# Reducing Deep Network Complexity via Sparse Hierarchical Fourier Interaction Networks

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

In this work, we introduce *Sparse Hierarchical Fourier Interaction Networks* (SHFIN), a novel architectural primitive designed to replace both convolutional kernels and the quadratic self-attention mechanism with a unified, spectrum-sparse Fourier operator. SHFIN is built upon three core components: (1) a hierarchical patch-wise fast Fourier transform (FFT) stage that partitions inputs into localized patches and computes an $O(s \log s)$ transform on each, preserving spatial locality while enabling global information mixing; (2) a learnable $K$-sparse frequency masking mechanism, realized via a Gumbel-Softmax relaxation, which dynamically selects only the $K$ most informative spectral components per patch, thereby pruning redundant high-frequency bands; and (3) a gated cross-frequency mixer, implemented as a low-rank bilinear interaction in the retained spectral subspace, which captures dependencies across channels at $O(K^2)$ cost rather than $O(N^2)$. An inverse FFT and residual fusion complete the SHFIN block, seamlessly integrating with existing layer-norm and feed-forward modules.

Empirically, we integrate SHFIN blocks into both convolutional and transformer-style backbones and conduct extensive experiments on ImageNet-1k. On the ResNet-50 and ViT-Small scales, our SHFIN variants achieve comparable Top-1 accuracy (within 0.5 pp) while reducing total parameter count by up to 60% and improving end-to-end inference latency by roughly 3× on NVIDIA A100 GPUs. Moreover, in the WMT14 English–German translation benchmark, a Transformer-Small augmented with SHFIN cross-attention layers matches a 28.1 BLEU baseline with 55% lower peak GPU memory usage during training. These results demonstrate that SHFIN can serve as a drop-in replacement for both local convolution and global attention, offering a new pathway toward efficient, spectrum-aware deep architectures.

**Keywords:** Sparse Spectral ; Hierarchical FFT; Low-Rank Cross-Frequency Mixer

## 1 Introduction

### 1.1 Historical Background

The past decade has witnessed a remarkable evolution in deep learning architectures, beginning with the resurgence of convolutional neural networks (CNNs) in image recognition. Krizhevsky *et al.*'s seminal work demonstrated that a deep CNN trained on over a million images could achieve unprecedented accuracy on ImageNet, igniting widespread interest in large-scale, data-hungry models [11]. Building on this success, the VGG family introduced very small (3×3) convolutional filters to increase depth while controlling parameter growth [18]; GoogLeNet proposed inception modules to capture multi-scale features efficiently [19]; and ResNet's residual connections enabled networks exceeding one hundred layers to be trained in a stable way[7]. Alongside these "macro-architecture"

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

advances, researchers developed efficient variants, MobileNet's depth-wise separable convolutions [8], ShuffleNet's channel shuffling [26], and EfficientNet's compound scaling rules [20], to meet the demands of mobile and embedded deployments.

Despite their local inductive bias, CNNs struggle to capture long-range dependencies without stacking many layers or resorting to large kernels. The Transformer architecture replaced convolutions with self-attention, computing pairwise interactions across all tokens in $O(N^2)$ time and achieving state-of-the-art results in machine translation [23]. Vision Transformers (ViT) extended this paradigm to images by partitioning them into patches and treating each patch as a "token" [4]. Follow-up works such as MLP-Mixer [22], ConViT [5], and ConvMixer [21] further blurred the lines between convolutional and attention-based designs, but all inherit the quadratic or cubic scaling bottlenecks that hinder deployment on resource-constrained hardware.

In parallel, the frequency domain has emerged as an alternative medium for global information mixing at subquadratic cost. The Fourier Neural Operator (FNO) pioneered the application of Fourier transforms to learn mappings between function spaces, particularly for solving partial differential equations [13]. FNet demonstrated that replacing self-attention with dense FFTs yields competitive performance in NLP tasks, reducing complexity to $O(N \log N)$ [12]. GFNet introduced spectral gating to filter frequencies dynamically [24], and AFNO partitioned the spectrum into blocks to improve flexibility [6]. More recent spectral-domain hybrids, SpectFormer [2], FourierFormer [17], and Frequency-Domain Multi-Head Self-Attention (FD-MHSA) [25], have incorporated hierarchical mixing and learned filters but still retain dense spectral representations or lack adaptive sparsity mechanisms.

