Q}_{\text{init}}\mathbf{K}_{\text{init}}^T\mathbf{X}^T) \approx \mathbf{H}. \tag{8}$$

Why using impulse filters? Any random convolution filters that satisfy proposition 1 could be a choice of initialization of attention maps to introduce inductive bias. However, random convolution filters $\mathbf{H}$ usually contain both positive and negative values, while the output of the softmax function is always positive, making Eq. (8) unreachable. One straightforward option is to use random positive convolution filters with a normalized sum of one, following the property of softmax. However, this approach often proves inefficient as the patterns are too complicated for a softmax function to handle with $\mathbf{QK}$ being of low rank. In [25], the authors found that the softmax attention map serves as a feature selection function, which typically tends to select a single related feature. In convolution filters, this selection can be parameterized as impulse filters. According to Corollary 2, random impulse filters are also f^2–k spanned. In conclusion, when initializing a softmax attention map, the most straightforward and suitable choice is random impulse convolution filters.

Pseudo input. The advantage of self-attention is that its spatial mixing map is learned from data. The real input to an attention layer is $\mathbf{P}+\mathbf{X}$ for the first layer and $\mathbf{X}$ (the intermediate output from the previous layer) for the following layers. However, during initialization, there is no prior information about the input. To address this problem, we simply use the initialization of positional encoding $\mathbf{P}$ as the pseudo input in the initialization of $\mathbf{Q}$ and $\mathbf{K}$, replacing the actual input data $\mathbf{P}+\mathbf{X}$ or intermediate outputs. Remember that this only happens when we solve the initialization to avoid data-dependent initialization, while in the training stage, we make no change to the ViT architecture.

With the use of impulse filters and the pseudo input, Eq. (8) becomes

$$\mathbf{M}_{\text{init}} = \text{softmax}(\mathbf{P}\mathbf{Q}_{\text{init}}\mathbf{K}_{\text{init}}^T\mathbf{P}^T) \approx \mathbf{H}_{\text{impulse}}. \tag{9}$$

4.2 Solving $\mathbf{Q}_{\text{init}}$ and $\mathbf{K}_{\text{init}}$

There exist numerous approaches to solve Eq. (9) for $\mathbf{Q}_{\text{init}}$ and $\mathbf{K}_{\text{init}}$ with known $\mathbf{H}_{\text{impulse}}$ and $\mathbf{P}$. Here, we apply an SVD-based method. First, we change Eq. (9) to exclude the Softmax function as

$$\mathbf{P}\mathbf{Q}_{\text{init}}\mathbf{K}_{\text{init}}^T\mathbf{P}^T = \alpha\mathbf{H}_{\text{impulse}} + \beta\mathbf{Z}\,, \tag{10}$$

where $\mathbf{Z}\sim\mathcal{N}(0,\frac{1}{D}\mathbf{I})$. Then to solve for $\mathbf{Q}_{\text{init}}$ and $\mathbf{K}_{\text{init}}$, we put $\mathbf{P}$ to the right-hand side of the equation using the pseudo inverse $\mathbf{P}_{\text{inv}} = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T$—since $\mathbf{P}$ is randomly initialized, it will be

of full rank. Therefore, we get

$$\mathbf{Q}_{\text{init}}\mathbf{K}_{\text{init}}^{T} = \mathbf{P}_{\text{inv}}(\alpha\mathbf{H}_{\text{impulse}} + \beta\mathbf{Z})\mathbf{P}_{\text{inv}}^{T}\,. \tag{11}$$

With Eq. (11), we do SVD on the right-hand side and get a low-rank approximation of $\mathbf{Q}_{\text{init}}$ and $\mathbf{K}_{\text{init}}$:

$$\mathbf{U}, \mathbf{s}, \mathbf{V}^{T} = \text{SVD}\left(\mathbf{P}_{\text{inv}}\left(\alpha\mathbf{H}_{\text{impulse}} + \beta\mathbf{Z}\right)\mathbf{P}_{\text{inv}}^{T}\right), \tag{12}$$

$$\widetilde{\mathbf{Q}}_{\text{init}} = \mathbf{U}\sqrt{\mathbf{s}}\,[:, :d], \quad \widetilde{\mathbf{K}}_{\text{init}} = \mathbf{V}\sqrt{\mathbf{s}}\,[:, :d]\,. \tag{13}$$

Finally, we do a normalization to get the final $\mathbf{Q}_{\text{init}}$ and $\mathbf{K}_{\text{init}}$ as

$$\mathbf{Q}_{\text{init}} = \frac{\gamma}{\|\widetilde{\mathbf{Q}}_{\text{init}}\|_F)}\widetilde{\mathbf{Q}}_{\text{init}}, \quad \mathbf{K}_{\text{init}} = \frac{\gamma}{\|\widetilde{\mathbf{K}}_{\text{init}}\|_F}\widetilde{\mathbf{K}}_{\text{init}}\,. \tag{14}$$

In ViT models with $d = 64$, we use a 3×3 convolution filters by setting $f = 3$. We find that the random values in $\mathbf{Z}$ are not decisive for our initialization strategy. To ensure a clear impulse structure, the initialization is primarily governed by selecting an appropriate ratio of α to β. Empirically, we choose a large ratio such that $\alpha{:}\beta = 40{:}1$ with $\gamma = 2$. Notably, the exact values of α and β are not strictly constrained due to the normalization step, which scales these parameters. The pseudo code for our initialization strategy can be found in the appendix.

5 Experiments and Analysis

In this section, we show that our impulse initialization can improve the performance of ViT on small-scale datasets in Sec. 5.1, while it does not limit the learning flexibility of ViT models on large-scale datasets in Sec. 5.2. In Sec. 5.6, we show that even for a pretrained method like weight selection [33], the pretrained model initialized with our impulse structure can provide better pretrained weights with a relatively smaller-scale dataset. Finally, in Sec. 5.4, we show that besides ViTs, the concept of introducing architectural bias into initializations can also be generalized to other models like Swin Transformers and MLP-Mixer. Note that all experiments were conducted on a single node with 8 Tesla V100 SXM3 GPUs, each with 32GB of memory, if not specified. Specifically, all the experiments on the small-scale datasets took about three hours to train each model, while the experiments on the ImageNet-1K took about two days. Please note that all results were reported based on our experiments with retrained models. As a result, minor discrepancies may exist between our reported results and those in published papers. However, the key focus of this analysis remains on the improvements achieved through different initialization strategies.

5.1 Small and Medium-Scale Datasets

In this section, we compare the performance of ViT-Tiny under three initialization strategies: the default [31], mimetic [28], and our impulse initialization. Experiments are conducted on medium-scale datasets ($\sim$50K training images), including Food-101 [2], CIFAR-10, and CIFAR-100 [13], as well as small-scale datasets ($\sim$5K training images), such as STL-10 [4], Flowers [19], and Pets [22]. We follow the training recipe from [33], which is proven to be useful in training ViT-Tiny on small and medium-scale datasets. Their codes are based on the timm library [31]. By default, all the weights are initialized with a truncated normal distribution. For a fair comparison, all the experiments were run with identical codes except for the choice of initialization methods. Please also note that although [33] initializes the model weights from large pretrained models, we did not adopt any pretraining step for this experiment. In contrast, we apply default, mimetic, and our impulse initialization methods to ViT-Tiny models and training these models from scratch.

