SPECTRAL TOKENIZATION FOR SCALABLE VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

At 224×224, a standard ViT converts RGB patches into tokens; processing a finer image with similar encoder cost requires a different way to construct those tokens. Inspired by the blockwise discrete cosine transform (DCT) used in JPEG compression, we introduce SC-ViT, which tokenizes selected coefficients from a dense grid of image blocks instead of embedding RGB patches directly. It retains a limited spectral bandwidth while sampling the image at 448×448: with $K = 16$ coefficients per channel and block, the retained tensor has the same number of scalars as a 224×224 RGB image, and both models form 196 image tokens. Separate luminance and chrominance projections and a finer path for DC coefficients form the token embeddings. After 50 epochs on ImageNet-1K, SC-ViT reaches 77.36% top 1 accuracy at 16.95 GMACs, versus 74.42% at 17.47 GMACs for ViT-B/16 trained with the same recipe. A separate direct 300 epoch run reaches 78.44%. A progressive configuration yields three inference budgets from one checkpoint, reaching 77.2% at 8.62 GMACs. On KADID-10k, no reference quality prediction reaches 0.7064 Spearman correlation, versus 0.5353 for a spatial ViT-B/16 at 384×384. The spectral tokenizer also fits a hierarchical Swin encoder, where SC-Swin reaches 79.44% on ImageNet-1K after 50 training epochs.

1 INTRODUCTION

For a 224×224 RGB image, ViT-B/16 divides the image into 16×16 pixel patches and embeds them as a 14×14 grid of 196 image tokens (Dosovitskiy et al., 2021). Smaller patches expose finer spatial detail but create more tokens and raise attention cost. Keeping 196 image tokens while increasing the image to 448×448 would instead require 32×32 RGB patches, each compressed into one token. Thus the information selected and combined before the encoder affects the accuracy and compute tradeoff.

Transform coding offers another way to control the representation. Natural images concentrate much of their energy at low spatial frequencies (Field, 1987), and JPEG exploits this structure through a block discrete cosine transform (DCT) (Wallace, 1991). Spatial sampling and spectral bandwidth can therefore be chosen separately. A model can operate on a finer raster while retaining only a subset of the coefficients from each local block. This does not recover detail absent from the source image, and coefficient count is not an information theoretic rate. It does, however, provide an explicit allocation of spatial and spectral detail before the transformer.

We study this allocation through *spectral tokenization*. The input is decoded to RGB, converted to YCbCr, and transformed in local blocks. The first K zigzag coefficients are retained. A learned stem normalizes their different scales, weights individual coefficients, gates chrominance, and projects neighbouring blocks into tokens. In the flat model, DC coefficients also enter a finer spatial branch before being combined with AC features. Thus, local intensity structure and oscillatory detail can have different representations without increasing the sequence length.

Our main model, SC-ViT, maps a 448×448 image to a 14×14 grid of image tokens of width 768, matching the spatial token grid and embedding width of ViT-B/16 at 224×224. With $K = 16$, it also retains $150{,}528$ DCT values, equal to the number of scalar values in a 224×224 RGB image. Its tokenizer changes how local image information reaches the encoder, while preserving a standard token interface. To show that this input design can accompany a different transformer architecture, we also adapt it to Swin. The resulting SC-Swin uses shifted windows and patch merging (Liu et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2021); its stem maps each DCT block to one token to fit that hierarchy. Both models operate on decoded images and require no compressed bitstream.

The principal experiments concern fixed inference budgets. With 50 epochs of training on ImageNet-1K, SC-ViT with $K = 16$ at 448 improves on the evaluated ViT-B/16 by 2.94 percentage points, and SC-Swin improves on the evaluated Swin-B by 0.97 points. These are comparisons of complete models under a common optimizer and augmentation recipe. They do not isolate the effect of the DCT alone. At the same 224 resolution as ViT-B/16, SC-ViT reaches its highest measured accuracy with $K = 32$, between the tested $K = 16$ and $K = 64$ settings. The larger gain at 448 combines a denser raster with bandwidth selection. A separate 300 epoch direct run reaches 78.44%, numerically 0.53 points above the original ViT-B/16 study's 77.91% (Dosovitskiy et al., 2021). That study fine-tunes at 384×384 after training on ImageNet, and its protocol differs from ours.

We also examine two consequences of the representation. First, spectral groups can be introduced alongside encoder depth during training, producing several inference budgets from one checkpoint. Second, a representation that exposes local frequency content may be useful for image quality assessment (IQA). We report these as separate experiments, since their training schedules and model configurations differ from the fixed budget study.

Our contribution is SC-ViT: a DCT based transformer whose tokenizer treats coefficient scale, colour channel, and DC spatial structure explicitly. We study its fixed budget recognition, progressive inference, and quality prediction capabilities in separate experiments. We also test the spectral stem with Swin to see how it performs with a hierarchical encoder. The method below focuses on SC-ViT; the appendix gives the extension's implementation details.