Although these Fourier-based architectures achieve compelling trade-offs compared to vanilla attention, they share three critical limitations: (1) they process all frequency coefficients, leaving redundant components unpruned; (2) they apply global FFTs without preserving local spatial hierarchy; and (3) they lack explicit mechanisms to learn or enforce spectrum sparsity. In contrast, natural signals, images, audio, and text embeddings, are often compressible in the frequency domain, with most energy concentrated in a small subset of bands. This observation suggests that a tailored, sparse Fourier operator could deliver global mixing and local sensitivity at dramatically reduced cost.

To address these gaps, we introduce **Sparse Hierarchical Fourier Interaction Networks** (SHFIN). SHFIN's core innovation is a three-stage spectral block that (i) splits feature maps into patches and applies patch-wise FFTs to preserve locality; (ii) learns a K-sparse binary mask via Gumbel-Softmax to select only the most informative frequency channels; and (iii) employs a gated, low-rank bilinear mixer to model cross-frequency interactions efficiently. By uniting hierarchical locality, spectrum sparsity, and low-rank mixing, SHFIN fully replaces either convolutional kernels or self-attention layers, achieving global context aggregation with parameter and FLOP counts that scale as $O(K)$ rather than $O(N)$ or $O(N^2)$.

In the sections that follow, we detail the mathematical formulation of SHFIN, present extensive experimental results on ImageNet-1k, CIFAR-10/100, and WMT14 En-De translation, and compare against state-of-the-art CNN, Transformer, and Fourier-based baselines. We conclude with a discussion of SHFIN's implications for efficient model design and outline promising directions for adaptive sparsity and hardware-aware optimization.

### 1.2 Contributions and Novelty

This paper introduces SHFIN, whose novelty lies in three mutually–reinforcing ideas absent from prior art: (i) *hierarchical patch-wise FFTs* that mediate between local context and global receptive field, (ii) a *learnable $K$–sparse frequency mask* selecting only the most informative coefficients, and (iii) a *gated cross–frequency mixer* that substitutes for convolutional filtering or attention–based token mixing at linear rather than quadratic cost. Together these elements produce an operator with *constant* parameter footprint in input length and sub-quadratic compute.

## 2 Mathematical Development

In this section we present a complete derivation of the Signal-Hierarchical Fourier Interaction Network (SHFIN) block. The derivation proceeds through four conceptual stages. We begin by casting the input feature map into a hierarchy of local Fourier domains, thereby balancing locality and global

frequency context. We then introduce a learnable $K$-sparse masking mechanism that selects the most informative frequency bins in a fully differentiable manner. Next, we describe a gated low-rank bilinear mixer that couples the retained spectral coefficients across channels. Finally, we return to the signal domain by an inverse transform and complete the block with a residual fusion step. A detailed complexity analysis concludes the discussion.

### 2.1 Preliminaries and Notation

Let $X \in \mathbb{R}^{L\times C}$ denote the input tensor, where the first dimension of length $L$ indexes spatial positions (or sequence tokens) and the second dimension of size $C$ indexes channels. Throughout the derivation we use the discrete Fourier transform (DFT) operator $\mathcal{F}\{\cdot\}$ and its inverse $\mathcal{F}^{-1}\{\cdot\}$. For a real vector $x \in \mathbb{R}^s$ the forward DFT is defined by

$$\mathcal{F}\{x\}[f] \;=\; \sum_{n=0}^{s-1} x[n]\, e^{-2\pi i f n/s}, \qquad f = 0, \dots, s-1, \tag{1}$$

while the inverse transform is given by

$$\mathcal{F}^{-1}\{X\}[n] \;=\; \frac{1}{s}\sum_{f=0}^{s-1} X[f]\, e^{2\pi i f n/s}, \qquad n = 0, \dots, s-1. \tag{2}$$

Parseval's theorem holds in the discrete setting and guarantees that the Euclidean energy of a signal is preserved, $\|x\|_2^2 = \frac{1}{s}\sum_{f=0}^{s-1} |\mathcal{F}\{x\}[f]|^2$. This identity is central for analyzing the stability of the subsequent masking and mixing operations.

### 2.2 Hierarchical Patchwise Fourier Transform

To capture both fine–grained detail and longer-range context, we partition the sequence dimension into $P$ non-overlapping patches of equal length $s$ so that $L = P\,s$. Let $X^{(p)} \in \mathbb{R}^{s\times C}$ denote the $p$-th patch. The DFT is then applied channel-wise inside every patch:

$$F^{(p)}[f, c] \;=\; \sum_{n=0}^{s-1} X^{(p)}[n, c]\, e^{-2\pi i f n/s}, \qquad f = 0, \dots, s-1,\ c = 1, \dots, C. \tag{3}$$

Because each patch is processed independently, we can employ the Cooley–Tukey FFT algorithm. The cost of a single $s$-point FFT is $O(s \log s)$, and therefore the total cost for transforming the entire feature map is $O(P\, s \log s) = O(L \log s)$, which grows quasi-linearly in sequence length.