The experimental results are presented in Tab. 1. For simplicity, we keep the statistical results on using different seeds with stochastic filter generations with 5 different runs in Appendix F and Fig. 6. Our method consistently yields substantial improvements over the default initialization across all evaluated datasets. On medium-scale datasets, it achieves performance gains of $2\%{\sim}5\%$, while on small-scale datasets, improvements can reach approximately 8%, and in some cases, up to 20%. While mimetic initialization also improves the performance, our impulse initialization shows superior efficacy, attributed to the convolutional structure integrated in the attention initialization. Notably, as the dataset size decreases, the performance gap between our method and the default (or mimetic) initialization gets larger. This observation validates that the convolutional inductive bias introduced by our initialization becomes increasingly important when there is less data for the model to learn the spatial dependencies.

Table 1: Classification accuracy of ViT-Tiny with different initialization methods on different datasets. **Green** number indicates an increase in accuracy. Note that we compare the performance to the default initialization method (shaded in gray). ● represents small-scale datasets, and ▲ represents medium-scale datasets. The datasets are ranked based on their training scales.

Method \ Data↓	▲ Food-101		▲ CIFAR-10		▲ CIFAR-100		● STL-10		● Flowers		● Pets	
Default [31]	77.95		92.29		71.67		61.86		64.60		26.58	
Mimetic [28]	81.78	3.83↑	93.50	1.21↑	75.16	3.49↑	68.54	6.68↑	71.62	7.02↑	47.63	21.05↑
Ours (impulse)	**81.85**	**3.90↑**	**94.67**	**2.38↑**	**77.02**	**5.35↑**	**70.21**	**8.35↑**	**73.18**	**8.58↑**	**50.84**	**24.26↑**

Table 2: Classification accuracy of ViT-Tiny, ViT-Small and ViT-Base on ImageNet-1K dataset with different initialization methods. ■ indicates large-scale datasets. Please note that for the last column (shaded in yellow), we report the experimental results on ViT-Base with specific settings to make the default initialization-based ViT comparable to the results reported in concurrent papers.

Method \ Model	■ ViT-Tiny		■ ViT-Small		■ ViT-Base		■ ViT-Base*	
Default [31]	72.71		79.68		81.24		81.89	
Mimetic [28]	72.90	0.19↑	80.26	0.58↑	80.56	0.68↓	80.56	1.33↓
Ours (impulse)	**72.76**	**0.05↑**	**80.40**	**0.72↑**	**81.83**	**0.59↑**	**82.13**	**0.24↑**

5.2 Large-Scale Datasets

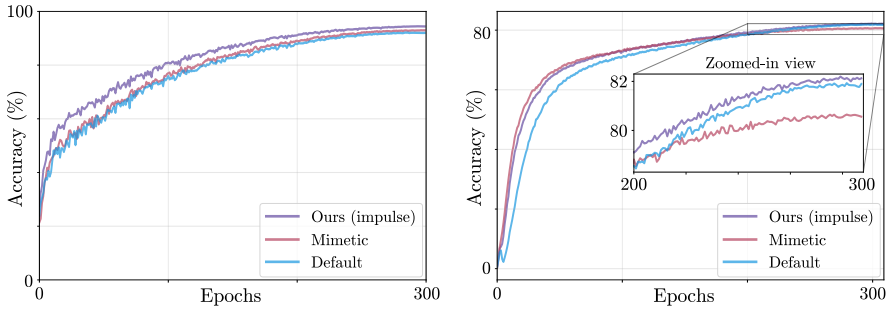
In this section, we compare the performance of ViT-Tiny, ViT-Small, and ViT-base with default, mimetic, and our impulse initialization on a large-scale dataset—ImageNet-1K (over 1M training images). We follow the training recipe from DeiT [26]—a classic and efficient training recipe for training ViT models on ImageNet-1K. We directly use the original ViT structure and training codes in the timm library, except for adding our implementation of initialization. All the models were trained with the same hyperparameters starting from scratch without any pretraining or distillation.

We show the comparison results on the ImageNet-1K dataset in Tab. 2. Detailed training hyperparameters and training curves can be found in Appendix E. In particular, we find that the rapid update for the baseline ViT codebase (*i.e.*, timm library) and the difference in GPU hardware settings result in a small discrepancy in the default initialization-based ViT-Base model between our main result ($\sim$81.24) and the results reported in concurrent papers ($\sim$81.89). Therefore, we have included an additional column (shaded in yellow) in Tab. 2 for ViT-Base* model that uses 16 Tesla V100 GPUs with an extra 0.3 color jittering data augmentation for a clearer comparison.

Despite different training settings and comparisons, our method maintains comparable performance with default initialization, demonstrating that the convolutional inductive bias introduced during initialization does not hinder the model's flexibility in learning data-driven dependencies. This indicates that while the transformer architecture begins training with structurally imposed spatial priors, the attention mechanism retains full capacity to learn optimal feature representations when sufficient training data is available. Furthermore, the convolutional structure we introduced in the attention map initialization not only accelerates early convergence but also improves the robustness to variations in training hyperparameters. We provide more quantitative results in Appendix E.

5.3 Training Curves and Analysis

In Fig. 3, we show the training accuracy curves across 300 epochs for different initialization methods of ViT-Tiny on CIFAR-10 and ViT-Base on ImageNet-1K. For the small model on medium-scale dataset, our impulse initialization consistently outperforms the default or mimetic initialization throughout the entire training process. On the large-scale dataset with training large ViT-Base model, our impulse initialization method and the mimetic initialization have shown faster convergence rate than the default initialization at the beginning of the training. However, the mimetic initialization shows degraded performance even to the default initialization at the last 100 training epochs, indicating limited ability in large-scale model training. On the contrary, our method does not limit the learning ability of the large-scale ViT models, showing a significant advantage in the final training stage where the performance surpass both the default and the mimetic initialization methods.



(a) ViT-Tiny on CIFAR-10 (b) ViT-Base on ImageNet-1K

Figure 3: Training curves of ViT-Tiny on CIFAR-10 and ViT-Base on ImageNet-1K using default, mimetic, and impulse initialization. The zoomed-in box shows the training curve in the final training stage from epoch 200 to epoch 300.

Table 3: Classification accuracy of different Swin Transformers and MLP-Mixer on different datasets with different initialization methods. **Red**/**green** number indicates accuracy decrease/increase. We compare the performance to the default initialization method (shaded in gray).

Model&Data / Method	Swin-B on ■ ImageNet-1K	Swin-T on ▲ CIFAR-10	MLP-Mixer on ▲ CIFAR-10
Default [31]	83.14	89.85	87.00
Mimetic [28]	83.14 —	89.24 0.61↓	—
Ours (impulse)	**83.55** **0.41↑**	**91.19** **1.34↑**	**88.78** **1.78↑**

In summary, experiments on both small-scale and large-scale datasets show that our initialization method effectively balancing prior architectural knowledge integration with data adaptability—incorporating a convolutional inductive bias during initialization provides structural guidance to capture dependencies when training data is limited, while still allowing the attention mechanism to retain full learning flexibility in scenarios with abundant data.