2 Related work

Learning from frequency representations. Learning in the frequency domain has been explored for convolutional models (Xu et al., 2020), channel attention (Qin et al., 2021), and global filtering (Rao et al., 2021). DCFormer (Li et al., 2023) is particularly close to our work: it constructs frequency representations for transformers and studies input compression by removing high frequency components, including with a Swin backbone. DCT-ViT (Lee & Kim, 2024) also studies frequency pruning, while Pan et al. (2024) use DCT structure in attention projections. Our contribution is therefore not the first use of DCT inputs or spectral truncation in a transformer. We focus on the learned construction of tokens from those inputs, including coefficient weighting, chrominance gating, and a separate spatial path for DC coefficients, together with progressive bandwidth and depth selection.

Changing inference cost. FlexiViT (Beyer et al., 2023) trains a transformer to support different patch sizes. Slimmable Networks (Yu et al., 2019) and Once-for-All (Cai et al., 2020) train shared parameters for multiple network configurations. Adaptive depth (Elbayad et al., 2020) changes the amount of encoder computation. Our progressive model combines a retained coefficient budget with an exit depth while keeping the spatial token grid fixed. The resulting operating points are selected by the user; the model does not learn a policy that routes each image to an exit.

3 Spectral tokenization

3.1 Representation and coefficient normalization

Let an RGB image have height H and width W, both divisible by the DCT block side b. After conversion to YCbCr in the $[0, 255]$ convention and subtraction of 128, each block $X_{c,i,j} \in \mathbb{R}^{b\times b}$ is transformed as

$$C_{c,i,j} = A_b X_{c,i,j} A_b^\top, \qquad (A_b)_{u,x} = a_u \cos\left[\frac{\pi(2x+1)u}{2b}\right], \tag{1}$$

where $a_0 = 1/\sqrt{b}$, $a_u = \sqrt{2/b}$ for $u > 0$, and $c \in \{Y, C_b, C_r\}$. The block coordinates are i, j; within a block, x, y index pixels and u, v index transform frequencies. Let $\pi(k) = (u_k, v_k)$ be the fixed zigzag ordering of the b^2 frequency pairs. Retaining $0 \le k < K \le b^2$ gives

$$D_{c,k,i,j} = (C_{c,i,j})_{u_k,v_k}, \qquad D \in \mathbb{R}^{3\times K\times B_h\times B_w}, \qquad B_h = H/b, \quad B_w = W/b. \tag{2}$$

We omit the batch axis throughout. All blocks use the same retained ranks: selection is fixed for a chosen K, not adapted to image content. In the principal experiments $b = 8$. At inference, the model applies the transform to decoded images without entropy coding or coefficient quantization. Optional training augmentation is described in Appendix B.

Normalization. DC and AC coefficients have different distributions. We use signed logarithmic compression for DC and an affine transform followed by smooth clipping for AC:

$$\widehat{D}_{c,0,i,j} = s^{\mathrm{dc}}_c \, \mathrm{sign}(D_{c,0,i,j}) \log(1 + |D_{c,0,i,j}|) + \beta^{\mathrm{dc}}_c, \tag{3}$$

$$\widehat{D}_{c,k,i,j} = \tau_k \tanh\left(\frac{s^{\mathrm{ac}}_{c,k} D_{c,k,i,j} + \beta^{\mathrm{ac}}_{c,k}}{\tau_k} \right), \quad k > 0. \tag{4}$$

The DC scale and bias are learned per channel; the AC biases are learned per channel and rank. The AC scale is kept positive as $s^{\mathrm{ac}}_{c,k} = \exp(\theta_{c,k})$, and the AC threshold $\tau_k = \mathrm{softplus}(\gamma_k)$ is learned per rank and shared across channels.[1] The transform smoothly limits the magnitude of the affine AC activations.

Spatial resampling of coefficient maps. Before projection, SC-ViT reduces selected normalized coefficient maps to a grid of width $r_{c,k}$ and expands them back to the block grid:

$$\overline{D}_{c,k} = \mathcal{R}_{r_{c,k}}(\widehat{D}_{c,k}), \qquad \mathcal{R}_r(U) = \mathrm{Nearest}_{B_h \times B_w}\left(\mathrm{AdaptiveAvgPool}_{r \times r}(U)\right). \tag{5}$$

Here $\mathrm{AdaptiveAvgPool}$ returns an $r \times r$ grid. Unselected maps use the identity operator. At 448, chrominance AC maps use $r = 7$, luminance ranks 22 through 31 use $r = 14$ when they are retained ($K > 22$), and DC maps retain their full spatial grid. These resampling ranges are distinct from the projection groups below. This operation controls the spatial granularity of selected frequencies without changing the tensor shape. SC-Swin uses no coefficient map resampling.