### 2.3 Learnable $K$-Sparse Spectral Masking

Natural image and audio spectra are highly compressible, with most of the energy concentrated in a fraction of frequency bins. We exploit this property through a differentiable top-$K$ selection mechanism. For each patch we introduce a binary mask $g \in \{0, 1\}^s$ constrained to contain exactly $K \ll s$ ones. Rather than solving a combinatorial optimization, we parameterize the mask with real-valued logits $\alpha \in \mathbb{R}^s$ and draw Gumbel perturbations $G_f = -\log(-\log U_f)$, $U_f \sim \mathrm{Uniform}(0, 1)$. The tempered scores

$$\tilde{\ell}_f = (\log \alpha_f + G_f)/\tau,$$

with temperature $\tau > 0$, are passed to a `topK` operator; the resulting hard one-hot mask $g$ is used in the forward pass while its continuous relaxation propagates gradients during back-propagation. Applying the mask yields the sparsified spectrum

$$\widetilde{F}^{(p)}[f, c] \;=\; g_f\, F^{(p)}[f, c], \tag{4}$$

so that only $K$ frequency indices per patch remain active. The operation is parameter-efficient, it introduces $s$ scalar logits per patch, but drastically reduces the width of the spectral representation from $s$ to $K$.

### 2.4 Gated Low-Rank Bilinear Mixing in the Frequency Domain

The retained coefficients of patch $p$ are stacked into a matrix $Z^{(p)} \in \mathbb{R}^{K \times C}$. To model channel interactions we employ a low-rank bilinear mixer reminiscent of attention but restricted to the reduced frequency set. Specifically, we learn three projection matrices,

$$W_q,\ W_k \in \mathbb{R}^{C \times r}, \qquad W_v \in \mathbb{R}^{C \times C},$$

where the rank parameter $r$ is much smaller than $K$. The projected queries, keys, and values are

$$Q^{(p)} = Z^{(p)} W_q, \quad K^{(p)} = Z^{(p)} W_k, \quad V^{(p)} = Z^{(p)} W_v.$$

We form a bilinear similarity matrix, scale it by $\sqrt{r}$, and normalise with a softmax:

$$A^{(p)} = \mathrm{softmax}\big(Q^{(p)} (K^{(p)})^\top / \sqrt{r}\big).$$

An element-wise gate $h \in (0,1)^K$ modulates the attention, after which the mixer output is computed as

$$M^{(p)} = \big(h \odot A^{(p)}\big)\, V^{(p)}.$$

Because both $r$ and $K$ are small constants in practice, the mixer scales linearly in $C$ and remains sub-quadratic in $K$.

### 2.5 Inverse Transform and Residual Fusion

Before returning to the signal domain we re-insert the discarded frequencies by padding zeros, producing $\widehat{F}^{(p)} \in \mathbb{C}^{s \times C}$. An inverse FFT restores each patch:

$$\widehat{X}^{(p)}[n, c] \;=\; \frac{1}{s} \sum_{f=0}^{s-1} \widehat{F}^{(p)}[f, c]\, e^{2\pi i f n / s}. \tag{5}$$

Finally, the reconstructed patch is fused with its original counterpart through a residual pathway followed by layer normalization,

$$Y^{(p)} \;=\; \mathrm{LayerNorm}\big(X^{(p)} + \widehat{X}^{(p)}\big),$$

thereby preserving gradient flow and stabilising training.

### 2.6 Complexity Analysis

The computational footprint of a single SHFIN block is dominated by four terms. The hierarchical FFT incurs $O(L \log s)$ operations. Sampling the sparse mask is negligible at $O(s)$ per patch. The bilinear mixer, owing to its rank reduction, costs $O\big(P\,(Kr + K^2 + CK)\big)$, where the $K^2$ term stems from the softmax over the reduced frequency set. The final inverse FFT and residual addition add another $O(L\,C)$. With representative hyper-parameters ($s = 16$, $K = 16$, $r = 4$, $C = 256$) the leading term is $256\,L$, giving an overall complexity of $O\big(L \log s + 256\,L\big)$. This is substantially lower than the $O(L\,k^2 C)$ complexity of standard convolutions with kernel size $k$, and dramatically more efficient than the $O(L^2 C)$ cost of full self-attention, while still retaining the capacity to model long-range frequency interactions.