5.4 ViT Variants

Beyond the original ViT architectures, we extend a simplified version of our initialization strategy to the Swin Transformer. Notably, Swin Transformers incorporate relative positional encoding within each attention block—a design choice that aligns with our core objective of instilling convolutional structure into attention maps. For these models, we achieve this by directly initializing the relative positional embeddings with our impulse pattern.

We evaluated our approach with Swin Transformer-Base (Swin-B) architecture on the ImageNet-1K following the training recipe in the original Swin Transformer paper [17]. To further demonstrate the flexibility of our approach, we applied our impulse initialization to the MLP-Mixer. For experiments with Swin Transformer-Tiny (Swin-T) and MLP-Mixer, we follow training settings in [34], which is specifically designed for training different models on CIFAR-10.

The results are summarized in Tab. 3. For models with strong learning capacity but limited inherent inductive biases at initialization, introducing a structured initialization consistently enhances performance without compromising the model's capacity to learn complex data dependencies. In particular, while Swin Transformer incorporates convolutional inductive bias through windowed self-attention, applying our structured initialization further improves the performance by 1.34% on smaller-scale datasets such as CIFAR-10, with no degradation on large-scale datasets like ImageNet-1K. MLP-Mixer, which replaces the spatial convolutions in ConvMixer with MLP layers, typically struggles to train effectively on small-scale datasets. However, initializing the spatial MLPs with a convolutional structure leads to a 1.78% performance gain on CIFAR-10. In contrast, mimetic initialization—designed to replicate empirical weight distributions from pretrained ViTs—shows negligible benefits or degrades performance, highlighting its limited generalizability outside the specific pretrained ViT structures.

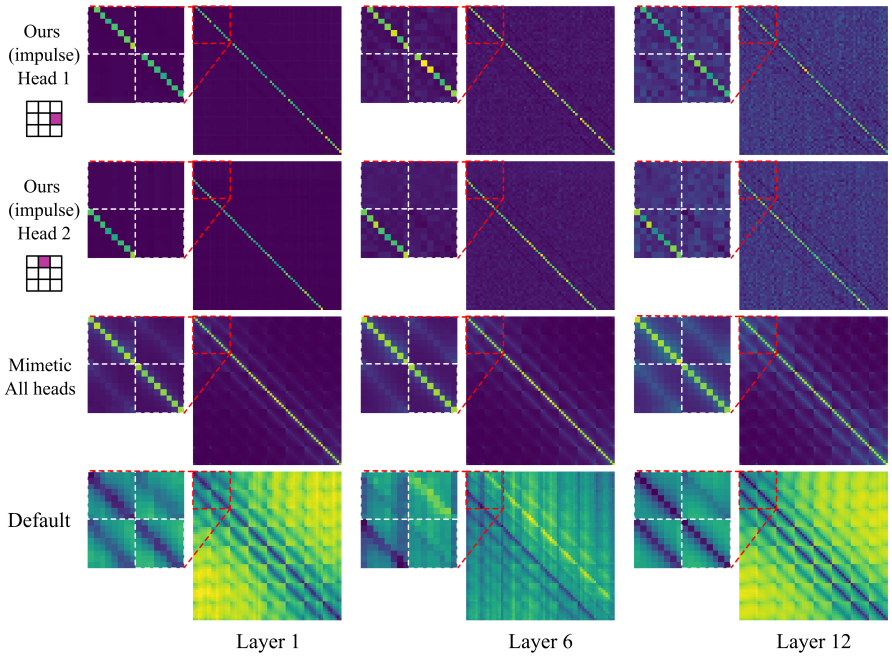


Figure 4: Visualization of attention maps in ViT-T using ours, mimetic [28], and default [33] initializations. Red boxes highlight zoomed-in details of the 16×16 upper left corner in attention maps. White boxes indicate the 8×8 sub-blocks of the zoomed-in attention maps. Our structured initialization method offers distinct attention peaks aligned with the impulse structures across different heads. Head 1 offers a peak at $+1$ offset from the main diagonal. Head 2 offers a peek at -8 offset (equivalent to $-$image_size) from the main diagonal. Both mimetic and random initialization methods initialize all the attention heads identically. Specifically, mimetic initialization primarily strengthens the main diagonal of the attention map for each head, while random initialization shows minimal structural patterns with flatter peak values.

5.5 Attention Maps

The initialized attention maps produced by default, mimetic, and our impulse initializations are shown in Fig. 4. For better visualization, we use 32×32 CIFAR-10 images as input with a 4×4 patch size, yielding the token size as $8\times8=64$. We observe that the default initialization generates near-identical attention values with little distinguishable spatial structure, while the mimetic initialization introduces diagonally dominant values by adding an identity matrix during initialization. Notably, these two initialization methods initialize all the attention heads in the same way. In contrast, our initialization assigns different impulse patterns to each attention head, producing spatially diverse activations with off-diagonal peaks on specific impulse positions.

5.6 Pre-trained Model

In this section, we demonstrate that our initialization method remains compatible with existing pretraining pipelines. Building on the work of Xu *et al.* [33], who showed that weight selection from ImageNet-21K-pretrained ViT-Small models effectively initializes ViT-Tiny architectures, we extend this approach to ImageNet-1K pertaining. We pretrained three ViT-Small models on ImageNet-1K using three initialization strategies: default, mimetic, and our proposed impulse initialization. From these, we derived three ViT-Tiny initialization variants—termed "1K-default", "1K-mimetic", and "1K-impulse"—using the same weight selection methodology on smaller-scale datasets. When training these initialized ViT-Tiny models on small and medium-scale datasets (results shown in Tab. 4), we observed that: (1) Switching from ImageNet-21K to ImageNet-1K—less training data—pretraining with default initialization typically incurs a performance drop of $\sim1\%$. (2) Mimetic initialization fails to mitigate this data degradation, (3) The ImageNet-1K pretraining model with our impulse-based

Table 4: Classification accuracy of ViT-Tiny on small-scale datasets with the weight selection method. Here, the weights selected for the experiments were pretrained on the ImageNet-1K (shortened as 1K, shown above the **purple** dashed line) and ImageNet-21K (shortened as 21K, shown below the **purple** dashed line) datasets with different initialization methods. Please note that pretraining on ImageNet-21K with default initialization is the original weight selection method [33]. We compare the performance to the default initialization method pretrained on ImageNet-1K (shaded in **gray**).

Method+Model \ Data↓	▲ Food-101		▲ CIFAR-10		▲ CIFAR-100		● STL-10	
1K+Default [33]	85.42		96.61		79.64		82.58	
1K+Mimetic [28]	86.32	0.90↑	96.37	0.24↓	79.86	0.22↑	82.39	0.19↓
1K+Ours (impulse)	**87.43**	**2.01**↑	**97.19**	**0.58**↑	80.92	1.28↑	**83.89**	**1.31**↑
21K+Default [33]	87.14	1.72↑	97.07	0.46↑	**81.07**	**1.43**↑	83.23	0.65↑

initialization achieves performance parity with ImageNet-21K pretrained baselines. This highlights the robustness of our method even under reduced pretraining data regimes.