Coefficient scales and chrominance gates. The stem applies a positive scale and a sigmoid gate to the prepared maps:

$$\widetilde{D}_{c,k,i,j} = \exp\big(\mathrm{clip}(\ell_{c,k}, -4, 2)\big) g_{c,k} m_k \overline{D}_{c,k,i,j}, \qquad g_{c,k} = \begin{cases} 1, & c = Y, \\ \mathrm{sigmoid}(\eta_{c,k}), & c \in \{C_b, C_r\}. \end{cases} \tag{6}$$

The learned parameters ℓ and η are shared over block positions in SC-ViT. SC-Swin learns position dependent versions $\ell_{c,k,i,j}$ and $\eta_{c,k,i,j}$. These gates are model parameters, not predictions conditioned on the input. For SC-ViT, the mask has $m_0 = 1$; during training each AC rank has $m_k \sim \mathrm{Bernoulli}(0.95)$, sampled once per step and shared across images, channels, and block positions. At evaluation all $m_k = 1$. The mask is applied after resampling and before stem weighting. SC-Swin instead samples its AC mask by block position and rank (Appendix H).

3.2 SC-ViT: SEPARATE DC AND AC PATHS

For a square image, let p be the token grid width and $t = B_h/p$ the number of blocks along each side of a token's spatial tile. At $H = 448$, $b = 8$, and $p = 14$, a token covers a 4×4 block tile, corresponding to a 32×32 pixel region. There are $N = p^2 = 196$ tokens. The DC branch uses grid width $2p = 28$, so four fine DC features are associated with each final token (Figure 1).

AC path. Partition retained AC ranks into groups $\mathcal{G}_g$. For token n, flatten each group's coefficients within its $t \times t$ tile into a luminance vector $y_{g,n}$ and a chrominance vector $c_{g,n}$. Their dimensions are $t^2|\mathcal{G}_g|$ and $2t^2|\mathcal{G}_g|$, respectively. Separate projections produce complementary feature subspaces:

$$z_{g,n} = \mathrm{LN}_g\left([W^Y_g y_{g,n} + b^Y_g \, ; \, W^C_g c_{g,n} + b^C_g]\right), \qquad a_n = \mathrm{LN}_{\mathrm{AC}}\left(\sum_g \alpha_g z_{g,n} \right). \tag{7}$$

Here $[\,;\,]$ denotes concatenation and the group outputs have width $D_{\mathrm{AC}} = 640$. For the principal $K = 16$ setting, the AC groups cover ranks 1 to 5, 6 to 9, and 10 to 15, with luminance allocations of

[1] SC-Swin learns $s^{\mathrm{ac}}_{c,k}$ directly as an unconstrained parameter, so it may change sign, and uses $\tau_k = \mathrm{softplus}(\gamma_k) + 10^{-3}$.

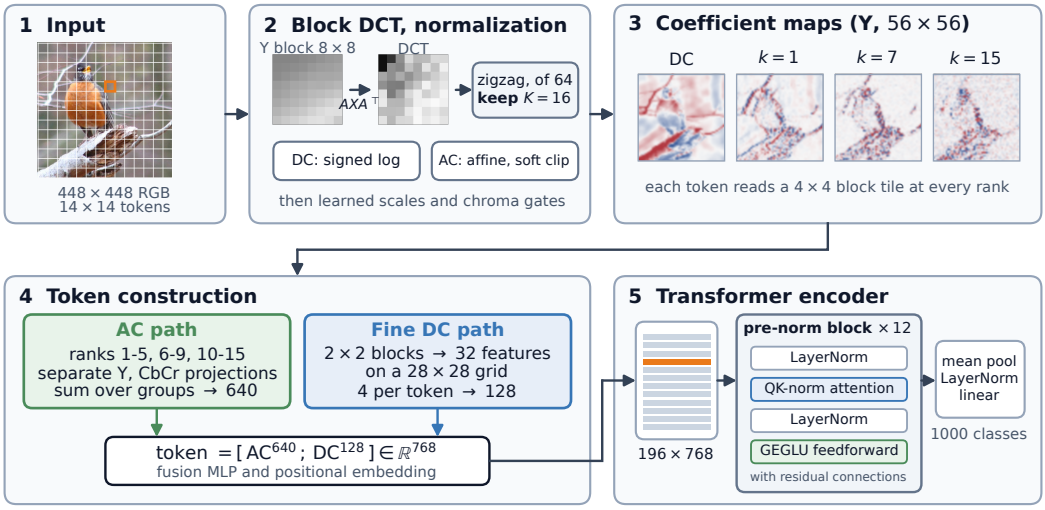

Figure 1: SC-ViT at 448×448, $b = 8$, and $K = 16$. The block DCT produces a 56×56 grid of blocks, of which the first 16 zigzag ranks are retained and normalized. Each of the 196 tokens reads a 4×4 tile of blocks: separate luminance and chrominance projections form three AC group features, normalized and summed to width 640, and four fine DC features of width 32 supply the remaining 128 dimensions. A residual fusion MLP and positional embeddings precede the 12 block encoder. The coefficient maps show selected ranks of the luminance channel before learned processing; the progressive configuration is specified separately.

