A Dual-Domain Vision Transformer with Spectral Positional Bias

Abdulaziz Alshamsi Abdulla Alghfeli Chenghua Lin Hujun Yin
The University of Manchester, Manchester, UK
{abdulaziz.alshamsi, abdulla.alghfeli-2}@postgrad.manchester.ac.uk
{chenghua.lin, hujun.yin}@manchester.ac.uk

Abstract

Positional encoding in Vision Transformers presents a recurring trade-off: spatial relative-position tables (Swin) carry thousands of parameters per block and require ad-hoc interpolation to transfer across resolutions, while no-bias designs leave the attention logits geometry-agnostic. We propose a relative positional bias parameterized directly in the frequency domain. This parameterization yields three benefits over the standard spatial-table position encoding: (i) it expresses the same translation-equivariant bias with $3.25\times$ fewer parameters than Swin's spatial table at the same head count; (ii) it admits a closed-form, structurally exact transfer to a different patch grid (zero-padding to a finer grid, anti-aliased truncation to a coarser one), in contrast to interpolation of spatial tables, which replaces the learned kernel with a smoothed approximation; and (iii) it shares its parameterization with a frequency-domain spectral mixer, so that one transfer rule moves the positional bias and the token mixer together. We demonstrate the bias inside our **Dual-Domain Residual (DDR)** framework, in which a spectral mixer and a gated attention head act as parallel residuals within each block. In recipe-controlled comparisons at Tiny scale, the frequency-domain bias (SCQB) outperforms a Swin-style spatial table, with $3.25\times$ fewer bias parameters, by $+1.98$ pp, and a no-bias baseline by $+2.74$ pp; removing the entire attention branch (which carries the bias) drops accuracy by 5.35 pp, while removing the spectral branch drops it by 2.22 pp. At Small scale SCQB ties the spatial table and both exceed no bias, and a parameter-free 2D-RoPE arm ties SCQB at Tiny but loses 7.3 to 11.6 pp more at 384^2 and 448^2. Across Tiny, Small and Base, DDR's ImageNet-1K accuracy is consistent with published spectral hybrids (78.9% at 10.7M, 82.5% at 63.1M), though recipe differences prevent a strict comparison. The learned per-layer gates concentrate spectral processing in early blocks and attention in late blocks, qualitatively matching the schedules that are hand-coded in the previous hybrid models.

1 Introduction

Vision Transformers [Dosovitskiy et al., 2021, Touvron et al., 2021] treat an image as a set of patch tokens, ignoring their explicit spatial information. Self-attention is permutation-equivariant, hence a ViT cannot distinguish two images that share the same patches in different spatial layouts unless an external positional signal is injected into the token representations or the attention logits. The choice of positional signal is therefore a first-order architectural decision. Existing approaches parameterize positions as if they were unstructured: learned absolute embeddings [Dosovitskiy et al., 2021], relative-position tables that store an independent scalar for every displacement [Shaw et al., 2018, Liu et al., 2021], or bucketed scalar biases [Raffel et al., 2020]. *None of these reflect the spatial structure of natural images.* They neither bandlimit the bias to the smooth spatial relationships that dominate natural-image statistics, nor encode the translation equivariance of the patch grid as a hard

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

constraint. They also do not admit a structurally exact rule for transferring the learned bias to a different patch resolution. A recurring symptom is that fine-tuning a Swin model at a new resolution requires *ad hoc* bilinear interpolation of the relative-position table [Liu et al., 2021], an operation that has no closed-form justification and would smooth the learned structure.

We argue that the spatial frequency domain is the natural representational substrate for position encoding in vision, on grounds that go beyond computational efficiency. First, *translation equivariance is native*: a kernel parameterized by Fourier coefficients on the discrete patch torus is automatically circulant in the spatial domain and depends only on relative displacement. Second, *bandlimited smoothness is built into the parameterization*: a finite Fourier basis cannot represent arbitrarily sharp spatial transitions, matching the empirical regularity of vision biases at typical patch counts. Third, *resolution transfer is closed-form*: zero-padding the Fourier coefficients to a finer grid recovers the unique torus-periodic bandlimited extension of the learned bias, a structurally exact operation that bilinear interpolation of a spatial table cannot provide. These properties matter wherever training and test grids differ or the parameter budget is tight: variable-resolution backbones for detection and segmentation, train-small/test-large pipelines, and modalities with native spectral or multi-resolution structure such as audio spectrograms and medical or remote-sensing imagery. Computational savings are another benefit: spectral mixing [Rao et al., 2021, Lee-Thorp et al., 2022] achieves global token interaction at $\mathcal{O}(Nd \log N)$ per layer[1] versus $\mathcal{O}(N^2 d)$ for self-attention. Attention adapts to input content, while spectral mixing supplies a strong, economical, content-independent geometric prior. The two are complementary.

Existing hybrids that combine spectral mixing with attention commit each layer to a single mixing primitive through a hand-coded depth schedule, e.g., spectral in early blocks and attention in later blocks [Patro et al., 2025], or fixed spectral mixing as a front end for attention [Patro and Agneeswaran, 2023]. The schedule is a discrete, architectural hyperparameter chosen *a priori* by the architect, not learned from data. Moreover, these hybrids inherit standard spatial position-encoding tables for their attention layers, so the position encoding and the spectral mixer live in different representational spaces. We argue that the schedule should be *learned* and the position encoding should share a representational substrate with the mixing operator.

Our **Dual-Domain Residual (DDR)** block replaces the discrete depth schedule with a continuous, per-layer scalar gate that the architecture sets at training time:

$$\mathbf{x} \leftarrow \mathbf{x} + \mathrm{DropPath}\big(\mathcal{F}(\mathrm{RMSNorm}(\mathbf{x}))\big), \qquad \text{(Spectral)}$$

$$\mathbf{x} \leftarrow \mathbf{x} + \mathrm{DropPath}\big(\sigma(\alpha_\ell) \cdot \mathrm{Attn}_{\mathrm{SCQB}}(\mathrm{RMSNorm}(\mathbf{x}))\big), \qquad \text{(Gated Attn)}$$

$$\mathbf{x} \leftarrow \mathbf{x} + \mathrm{DropPath}\big(\mathrm{SwiGLU}(\mathrm{RMSNorm}(\mathbf{x}))\big), \qquad \text{(FFN)}$$

The DDR block applies three sequential residual updates per layer (spectral, gated attention with SCQB, FFN). The departure from prior hybrids is *schedule learnability*: a learned per-layer scalar gate $\sigma(\alpha_\ell) \in (0, 1)$ on the attention branch (initialized at $\sigma(0)$=0.5) lets the architecture decide at training time how to allocate processing across depth. The spectral branch is ungated and supplies shift-equivariant global mixing at every layer.

Inside the attention branch we introduce **Spectral-Conditioned QK Bias (SCQB)**, a relative position bias that realizes the frequency-domain parameterization as argued above. SCQB stores 896 learned Fourier coefficients per block (versus 2,916 for Swin-style relative position bias [Liu et al., 2021]) and recovers the $N \times N$ attention bias matrix through an inverse FFT followed by circulant indexing. By construction, it inherits translation equivariance, bandlimited smoothness, closed-form resolution transfer, and shares the same Fourier-domain parameterization as the spectral branch.

We evaluate SCQB in a controlled series in which every positional mechanism (SCQB, a Swin-style table, axial 2D-RoPE, and no bias) is trained in one backbone under one recipe, at Tiny scale (with SCQB, the table and no bias repeated at Small scale), and at eight test resolutions at Tiny scale. On ImageNet-1K from scratch, DDR's accuracy is consistent with published spectral hybrids at Tiny, Small and Base (Table 2); at Tiny it exceeds SpectFormer-Ti [Patro et al., 2025] by $+2.0$ pp, but these are cross-paper numbers under different recipes, and we rest no claim on them.

Our contributions are:

1. **SCQB: a frequency-domain relative position bias.** SCQB compresses Swin-style spatial RelPos into 896 Fourier coefficients per block, a $3.25\times$ reduction at h=4. In a controlled ablation at Tiny

[1] Per channel; the total spectral cost is $\mathcal{O}(DN \log N)$ per block.

scale it recovers $+2.74$ pp over no positional bias and $+1.98$ pp over Swin RelPos. The margin over the table is capacity dependent: at Small scale the two tie, and both exceed no bias. The closed-form transfer is structurally exact, and at 384^2 and 448^2 the learned biases lose 7.3 to 11.6 pp less than 2D-RoPE.

2. **Dual-domain residuals in each block.** Spectral filtering and gated attention act as parallel residuals, replacing SpectFormer's hand-coded depth schedule. A seven-variant ablation isolates three load-bearing effects: the spectral branch (-2.22 pp, A1), SCQB versus Swin RelPos (-1.98 pp, B1), and attention-plus-SCQB jointly (-5.35 pp, A2). Gating (A3) and backbone choices (A4) each move accuracy by less than 0.2 pp at Tiny scale and we draw no conclusion from them.
3. **An emergent schedule and a measured cost.** The learned per-layer gates concentrate the spectral branch in early layers and attention in late layers, qualitatively matching the schedule that prior work hand-coded ($|\rho| = 0.869$ vs. SpectFormer's binary schedule, $n = 12$), and they exceed gates frozen to that schedule by $+0.3$ pp on ImageNet-100. The SCQB bias costs 1.9 to 2.0% of end-to-end throughput at 224^2 against a FlashAttention-2 reference.

2 Related Work

Vision Transformers. ViT [Dosovitskiy et al., 2021] showed that a pure self-attention architecture matches or surpasses convolutional networks on image recognition. DeiT [Touvron et al., 2021] and DeiT III [Touvron et al., 2022] closed the data-efficiency gap via distillation and improved training recipes. Swin Transformer [Liu et al., 2021] introduced hierarchical feature maps and shifted window attention with relative position bias. DDR retains the isotropic structure of DeiT while replacing the attention-only block with a dual-domain residual. Token-mixer alternatives such as MLP-Mixer [Tolstikhin et al., 2021], ConvMixer [Trockman and Kolter, 2023], and the MetaFormer abstraction [Yu et al., 2022] contextualize attention as one mixing primitive among several.

Spectral and frequency-domain methods. FNet [Lee-Thorp et al., 2022] replaces self-attention with a parameter-free Fourier transform along the sequence and hidden axes. GFNet [Rao et al., 2021] adds learnable frequency filters via element-wise multiplication in the Fourier domain; equivalently, this is a learned circular convolution on the patch grid. AFNO [Guibas et al., 2022] extends GFNet with block-diagonal channel mixing and soft-thresholding in frequency space, retaining circulant token mixing. Recent state-space alternatives such as Vision Mamba [Zhu et al., 2024] and Hyena [Poli et al., 2023] pursue a different efficient-mixing direction outside the Fourier domain. DDR's spectral branch builds on GFNet, applying learned BCCB (Block Circulant with Circulant Blocks) filters via 2D FFT/iFFT. The key departure is that the spectral branch is one of two complementary residual terms per block, not a standalone mixer.