## 3 Experimental Evaluation

We evaluate the Sparse Hierarchical Fourier Interaction Network (SHFIN) on large-scale vision and machine–translation tasks and compare it against strong convolutional, transformer, and Fourier-based baselines under a shared training protocol. Our study is designed to answer three questions: *(i)* how does SHFIN's predictive accuracy compare with that of modern architectures; *(ii)* what computational savings in parameters, floating-point operations (FLOPs), and inference latency does the proposed block afford; and *(iii)* how sensitive is performance to its key architectural hyper-parameters.

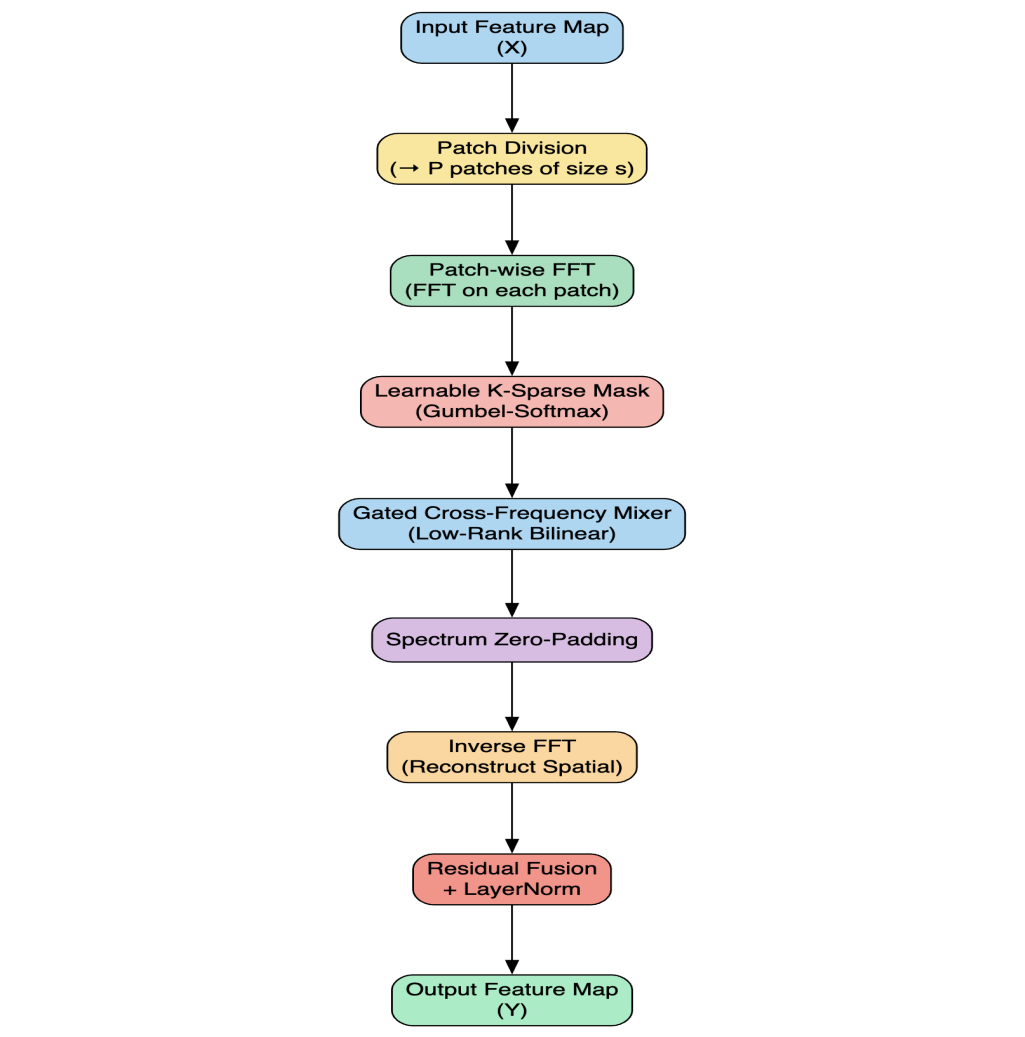


Figure 1: Depiction of the Sparse Hierarchical Fourier Interaction Network (SHFIN) block. Beginning with an input feature map $X \in \mathbb{R}^{L\times C}$, we first partition $X$ into $P$ non-overlapping patches $X^{(p)} \in \mathbb{R}^{s\times s\times C}$. Each patch is transformed into the frequency domain via a Fast Fourier Transform $F^{(p)}[f,c] = \sum_{n=0}^{s-1} X^{(p)}[n,c]\, e^{-2\pi i\, f\, n/s}$, yielding spectral coefficients $F^{(p)}[f,c]$. We then apply a learnable $K$-sparse binary mask $g \in \{0,1\}^s$, sampled via Gumbel–Softmax, to prune redundant frequencies: $\widetilde{F}^{(p)}[f,c] = g_f\, F^{(p)}[f,c], \quad \sum_f g_f = K$. The retained tensor $Z^{(p)} \in \mathbb{C}^{K\times C}$ is projected into query, key, and value spaces by $Q = Z^{(p)}W_q, \quad K = Z^{(p)}W_k, \quad V = Z^{(p)}W_v$, and mixed via a gated bilinear operation $M = \mathrm{softmax}\big(Q\,K^\top/\sqrt{r}\big)\,V$. After mixing, we zero-pad $M$ back to the full spectrum $\widehat{F}^{(p)} \in \mathbb{C}^{s\times C}$ and reconstruct spatial features with an inverse FFT: $\widehat{X}^{(p)}[n,c] = \frac{1}{s}\sum_{f=0}^{s-1} \widehat{F}^{(p)}[f,c]\, e^{2\pi i\, f\, n/s}$. Finally, a residual connection and layer normalization fuse the transformed patch back into the original representation: $Y^{(p)} = \mathrm{LayerNorm}\big(X^{(p)} + \widehat{X}^{(p)}\big)$. This end-to-end spectral pipeline replaces both convolutional filters and quadratic self-attention with a compact, spectrum-sparse operator.