6 Limitations and Broader Impacts

There exist several limitations in our initialization method: **(1) Positional encoding.** Although positional encoding is a natural and effective choice for pseudo-inputs—due to its simplicity and data-independent nature—even simpler alternatives may exist for initializing the $\mathbf{Q}$ and $\mathbf{K}$ matrices. In this work, we focus solely on the initialization of $\mathbf{Q}$ and $\mathbf{K}$, but a more comprehensive initialization strategy that also considers patch embeddings and positional encodings could offer greater control over the structure of attention maps. Notably, since positional encoding is only added at the input layer, the influence on attention maps may diminish with increasing network depth, as illustrated in Fig. 4. **(2) Hard constraints.** Our Corollary 2 of Proposition 1 is based on the presumption that the filters are f^2–k spanned, which is usually a characteristic inherent in CNNs. However, in ViTs, the limited number of heads may be inadequate to span the filter space of a small kernel. Finding better adaptations in this scenario remains a challenge. **(3) Value initialization.** Our method does not consider the initialization for the value weights $\mathbf{V}$ and the projection matrix.

Broader impacts. This work advances the understanding of how structured initialization influences the performance of transformers, particularly in resource-constrained settings. Our research findings enables more efficient training of neural networks on small-scale datasets, which may benefit domains such as medical imaging, environmental monitoring, robotics, or education, where data is limited or expensive to collect. Furthermore, improving initialization strategies can reduce computational costs, contributing to more sustainable AI practices.

However, as with most advances in artificial intelligence, these techniques carry a risk of misuse or harmful societal applications. For example, by applying our initialization method to more powerful models with fewer resources could make larger models more easily accessible for malicious purposes. These concerns further remind us to carefully consider the broader societal impacts of our research and make sure its benefits outweigh potential harms.

7 Conclusion

In this paper, we propose a structured initialization method with convolutional impulse filters for attention maps in ViTs. Our method preserves both the advantage of locality within CNNs and the capacity to learn long-range dependencies inherited from ViTs. We also provide a thorough theoretical explanation of the spatial and channel mixing in ConvMixer and ViT, building connections between the structural bias in CNNs and the initialization of ViTs. Our results on small-scale datasets validate the effectiveness of the convolutional structural bias, while on-par performance on large-scale datasets indicates the preservation of architectural flexibility. Our initialization also accelerates early-stage convergence but also enhances the model's robustness to variations in training hyperparameters. Furthermore, we demonstrate that our method consistently provides benefits across a range of architectures and even under pre-training strategies.

References

[1] Andreoli, J.M.: Convolution, attention and structure embedding. arXiv preprint arXiv:1905.01289 (2019)

[2] Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13. pp. 446–461. Springer (2014)

[3] Cazenavette, G., Julin, J., Lucey, S.: Rethinking the role of spatial mixing. arXiv preprint arXiv:2503.16760 (2025)

[4] Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 215–223. JMLR Workshop and Conference Proceedings (2011)

[5] Cordonnier, J.B., Loukas, A., Jaggi, M.: On the relationship between self-attention and convolutional layers. In: Eighth International Conference on Learning Representations-ICLR 2020 (2020)

[6] Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 702–703 (2020)

[7] Dai, Z., Liu, H., Le, Q.V., Tan, M.: Coatnet: Marrying convolution and attention for all data sizes. Advances in neural information processing systems **34**, 3965–3977 (2021)

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

[9] d'Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: Improving vision transformers with soft convolutional inductive biases. In: International Conference on Machine Learning. pp. 2286–2296. PMLR (2021)

[10] Feng, R., Zheng, K., Huang, Y., Zhao, D., Jordan, M., Zha, Z.J.: Rank diminishing in deep neural networks. Advances in Neural Information Processing Systems **35**, 33054–33065 (2022)

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

[12] Ji, Y., Saratchandran, H., Moghaddam, P., Lucey, S.: Always skip attention. arXiv preprint arXiv:2505.01996 (2025)

[13] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

[14] Li, A., Tian, Y., Chen, B., Pathak, D., Chen, X.: On the surprising effectiveness of attention transfer for vision transformers. Advances in Neural Information Processing Systems **37**, 113963–113990 (2024)

[15] Li, K., Wang, Y., Zhang, J., Gao, P., Song, G., Liu, Y., Li, H., Qiao, Y.: Uniformer: Unifying convolution and self-attention for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)

[16] Liu, Y., Sangineto, E., Bi, W., Sebe, N., Lepri, B., Nadai, M.: Efficient training of visual transformers with small datasets. Advances in Neural Information Processing Systems **34**, 23818–23830 (2021)

[17] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)

[18] Naderi, A., Saada, T.N., Tanner, J.: Mind the gap: a spectral analysis of rank collapse and signal propagation in transformers. arXiv preprint arXiv:2410.07799 (2024)

[19] Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian Conference on Computer Vision, Graphics and Image Processing (Dec 2008)

[20] Noci, L., Anagnostidis, S., Biggio, L., Orvieto, A., Singh, S.P., Lucchi, A.: Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. Advances in Neural Information Processing Systems **35**, 27198–27211 (2022)

[21] Pan, X., Ge, C., Lu, R., Song, S., Chen, G., Huang, Z., Huang, G.: On the integration of self-attention and convolution (2021)

[22] Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)

[23] Samragh, M., Farajtabar, M., Mehta, S., Vemulapalli, R., Faghri, F., Naik, D., Tuzel, O., Rastegari, M.: Weight subcloning: direct initialization of transformers using larger pretrained ones. arXiv preprint arXiv:2312.09299 (2023)

[24] Samragh, M., Mirzadeh, I., Vahid, K.A., Faghri, F., Cho, M., Nabi, M., Naik, D., Farajtabar, M.: Scaling smart: Accelerating large language model pre-training with small model initialization. arXiv preprint arXiv:2409.12903 (2024)

[25] Tarzanagh, D.A., Li, Y., Thrampoulidis, C., Oymak, S.: Transformers as support vector machines. In: NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning (2023)

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

[27] Trockman, A., Kolter, J.Z.: Patches are all you need? arXiv preprint arXiv:2201.09792 (2022)

[28] Trockman, A., Kolter, J.Z.: Mimetic initialization of self-attention layers. In: International Conference on Machine Learning. pp. 34456–34468. PMLR (2023)

[29] Trockman, A., Willmott, D., Kolter, J.Z.: Understanding the covariance structure of convolutional filters. arXiv preprint arXiv:2210.03651 (2022)

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

[31] Wightman, R.: Pytorch image models. `https://github.com/rwightman/pytorch-image-models` (2019). https://doi.org/10.5281/zenodo.4414861

[32] Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 22–31 (2021)

[33] Xu, Z., Chen, Y., Vishniakov, K., Yin, Y., Shen, Z., Darrell, T., Liu, L., Liu, Z.: Initializing models with larger ones. In: International Conference on Learning Representations (ICLR) (2024)

[34] Yoshioka, K.: vision-transformers-cifar10: Training vision transformers (vit) and related models on cifar-10. `https://github.com/kentaroy47/vision-transformers-cifar10` (2024)

[35] Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., Wu, W.: Incorporating convolution designs into visual transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 579–588 (2021)

[36] Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6023–6032 (2019)