Hybrid spectral-attention architectures. Several recent works combine spectral and attention processing, but differ from DDR in how the two domains interact. SpectFormer [Patro et al., 2025] stacks spectral layers in early blocks and attention in later blocks, with the transition selected per-architecture as a discrete hyperparameter. SVT [Patro and Agneeswaran, 2023] applies scattering transforms before attention as a fixed front end. iFormer [Si et al., 2022] operates in the spatial domain via a high-frequency mixer derived from convolution and pooling, paired with low-frequency self-attention. Wave-ViT [Yao et al., 2022] unifies wavelet down-sampling with attention. HorNet [Rao et al., 2022] approximates high-order spatial interactions through recursive gated convolutions, an interpretation that admits a frequency-domain reading via its convolutional substrate. DDR differs in two ways. First, every layer runs both branches in parallel, with a learned scalar gate, rather than committing each layer to one mode. Second, SCQB places the attention bias on the same frequency-domain substrate as the spectral branch, instead of bolting a spatial lookup table onto a frequency-domain mixer.

Positional encoding for vision transformers. Learned positional embeddings [Dosovitskiy et al., 2021] were the original choice for ViT. Shaw et al. [2018] introduced relative position bias by adding a learned scalar to attention logits indexed by displacement. T5 [Raffel et al., 2020] popularized the bucketed-displacement scalar bias. Swin [Liu et al., 2021] popularized the spatial-table form ($(2H-1)\times(2W-1)$ entries per head) that DDR replaces. Swin v2 [Liu et al., 2022] parameterizes the bias as a small MLP over log-spaced relative coordinates, a *spatial* continuous parameterization. SCQB instead parameterizes the bias via a *frequency-domain* basis, which (i) gives a closed-form

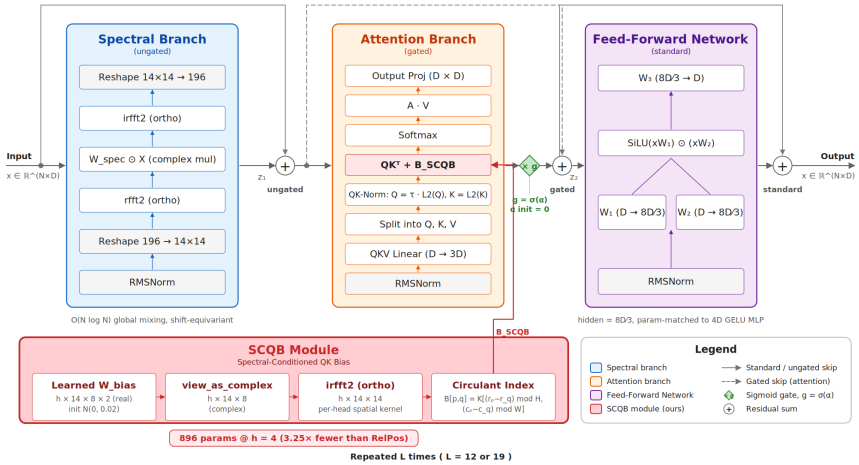


Figure 1: **DDR block.** Three residual branches: a spectral filter (left, blue) at $\mathcal{O}(N \log N)$ cost, a self-attention head with SCQB position bias (center, orange) gated by a per-layer sigmoid $\sigma(\alpha_\ell)$, and a SwiGLU FFN (right, purple). The spectral branch is ungated; the attention branch carries the gate. The SCQB module (red inset) learns $h \times 14 \times 8$ complex Fourier coefficients, recovers a 14×14 spatial kernel through `irfft2`, and assembles the $N \times N$ bias by circulant displacement indexing: 896 parameters per block versus 2,916 for Swin RelPos (num_heads=4). The block is repeated L times (L=12 or 19).

bandlimited resolution-transfer rule and (ii) shares the FFT parameterization with the spectral branch. RoPE [Su et al., 2024] encodes position multiplicatively through query/key rotation in complex space, and a 2D variant for vision was introduced by Heo et al. [2024]. ALiBi [Press et al., 2022] uses fixed linear distance-decay biases, while CPE [Chu et al., 2023] generates positional signals conditionally on input tokens. The building blocks of SCQB pre-exist, and we state its delta precisely. Learnable Fourier Features [Li et al., 2021] learn Fourier features of *absolute* coordinates and add them to token content. GFNet [Rao et al., 2021] and SpectFormer [Patro et al., 2025] learn frequency-domain *mixing* filters that carry no relative bias. SCQB's contribution is the combination of (a) a Fourier-parameterized circulant kernel placed in the attention logits as a relative bias and (b) a parameterization shared with the spectral mixer. This placement makes translation equivariance, the parameter reduction, and the closed-form transfer structural properties rather than heuristics; we claim no more than that.

Modern transformer components. SwiGLU [Shazeer, 2020] replaces the standard MLP with a gated linear unit. RMSNorm [Zhang and Sennrich, 2019] substitutes LayerNorm's mean-centering with pure RMS rescaling. QK-Norm [Dehghani et al., 2023] normalizes query and key projections to stabilize attention logits when additive biases shift the pre-softmax distribution.

3 Method

3.1 Overview

Given an image $\mathbf{X} \in \mathbb{R}^{H_{\text{img}} \times W_{\text{img}} \times 3}$, we partition it into $N = H_{\text{img}} W_{\text{img}} / P^2$ non-overlapping patches of size $P \times P$ and linearly embed them into D-dimensional tokens $\mathbf{z}_0 \in \mathbb{R}^{N \times D}$. A learned positional embedding $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{N \times D}$ is added; this is the only place where absolute position enters the network, because the spectral branch and SCQB are translation equivariant by construction. The token sequence is processed by L DDR blocks, followed by RMSNorm [Zhang and Sennrich, 2019], global average pooling, and a linear classifier. With P=16 and 224×224 input, $N = 196$ tokens are arranged on a 14×14 spatial grid. The spectral branch operates on the spatial token grid; the class token is routed through the residual unchanged. Figure 1 summarizes the DDR block, which we describe formally next.

3.2 DDR Block

Let $\mathbf{z}^{(\ell)} \in \mathbb{R}^{N \times D}$ denote the input to block ℓ. Each DDR block applies three residual updates.

Spectral branch (ungated). The spectral branch applies learned filters in the 2D Fourier domain, implementing BCCB convolution [Rao et al., 2021]. For each channel, tokens are reshaped to an $H \times W$ grid ($H{=}W{=}14$, $HW{=}N$), transformed via real-valued 2D FFT, element-wise multiplied by learned complex weights, and inverse-transformed:

$$\hat{\mathbf{z}} = \mathrm{RMSNorm}(\mathbf{z}^{(\ell)}), \tag{1}$$

$$\mathbf{Z}_{\mathrm{spec}} = \texttt{irfft2}\Big(\mathbf{W}_{\mathrm{spec}} \odot \texttt{rfft2}\big(\mathrm{Reshape}_{N \to H \times W}(\hat{\mathbf{z}}),\ \mathrm{norm{=}ortho}\big),\ \mathrm{norm{=}ortho}\Big), \tag{2}$$

where $\mathbf{W}_{\mathrm{spec}} \in \mathbb{C}^{D \times H \times (W/2+1)}$ is a learned complex weight initialized from $\mathcal{N}(0, 0.02^2)$ (std$=$ 0.02) and $\odot$ denotes element-wise multiplication. For $H{=}W{=}14$, this requires $D \times 14 \times 8$ complex entries. Orthonormal FFT normalization preserves signal energy (Parseval's theorem), helping maintain consistent gradient magnitudes across the frequency and spatial domains. The spectral branch is *ungated*:

$$\mathbf{z}_1^{(\ell)} = \mathbf{z}^{(\ell)} + \mathcal{F}\big(\mathrm{RMSNorm}(\mathbf{z}^{(\ell)})\big). \tag{3}$$

Attention branch with SCQB (gated). Given $\hat{\mathbf{z}}_1 = \mathrm{RMSNorm}(\mathbf{z}_1^{(\ell)})$, we compute queries, keys, and values: $\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathrm{split}(\hat{\mathbf{z}}_1 \, \mathbf{W}_{QKV})$, reshaped to $\mathbb{R}^{h \times N \times d}$ for h heads with $d = D/h$. We apply ℓ_2-normalized QK with a learned temperature [Dehghani et al., 2023]:

$$\mathbf{Q} \leftarrow \tau \cdot \frac{\mathbf{Q}}{\|\mathbf{Q}\|_2}, \qquad \mathbf{K} \leftarrow \frac{\mathbf{K}}{\|\mathbf{K}\|_2}, \tag{4}$$

where $\tau \in \mathbb{R}$ is a learned scalar initialized to $\sqrt{d}$ and the normalization is applied along the last dimension. This bounds attention logits and stabilizes training when additive biases (SCQB) shift the pre-softmax distribution. Attention scores are:

$$\mathbf{A} = \mathrm{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathbf{B}_{\mathrm{SCQB}}\Big), \qquad \mathrm{Attn}(\hat{\mathbf{z}}_1) = \mathrm{Concat}(\mathbf{A}\mathbf{V})\, \mathbf{W}_O, \tag{5}$$

where $\mathbf{B}_{\mathrm{SCQB}} \in \mathbb{R}^{h \times N \times N}$ is the SCQB bias (Section 3.3). The attention branch is gated by a learned sigmoid:

$$\mathbf{z}_2^{(\ell)} = \mathbf{z}_1^{(\ell)} + \sigma(\alpha^{(\ell)}) \cdot \mathrm{Attn}_{\mathrm{SCQB}}\big(\mathrm{RMSNorm}(\mathbf{z}_1^{(\ell)})\big), \tag{6}$$

where $\alpha^{(\ell)} \in \mathbb{R}$ is a per-block scalar initialized to 0, giving $\sigma(0) = 0.5$.