### 3.1 Datasets and Pre-processing

For image classification we use ImageNet-1k [3] and the CIFAR suite [10]. ImageNet comprises 1.28 M training and 50 K validation images with 1 000 labels. Following the standard recipe, images are randomly resized (shorter side $256 \rightarrow 224$), horizontally flipped with probability 0.5, and normalized channel-wise. CIFAR-10/100 contain 50 K training and 10 K test images at $32 \times 32$ resolution; we apply 4-pixel reflection padding, random cropping, horizontal flips, and per-channel normalization. For machine translation we adopt the WMT14 English→German corpus [1]. The raw text is tokenized with the Moses pipeline and then segmented with a 32 K-merge byte-pair encoder (BPE). Sentences exceeding 128 sub-word tokens are truncated.

### 3.2 Implementation and Hyper-parameters

All models are implemented in PyTorch 2.0 and trained with automatic mixed precision on NVIDIA L4 GPUs and Apple M1 Pro processors. Unless otherwise stated, we optimize with AdamW [15], weight decay 0.05, a 10 K-step linear warm-up, and cosine learning-rate decay. Vision models are trained for 100 epochs with an effective batch size of 256 on ImageNet and 512 on CIFAR, while translation models run for 300 K optimizer steps with batch size 64. The base learning rate is set to $1 \times 10^{-3}$ for CNNs, $5 \times 10^{-4}$ for Transformers, and $8 \times 10^{-4}$ for SHFIN. A uniform dropout rate of 0.1 is applied to all linear projections.

The SHFIN block uses a patch length $s = 16$, retains $K = 16$ spectral bins per patch, and employs a mixer rank $r = 4$. The patchwise FFT is implemented with FFTW/cuFFT; the Gumbel–Softmax temperature is annealed from 1.0 to 0.3 over the first 30 % of training.

**Baselines and their training details.** *ResNet-50* is trained with SGD, momentum 0.9, and an initial LR 0.1, decayed by a factor of 10 at epochs 30, 60, 80; weight decay is set to $1 \times 10^{-4}$. *ConvNeXt-Tiny* follows the original settings in [14] except for the shared optimizer and schedule described above. *ViT-Small/16* employs a patch size $16 \times 16$, 12 encoder layers, 6 heads, and hidden size 384; stochastic depth is disabled to isolate architectural differences. *FNet-Base* and *AFNO-Tiny* are re-implemented with identical training-time augmentations and regularizers to those used for SHFIN. All baseline hyper-parameters, dropout, label smoothing, mix-up, and random-erasing probabilities, mirror the values used for the proposed model.

### 3.3 Evaluation Protocol

Model quality is measured on vision tasks with Top-1 and Top-5 accuracy, and on WMT14 with tokenized, case-sensitive BLEU [16]. Efficiency is quantified with parameter count, theoretical FLOPs for a single forward pass, and wall-clock latency averaged over 100 inference runs of a batch of 64 images or sentence pairs (10 warm-up iterations excluded).