[37] Zhang, Y., Backurs, A., Bubeck, S., Eldan, R., Gunasekar, S., Wagner, T.: Unveiling transformers with lego: a synthetic reasoning task. arXiv preprint arXiv:2206.04301 (2022)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have clearly summarized the contributions of our paper at the end of the introduction section. The theoretical and experimental results shown in the paper accurately reflect the contributions of our proposed method.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have clearly stated the limitations of our proposed method and the broader impact in the main paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We have provided the complete, correct, detailed proof and the full set of assumptions for every theoretical result in both the main paper and the appendix. And we have numbered and cross-referenced all the theorems, formulas, and proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have provided sufficient experiments and the detailed initialization strategies to validate the effectiveness of our proposed method in the main paper and the appendix. We have also released the full code to the public.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the data used in the paper are open-sourced datasets. We have provided the implementation details and have released the code to reproduce the results claimed in our paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided all the training and testing details, including datasets, hyperparameters, hardware settings *etc*. in the main paper and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: We have reported the error bars for the main results in the appendix. For larger model training, we skip the error bar report due to the expensive computation. Nevertheless, we have set the experimental seed for all the experiments and have ensured accurate and the same experimental settings and hyperparameters for different methods on the same experiment. We have also provided the detailed implementation strategies and have released the full code for reproducing the main results in the paper.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the computing resources we used in the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that our research, in every respect, conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: We have discussed both potential positive and negative societal impacts in the broader impact section in the main paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All the datasets, codes, and models used in the paper are open-sourced. And we have clearly referenced these assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We did not introduce any new assets in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our research does not involve any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our research does not involve any crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix: Structured Initialization for Vision Transformers

A Frequently Asked Questions

Q: Does introducing CNN structural biases to Transformer contradict the advantage of its structure?

A: Our method focuses on introducing the architectural inductive bias from CNNs to initialize the attention map without changing the Transformer architectures. We have emphasized this argument in the abstract and introduction, stating that unlike previous arts [35, 15, 7, 32, 16] that directly introduce convolutions into attention, potentially damaging the structural advantages of Transformers, our method only introduces architectural inductive bias in attention map initialization, maintains the inherent structural flexibility in ViTs. Therefore, our method preserves the Transformer architectures and still allows the ViTs to learn flexible, dynamic global relationships. This is also one of the central innovations of our method. In addition, we have also provided experiments on large-scale applications to validate the stable performance of our method that preserves the architectural flexibility of ViTs.

Q: Why is the proposed method better than transfer learning on large-scale pretrained models or a hybrid architecture combining CNN and attention?

A: As stated in Sec. 1, we would like to emphasize the advantages of our proposed structured initialization method: 1) Unlike previous methods that do transfer learning on large-scale pre-trained ViTs, our method involves no pre-training of large-scale models on large-scale datasets, which may not be readily available; 2) Our method shares the advantages of both CNNs and ViTs.

In general, introducing inductive bias in CNNs for initializing the attention map helps ViTs begin learning from a more reasonable/stable starting point, while the default random initialization can lead to a noisy starting point, especially when training on small-scale datasets. Notably, our initialization method does not alter the ViT structure, maintaining the advantages of the Transformer architectures in learning dynamic, long-dependent global features. In contrast, methods that directly combine the architectures of CNNs and ViTs alter the Transformer architectures, potentially compromising its architectural advantages. Please also refer to the theoretical analysis of random filters in Sec. 3 and the convolutional representation matrix in Appendix C of the main paper for more theoretical explanations.

For pretrained models, we noted in the introduction section that their reliance on access to pretrained models makes them impractical for domain-specific scenarios. For example, pretraining a large model on extensive datasets only to deploy a smaller model on limited data is inefficient and often unreasonable. Furthermore, their effectiveness heavily depends on the performance of the specific model architectures, offering no inherent advantages from using pretrained weights. Nevertheless, even on pretrained models, our initialization still outperforms other methods. For additional quantitative evidence, please refer to the experimental analysis in Sec. 5.1 and Sec. 5.6.

Q: Why use an impulse filter? Does enforcing a strict structure in initialization degrade performance?

A: While one might expect that a rigid initialization could limit the model's flexibility, our experimental results in Sec. 5 demonstrate that the impulse structure does not hurt the training. On the contrary, it consistently improves the performance, particularly on small and medium-scale datasets. This is attributed to the structured initialization introducing a beneficial inductive bias, which guides the model toward learning useful representations in the early stage of training.

Moreover, the hyperparameters α, β, and γ control the norms of $\mathbf{Q}$ and $\mathbf{K}$, ensuring that the attention maps exhibit a well-defined convolutional structure at initialization, while maintaining sufficiently flexible to adapt and learn from data during training. This design shows a central innovation of our method, and its effectiveness is consistently supported by experiments across both small and large-scale datasets.

Although, as suggested by Proposition 1, any set of random convolutional filters can be used to initialize attention maps, it is difficult to obtain both a clear attention map structure and sufficient

training flexibility from random non-impulse filters, especially when analyzed through low-rank approximation via singular value decomposition (SVD) and the Softmax operation. Thus, when aiming to initialize attention maps with a convolutional prior, impulse filters remain the most straightforward and robust choice.

B Proof for Proposition 1

Let $\mathbf{w} = [w_1, w_2, \ldots, w_D]^T \in \mathbb{R}^{D\times 1}$ be the channel mixing weights for one output channel and $\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_D$ are the corresponding spatial convolution filters for each channel. Therefore, the result $\mathbf{y} \in \mathbb{R}^N$ after spatial and channel mixing can be represented as,

$$\mathbf{y} = \sum_{i=1}^{D} w_i \mathbf{H}_i \mathbf{x}_i, \tag{15}$$

With Remark 1, we can suppose the rank of $\mathbf{X} \approx \mathbf{ZA}$ is k, where $\mathbf{Z} = [\mathbf{z}_1, \ldots, z_k]$ and $k \ll D$, as illustrated in Fig. 2. We then obtain

$$\mathbf{y} \approx \sum_{i=1}^{D}\sum_{j=1}^{k} w_i a_{ji} \mathbf{H}_i \mathbf{z}_j = \sum_{j=1}^{k} \tilde{\mathbf{H}}_j \mathbf{z}_j, \tag{16}$$

where a_{ji} refers to the row j, column i element of $\mathbf{A}$, and $\tilde{\mathbf{H}}_j = \sum_{i=1}^{D} w_i \, a_{ji} \, \mathbf{H}_i$.

Remember that a linear combination of f^2 linearly independent filters $\mathbf{h}$ can express any arbitrary filter in filter space $\mathbb{R}^{f\times f}$, where $\mathbf{h}$ serves as the basis. Consequently, any desired $\tilde{\mathbf{H}}_1, \tilde{\mathbf{H}}_2, \ldots, \tilde{\mathbf{H}}_D$ can be achieved by only learning the channel mixing weights $\mathbf{w}$. Therefore, we obtain the following proposition.

C Convolutional Represetation Matrix

In Sec. 3, we interchangeably use the terms convolution filter $\mathbf{h}$ and convolution matrix $\mathbf{H}$. Additionally, we represent the impulse filter as a convolutional matrix. Here, we offer a detailed explanation of the relationship between the convolutional filters and the convolutional matrices.