Feed-forward branch. The FFN uses SwiGLU activation [Shazeer, 2020]:

$$\mathrm{SwiGLU}(\mathbf{x}) = \big[\mathrm{SiLU}(\mathbf{x}\mathbf{W}_1) \odot (\mathbf{x}\mathbf{W}_2)\big]\mathbf{W}_3, \tag{7}$$

$$\mathbf{z}^{(\ell+1)} = \mathbf{z}_2^{(\ell)} + \mathrm{SwiGLU}\big(\mathrm{RMSNorm}(\mathbf{z}_2^{(\ell)})\big), \tag{8}$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{D \times D_{\mathrm{ff}}}$, $\mathbf{W}_3 \in \mathbb{R}^{D_{\mathrm{ff}} \times D}$, and $D_{\mathrm{ff}} = \lceil 8D/3 \rceil_8$ (rounded up to the nearest multiple of 8). This gated activation yields the same parameter count as a standard $4D$-hidden MLP: $3 \times D_{\mathrm{ff}} \times D \approx 2 \times 4D \times D = 8D^2$.[2]

3.3 Spectral-Conditioned QK Bias (SCQB)

SCQB parameterizes the relative position bias inside each attention head directly in the frequency domain. The spectral branch already expresses token mixing through learned Fourier coefficients; SCQB applies the same FFT parameterization to the attention's positional bias. RoPE [Su et al., 2024, Heo et al., 2024] and ALiBi [Press et al., 2022] achieve relative position information without learned bias parameters; we compare SCQB against axial 2D-RoPE directly in a controlled arm (Table 3, B3). Among learned biases, SCQB is most directly compared against the Swin family [Liu et al., 2021], where its $3.25\times$ parameter reduction (896 versus 2,916 per block at $h{=}4$) comes with $+1.98$ pp accuracy at Tiny scale and a tie at Small scale (Section 4.3; Appendix D).

[2] $D_{\mathrm{ff}}{=}\lceil 8D/3 \rceil_8$ matches the *parameter count* of a $4D$-hidden GELU MLP ($8D^2$) when $8D/3$ is a multiple of 8 (e.g. $D{\in}\{192, 384, 768\}$); for $D{\in}\{232, 256\}$ rounding inflates the count by $\sim$0.8%. FLOPs differ from a $4D$ MLP by the SwiGLU gating branch.

Formulation. For each attention head $i \in \{1, \ldots, h\}$ we learn a complex tensor $\mathbf{W}^{(i)}_{\text{bias}} \in \mathbb{C}^{H\times(W/2+1)}$, stored as a real tensor of shape $(h, H, W/2{+}1, 2)$ initialized from $\mathcal{N}(0, 0.02^2)$ and reinterpreted as complex on each forward pass. The spatial bias kernel is recovered through an inverse FFT:

$$\mathbf{K}^{(i)} = \texttt{irfft2}\big(\mathbf{W}^{(i)}_{\text{bias}},\ s = (H, W),\ \text{norm=ortho}\big) \in \mathbb{R}^{H\times W}. \tag{9}$$

The $N\times N$ attention bias is assembled by circulant displacement indexing:

$$\mathbf{B}^{(i)}_{\text{SCQB}}[p, q] = \mathbf{K}^{(i)}\big[(r_p - r_q) \bmod H,\ (c_p - c_q) \bmod W\big], \tag{10}$$

where (r_p, c_p) and (r_q, c_q) are the 2D grid coordinates of tokens p and q. The displacements $\Delta_r[p, q]$ and $\Delta_c[p, q]$ are precomputed once at construction and cached as non-persistent buffers. We approximate the patch grid as a discrete torus through modular displacement; boundary effects are absorbed into the learned coefficients, and at test time the same approximation is used at the target resolution. We use directional displacements on the discrete torus: after the modulo, raw displacements lie in $[0, H{-}1]$, but the kernel itself is not constrained to be symmetric, so SCQB can express directional bias, for example stronger attention upward than downward, that a symmetric distance-only bias cannot capture. When a class token is present, SCQB applies only to patch-patch pairs and zeros are inserted on the corresponding row and column.

Parameter count. A Swin-style relative position table [Liu et al., 2021] stores $(2H{-}1)^2$ entries per head: $h \cdot 27^2 = 2{,}916$ real parameters for $h{=}4$ heads at $H{=}W{=}14$. SCQB stores $h \cdot H \cdot (W/2{+}1)$ complex values, i.e., $h \cdot 14 \cdot 8 \cdot 2 = 896$ real parameters for $h{=}4$, a $3.25\times$ reduction. The ratio is $3.25\times$ at every head count (e.g., 1,344 versus 4,374 at $h{=}6$ for DDR-S; head counts per variant are in Table 1). The stored $(h, H, W/2{+}1, 2)$ tensor encodes 224 reals per head, of which only $H \cdot W = 196$ are independent: 28 reals per head sit in the Hermitian-symmetric columns $c \in \{0, W/2\}$ and `irfft2` discards them implicitly.

Function class. By construction $\mathbf{B}^{(i)}_{\text{SCQB}}$ is a Block Circulant with Circulant Blocks (BCCB) matrix on the $H\times W$ patch grid: it is fully determined by a single $H\times W$ kernel and is diagonalized by the 2D DFT. SCQB exactly parameterizes the BCCB subclass of relative position biases on the discrete patch torus. Whether the optimal bias lies in this subclass is an empirical question; our ablations support it at Tiny scale, where SCQB at 896 parameters per block exceeds Swin RelPos at 2,916 parameters per block by $+1.98$ pp; at Small scale the two tie. Storing only $H \cdot (W/2{+}1)$ complex coefficients (rather than all HW) further bandlimits the kernel along W, which is consistent with the empirical observation that learned biases are spatially smooth at 14×14 resolution.

Resolution transfer. At inference time, SCQB can evaluate at a different patch grid $H'\times W'$ by zero-padding the Fourier coefficients to shape $H'\times(W'/2{+}1)$ and recomputing the displacement buffers. Zero-padding the Fourier coefficients recovers the unique torus-periodic bandlimited extension at any target grid; in practice we rescale the inverse FFT output by $\sqrt{N'/N}$ to preserve the kernel peak amplitude across resolutions. To a coarser grid (a lower resolution or a larger patch), the closed form is coefficient truncation, the adjoint of zero-padding: it is exact for every retained frequency and anti-aliased by construction. Because patch size and resolution act through the same token grid, the rule also covers a change of patch size: a $P{=}16$ model at 448^2 sees the same grid as $P{=}8$ at 224^2. The spectral mixer filters move with the same frequency-bin placement, with the per-bin gain kept unchanged (Appendix E). Spatial relative position bias has no analogous structure: its standard fine-tuning recipe [Liu et al., 2021] relies on bilinear interpolation of the learned table, a piecewise-linear approximation that smooths the bandlimited structure. We measure the consequence at eight test resolutions in Appendix E.

Inference-time cost (measured). The bias matrix $\mathbf{B}^{(i)}_{\text{SCQB}} \in \mathbb{R}^{N\times N}$ is a deterministic function of the learned coefficients, so at inference time it can be precomputed once per layer at model load. Precomputation alone gives no measurable speedup, because building the bias is a negligible fraction of the forward pass. The gain is that a constant bias can be handed to a fused attention kernel: on DDR-Ti-SF this lifts throughput from 2,290 to 2,818 img/s. Against the same model with the bias removed on the FlashAttention-2 kernel, the bias costs 1.9 to 2.0% end-to-end at 224^2 and 3.8 to 4.1% at 384^2 (Appendix C).

3.4 Architecture Variants

DDR follows an isotropic architecture: all blocks operate at the same resolution and embedding dimension. Table 1 summarizes the model family. The three reported headline configurations are DDR-Ti-SF (Tiny), DDR-S-Deep (Small), and DDR-B-Deep (Base); DDR-Ti and DDR-Ti+ serve as ablation-purpose variants at reduced parameter counts, and DDR-S provides an intermediate Small baseline. Variants labeled "-SF" or "-Deep" match SpectFormer [Patro et al., 2025] configurations in embedding dimension, head count, and depth for controlled comparison.

Table 1: **DDR model variants.** All use patch size 16×16 at 224^2 resolution. FFN hidden dimension is $\lceil 8D/3 \rceil_8$. Parameters measured for 1000-class ImageNet.

Model	D	h	L	**Params (M)**	**FLOPs (G)**
DDR-Ti	192	3	12	6.23	1.3
DDR-Ti+	232	4	12	8.91	1.8
DDR-Ti-SF	256	4	12	10.71	2.2
DDR-S	384	6	12	23.07	4.7
DDR-S-Deep	384	6	19	36.09	7.3
DDR-B-Deep	512	8	19	63.14	12.7

4 Experiments

4.1 Setup

Dataset. We evaluated on ImageNet-1K [Deng et al., 2009] (1.28M training images, 50K validation images, 1,000 classes). All models were trained from scratch without external data or distillation.

Training. We trained all main and ablation models for 300 epochs on ImageNet-1K (exceptions in Appendix A) with AdamW [Loshchilov and Hutter, 2019], a cosine schedule, RandAugment, Mixup/CutMix, label smoothing and stochastic depth on 4×NVIDIA RTX A6000 GPUs; full hyperparameters are in Appendix A. The training recipe was identical across all DDR variants and all controlled arms; the only variable was the architecture. Each arm is a single run (seed 42). The results reported for SpectFormer [Patro et al., 2025], GFNet [Rao et al., 2021], FNet [Lee-Thorp et al., 2022], DeiT [Touvron et al., 2021], DeiT-III [Touvron et al., 2022], and the published position encodings are quoted from the respective papers and marked "pub." in Table 2. A 1–2 pp gap between the DeiT and DeiT-III recipes on the same backbone is well documented, and the Small and Base margins over SpectFormer fall inside it. Table 2 is therefore context, not controlled evidence. The controlled evidence is the positional series (Table 3, B1 to B3, and Table 5), in which every arm is retrained in one backbone under one recipe.

4.2 ImageNet-1K Results

Table 2 reports top-1 accuracy on ImageNet-1K. We group rows by parameter scale and match DDR variants to SpectFormer configurations at comparable parameter budget (embedding dimension, head count, and depth are approximately matched).

Across scales. At Tiny (DDR-Ti-SF, 10.7M, 78.9%), DDR is $+2.0$ pp above SpectFormer-Ti (9.0M, 76.9%), a margin larger than the documented recipe gap, though the recipe component is not isolated. At Small (DDR-S-Deep, 82.3%) and Base (DDR-B-Deep, 82.5%), the margins over SpectFormer ($+0.6$ and $+0.4$ pp) lie inside the recipe gap, so we read them only as consistency with published spectral hybrids. Against DeiT-III at the same 300-epoch budget, DDR-S leads ViT-S by 1.3 pp, while ViT-B leads DDR-B-Deep by 0.3 pp with 37% more parameters; we do not claim state of the art. The spectral branch and SCQB together account for $\sim$3% of total FLOPs at Tiny and $\sim$1% at Base (Appendix B); wall-clock costs are measured in Appendix C.