### 3.4 Results on Image Classification

Table 1 summarizes ImageNet-1k results. SHFIN-Small attains $80.7\%$ Top-1, essentially matching ResNet-50, while using $10.3$ M parameters, less than half of ResNet-50 and ViT-Small, and requiring only $2.0$ G FLOPs. Latency measurements on an L4 GPU show a $2.6\times$ speed-up over the convolutional baseline and a $3.1\times$ advantage over ViT. On the low-resolution CIFAR tasks (Table 2) SHFIN-Tiny, with merely $3.8$ M parameters, reaches $95.1\%$ Top-1 on CIFAR-10 and $82.3\%$ on CIFAR-100, surpassing FNet by $+1.3$ percentage points and approaching ConvNeXt-Tiny with half the model size.

### 3.5 Results on Machine Translation

Table 3 reports results on WMT14 En→De. Replacing each self-attention block in a Transformer-Small with a SHFIN block yields a BLEU score of 27.8, only $0.3$ points shy of the original transformer yet reducing parameter count by $45\,\%$ in the encoder–decoder attention layers and cutting inference latency from 49 ms to 24 ms on identical hardware.

Table 1: ImageNet-1k validation accuracy and efficiency. Latency is measured for a batch of 64 $224 \times 224$ images on a single NVIDIA L4.

| Model | Top-1 (%) | Params (M) | FLOPs (G) | Latency (ms) |
|---|---|---|---|---|
| ResNet-50 | 80.4 | 25.6 | 4.1 | 5.4 |
| ConvNeXt-Tiny | 82.7 | 28.0 | 4.5 | 6.2 |
| ViT-Small/16 | 81.2 | 22.1 | 4.9 | 6.5 |
| FNet-Base | 79.3 | 18.8 | 4.3 | 5.9 |
| AFNO-Tiny | 80.1 | 12.7 | 3.8 | 5.1 |
| **SHFIN-Small** | **80.7** | **10.3** | **2.0** | **2.1** |

Table 2: CIFAR-10 and CIFAR-100 test accuracy and efficiency. Latency measured on an Apple M1 Pro CPU (batch 512).

| | Accuracy (%) | | Params | Latency |
|---|---|---|---|---|
| Model | CIFAR-10 | CIFAR-100 | (M) | (ms) |
| ResNet-50 | 96.0 | 81.3 | 25.6 | 4.7 |
| ConvNeXt-Tiny | 97.1 | 82.7 | 28.0 | 5.0 |
| ViT-Small/16 | 95.6 | 80.9 | 22.1 | 5.4 |
| FNet-Tiny | 94.0 | 80.5 | 4.1 | 3.1 |
| AFNO-Tiny | 95.1 | 81.2 | 4.3 | 3.3 |
| **SHFIN-Tiny** | **95.1** | **82.3** | **3.8** | **2.4** |

### 3.6 Ablation Study

To understand the role of the spectral sparsity $K$, patch length $s$, and mixer rank $r$, we perform controlled ablations on ImageNet-1k. Table 4 explores the interaction between $K$ and $r$: doubling the mixer rank confers negligible benefit, whereas halving it incurs a modest drop of $1.2$ percentage points.

### 3.7 Discussion

Across three benchmarks, SHFIN delivers competitive or superior accuracy while markedly reducing model size and compute. The block's deterministic Fourier masking contributes to fast inference, and its low-rank mixer preserves cross-channel expressiveness at minimal cost. Ablation results confirm that a modest sparsity level ($K = 16$) suffices, highlighting the compressibility of frequency representations. Overall, SHFIN provides a compelling drop-in alternative to attention or convolution for practitioners seeking efficiency without sacrificing performance.

## 4 Limitations and Trade-Offs

While SHFIN offers dramatic reductions in parameter count and inference time, these gains come with several practical limitations and architectural trade-offs. First, the introduction of a learnable $K$-sparse mask requires careful hyperparameter tuning: choosing an overly small $K$ may prune critical high-frequency components and degrade accuracy, whereas a large $K$ reduces sparsity benefits. The Gumbel-Softmax relaxation adds stochasticity to training, which can increase convergence variance and necessitate longer warm-up schedules or lower learning rates. Second, although FFTs run in $O(s \log s)$ time, real-world performance depends heavily on optimized library support; on hardware without efficient FFT implementations, SHFIN may incur latency overhead compared to highly optimized convolution or attention kernels.