Let us define a 2D convolution filter as $\mathbf{h} \in \mathbb{R}^{f\times f}$ with elements

$$\mathbf{h} = \begin{pmatrix} h_{11} & \cdots & h_{1f} \\ \vdots & \ddots & \vdots \\ h_{f1} & \cdots & h_{ff} \end{pmatrix}. \tag{17}$$

When $\mathbf{h}$ is convolved with an image $\mathbf{x} \in \mathbb{R}^{H\times W}$, this convolution operation is equivalent to a matrix multiplication

$$\text{vec}(\mathbf{h} * \mathbf{x}) = \mathbf{H}\,\text{vec}(\mathbf{x}), \tag{18}$$

where $\mathbf{H}$ is composed from the elements in $\mathbf{h}$ and zeros in the following format:

$$\mathbf{H} = \begin{pmatrix} \mathbf{F_1} & \mathbf{F_2} & \cdots & \mathbf{F_f} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{F_1} & \mathbf{F_2} & \cdots & \mathbf{F_f} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{F_1} & \mathbf{F_2} & \cdots & \mathbf{F_f} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{F_1} & \mathbf{F_2} & \cdots & \mathbf{F_f} \end{pmatrix}, \tag{19}$$

where

$$\mathbf{F_i} = \begin{pmatrix} h_{i1} & h_{i2} & \cdots & h_{if} & 0 & 0 & \cdots & 0 \\ 0 & h_{i1} & h_{i2} & \cdots & h_{if} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{i1} & h_{i2} & \cdots & h_{if} & 0 \\ 0 & \cdots & 0 & 0 & h_{i1} & h_{i2} & \cdots & h_{if} \end{pmatrix}, \tag{20}$$

for $i = 1, 2, \ldots, f$. $\mathbf{F_i}$s are circulant matrices and $\mathbf{H}$ is a block circulant matrix with circulant block (BCCB). Note that convolutions may employ various padding strategies, but the circulant structure remains consistent. Here, we show the convolution matrix without any padding as an example.

Table 5: Ablation study on training settings of ViT-Base on ImageNet-1K dataset. Note that we compare the performance to the default initialization method (shaded in gray).

Scaled LR	Repeated Augment	Default	Mimetic	Ours (impulse)
7e-4	1.0	76.25	79.12 2.87↑	**80.48** **4.23↑**
1e-3	1.0	80.09	80.17 0.08↑	**81.55** **1.07↑**
1e-3	3.0	81.24	80.56 0.68↓	**81.83** **0.59↑**

D Pseudo Code for Solving $\mathbf{Q}_{\text{init}}$ and $\mathbf{K}_{\text{init}}$

Algorithm 1 Convolutional Structured Impulse Initialization for ViT

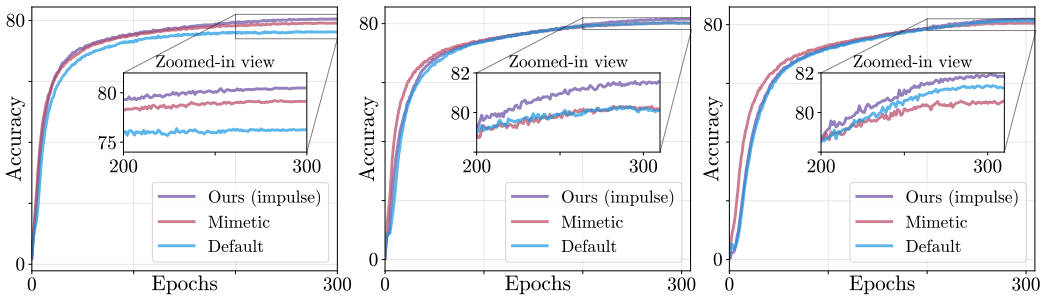
Input: $\mathbf{P}$ ▷ Input positional encoding
Input: $d, f, \alpha, \beta, \gamma$ ▷ Hyperparameters
Output: $\mathbf{Q}_{\text{init}}, \mathbf{K}_{\text{init}}$ ▷ Initialized attention parameters

$N, D \leftarrow \text{shape}(\mathbf{P})$
$\mathbf{H}_{\text{impulse}} \leftarrow \text{ImpulseConvMatrix}(N, f)$ ▷ Build 2D impulse convolution matrix
$\widetilde{\mathbf{M}} \leftarrow \alpha \mathbf{H}_{\text{impulse}} + \beta \mathbf{Z}$ ▷ Get ideal map
$\widetilde{\mathbf{X}} \leftarrow \text{LayerNorm}(\mathbf{P})$ ▷ Get pseudo input
$\mathbf{P}_{\text{inv}} \leftarrow (\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})^{-1} \widetilde{\mathbf{X}}$ ▷ Get pseudo inverse of $\mathbf{P}$
$\hat{\mathbf{M}} \leftarrow \mathbf{P}_{\text{inv}} \widetilde{\mathbf{M}} \mathbf{P}_{\text{inv}}^\top$ ▷ Get patterns $\mathbf{QK}^T$
$\mathbf{U}, \mathbf{s}, \mathbf{V}^\top \leftarrow \text{SVD}(\hat{\mathbf{M}})$ ▷ SVD of $\hat{\mathbf{M}}$
$\widetilde{\mathbf{Q}}_{\text{init}} \leftarrow \mathbf{U}\sqrt{\mathbf{s}}, \quad \widetilde{\mathbf{K}}_{\text{init}} \leftarrow \mathbf{V}\sqrt{\mathbf{s}}$
$\widetilde{\mathbf{Q}}_{\text{init}} \leftarrow \widetilde{\mathbf{Q}}_{\text{init}}[:, : d], \quad \widetilde{\mathbf{K}}_{\text{init}} \leftarrow \widetilde{\mathbf{K}}_{\text{init}}[:, : d]$
$\widetilde{\mathbf{Q}}_{\text{init}} \leftarrow \widetilde{\mathbf{Q}}_{\text{init}} / \text{norm}(\mathbf{Q}), \quad \widetilde{\mathbf{K}}_{\text{init}} \leftarrow \widetilde{\mathbf{K}}_{\text{init}} / \text{norm}(\mathbf{K})$
$\mathbf{Q}_{\text{init}} \leftarrow \gamma \widetilde{\mathbf{Q}}_{\text{init}}, \quad \mathbf{K}_{\text{init}} \leftarrow \gamma \widetilde{\mathbf{K}}_{\text{init}}$
return $\mathbf{Q}_{\text{init}}, \mathbf{K}_{\text{init}}$

E Training Details and Training Curves for ViT-Base

At first, we found it challenging to reproduce the ViT-Base performance reported in DeiT [26], even when strictly following their specified training settings. Upon a careful examination of code differences across various versions of the timm [31] library, we identified two critical discrepancies that likely contributed to the performance gap: **(a) Learning rate scaling strategy:** In DeiT, the learning rate is linearly scaled with respect to the batch size: $\text{lr}_{\text{scaled}} = \text{lr}_{\text{base}} \times \frac{\text{batch size}}{512}$. In contrast, the current version of timm uses a square root scaling rule as the default for the AdamW optimizer: $\text{lr}_{\text{scaled}} = \text{lr}_{\text{base}} \times \sqrt{\frac{\text{batch size}}{512}}$. This discrepancy in the default setting leads to different effective learning rates, even when all the other hyperparameters are identical, and can substantially affect performance. **(b) Repeated data augmentation setting:** DeiT emphasizes the importance of the repeated data augmentation strategy, stating a drop in top-1 accuracy from 81.8% to 76.5% when it is disabled. However, they did not specify the exact augmentation weighting value in their original paper. After inspecting their specific version of code, we discovered that the default value for the repeated data augmentation was set to `3`, whereas we used `1` in our main experiments, which may partially explain the performance discrepancy. We also aligned several other minor settings, such as the minimum learning rate during warm-up and at the end of training, as well as an additional 10 epochs for cool-down. However, we believe that these factors have only a marginal effect on the final performance, while the two aforementioned reasons remain the primary contributors to the observed discrepancy.