Table 2: **ImageNet-1K classification.** No distillation, no extra data. "Recipe": *pub.* = quoted from the cited paper (recipe differs; context, not controlled evidence); *ours* = our 300-epoch recipe (Appendix A). Groups are the Tiny, Small and Base parameter-scale sub-headers; **bold** marks the best Top-1 within each group. Shaded rows are ours; the B3 row is the controlled 2D-RoPE arm in the DDR-Ti-SF backbone.

Model	Recipe	Params (M)	FLOPs (G)	Top-1 (%)	Top-5 (%)
Tiny scale					
DeiT-Ti [Touvron et al., 2021]	pub.	5	1.2	72.2	91.1
FourierFormer [Nguyen et al., 2022]	pub.	n/a	n/a	73.3	91.7
CPVT-Ti-GAP (CPE) [Chu et al., 2023]	pub.	6	n/a	74.9	92.6
GFNet-Ti [Rao et al., 2021]	pub.	7	1.3	74.6	92.2
SpectFormer-Ti [Patro et al., 2025]	pub.	9	1.8	76.9	93.4
DDR-Ti (ours)	ours	6	1.3	75.2	92.7
DDR-Ti+ (ours)	ours	9	1.8	77.9	94.0
DDR-Ti-SF (ours)	ours	11	2.2	78.9	94.5
DDR-Ti-SF, SCQB $\rightarrow$ 2D-RoPE (B3)	ours	11	2.2	**79.0**	94.6
Small scale					
DeiT-S [Touvron et al., 2021]	pub.	22	4.6	79.8	95.0
DeiT-III ViT-S, 300 ep [Touvron et al., 2022]	pub.	22	4.6	79.9	n/a
DeiT-III ViT-S, 800 ep[‡] [Touvron et al., 2022]	pub.	22	4.6	81.4	n/a
ViT-S + 2D-RoPE, 400 ep[§] [Heo et al., 2024]	pub.	22	4.6	80.9	n/a
Swin-T (RelPos) [Liu et al., 2021]	pub.	28	4.5	81.3	n/a
FNet-S[†] [Lee-Thorp et al., 2022]	pub.	15	2.9	71.2	n/a
GFNet-XS [Rao et al., 2021]	pub.	16	2.9	78.6	94.2
GFNet-S [Rao et al., 2021]	pub.	25	4.5	80.0	94.9
SpectFormer-XS [Patro et al., 2025]	pub.	20	4.0	80.2	94.7
SpectFormer-S [Patro et al., 2025]	pub.	32	6.6	81.7	95.6
DDR-S (ours)	ours	23	4.7	81.2	95.5
DDR-S-Deep (ours)	ours	36	7.3	**82.3**	96.0
Base scale					
DeiT-B [Touvron et al., 2021]	pub.	86	17.5	81.8	95.6
DeiT-III ViT-B, 300 ep [Touvron et al., 2022]	pub.	87	17.5	82.8	n/a
DeiT-III ViT-B, 800 ep[‡] [Touvron et al., 2022]	pub.	87	17.5	**83.8**	n/a
GFNet-B [Rao et al., 2021]	pub.	43	7.9	80.7	95.1
SpectFormer-B [Patro et al., 2025]	pub.	57	11.5	82.1	95.7
DDR-B-Deep (ours)	ours	63	12.7	82.5	96.0

[†] FNet is an NLP architecture; the ImageNet-1K number is reproduced from Patro et al. [2025]. [‡] 800 epochs with 3-Augment, LayerScale, BCE loss and LAMB; the ViT-B row also uses 192^2 pre-training and fine-tuning at 224^2. [§] Axial 2D-RoPE under the DeiT-III 400-epoch recipe. We found no published ImageNet-1K ViT result for ALiBi [Press et al., 2022] under a comparable recipe, so ALiBi has no row.

4.3 Ablation Study

We ablated the architectural decisions of DDR-Ti-SF (256/4/12, 10.71M parameters), trained on ImageNet-1K for 300 epochs with the same recipe as the main results. The ablation is organized into branch ablations (A1 to A4) and positional bias ablations (B1 to B3). Table 3 reports top-1 and top-5 accuracies and changes relative to the full model (78.88% top-1). The headline effects (-5.35, -2.74, -2.22, -1.98 pp) are all multi-pp; rows A3 ($+0.12$), A4 (-0.19) and B3 ($+0.15$) show accuracy changes by less than 0.2 pp and we draw no accuracy conclusion from them.

Three load-bearing components. DDR-Ti-SF is, at its core, three co-operating components: a spectral branch, a gated attention branch, and SCQB. Removing the spectral branch (A1) costs 2.22 pp, comparable to the full margin over SpectFormer-Ti; the combined ablation A1+A2 ($2.22+5.35=7.57$ pp) exceeds either individual loss. A2 removes the attention branch (and SCQB with it); the -5.35 pp drop is an upper bound on their joint value, not a decomposition. SCQB's isolated contribution is given by B1 ($+1.98$ pp over Swin RelPos, $3.25\times$ fewer parameters) and B2 ($+2.74$ pp over no bias). The gating formulation (A3, $+0.12$ pp) and modern FFN/norm bundle (A4, -0.19 pp) are below the 0.2 pp threshold and we draw no conclusion from them.

Table 3: **Ablation study** on DDR-Ti-SF (ImageNet-1K, 300 epochs). The full model reaches 78.88% top-1, 94.53% top-5. Δ is reported in absolute percentage points. Rows A3, A4 and B3 are below the 0.2 pp threshold.

	Variant	Params (M)	Top-1 (%)	Top-5 (%)	Δ
	DDR-Ti-SF (full)	10.711	**78.88**	**94.53**	ref
Branch ablations					
A1	w/o spectral branch	10.020	76.66	92.40	-2.22
A2	w/o attention branch	10.700	73.53	91.41	-5.35
A3	Symmetric gating	10.711	79.00	94.50	$+0.12$
A4	w/o SwiGLU, RMSNorm, QK-Norm	10.687	78.69	94.42	-0.19
Positional bias ablations					
B1	SCQB $\rightarrow$ RelPos	10.735	76.90	92.79	-1.98
B2	SCQB $\rightarrow$ None	10.700	76.14	92.05	-2.74
B3	SCQB $\rightarrow$ 2D-RoPE	10.700	79.03	94.58	$+0.15$

SCQB vs. RelPos: function class, not parameter count. SCQB constrains the bias to BCCB matrices parameterized by 896 Fourier coefficients per block (h=4); Swin RelPos uses an unconstrained $(2H-1)^2$ lookup table (2,916 parameters for full-grid coverage). At matched budget (896 vs. 900), Swin RelPos covers only ± 7 displacements while the 14×14 grid requires ± 13; SCQB covers the full grid by construction (Table 4; iso-budget kernels in Appendix D, Figure 4). The -1.98 pp gap (B1) is consistent with co-adaptation between SCQB and the spectral branch, though B1 alone does not isolate the cause.

The positional margin is capacity dependent. We repeated the controlled series at Small scale (DDR-S, 300 epochs, same recipe and seed; Table 5). SCQB and the Swin table tie (0.1 pp, below our 0.2 pp threshold), and both exceed no bias by 0.5 to 0.6 pp. At Tiny, a parameter-free axial 2D-RoPE arm [Heo et al., 2024] (θ=100; B3) also ties SCQB ($+0.15$ pp). SCQB's accuracy margin at the training grid is therefore not universal: it separates from the table at Tiny and ties it at Small. What persists across scale is the $3.25\times$ parameter reduction against the table and the closed-form transfer, whose consequence we measure at eight resolutions (Appendix E). Under upscale transfer, 2D-RoPE's native extension loses 7.3 to 7.6 pp more than the learned biases at 384^2 and 10.9 to 11.6 pp more at 448^2.

Table 4: **Position bias at Tiny scale** (DDR-Ti-SF, per block, h=4). SCQB recovers more accuracy than Swin RelPos with $3.25\times$ fewer parameters.

Method	Params/block	Δ over None	Top-1 (%)
None (B2)	0	ref	76.14
RelPos (B1)	2,916	$+0.76$	76.90
SCQB (ours)	**896**	$\mathbf{+2.74}$	**78.88**

Table 5: **Controlled series at Small scale** (DDR-S, 300 epochs). Same backbone, recipe, seed and schedule; only the positional mechanism changes.

Mechanism	Params (M)	Params/block	Top-1 (%)
None	23.05	0	80.7
Swin RelPos	23.11	4,374	81.3
SCQB (ours)	23.07	1,344	81.2

Published RoPE, CPE and Swin numbers appear in Table 2 as quoted context (Figure 5).

Gate form versus gate learning. A3 swaps the asymmetric gating for a symmetric form $(1-\sigma(g))\cdot \text{spec} + \sigma(g)\cdot \text{attn}$; the two perform indistinguishably ($+0.12$ pp). A3 therefore compares two *learned* gate forms; it does not test whether learning the schedule matters. We retain the asymmetric form because it cleanly separates the ungated structural prior (spectral) from the learned modulation (attention). To test learning directly, we trained the DDR-Ti-SF backbone on ImageNet-100 for 300 epochs with gates frozen to SpectFormer's binary schedule (4 spectral + 8 attention blocks) and with learned gates: the learned gates reach 88.3% versus 88.0% ($+0.3$ pp, single seed; Appendix A). The learned gate thus removes SpectFormer's per-model spectral-depth hyperparameter and, on this benchmark, also exceeds the hand-coded schedule it replaces; we present this as supporting evidence rather than proof.

Modern components are not critical at Tiny scale. A4 reverts SwiGLU to a $4D$-hidden GELU MLP, RMSNorm to LayerNorm, and removes QK-Norm; the combined effect is -0.19 pp. Prior work shows these components yield larger gains at higher capacities [Shazeer, 2020, Zhang and Sennrich, 2019, Dehghani et al., 2023].

4.4 Analysis: what the trained model learned

The gates qualitatively match SpectFormer's hand-coded schedule. Across the 12 layers of DDR-Ti-SF, layers 0 to 2 hold gate values $\sigma(\alpha^{(\ell)})$ in the range ≈ 0.58 to 0.80 (substantial spectral contribution), while layers 3 to 11 saturate to $\sigma(\alpha) \geq 0.93$. The Spearman rank correlation between this learned profile and SpectFormer-Ti's binary schedule is $|\rho| = 0.869$ on $n = 12$ layers,[3] consistent with the gate having found a similar coarse-to-fine allocation without supervision (Appendix F, Figures 10 and 9). The ungated spectral branch keeps contributing in deeper layers, so DDR's allocation is not a strict copy of the binary schedule.