70%, 72%, and 78%, respectively. The widths are rounded to integer dimensions and remain fixed during training; the chrominance gates instead learn continuous input weights. The group gates α_g equal one for direct training and control the introduction of spectral groups in progressive training. Appendix A gives the projection widths and the additional groups used at $K = 32$ and $K = 64$.

Fine DC path. At the 28×28 DC grid, each feature is a learned projection of the three channels' DC values in a 2×2 block region, followed by LayerNorm. Denote this 32 dimensional feature by u_m. For each final token n, let $\mathcal{N}(n)$ contain its four fine DC locations in raster order. The combined embedding is

$$v_n = [a_n \,;\, (u_m)_{m \in \mathcal{N}(n)}] \in \mathbb{R}^{768}, \qquad h_n = v_n + f(v_n) + e_n, \tag{8}$$

where e_n is a learned positional embedding and $f(v) = W_2\,\mathrm{GELU}(W_1\,\mathrm{LN}(v) + b_1) + b_2$. Initializing W_2 and b_2 to zero makes this residual branch initially contribute zero. The DC path preserves spatial variation within a token through separate projected features; it adds feature dimensions, not attention tokens.

The embeddings enter a stack of 12 pre-norm transformer blocks with width 768 and 12 attention heads, the layout of ViT-B/16. Two components of each block are adapted to the statistics of spectral inputs. AC coefficients are sparse: most are close to zero, and a few carry large values at edges and texture. Projected from such inputs, queries and keys can have widely varying norms, and a few large dot products can saturate the softmax. We therefore use query and key normalization (Henry et al., 2020): queries and keys are ℓ_2 normalized per head, which bounds each attention logit, and a learned per-head temperature, initialized to $1/\sqrt{64}$, sets the sharpness of attention. The feedforward layer is a GEGLU (Shazeer, 2020), whose multiplicative gate lets features derived from near-zero coefficients remain small instead of being passed through a fixed nonlinearity. Its hidden width of 2048 matches the weight-matrix parameter count of a GELU MLP with hidden width 3072, up to bias terms. In a separate 90 epoch study using the vanilla training setting of Appendix D, replacing GEGLU with a standard GELU MLP lowers accuracy from 76.1% to 75.3%. Both components are drop-in changes inside the block and are increasingly common in large transformers. The final tokens are averaged, then passed through LayerNorm and a linear classifier. Configured stochastic depth (Huang et al., 2016) rates increase linearly to 0.1; Appendix A notes that branch dropping may not have been active in the earlier direct implementation. The tokenizer provides a fixed 196 image

token grid of width 768, which can feed a ViT sized encoder. The evaluated flat model also changes attention and feedforward components, so its measured gain does not isolate the tokenizer alone.

3.3 Representation size and computation

With all coefficients retained, an orthonormal DCT is an invertible change of basis. An unconstrained affine projection after the full transform could be absorbed into an affine RGB patch projection, assuming the same input region is used. The substantive choices here are therefore the retained spectral subspace, nonlinear normalization, separate DC and AC paths, spatial resampling, and constrained channel projections. This distinction also motivates comparison with prior DCT tokenizers rather than attributing the gains to a basis change alone.

The retained tensor contains $M = 3KHW/b^2$ scalar values before further coefficient map resampling. With $b = 8$, 448 input, and $K = 16$, $M = 150{,}528$, equal to the number of RGB values at 224. At $K = 32$, $M = 301{,}056$, twice that number. Equal scalar counts do not imply equal entropy, encoded size, or semantic information.

For SC-ViT, the encoder sequence remains 196 tokens at either bandwidth. The principal reduction in attention cost comes from grouping blocks into this fixed grid. For SC-Swin, the initial 56×56 grid matches that of Swin-B with 4×4 RGB patches at 224. These constructions make encoder cost comparable across different input rasters. They do not eliminate the cost of decoding, resizing, colour conversion, the DCT, or stem projection. We report the recorded multiply accumulate counts in GMACs; one MAC corresponds to two floating point operations under that convention. Profiler coverage and parameter counting require separate attention (Appendix B).

4 Experiments

4.1 ImageNet classification at a fixed budget

We evaluate on ImageNet-1K (Deng et al., 2009), with approximately 1.28 million training images and 50,000 validation images. The fixed budget runs train from scratch for 50 epochs. Their common settings are AdamW, peak learning rate 5×10^{-4}, cosine decay to 10^{-5}, five warmup epochs, weight decay 0.05, RandAugment with two operations and magnitude 10, Mixup 0.8, CutMix 1.0, and configured maximum stochastic depth 0.1. Input channels use mean and standard deviation 0.5. The training pipeline also includes colour jitter and Gaussian blur. Direct SC-ViT training fixes the final spectral budget and depth from the first epoch, without budget distillation or an auxiliary prediction loss. Hierarchical models such as Swin are usually trained with their own recipe, which uses a higher learning rate, random erasing, and no colour jitter or blur. We also trained SC-Swin with such a recipe and obtained 79.33%, close to the 79.44% it reaches under the common recipe. Every result in Table 1 uses the common recipe, which differs from the published recipes of the baseline architectures. Other experiments and the additional 79.33% SC-Swin run use the protocols stated where they appear.