We present an ablation study on these two main settings difference in Tab. 5. In Fig. 5, we also present the training accuracy curves across epochs for different initialization methods with three different training configurations. Our impulse initialization consistently outperforms the default initialization throughout the entire training process, with a particularly significant advantage in the final training stage. While the mimetic initialization shows relatively faster initial convergence, it



(a) Scaled LR: 7e-4 Repeated Augementation: 1.0 (b) Scaled LR: 1e-3 Repeated Augmentation: 1.0 (c) Scaled LR: 1e-3 Repeated Augmentation: 3.0

Figure 5: Training curves of ViT-Base using default, mimetic, and impulse initialization under three different training configurations. The zoomed-in box shows the training curve in the final training stage from epoch 200 to epoch 300.

Table 6: Detailed scale of all the datasets used in the paper. ■ represents large-scale datasets, ▲ represents medium-scale datasets, and ● represents small-scale datasets. The datasets are ranked based on their scales following [33]. A '+' means we use both train and validation dataset in training. Note for Flowers and Pets datasets, we use both training and validation data for training.

Dataset ↓	#Classes	#Train images in each class	#Train images in total	#Images in total
■ ImageNet-1K	1K	∼ 1K	1.3M	1.4M
▲ Food-101	101	750	75K	101K
▲ CIFAR-10	10	5K	50K	60K
▲ CIFAR-100	100	500	50K	60K
● STL-10	10	500	5K	13K
● Flowers	102	10+10	2K	8K
● Pets	37	50+50	3.7K	7.3K

ultimately degrades performance in the final training stage. Our structured initialization method demonstrates robustness across different training configurations, consistently yielding over 80% accuracy in all cases.

F Additional Results for Small and Medium-Scale Datasets

F.1 Dataset Scale

For completeness, we provide the detailed dataset scales used in the main paper in Tab. 6. The ordering of the datasets follows that in [33].

F.2 Additional Statistical Results

Here we provided the mean and standard deviation of 5 runs for ViT-Tiny with different initialization methods on small and medium-scale datasets in Fig. 6, which serves as additional statistical results of Tab. 1 in the main paper. Notably, our initialization method still outperforms other methods across small and medium-scale datasets. Especially for the smaller-scale datasets, our methods shows larger performance improvements, aligning with our findings in the main paper.

F.3 Model Convergence Rate

We would like to clarify that while impulse filters mimic convolutional locality, they do not reuse weights like CNNs do—which may explain the fast convergence of CNNs. Interestingly, we did observe a faster convergence when using the mimetic or impulse initialization to replace the vanilla ViT model. We give an example of the training accuracy of ViT-Tiny with different initialization methods on CIFAR-10 during the optimization stage, as shown in Tab. 7. We can clearly see a faster

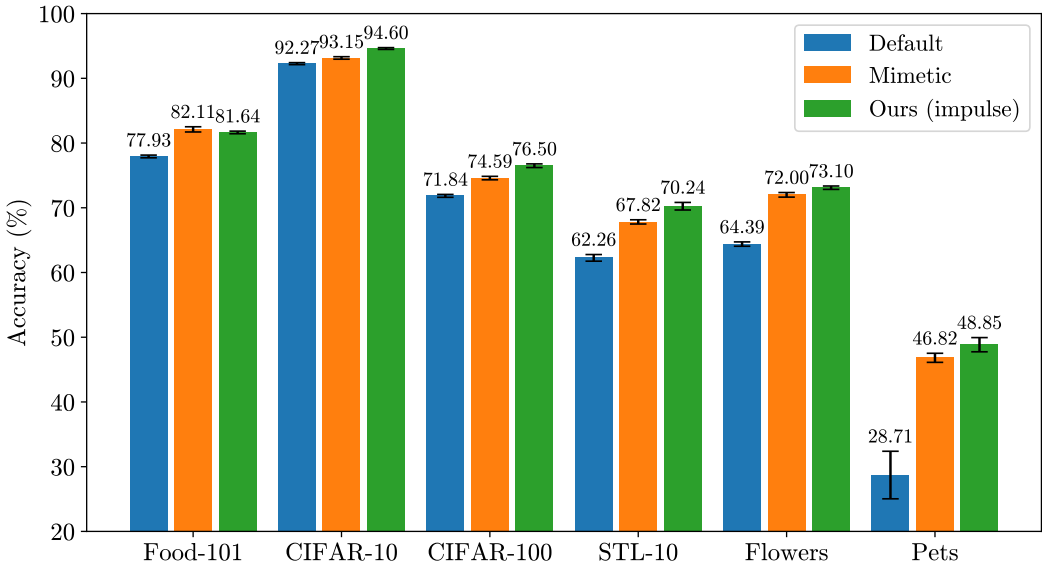


Figure 6: Classification accuracy of ViT-Tiny with different initialization methods on different datasets (mean of 5 runs). Our method consistently outperforms the default and the mimetic initializations. Except for the smallest-scale dataset Pets, all the methods show robust performance across different datasets with small error bars. Note that from left to right, the dataset scale decreases.

Table 7: Comparison of performance when training ViT-Tiny on CIFAR-10 at different epochs. Our method consistently achieves higher accuracy than both default and mimetic initialization baselines.

Epoch	Default	Mimetic	Ours (Impulse)
10	47.95	49.05	**54.62**
50	61.51	63.62	**69.51**
100	76.87	77.64	**81.87**

convergence of mimetic and our impulse initialization methods compared to the default initialization used in vanilla ViTs. One intuitive explanation is that the mimetic initialization yields $\sim 4\times$ the norm values of the default initialization, while our initialization method has $\sim 2\times$ the norm values of the default initialization. Different norm values will affect the gradient magnitudes and early training dynamics.

F.4 Performance Improvement Increases as the Model Size Increases.

In the main paper Tab. 2, we also observe that as the model scale increase (*i.e.*, the number of heads increases), our method becomes relatively more expressive. Theoretically, as explained in Sec. 3, to replace the spatial kernels with fixed filters, the initialized attention heads must span the spatial filter basis. Therefore, for a 3×3 filter used in the experiments, this requires at least 9 independent heads to span the kernel space. Please note that in ConvMixer, each channel uses an independent filter, which satisfies this criterion. However, ViT models share the same filters within each head, which need a separate discussion: 1) ViT-Tiny has 3 heads, ViT-Small has 6 heads—insufficient to span to the kernel space; 2) ViT-Base has 12 heads—sufficient for replacing the spatial kernels. This analysis fully supports this performance improvement increase observation. However, as the motivation of the paper also suggested, training even larger ViT models on small datasets is notoriously challenging due to optimization instability and is out of the scope of this paper.