5 Conclusions

We presented SCQB, a relative position bias parameterized directly in the frequency domain, and DDR, the dual-domain residual block in which SCQB shares a Fourier substrate with a spectral mixer. In recipe-controlled comparisons at Tiny scale, SCQB outperforms a Swin-style spatial table, with $3.25\times$ fewer bias parameters, by $+1.98$ pp, and a no-bias baseline by $+2.74$ pp. Its accuracy margin over the table is capacity dependent (a tie at Small), and a parameter-free 2D-RoPE ties it at Tiny; what persists is the parameter reduction and a closed-form, structurally exact transfer, under which the learned biases degrade far less than 2D-RoPE on upscale grid change. Across Tiny, Small and Base, DDR's accuracy is consistent with published spectral hybrids, though recipe differences prevent a strict comparison. The design lesson is that position encoding and token mixing can be co-designed in one representational domain, so that one transfer rule moves both. Learned per-layer gates concentrate the spectral branch in early layers and attention in late layers without supervision, qualitatively matching the schedules that are hand-coded in prior hybrids.

Limitations and future work. **Comparison**: recipe-mismatched baselines (SpectFormer, GFNet, FNet, DeiT, DeiT-III, published position encodings) are quoted; controlled comparisons are shown in Table 3 (B1 to B3) and Table 5. **Scale and seeds**: branch ablations are at Tiny only, every arm is a single run, and the gate-learning test uses ImageNet-100 only. **Decomposition**: A2 conflates attention removal with SCQB removal. **Transfer**: claim (ii) is structural, not a general accuracy advantage; at moderate upscale (288^2 to 320^2) the closed form trails bilinear smoothing of the same kernel by 1.0 to 1.7 pp, and the unconstrained table is the most robust arm at 128^2. **Absolute position**: both branches are translation equivariant, and a linear probe shows that absolute position decays with depth (R^2 from 0.74 to 0.04); making it available at every layer, as a reviewer suggested, is a promising direction. **Throughput**: stock FlashAttention-2 does not accept an arbitrary additive bias; on the fused SDPA kernel SCQB costs 1.9 to 2.0% at 224^2 and 3.8 to 4.1% at 384^2 (FlexAttention matches SDPA at Tiny and Small, not yet at Base), a cost that grows with resolution and is not negligible against scaling the base architecture. The FFT branch cuts wall-clock throughput by about $3\times$ against a plain ViT of similar size (Appendix C), a cost shared with GFNet and a practical limit on adoption. **Architecture**: static gates at inference and classification-only evaluation. Next steps: a transfer-dataset version of the gate-learning test, input-dependent gating, layer-level absolute position, fused-kernel SCQB, and dense prediction.

[3] The binary nature of the SpectFormer schedule means a point-biserial statistic is more appropriate; we report Spearman for comparability.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers and the area chair for constructive feedback that shaped the controlled experiments in this version.

Funding. Abdulaziz Alshamsi and Abdulla Alghfeli are PhD students at The University of Manchester, funded by the Technology Innovation Institute (TII), Abu Dhabi, UAE. This work received no other third-party funding.

Competing interests. The authors declare no competing interests.

References

Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR-W)*, 2020.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning (ICML)*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

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 (ICLR)*, 2021.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive Fourier neural operators: Efficient token mixers for transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision (ECCV)*, 2024.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision (ECCV)*, pages 646–661, 2016.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. FNet: Mixing tokens with Fourier transforms. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2022.

Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. Learnable Fourier features for multi-dimensional spatial positional encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer V2: Scaling up capacity and resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Tan Nguyen, Minh Pham, Tam Nguyen, Khai Nguyen, Stanley Osher, and Nhat Ho. FourierFormer: Transformer meets generalized Fourier integral theorem. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Badri N. Patro, Vinay P. Namboodiri, and Vijay S. Agneeswaran. SpectFormer: Frequency and attention is what you need in a vision transformer. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Badri Narayana Patro and Vijay Srinivas Agneeswaran. Scattering vision transformer: Spectral mixing matters. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning (ICML)*, 2023.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations (ICLR)*, 2022.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 21(140):1–67, 2020.

Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser-Nam Lim, and Jiwen Lu. HorNet: Efficient high-order spatial interactions with recursive gated convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. Inception transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-Mixer: An all-MLP architecture for vision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, 2021.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: Revenge of the ViT. In *European Conference on Computer Vision (ECCV)*, 2022.

Asher Trockman and J. Zico Kolter. Patches are all you need? *Transactions on Machine Learning Research (TMLR)*, 2023. arXiv:2201.09792.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Ting Yao, Yingwei Pan, Yehao Li, Chong-Wah Ngo, and Tao Mei. Wave-ViT: Unifying wavelet and transformers for visual representation learning. In *European Conference on Computer Vision (ECCV)*, 2022.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. MetaFormer is actually what you need for vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision Mamba: Efficient visual representation learning with bidirectional state space model. In *International Conference on Machine Learning (ICML)*, 2024.

A Training Details

All main and ablation models were trained for 300 epochs (exceptions below) with AdamW [Loshchilov and Hutter, 2019] (lr $= 5 \times 10^{-4}$, weight decay $= 0.05$), effective batch size 1,024 (256 per GPU $\times$ 4 GPUs), cosine schedule with 20 warmup epochs, RandAugment [Cubuk et al., 2020] (`rand-m9-mstd0.5-inc1`), Mixup [Zhang et al., 2018] ($\alpha = 0.8$), CutMix [Yun et al., 2019] ($\alpha = 1.0$), label smoothing ($\epsilon = 0.1$), and gradient clipping at 1.0. Stochastic depth [Huang et al., 2016] rates: 0.1 (Ti/Ti+/Ti-SF), 0.2 (S), 0.25 (S-Deep), 0.3 (B-Deep). Resolution 224×224 with 16×16 patches. Implementation was built on the `timm` library [Wightman, 2019].

Ablation configurations. Table 6 details each ablation variant. Each arm is a single run with seed 42.

Additional controlled runs. *Small series* (Table 5): DDR-S with SCQB replaced by a Swin-style table (h=6, 4,374 parameters per block) or removed, under the DDR-S recipe, seed and schedule (effective batch 1,024, 300 epochs). *Gate learning*: DDR-Ti-SF on ImageNet-100 (a 100-class subset of ImageNet-1K) for 300 epochs under the same recipe, with learned gates or with gates frozen to SpectFormer-Ti's binary schedule (4 spectral + 8 attention blocks); best checkpoints 88.30 versus 87.98 (EMA 87.84 versus 87.56). *Cross-resolution pair*: because the transfer study needs checkpoints of both bias types under an identical schedule, the SCQB and Swin-table DDR-Ti-SF arms were retrained as a matched pair for 150 epochs each (224^2 top-1 77.0 for both); the 2D-RoPE arm is the 300-epoch B3 model.

Table 6: **Ablation variant specifications.** Each row describes the single change made to the DDR-Ti-SF baseline (256/4/12, 10.71M); all other settings are held fixed.

#	Name	Change from Baseline
A1	w/o spectral	Spectral branch removed, gate frozen at $\sigma(100) \approx 1$
A2	w/o attention	Gate frozen at $\sigma(-100) \approx 0$, no SCQB
A3	Symmetric gating	Both branches gated: $(1-g) \cdot \text{spectral} + g \cdot \text{attention}$
A4	w/o modern	GELU MLP ($4D$ hidden), LayerNorm, no QK-Norm
B1	SCQB $\rightarrow$ RelPos	Swin-style relative position bias (2,916 params/block, num_heads=4)
B2	SCQB $\rightarrow$ None	No QK positional bias; only absolute positional embedding
B3	SCQB $\rightarrow$ 2D-RoPE	Axial 2D-RoPE [Heo et al., 2024] on Q and K (θ=100, head dimension split across the two image axes); identity rotation for non-spatial tokens; zero positional parameters

B Compute Breakdown

This section supports the body's claim that the dual-domain design adds only a small compute overhead. We report the per-branch FLOPs share for each DDR variant from a single forward pass at 224^2 input, using the MAC convention (1 multiply-accumulate counted as 1 FLOP, matching `fvcore` / `thop`). The per-layer cost decomposes as:

- **Attention:** $2N^2D$ for the QK and AV matmuls (each $\sum_h N \cdot d \cdot N$ MACs summed across heads), plus $4ND^2$ for the QKV and output projections.[4]
- **FFN (SwiGLU):** $3ND \cdot D_{\text{ff}}$ with $D_{\text{ff}} = \lceil 8D/3 \rceil_8$, i.e., $\sim 8ND^2$ per layer.
- **Spectral branch:** a 2D FFT/iFFT pair on D channels at $H \times W$=14×14, plus the complex element-wise filter: $\sim 10DN \log_2 N + 4DH(W/2+1)$ per layer.[5]
- **SCQB:** a single 2D inverse FFT on $h \cdot H \cdot (W/2+1)$ Fourier coefficients ($\sim 5hHW \log_2 N$) plus the additive bias (hN^2 adds) per layer. The kernel is shared across the batch.

[4] We do not count softmax, layer-norm reductions, or the bias addition; these are sub-leading at N=196.

[5] Cooley–Tukey radix-2: $\sim 2.5\, N \log_2 N$ MACs per 1D pass; a 2D FFT/iFFT pair costs $\sim 10\, DN \log_2 N$ MACs across D channels.

Substituting the variant configurations from Table 1 ($D \in \{192, 232, 256, 384, 384, 512\}$, $h \in \{3, 4, 4, 6, 6, 8\}$) yields the per-branch fractions in Table 7; Figure 2 visualizes the same data. Reconstructed totals match Table 1 within rounding.

Table 7: **Spectral plus SCQB cost only ∼3% of FLOPs at Tiny and ∼1% at Base.** Per-branch FLOPs share by variant. The fraction decreases with scale because attention and FFN carry $\mathcal{O}(ND^2)$ terms while spectral and SCQB do not. Rows may not sum to exactly 100% due to rounding.