Table 1: ImageNet-1K classification after 50 training epochs. The ViT-B/32 row evaluates a spatial model on the same 448 raster as SC-ViT. ‡Measured with `fvcore` at the stated input. Parameter conventions are detailed in Appendix B.

Model	Input	Params (M)	GMACs	Top 1	Top 5
ViT-B/16	224	86.57	17.47	74.42	91.79
ViT-B/32	448	88.34	17.21‡	74.70	91.92
SC-ViT, $K = 16$	224	87.23	16.92	73.13	91.10
SC-ViT, $K = 32$	224	87.28	16.93	75.37	92.71
SC-ViT, $K = 64$	224	87.37	16.95	73.45	91.23
SC-ViT, $K = 8$	448	87.31	16.94	77.32	93.40
SC-ViT, $K = 16$	448	87.37	16.95	**77.36**	**93.48**
SC-ViT, $K = 32$	448	87.56	16.99	76.61	93.03
Swin-B	224	87.77	15.43	78.47	94.11
SC-Swin, $K = 16$	448	88.05	15.53	**79.44**	**94.65**

DCT based vision transformer. Table 1 reports an improvement of 2.94 percentage points for SC-ViT with $K = 16$ over ViT-B/16, at 16.95 versus 17.47 reported GMACs. These numbers describe our short training protocol rather than the published performance of the baseline architectures under their original recipes.

A spatial model on the same raster. To give a spatial model the same 448 input, we train ViT-B/32 at 448 (Table 1). ViT-B/32 at 448 also produces 196 image tokens and has nearly the same parameter count and compute as SC-ViT. It reaches 74.70%, only 0.28 points above ViT-B/16 at 224, so the larger raster alone gives a plain patch tokenizer little benefit at a fixed sequence length. SC-ViT with $K = 16$ exceeds it by 2.66 points.

Benefit from a hierarchical backbone. Our main experiment uses a plain ViT encoder, but the spectral input design can also feed another transformer architecture. To show that improvements to the encoder carry over, we adapt it to Swin, whose windowed attention and hierarchy improve on the plain ViT. SC-Swin uses one token per DCT block and the Swin-B hierarchy, with implementation details in Appendix H. It reaches 79.44%, above SC-ViT with $K = 16$ at 77.36%, and 0.97 points above the spatial Swin-B at 78.47%, with parameter and MAC increases of approximately 0.32% and 0.65%. The improvement that a stronger encoder brings to spatial inputs therefore also appears with spectral inputs, and the spectral design adds to it. Encoders with other attention designs, such as MaxViT, are natural candidates for the same combination.

The spectral stem at 224. At 224, SC-ViT with all 64 coefficients receives the same image and 150,528 scalar values as ViT-B/16, with the same 196 image token positions. It reaches 73.45%, compared with ViT-B/16's 74.42%. Selecting fewer coefficients changes the result: $K = 16$ reaches 73.13%, while $K = 32$ reaches 75.37%, 0.95 points above ViT-B/16 at the same image resolution. The $K = 32$ input retains 75,264 scalars, half as many as $K = 64$; $K = 16$ retains 37,632. One possible explanation is that ranks 32 through 63 add little useful signal: high frequency coefficients often have low energy and can be weakened by JPEG compression of the ImageNet source images. Omitting them may make it easier for the stem to weight informative coefficients, whereas also omitting ranks 16 through 31 may discard detail needed at 224. Among the tested bandwidths, $K = 32$ performs best for this architecture and task, but these single runs do not establish the cause of the differences or an optimum between tested values. All settings retain 196 image tokens; fewer coefficients reduce the stem input and slightly lower reported compute, from 16.95 GMACs at $K = 64$ to 16.93 at $K = 32$ and 16.92 at $K = 16$.

Retained bandwidth at 448. Image codecs reduce data in two complementary ways: lowering the spatial resolution, and coarsening high frequency coefficients within each block, which JPEG does through quantization (Wallace, 1991). Standard RGB patch tokenization controls spatial grouping, while SC-ViT also selects spectral bandwidth: it keeps the 448 raster and discards the high frequency ranks outright, the limit of coarse quantization. At $K = 16$ the retained tensor has 150,528 values, exactly the size of an RGB image at 224, yet it represents an image rasterized at four times the area. With this input SC-ViT reaches 77.36%, 2.94 points above ViT-B/16 while using the same number of input scalars and tokens. At $K = 8$ the retained tensor is half that size, 75,264 values, and SC-ViT reaches 77.32%, 2.90 points above ViT-B/16. Retaining more ranks does not help: $K = 32$ doubles the input to 301,056 values and reaches 76.61%. High frequency content can be weakened by JPEG compression of the source images, and many source images have a shorter side below 448 (Appendix E.1), so resizing can alter their high frequency content. The $K = 16$ result is consistent with these extra ranks being less useful in this setting, but does not establish why accuracy falls at $K = 32$. Because the raster and token grid are fixed, all three settings cost between 16.94 and 16.99 GMACs, so the retained bandwidth trades input size against accuracy at essentially constant encoder cost.