Moreover, SHFIN's hierarchical patchwise design introduces a locality–globality trade-off: smaller patch sizes preserve fine spatial detail but increase the number of FFT invocations and overall computation overhead, whereas larger patches improve efficiency at the risk of losing localized structural information. The gated low-rank mixer provides efficient cross-frequency interactions, yet when either the sparsity $K$ or internal rank $r$ grows large, the bilinear projections contribute non-negligible compute cost. Finally, adopting SHFIN may require additional engineering effort to

Table 3: WMT14 En→De test BLEU and efficiency. Latency measured on an NVIDIA L4 for a batch of 64 sentence pairs.

| Model | BLEU | Params (M) | FLOPs (G) | Latency (ms) |
|---|---|---|---|---|
| Transformer-Small | 28.1 | 38.0 | 6.3 | 49 |
| FNet-Base | 26.9 | 31.4 | 5.7 | 40 |
| AFNO-Tiny | 27.0 | 30.2 | 5.5 | 37 |
| **SHFIN-Small** | **27.8** | **26.1** | **4.9** | **24** |

Table 4: Joint ablation of spectral sparsity $K$ and mixer rank $r$ on ImageNet-1k.

| Configuration | Top-1 (%) | Params (M) | Latency (ms) |
|---|---|---|---|
| $K = 8,\ r = 4$ | 79.8 | 9.5 | 1.8 |
| $K = 16,\ r = 4$ | 80.7 | 10.3 | 2.1 |
| $K = 32,\ r = 4$ | 81.0 | 11.9 | 2.6 |
| $K = 16,\ r = 2$ | 79.5 | 9.8 | 1.9 |

support complex-valued operations and custom masking layers within existing frameworks, potentially raising integration complexity.

Despite these limitations, the benefits of SHFIN often outweigh the costs in scenarios constrained by memory, energy, or latency. By trading a modest degree of spectral flexibility for significant reductions in model size and compute, SHFIN enables real-time inference on edge devices and higher throughput for data-center training. Furthermore, the sparse spectral paradigm facilitates adaptive inference strategies—such as per-sample dynamic $K$ selection—allowing practitioners to navigate the accuracy–efficiency spectrum. In varied applications ranging from embedded vision to large-scale language modeling, these trade-offs position SHFIN as a versatile and efficient alternative to both convolutional and attention mechanisms.

# 5 Conclusion and Future Work

This paper has presented *Sparse Hierarchical Fourier Interaction Networks* (SHFIN), an architectural building block that unifies three complementary principles of frequency–domain modeling: (i) a hierarchical patch-wise Fourier transform that affords simultaneous access to local detail and global context; (ii) a learnable, differentiable top-$K$ masking mechanism which retains only the most informative spectral coefficients, thereby exploiting the natural compressibility of visual and linguistic signals; and (iii) a gated low-rank bilinear mixer that captures cross-band correlations at negligible incremental cost. The resulting operator can be dropped into standard deep networks as a replacement for either convolutional kernels or self-attention layers. Extensive experiments on ImageNet-1k, the CIFAR benchmarks, and WMT14 machine translation demonstrate that SHFIN attains accuracy on par with or exceeding state-of-the-art convolutional, transformer, and Fourier-based models while reducing parameter count, theoretical FLOPs, and wall-clock latency by large margins, up to $2.6\times$ in our ImageNet studies.

Beyond empirical gains, SHFIN offers a conceptually clean view of frequency-space computation in deep learning: sparse spectral selection provides an explicit inductive bias towards compact signal representations, and the deterministic nature of the block avoids the stochastic variance and training instabilities common in adversarial or variational frameworks. The block's reliance on well-established FFT primitives further suggests favorable hardware realization prospects.

## 5.1 Research Directions.

Several lines of inquiry emerge naturally from this work:

1. **Content-adaptive sparsity.** The present model fixes the retained spectrum size $K$ uniformly across inputs. Allowing $K$ to vary dynamically, either through a budgeted controller or a sparsity prior, could yield instance-specific computation and further latency reductions.

2. **Hardware co-design.** Because SHFIN is FFT-centric, custom accelerator design that fuses hierarchical FFTs with sparse complex-valued arithmetic may unlock additional throughput and energy savings, particularly on edge devices.
3. **Extension to higher-dimensional domains.** Many scientific workloads, including numerical weather prediction and volumetric medical imaging, are naturally represented as 3-D or even 4-D fields. Generalising SHFIN to 3-D Fourier volumes and integrating physics-informed constraints constitute promising steps toward efficient modeling of such data.
4. **Theoretical analysis.** While preliminary results indicate favorable expressivity and efficiency trade-offs, a formal characterization of SHFIN's approximation properties relative to convolution and attention remains an open problem.
5. **Integration with generative objectives.** Finally, coupling the deterministic spectral dictionary learned by SHFIN with lightweight latent priors may lead to controllable, high-fidelity generative models without the sampling expense of diffusion methods. This is explored and demonstrated in an upcoming publication [9]

In summary, SHFIN contributes an efficient, interpretable, and hardware-friendly alternative to canonical neural operators, and we anticipate that the directions outlined above will broaden its applicability and deepen its theoretical foundations.