Variant	D	L	Attn %	FFN %	Spec %	SCQB %
DDR-Ti	192	12	41.8	55.3	2.82	0.13
DDR-Ti+	232	12	40.3	57.2	2.40	0.12
DDR-Ti-SF	256	12	39.8	57.9	2.20	0.10
DDR-S	384	12	37.9	60.4	1.54	0.07
DDR-S-Deep	384	19	37.9	60.4	1.54	0.07
DDR-B-Deep	512	19	36.8	61.9	1.18	0.06

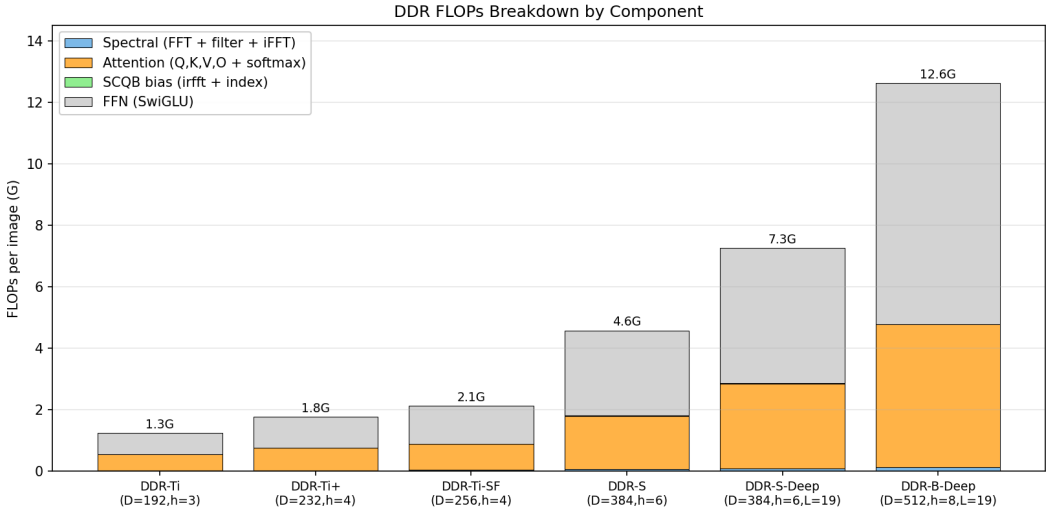


Figure 2: **Compute overhead of dual-domain processing is small (1.2% to 3.0% of total FLOPs) and shrinks with scale.** Per-branch FLOPs share across DDR variants at 224^2 input. Spectral plus SCQB occupy ∼3% at Tiny and ∼1% at Base; the FFN dominates compute at all sizes. The shrinkage with scale is structural: attention and FFN carry $\mathcal{O}(ND^2)$ terms while spectral and SCQB do not, so the dual-domain overhead becomes negligible at large D.

The dual-domain design costs between 1.2% and 3.0% of total FLOPs depending on scale. The overhead shrinks at larger scales because the FFN's $\mathcal{O}(D^2)$ cost dominates: spectral and SCQB carry no D^2 term. FLOPs understate the wall-clock cost of FFTs, which we measure next.

C Wall-Clock Throughput

Table 8 reports inference throughput on a single RTX A6000 (bf16, batch 256, median of 5 repeats of 50 timed iterations after 20 warm-up iterations). *LEVEL* runs every bias-carrying model on the strongest PyTorch SDPA backend that accepts an additive bias (memory-efficient), so all variants have an equal level of optimization; bias-free models run their native fused kernel. *REF* is each model's best kernel: FlashAttention-2 for bias-free models, and for DDR the same model with the bias *removed* on that kernel, which isolates the price of the bias. *FLEX* compiles the precomputed SCQB bias into a fused FlexAttention kernel. Bias cost is $(\text{REF} - \text{LEVEL})/\text{REF}$, computed from unrounded throughputs.

Precomputation. Within the unfused implementation, recomputing the bias on every forward pass and precomputing it at load give the same throughput (2,290 versus 2,234 img/s, within the run-to-run

Table 8: **Inference throughput (img/s).** The SCQB bias costs 1.9 to 2.0% at 224^2 and 3.8 to 4.1% at 384^2. The gap between DDR and plain ViTs is the spectral branch's FFT cost, shared with GFNet. *Bias-free models run their native fused kernel in both columns; sub-0.5% differences are run-to-run variance. †FlexAttention is not yet competitive at Base-scale head dimensions; SDPA is the deployment path there.

Model	LEVEL	REF (best kernel)	FLEX	Bias cost
224^2				
DeiT-Ti (5.7M)*	10,109	10,119	n/a	n/a
GFNet-Ti-wide*	3,963	3,963	n/a	n/a
DDR-Ti-SF (10.7M)	2,818	2,874	2,822	2.0%
DeiT-S (22M)*	4,648	4,648	n/a	n/a
DDR-S (23.1M)	1,806	1,842	1,809	2.0%
DeiT-B (86M)*	1,822	1,820	n/a	n/a
DDR-B-Deep (63.1M)	834	850	480†	1.9%
384^2				
DDR-Ti-SF	981	1,020	1,000	3.8%
DDR-S	633	660	639	4.1%
DDR-B-Deep	283	295	96†	3.9%

noise of the unfused path), because building the bias is a negligible fraction of the forward pass. The gain from precomputation is eligibility for a fused kernel: a constant bias tensor can be passed to SDPA, which lifts DDR-Ti-SF from 2,290 to 2,818 img/s.

D Parameter Efficiency vs Swin RelPos

This section supports the body's claim that SCQB's 896 Fourier coefficients are a $3.25\times$ parameter reduction over Swin RelPos at iso-coverage (h=4, 14×14 grid). The supporting evidence is presented in four steps. **(D.1)** A windowed Swin RelPos table needs $w^2 \cdot h$ parameters to cover a $w\times w$ displacement window; we tabulate this and identify the smallest window that covers the full ±13-displacement range required by a 14×14 grid (Table 9). **(D.2)** The same data is plotted against the 896-parameter SCQB budget and the full-coverage threshold (Figure 3). **(D.3)** At iso-parameter budget, we visualize the kernels each method actually expresses, showing that SCQB sees the full grid while Swin sees only ±7 tokens (Figure 4). **(D.4)** A Pareto view places SCQB against published learned and parameter-free position encodings on a parameter-cost-vs-accuracy plot (Figure 5).

Table 9: **D.1: Swin RelPos needs $3.25\times$ more parameters than SCQB to cover the full 14×14 grid.** Window size, parameter count, and coverage radius for Swin-style relative position bias on a 14×14 grid with h=4 heads. Full ±13 coverage requires a 27×27 window at 2,916 parameters. SCQB covers the full grid with 896 parameters via its circulant Fourier structure.

Window size	Swin RelPos params	Coverage radius	Covers 14×14 grid?
13×13	676	±6	No
15×15	900	±7	No
17×17	1156	±8	No
19×19	1444	±9	No
21×21	1764	±10	No
23×23	2116	±11	No
25×25	2500	±12	No
27×27	2916	±13	**Yes**
SCQB (h=4, 14×14)	**896**	full grid	**Yes**

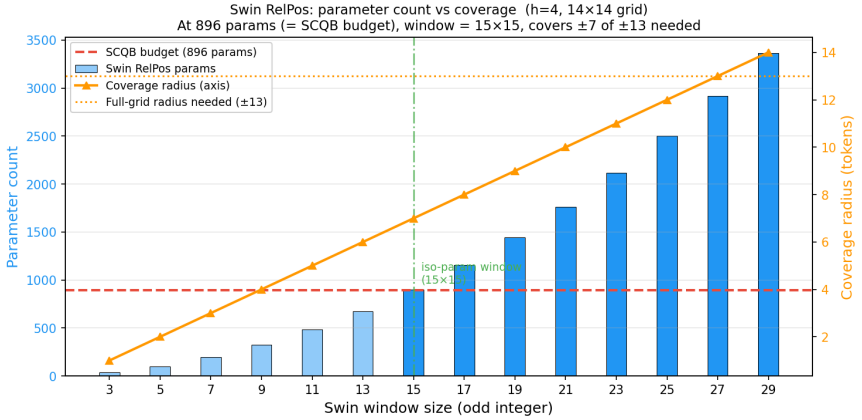


Figure 3: **D.2: Visualizing Swin's parameter-coverage tradeoff.** Blue bars: Swin RelPos parameter count vs window size (h=4, 14×14 grid). Orange line: coverage radius. Red dashed: SCQB's 896-parameter budget. Green dashed: full-grid coverage threshold (±13 tokens). At the SCQB budget, Swin reaches only the 15×15 window which covers ±7 of the required ±13 displacements; matching SCQB's full-grid coverage forces Swin up to 2,916 parameters.

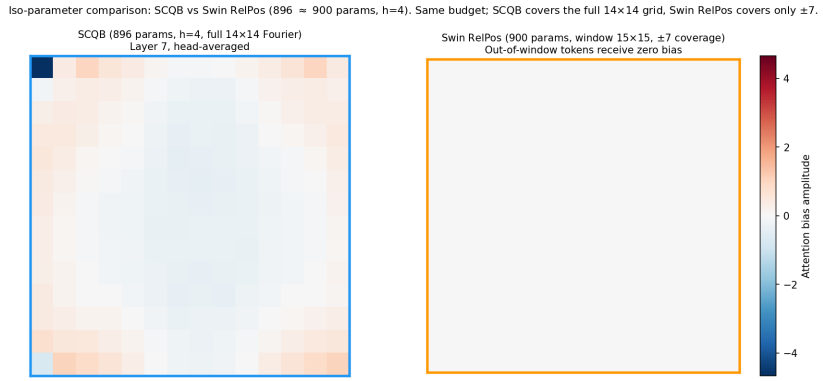


Figure 4: **D.3: At iso-parameter budget, SCQB sees the full grid; Swin RelPos sees only a ± 7 window.** Layer 7 head-averaged kernels for $h{=}4$ on a $14{\times}14$ grid. *Left*: SCQB at 896 parameters spans all displacements through its circulant Fourier parameterization. *Right*: Swin RelPos at the matched 900-parameter budget zeros out two-thirds of token displacements (out-of-window entries are exactly zero). Same parameter budget, qualitatively different expressive coverage. The coverage difference is consistent with the $+1.98$ pp head-to-head gap at Tiny scale (Table 3, row B1), but does not isolate its cause; at Small scale the full-coverage table and SCQB tie (Table 5).