Limits of the comparison. We interpret the gains as observed operating points under the stated settings, without claiming superiority under all training recipes. Appendix B records the provenance and differences between the direct and progressive experiments.

4.2 PROGRESSIVE INFERENCE FROM ONE CHECKPOINT

The progressive extension selects a triple (K_s, d_s, h_s): retained bandwidth, encoder exit depth, and classifier. The five settings in Table 2 share a 14×14 token grid and encoder weights. Newly introduced spectral groups are gated in over three epochs. Previously activated groups and blocks remain trainable, future capacity is frozen, and only the current exit head is updated directly. For a sampled earlier stage $r < s$, the classification loss is

$$\mathcal{L} = \mathrm{CE}(z_s, y) + \tfrac{1}{2}\,\mathrm{CE}(z_r, y) + \tfrac{1}{4} D_{\mathrm{KL}}(\mathrm{sg}[\mathrm{softmax}(z_s)] \,\|\, \mathrm{softmax}(z_r))\,. \tag{9}$$

Here z_s, z_r are logits, y is the target distribution (including Mixup or CutMix targets when used), CE is soft-target cross entropy, and sg stops the teacher gradient. The KL term is averaged over the minibatch. At stage zero only the first term is used. The current stage teaches the earlier stage through the KL term; freezing the earlier classifier still permits gradients through its shared features. The schedule and training procedure are given in Appendix C.

Table 2: Progressive SC-ViT trained for 300 epochs, with all exits evaluated from one checkpoint, and SC-ViT with $K = 16$ trained directly at full depth for 300 epochs. The progressive model omits the fine DC branch of Section 3.2. The direct row provides a full depth reference from a separate run. Stage parameter counts include the active stem, blocks, and every head up to the exit depth; the direct row reports trainable parameters. The first three progressive rows form the observed accuracy and MAC frontier of the checkpoint.

Stage	K_s	d_s	Params (M)	GMACs	Top 1
0	6	3	23.47	4.42	72.9
1	10	4	31.40	5.83	76.5
2	16	6	46.44	8.62	**77.2**
3	16	8	61.39	11.40	76.5
4	16	10	76.34	14.18	76.6
Direct	16	12	87.37	16.95	78.44

The best progressive exit, stage two, reaches 77.2% at 8.62 GMACs, 50.9% of the reported compute of the direct full depth reference. Stages zero and one reach 72.9% and 76.5% at 4.42 and 5.83 GMACs. Stages three and four do not improve on stage two, and every progressive exit remains below the 78.44% (top 5 92.66%) of the directly trained full depth reference after 300 epochs.

This ordering is consistent with the schedule, not necessarily a property of the retained spectrum: stages two through four all use $K = 16$. A progressive model is useful when its lighter exits already perform well, so the schedule deliberately gives the early stages the most weight. Stages are introduced in sequence, so the earliest stages are trained longest: the three blocks of stage zero train for all 300 epochs, whereas the blocks added at stages three and four start at epochs 144 and 216 and train for 156 and 84 epochs (Table 7). The later blocks also train under a lower learning rate, since the cosine schedule has decayed to about half of its peak by epoch 144 and to about a fifth by epoch 216; the learning rate boost at each stage transition offsets this only partly. Figure 2 in the appendix shows the validation accuracy of every exit over training. Each update combines the current exit with a sampled earlier exit, and the distillation term in Eq. 9 transfers knowledge from the current exit to the earlier one. The shared blocks are thus optimized to serve the low budget exits well. The deeper exits trade a small amount of peak accuracy for this, and other schedules could weight the stages differently, for example to favour the full depth exit.

The result is a single checkpoint with several inference budgets. The first three stages increase both the number of retained DCT coefficients and encoder depth; the last two increase depth only. This offers a different inference control from changing the patch size or exit depth alone. FlexiViT (Beyer et al., 2023) varies the patch size and hence the number of spatial tokens. In SC-ViT, the tile side t similarly controls the spatial token grid. Combining flexible token grids with progressive spectral budgets is a possible extension, which we have not tested. Early exit models, in turn, vary only the depth. Progressive SC-ViT adds higher zigzag ranks in the early stages while keeping the token grid fixed. The checkpoint provides three operating points at 4.42, 5.83, and 8.62 GMACs. The largest reaches 77.2%, about 1.2 points below direct full depth training at about half its compute,

and the smallest runs at about a quarter of that compute while reaching 72.9%. The schedule is also extensible: further stages can be appended and trained from this checkpoint.