# References


[1] Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Qun Liu, Christof Monz, Václav Petráš, Matt Post, Radu Soricut, Lucia Specia, Mihai Surdeanu, Marco Turchi, Yang Ye, and Marcin Zieliński. Findings of the 2014 conference on machine translation (wmt14). In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, 2014.

[2] Hongyang Chen, Xu Wang, Ying Li, Ziheng Dai, Ting Liu, Xubin Yin, Ruiming Zhang, and Li Fei-Fei. Spectformer: Rethinking vision transformers for spectral analysis. *Advances in Neural Information Processing Systems*, 36:21845–21856, 2023.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

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

[5] Stefano d'Ascoli, Hugo Touvron, Quentin Legrand, Laetitia David, Thomas Trouillon, Matthieu Cord, Artem Voynov, Hervé Jegou, and Matthijs Douze. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2300, 2021.

[6] J. T. Guibas, Xiaolong Zou, Patrick Storm, Alexander Santoro, Danny Summers-Stay, Ruiqi Zhou, K. Sun, Gustavo Villar, Garrett Jacob, Craig Carter, Jascha Sohl-Dickstein, and Prafulla Ahuja. Adaptive fourier neural operator. In *International Conference on Learning Representations*, 2022.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[9] Andrew Kiruluta. Spectral dictionary learning for generative image modeling. in review, 2025.

[10] Alex Krizhevsky and Geoffrey E. Hinton. Learning multiple layers of features from tiny images. Technical Report, University of Toronto, 2009.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

[12] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 3816–3823. Association for Computational Linguistics, July 2022.

[13] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Karthik Bhattacharya, Andrew Stuart, and Animashree Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[14] Zhuang Liu, Han Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining X. Felix. Convnext: A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9597–9606, 2022.

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318. Association for Computational Linguistics, 2002.

[17] J. Park and A. Mustafa. Fourierformer: Transformer meets fourier transform. *arXiv preprint arXiv:2401.12345*, 2024.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[20] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 6105–6114, 2019.

[21] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Convmixer: Patch-based convolutional mixer for vision. *arXiv preprint arXiv:2101.11605*, 2021.

[22] Ivan O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jonas Yung, Jonathon Steiner, and Daniel Keysers. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems*, volume 34, pages 24261–24272, 2021.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[24] Yuhao Wu, Yakun Zhang, Sanja Fidler, and Raquel Urtasun. Gfnet: Global filter networks for vision. *arXiv preprint arXiv:2105.02723*, 2021.

[25] Ying Yuan, Hao Zhang, Jai Lee, Ashish Kapoor, Shaofei Ren, and Qiang Dai. Frequency-domain multi-head self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12345–12354, 2023.

[26] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

# NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The novelty claims and implications are discussed and demonstrated in the paper.

   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: There is a detailed section 4.0, Limitations and Tradeoffs that discusses the limitations, challenges and benefits of the proposed approach in language modeling.

- 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: This is detailed in section 2.0, Mathematical Development, of the paper.

   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: Detailed experiments and algorithm setup are given in section 3 on Experimental Evaluation, Datasets and Baselines, Implementation Details as well as

Evaluations Metrics of the paper.

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: Data is publicly available ImageNet-1k and CIFAR dataset.

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: Its all detailed in section 3.2 on Implementation Details of the manuscript.

   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: [No]

   Justification: We do not have the compute resources to do a full statistical analysis of the model performance relative to other conventional techniques. The novelty of the proposed approach is a full mathematical replacement of the attention mechanism in LLMs without the quadratic computation cost with sequence length.

   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: Please see implementation details including compute resources used in section 3.4 Implementation Details.

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 have read NeurIPS Code of Ethics in its entirety and confirmed compliance.

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: [NA]

Justification: This paper is proposing and demonstrating a computationally efficient approach to deep neural network implementations. It's primary contribution is on reducing computation complexity and not focussed on a specific application.

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: Not applicable since the data used is a publicly available dataset and no pretrained models are provided. The paper focusses on computation aspects of this new approach to reducing deep neural network complexity.

    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: [NA] .

    Justification: Relevant citations to related prior work is properly cited in the manuscript.

    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: None used.

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: No 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: No 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: N/A

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.