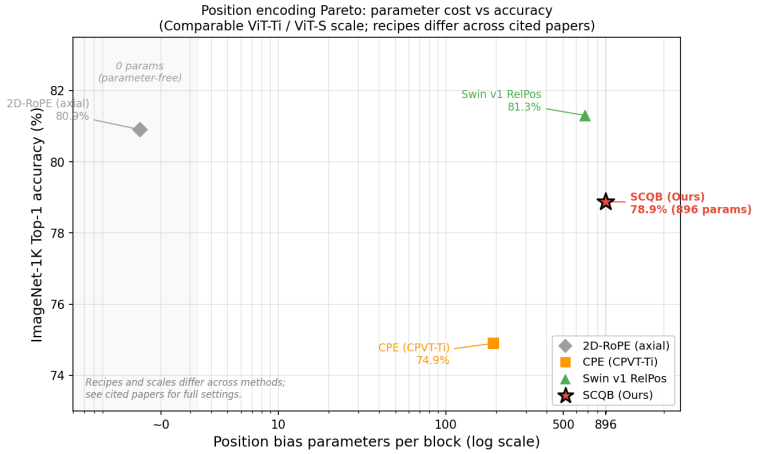


Figure 5: **D.4: Published position encodings by parameter cost and ImageNet-1K accuracy (context only).** Each point is a published result at ViT-Ti, ViT-S or Swin-T scale. Recipes, model sizes and epoch budgets differ (2D-RoPE uses the DeiT-III 400-epoch recipe on ViT-S; Swin-T is a 28M hierarchical model), so vertical positions reflect architecture, scale and recipe together, and the plot does not rank the methods. The quoted points also appear in Table 2. The recipe-controlled comparisons are Table 3 rows B1 to B3 (Tiny: $+1.98$ pp over Swin RelPos at $3.25\times$ fewer parameters; a tie with 2D-RoPE) and Table 5 (Small: a tie with Swin RelPos).

E SCQB Resolution Transfer

This section supports the body's claim that SCQB admits a structurally exact, closed-form rule for transferring the learned bias to a different patch grid. We compare frequency-domain zero-padding (the closed-form rule, Figure 6, top row) against bilinear upsampling (the standard fine-tuning recipe for spatial relative-position tables [Liu et al., 2021], bottom row), and quantify the $\sqrt{N'/N}$ amplitude rescaling we apply at inference (Figure 7).

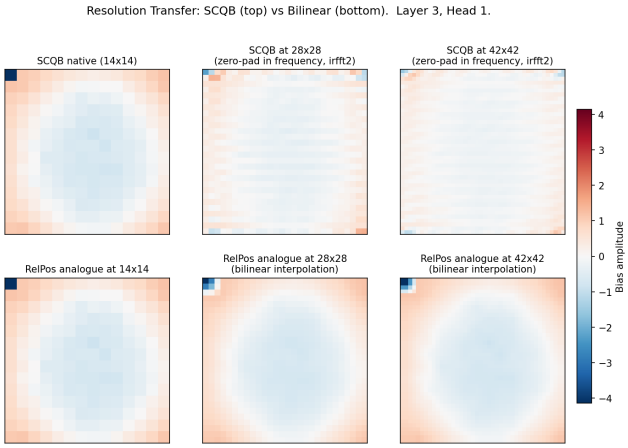


Figure 6: **Frequency-domain zero-padding preserves the bandlimited kernel; bilinear upsampling smooths it.** SCQB kernel learned on a 14×14 grid (DDR-Ti-SF) and evaluated at three resolutions. *Top row*: zero-padding the Fourier coefficients to the larger spectrum before `irfft2` recovers the unique torus-periodic bandlimited extension at the target grid; this is a closed-form operation that introduces no smoothing. *Bottom row*: bilinear upsampling, the standard fine-tuning recipe for spatial Swin-style biases, smooths the learned structure into a piecewise-linear approximation. Horizontal striations in the top row are sinc-interpolation artefacts of zero-padding, not training artefacts. Interpolation is a low-pass operation: its error concentrates in the high-frequency components where the learned structure lives, so the network sees a different kernel than the one it trained with. Zero-padding instead evaluates the trained bandlimited function on a denser grid. To a coarser grid (lower resolution or larger patch), the closed form is coefficient truncation, the adjoint of zero-padding: exact for every retained frequency and anti-aliased by construction.

Exactness. At every grid point that the source and target grids share, the transferred kernel agrees with the trained kernel to a maximum error of 5.4×10^{-7}; a bilinearly interpolated kernel deviates by up to 1.57 at the same points.

Spectral mixer filters. A network-level transfer also moves the spectral-branch filters. Under orthonormal `rfft2`, we place each filter's complex coefficients at the matching frequency bins of the larger (zero-padded) or smaller (truncated, Nyquist-folded) spectrum with the per-bin gain *unchanged*. The value-preserving rule that is correct for the additive bias (zero-padding with the $\sqrt{N'/N}$ rescale) would amplify each filter's response by the ratio of normalizations, and the error compounds across blocks.

Accuracy versus resolution. We evaluated trained Tiny networks zero-shot (no fine-tuning) on ImageNet-1K at eight resolutions with patch size 16, under four transfer rules: SCQB closed form, SCQB with bilinear interpolation of its own spatial kernel (a control that changes only the transfer rule), the Swin table with bilinear interpolation, and 2D-RoPE with its native extension. The absolute embedding is interpolated bicubically in every arm, so the positional transfer rule is the only variable. The SCQB and table arms are the matched 150-epoch pair (Appendix A); the RoPE arm is the 300-epoch B3 model. Each model is scored against its own 224^2 result (Table 10, Figure 8).

Four findings. (1) Near the training grid (192^2) every mechanism transfers gracefully (at most -1.6 pp). (2) Under upscale extrapolation, 2D-RoPE's native extension is the weakest: it trails the learned

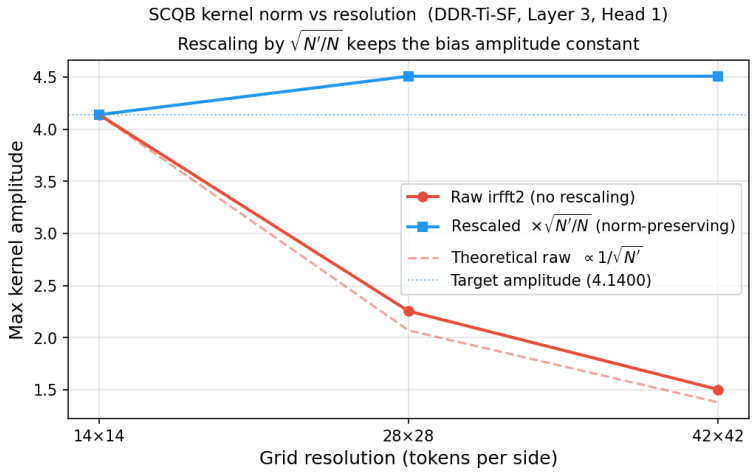


Figure 7: **The $\sqrt{N'/N}$ rescaling stabilizes kernel peak amplitude across resolutions.** SCQB kernel peak magnitude vs grid resolution (layer 3, head 1, DDR-Ti-SF). *Red*: raw `irfft2` output under `norm='ortho'` decays as $1/\sqrt{N'}$ (dashed: theoretical decay). *Blue*: multiplying by $\sqrt{N'/N}$ keeps amplitude approximately constant ($<10\%$ drift across 14×14, 28×28, 42×42). The near-flat blue curve confirms the rescaling prescription that we apply at inference when evaluating off the training grid.

Table 10: **Zero-shot cross-resolution transfer** (DDR-Ti-SF, ImageNet-1K top-1 change in pp against each model's own 224^2 result: 77.0 for the SCQB and table arms, 79.0 for 2D-RoPE). The RoPE arm was evaluated on six of the eight resolutions. A Small-scale table arm replicates the near-grid behavior (-1.2 pp at 192^2). Gaps quoted in the text are computed from unrounded accuracies.

Transfer rule	128	160	192	288	320	384	448
SCQB, closed form (ours)	-21.9	-6.3	-1.6	-6.4	-11.0	-17.1	-24.7
SCQB, bilinear control	-27.7	-6.1	-1.6	-5.3	-9.4	-17.0	-25.2
Swin table, interpolated	-15.1	-5.3	-1.2	-5.2	-9.4	-17.3	-24.4
2D-RoPE, native extension	-20.5	n/a	-1.6	-7.1	n/a	-24.6	-36.1

biases by 7.3 to 7.6 pp at 384^2 and 10.9 to 11.6 pp at 448^2, without the frequency rescaling used in long-context language models. (3) Between SCQB's own two rules, the closed form wins clearly at extreme downscale ($+5.8$ pp at 128^2, where its truncation is anti-aliased), ties at 384^2, is 0.5 pp better at 448^2, and trails by 1.0 and 1.7 pp at 288^2 and 320^2. A bias-magnitude sweep at 288^2 ($\times0.7$ to $\times1.15$) found no rescale that improves on the $\sqrt{N'/N}$ prescription, so we report this as a property, not a tuning artifact. (4) The unconstrained table is the most robust arm at 128^2, at $3.25\times$ the positional parameters and with no closed-form rule. SCQB stays within 1.6 pp of the most robust arm at every resolution above 128^2. We therefore state claim (ii) as structural (exactness, parameter count, anti-aliased downscale, no interpolation heuristic), not as a general accuracy advantage over interpolation; claim (iii) is the design principle that lets one rule move the bias and the mixer of every block together.

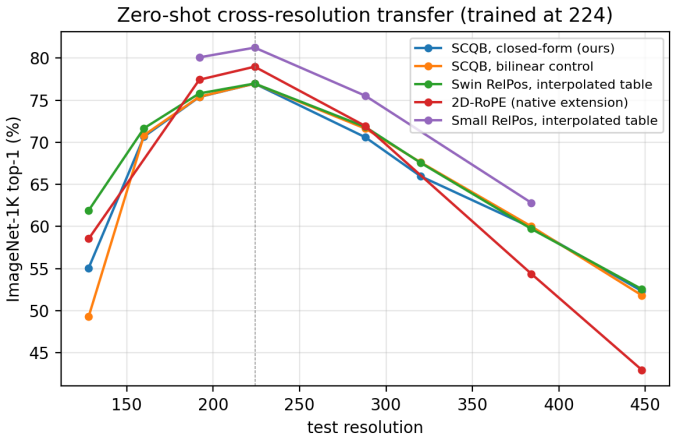


Figure 8: **Zero-shot cross-resolution transfer** (absolute top-1). Near the training grid every mechanism transfers gracefully; under upscale extrapolation 2D-RoPE's native extension degrades most. Purple: the DDR-S Swin-table arm (300 epochs).

F What the Trained Model Learned

This section documents the structure that DDR-Ti-SF discovers from data after 300 epochs on ImageNet-1K, in support of the body's claim that the learned per-layer gates recover a coarse-to-fine spectral-to-attention schedule without supervision and that both branches develop layerwise structure that evolves consistently with depth. The four views are presented in increasing granularity: **(F.1)** per-layer gate values, **(F.2)** their correlation with SpectFormer's hand-coded schedule, **(F.3)** the learned spectral filters and SCQB spatial kernels at four representative layers, and **(F.4)** the radial energy distribution of the SCQB Fourier coefficients.