4.3 Robustness to test-time JPEG re-encoding

A spectral input discards exactly the coefficients that lossy compression degrades most, which suggests that it may also be less sensitive to compression at test time. To test this, we re-encode every ImageNet validation image as JPEG at quality q after the resize and centre crop, at each model's own input resolution, and evaluate without retraining (Table 3). Lower q means stronger compression.

Table 3: ImageNet-1K top 1 accuracy under test-time JPEG re-encoding at quality q, grouped by input resolution. "None" is the unmodified validation pipeline, and Δ is the change from None to $q = 25$. Unmarked models use the common 50 epoch recipe. Progressive exits come from the single checkpoint of Table 2. †Trained with JPEG augmentation (probability 0.3, quality 60 to 100). *Public `timm` checkpoint trained with its original recipe, included as a reference outside the common protocol.

Model	Input	None	$q = 95$	$q = 75$	$q = 50$	$q = 25$	Δ
SC-ViT, $K = 32^{\dagger}$	224	75.37	75.25	74.10	73.56	69.88	−5.49
SC-ViT, $K = 64^{\dagger}$	224	73.45	73.25	71.82	70.23	65.60	−7.85
ViT-B/16	224	74.42	74.06	70.53	67.82	63.09	−11.33
ViT-B/16*	224	76.78	75.65	72.59	70.01	65.74	−11.04
SC-ViT, $K = 16^{\dagger}$	448	77.36	77.32	77.25	76.96	75.24	−2.12
Progressive exit 0, $K = 6^{\dagger}$	448	72.92	73.02	72.74	72.14	70.36	−2.56
Progressive exit 1, $K = 10^{\dagger}$	448	76.54	76.57	76.24	75.61	73.89	−2.65
Progressive exit 2, $K = 16^{\dagger}$	448	77.18	77.13	76.69	76.29	74.46	−2.72
Progressive exit 3, $K = 16^{\dagger}$	448	76.50	76.41	75.91	75.35	73.64	−2.86
Progressive exit 4, $K = 16^{\dagger}$	448	76.57	76.46	75.88	75.34	73.57	−3.00
ViT-B/32	448	74.70	74.61	73.80	72.01	68.91	−5.79

At the same 448 input resolution, ViT-B/32 loses 5.79 points at $q = 25$, while direct SC-ViT with $K = 16$ loses 2.12 points and the progressive exits lose 2.56 to 3.00 points. At 224, SC-ViT with $K = 32$ loses 5.49 points and $K = 64$ loses 7.85 points, compared with 11.33 for ViT-B/16. These are descriptive comparisons of complete models. The flat SC-ViT models and progressive exits were trained with JPEG augmentation, which may contribute to their smaller losses; the progressive checkpoint also has a longer training schedule. The trend of smaller losses for spectral inputs was also observed with Swin encoders, but those results are omitted here to keep the table focused on ViT.

4.4 Image quality assessment

KADID-10k (Lin et al., 2019) contains 10,125 distorted images derived from 81 references, with 25 distortion types and five severity levels. We evaluate both no reference prediction, which receives only the distorted image, and full reference prediction, which also receives the reference image. The from-scratch experiments use a split by reference content, with approximately 10% held out, and train for 100 epochs. Their spectral configuration uses 448 inputs and up to 64 coefficients. These runs are separate from the 60 epoch experiments initialized from ImageNet-1K classification checkpoints in the lower part of Table 4.

Against ViT-B/16 at 384, the spectral model improves NR SRCC by 0.1711 and FR SRCC by 0.0114 (Table 4). When both backbones start from ImageNet-1K classification checkpoints, SC-ViT with $K = 16$ improves NR SRCC from 0.7726 to 0.8040 and FR SRCC from 0.9194 to 0.9295 over ViT-B/16. With common NR and FR regression-head designs, these experiments ask how general-purpose visual backbones perform on the same quality-prediction task. SC-ViT has the highest reported correlations among the tested backbones in both the from-scratch and initialized settings. The initialized $K = 16$ result shows gains also with a smaller spectral budget. Dedicated IQA models may use specialized heads, contrastive objectives, or quality-specific pretraining; evaluating those models is a separate question. Appendix G records the differences in input resolution and training protocol.

Table 4: Quality prediction on KADID-10k. NR and FR denote no reference and full reference prediction. SRCC and PLCC are Spearman and Pearson correlations. Upper block: trained from scratch for 100 epochs. Lower block: initialized from ImageNet-1K classification checkpoints and trained for 60 epochs with a common recipe. The comparison uses common regression head designs; input resolutions and training schedules differ.