G Experimental Validation of the Propositions and Corollaries in Sec. 3

In this section, we conduct a series of preliminary experiments on CIFAR-10 to validate the propositions and corollaries in Sec. 3 of the main paper, with particular focus on demonstrating that in

Table 8: Classification accuracy(%) of ConvMixer (depth 8) with different filter sizes, embedding dimensions on CIFAR-10.

Kernel Size	Embedding Dimension = 256				Embedding Dimension = 512			
	Trained	Random	Impulse	Box	Trained	Random	Impulse	Box
3	91.76	90.72	90.68	81.70	92.82	92.15	92.20	81.90
5	92.69	90.87	90.41	80.57	93.90	92.72	91.91	81.19
8	92.34	88.12	87.82	78.95	92.96	90.09	89.61	80.10

ConvMixer, fixed random impulse spatial filters can achieve comparable performance to learned filters. All experiments adhere to the training protocol proposed in [34], which is specifically tailored for evaluating diverse architectures on small-scale datasets such as CIFAR-10.

To support our theoretical findings in Sec. 3 concerning the effectiveness of random filters, we train ConvMixer [27] models with an embedding dimension of 256, a depth of 8, and a patch size of 2 on the CIFAR-10 dataset, using spatial filter sizes of 3, 5, and 8. We also include a variant with an embedding dimension of 512 to examine the impact of feature width. We evaluate the end-to-end trained ConvMixer alongside three initialization strategies: random (Corollary 1), impulse (Corollary 2), and box (Corollary 3). Note that in all three cases, only the spatial convolution filters are initialized—the models are evaluated without any training. The box filters use all-one values, effectively performing average pooling. The results are summarized in Tab. 8.

In conclusion, these experimental results reveal several key insights. First, comparing across columns (*i.e.*, initialization methods), both random and impulse initializations achieve performance comparable ($\geq 90\%$) to that of fully trained models, whereas the box initialization leads to significantly worse performance ($\sim 80\%$). This discrepancy can be attributed to the deficient rank of the box filters, which fail to span the full f^2-dimensional filter space, unlike random and impulse filters that are capable of forming a complete basis.

Second, when comparing across rows (*i.e.*, kernel sizes) with an embedding dimension of 256, the performance gap between trained and untrained (random or impulse) filters grows with kernel size, from 1% (size 3) to 2% (size 5) and 5% (size 8). This occurs because larger kernels require more distinct filters to effectively span the filter space. However, the fixed embedding dimension constrains the number of such filters, reducing their ability to match the input rank as shown in Proposition 1. Notably, when the embedding dimension is doubled to 512, this performance gap narrows. In particular, for a kernel size of 3, the random and impulse initializations nearly match the performance of the trained filters, suggesting that sufficient embedding width compensates for the limitations of fixed filters.

H Attention Maps

Here we provide additional visualization of the attention maps for all 12 layers in Fig. 7. In particular, our structured initialization method offers different attention peaks on various heads, showing alignment with the impulse structures, while the mimetic initialization only presents main-diagonal peaks, and the default initialization shows little to no patterns. As stated in the main paper, both mimetic and default initialization methods use identical initialization for all attention heads.

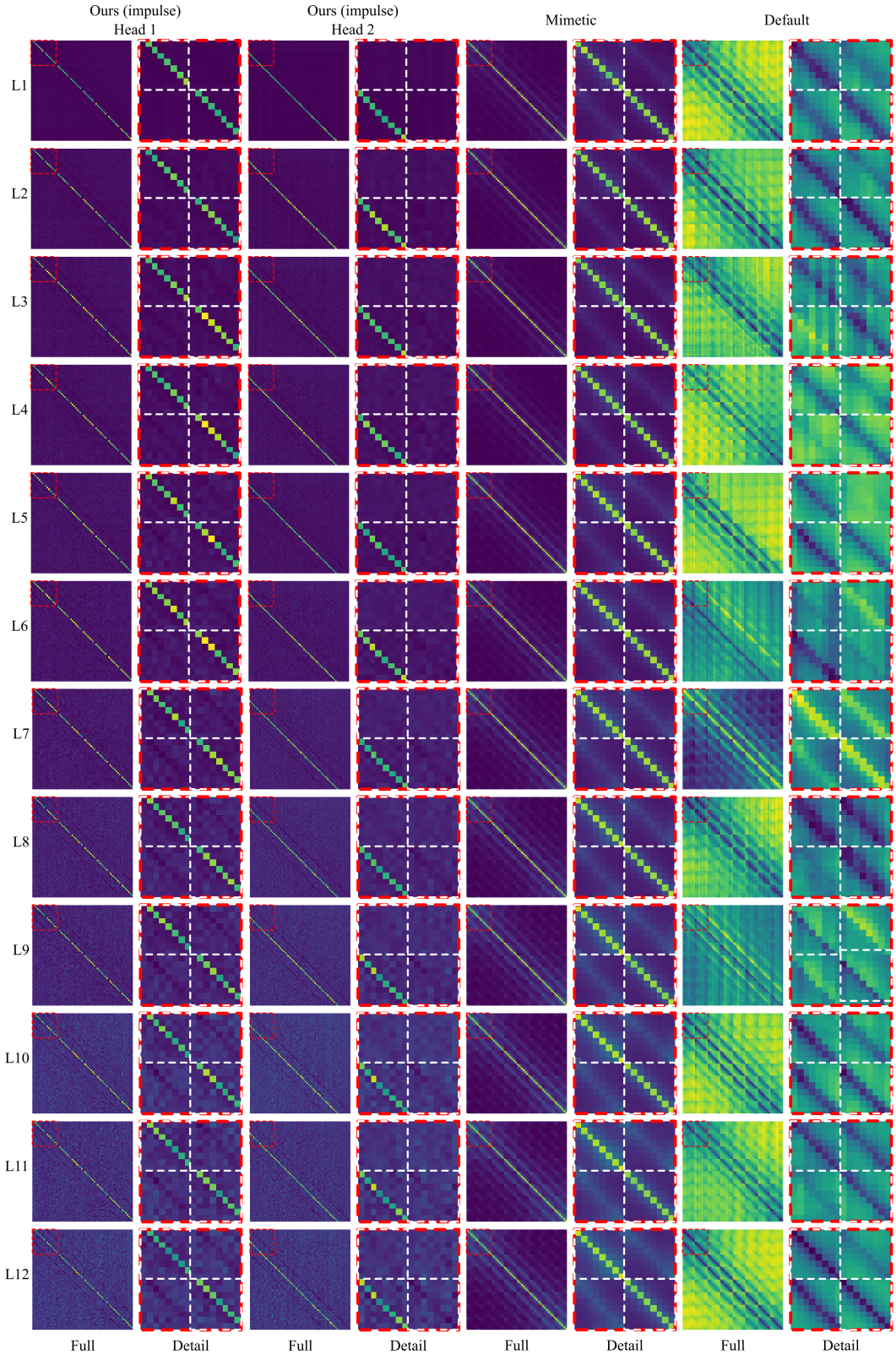


Figure 7: Visualization of attention maps in ViT-T using our impulse initialization method, mimetic [28], and default [31] initializations.