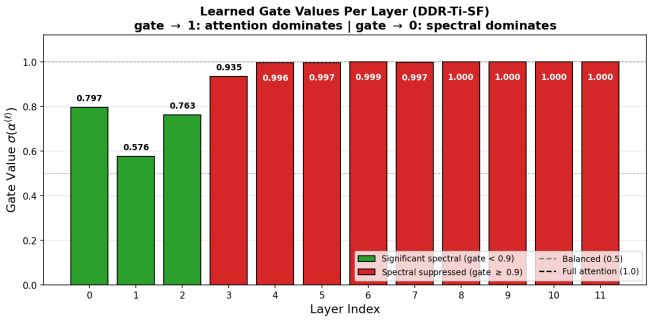


Figure 9: **F.1: The learned per-layer gates concentrate spectral processing in early layers and attention in late layers, without supervision.** Sigmoid gate values $\sigma(\alpha^{(\ell)})$ at convergence for DDR-Ti-SF on ImageNet-1K. Layers 0 to 2 hold gates in $[0.58, 0.80]$ (substantial spectral contribution); layers 3 to 11 saturate to ≥ 0.93 (attention-dominant). The schedule emerges from training; no curriculum, no sparsity penalty, no scheduling loss.

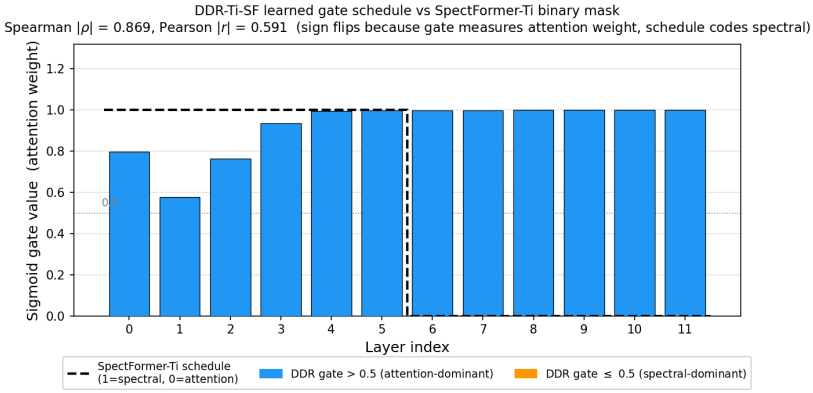
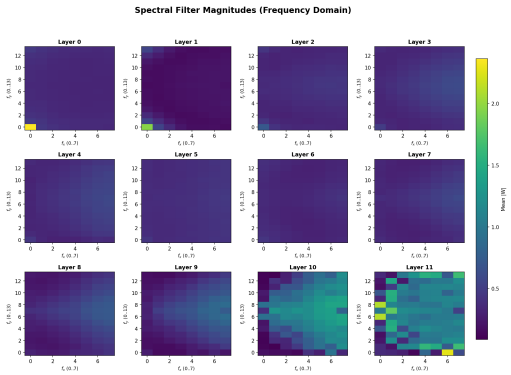
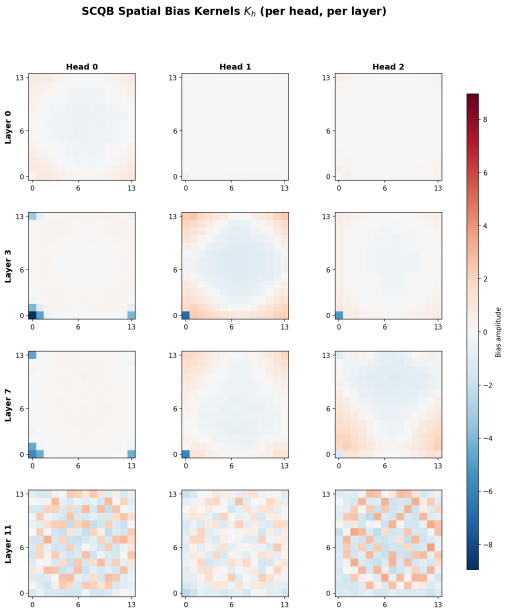


Figure 10: **F.2: The learned schedule matches SpectFormer's hand-coded one ($|\rho| = 0.869$).** Bars: per-layer DDR-Ti-SF gate values $\sigma(\alpha^{(\ell)})$ (blue = attention-dominant; orange = spectral-dominant, none observed). Dashed step-line: SpectFormer-Ti's hand-coded binary schedule (1 = spectral, 0 = attention). Spearman rank correlation $|\rho| = 0.869$ on $n = 12$ layers; the sign flips because the gate measures attention weight while the SpectFormer schedule codes spectral assignment. The match supports the claim that the coarse-to-fine spectral-to-attention allocation is recoverable as a learned outcome rather than a design constraint.

F.5: Absolute position decays with depth. The spectral branch and SCQB are translation equivariant, so absolute position enters only through the input embedding. A ridge-regression probe that decodes each patch token's (row, column) coordinates from frozen features after every block of the Tiny SCQB model gives $R^2 = 0.74$ after block 0, 0.36 after block 8 and 0.04 after block 11. Absolute position is therefore largely absent from late-layer representations; the reported accuracy is achieved without it, and making it available at every layer is a direction for future work.



(a) Spectral filters in the frequency domain



(b) SCQB spatial kernels

Figure 11: **F.3: Layerwise structure of the learned dual domain: spectral filters and SCQB spatial kernels evolve together with depth.** *(a)* Spectral filter magnitudes in the frequency domain: low-pass at layers 0 to 1, attenuation at layers 2 to 7, broadband high-frequency structure at layers 8 to 11. *(b)* SCQB spatial bias kernels at layers 0, 3, 7, 11 recovered through inverse FFT: broad isotropic bias in early layers, high-frequency near-checkerboard pattern by layer 11. The two views together show that the spectral branch and the SCQB position bias do complementary work even where the attention gate has saturated.

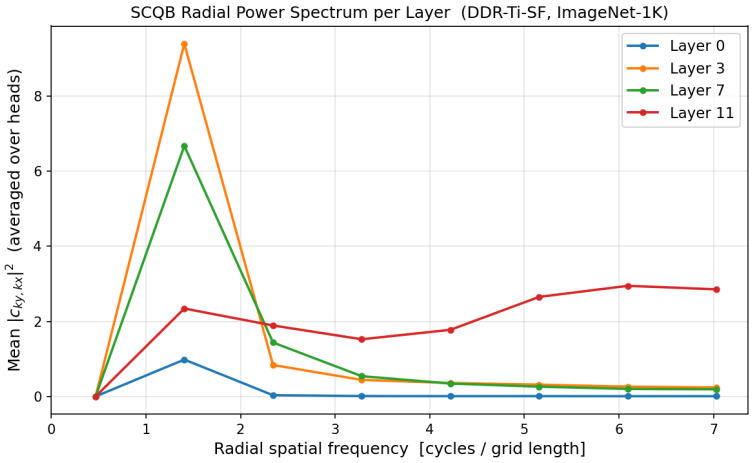


Figure 12: **F.4: SCQB Fourier coefficient energy shifts coarse-to-fine with depth.** Mean squared Fourier coefficient magnitude $|c_{k_y,k_x}|^2$ (head-averaged) vs radial spatial frequency for four representative layers of DDR-Ti-SF. The DC bin (k=0) is excluded as the constant offset is absorbed by softmax. Earlier layers concentrate energy at low frequencies; later layers shift toward high frequencies, matching the coarse-to-fine spectral evolution of the learned spectral filter shown in Figure 11(a). The two branches co-evolve consistent representations across depth.

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: The abstract and introduction state specific findings that are directly supported by Tables 2 and 3, e.g., in a controlled ablation at Tiny scale SCQB is +1.98 pp over Swin RelPos and +2.74 pp over no bias (Table 3). Limitations are discussed in Section 5.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Section 5 discusses limitations including quoted cross-recipe baselines, single-seed runs, Tiny-only branch ablations, the capacity dependence of SCQB's margin, the cases where the closed-form transfer trails interpolation, measured throughput costs, static gates at inference, and classification-only evaluation.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: The paper does not include formal theorems or proofs. The function class argument in Section 3.3 is an informal observation, not a formal theorem.

Guidelines:

- The answer [N/A] 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: Section 4.1 and Appendix A provide all hyperparameters, optimizer settings, augmentation details, and hardware specifications. The architecture is fully specified in Section 3 and Table 1; a reference implementation of the SCQB module is in the supplementary material.

Guidelines:

- The answer [N/A] 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: A reference implementation of SCQB is in the supplementary material; the full code and pretrained weights will be released publicly. ImageNet-1K is a publicly available dataset.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Full training recipe in Appendix A: optimizer (AdamW), learning rate, weight decay, batch size, augmentation parameters, drop path rate, warmup epochs. Hardware is specified in Section 4.1; the implementation library (`timm`) in Appendix A.

   Guidelines:

   - The answer [N/A] 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: The paper reports point-estimate Top-1 accuracies and per-block parameter and FLOPs counts; we do not report error bars. Each ablation row in Table 3 and Table 5 is a single training run (seed 42) on ImageNet-1K under an identical training recipe (Appendix A), so deltas between rows isolate the architectural change of interest. We treat Δ values whose magnitude is below 0.2 pp as within-variability and do not draw conclusions from them; rows above that threshold (A1, A2, B1, B2, and the Small no-bias arm) are the load-bearing comparisons.

Guidelines:

- The answer [N/A] 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: Section 4.1 specifies 4$\times$ NVIDIA RTX A6000 GPUs. Ablations use the same training recipe and compute as the main runs (Appendix A); see Section 4.3 for the ablation protocol.

   Guidelines:

   - The answer [N/A] 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: This work proposes a neural architecture for image classification using publicly available datasets. It does not involve human subjects, private data, or dual-use concerns.

   Guidelines:

- The answer [N/A] 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: [N/A]

    Justification: This is foundational research on vision transformer architectures. There is no direct path to negative societal applications beyond what applies to image classification in general.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The models are small-scale image classifiers (6–63M parameters) that do not pose misuse risks.

    Guidelines:

    - The answer [N/A] 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: ImageNet [Deng et al., 2009], DeiT [Touvron et al., 2021], GFNet [Rao et al., 2021], and the `timm` library [Wightman, 2019] are cited. All are publicly available for research use.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: Code and weights will be released upon publication. No new datasets are introduced.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This work does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This work does not involve human subjects research.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as a component of the core methodology. Any LLM assistance was limited to writing and editing.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.