		NR		FR	
Model	Params (M)	SRCC	PLCC	SRCC	PLCC
ViT-S/16, 224	22	0.4172	0.4146	0.8706	0.8700
ViT-B/16, 224	86	0.4172	0.4146	0.8835	0.8847
ViT-B/16, 384	86	0.5353	0.5335	0.8891	0.8894
SC-ViT, 448, $K = 64$	91	**0.7064**	**0.6755**	**0.9005**	**0.9011**
ViT-B/16, 224, pretrained	86	0.7726	0.7683	0.9194	0.9175
SC-ViT, 448, $K = 16$, pretrained	87	**0.8040**	**0.7779**	**0.9295**	**0.9305**

4.5 Design sensitivity

The additional experiments explore DCT block size, projection design, and channel capacity. They use a vanilla training setting shared within each study and, for the block and projection comparisons, a generalized tokenizer without the fine DC stem of the principal SC-ViT result. Appendix D separates these configurations; their absolute accuracies should not be compared with the main results as if only one component had changed. In the reported generalized configurations, a 16×16 block setting reaches 76.02% and a 32×32 setting reaches 75.35%. Projection variants span 75.35% to 75.71%. These results suggest that spatial grouping and projection matter, but do not isolate the effect of block size relative to the principal model. A separate 90 epoch channel allocation study reaches 75.6%, 76.1%, and 75.9% with luminance allocations of 50%, 62.5%, and 75%, respectively. We use it as evidence for treating colour channels separately, not as proof that one ratio is optimal for every stem.

5 Discussion and conclusion

ImageNet images are stored as JPEG, so compression can affect the coefficients available after decoding. SC-ViT forms tokens from a selected set of DCT coefficients on a dense block grid, combining spatial sampling with bandwidth selection at a fixed image token count. The 50 epoch results favour this complete model at 448, but the experiments also change components inside the encoder and do not isolate the tokenizer as the sole cause of the gain.

The shared 50 epoch recipe makes the main comparisons easier to interpret, although it produces lower accuracies than longer training schedules. The Swin experiment shows that the spectral stem can also be used with a hierarchical encoder.

Results come from single runs, and complexity is reported in MACs; latency, memory, and energy depend on the implementation. The image-token count remains fixed, while the spectral stem adds its own processing cost. In the recorded ImageNet training set, 87.91% of source images have a shorter side below 448 pixels. This statistic does not determine the fraction upsampled by the training crop pipeline, but resizing can affect the high frequency content seen by the model. Video motivated the progressive design and is a direct next experiment: a model could allocate greater spectral bandwidth and encoder depth to keyframes, then use lower budgets for intermediate frames that primarily convey motion or other temporal information. Whether this allocation improves video quality or efficiency requires experiments with temporal models and realistic latency measurements. Arbitrary resolutions and self supervised training are further directions.

SC-ViT brings a principle from image compression into transformer design: spatial sampling and retained frequencies can be chosen before the image reaches the encoder. Its recognition results, the Swin extension, and the quality-prediction experiments suggest that this way of allocating input information has value across visual tasks. We see particular promise in settings where the useful spatial and spectral detail changes with the task or over time.

REPRODUCIBILITY STATEMENT

The transform and token construction are specified in Section 3. Appendix B records training settings, complexity conventions, and differences between the experimental protocols. Appendix C specifies the progressive schedule, and Appendix G describes quality prediction. Results are reported as individual runs; no confidence intervals are inferred from them. Upon acceptance, we will release all training scripts and trained model checkpoints used in this work to support reproducibility.

AI USE STATEMENT

We used generative AI tools to help find and check related work, write evaluation and utility scripts, edit training code, and set up and run experiments. We also used them to draft, revise, and polish portions of this paper. The method, experimental design, training runs, and interpretation of the results are the authors' own. The authors take responsibility for the final paper and for checking every reported result against the original records before submission.

REFERENCES

Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. FlexiViT: One model for all patch sizes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations (ICLR)*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 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.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. In *International Conference on Learning Representations (ICLR)*, 2020.

David J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394, 1987.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

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

Jongho Lee and Hyun Kim. DCT-ViT: High-frequency pruned vision transformer with discrete cosine transform. *IEEE Access*, 12:80386–80396, 2024.

Xinyu Li, Yanyi Zhang, Jianbo Yuan, Hanlin Lu, and Yibo Zhu. Discrete cosin transformer: Image modeling from frequency domain. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5468–5478, 2023.

Hanhe Lin, Vlad Hosu, and Dietmar Saupe. KADID-10k: A large-scale artificially distorted IQA database. In *International Conference on Quality of Multimedia Experience (QoMEX)*, 2019.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021.

Antonio Ortega and Kannan Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 15(6):23–50, 1998.

Hongyi Pan, Emadeldeen Hamdan, Xin Zhu, Ahmet Enis Cetin, and Ulas Bagci. Discrete cosine transform based decorrelated attention for vision transformers. *arXiv preprint arXiv:2405.13901*, 2024.

Zequn Qin, Pengyi Zhang, Fei Wu, and Xi Li. FcaNet: Frequency channel attention networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

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)*, volume 34, pp. 980–993, 2021.

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

Gregory K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34 (4):30–44, 1991.

Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1740–1749, 2020.

Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.