VISION FLUXER WITH DYNAMIC GLOBAL RECEPTIVE FIELD

Anonymous authors
Paper under double-blind review

ABSTRACT

A token mixer should provide a global receptive field and input-dependent mixing at tractable cost: convolution, with its local, static kernel, offers neither, and self-attention offers both at a cost quadratic in the number of tokens. Frequency-domain convolution attains a global receptive field at $O(N \log N)$, but its filter is fixed after training. Conditioning the filter pointwise on the input admits two constructions: the filter may be generated in frequency from the spectrum, or in space from the signal and then transformed. A linear generator cannot distinguish the two, and the choice between them has gone unexamined. Here we show that once the generator is nonlinear the two constructions are no longer interchangeable, and adopt the spatial one: Fluxer generates a full-resolution kernel as a pointwise projection of the input, so that the mixing amounts to a bilinear interaction between two nonlinear views of a single feature map. Three separations position Fluxer relative to static spectral filters, local convolution, and frequency-pointwise masks; our experiments bear most on the last: in the same backbone, a frequency-pointwise generator recovers only a small part of the gap between a static filter and Fluxer on ImageNet-1K. As a drop-in backbone, Vision Fluxer (ViF) achieves the most favorable accuracy–compute trade-off among representative single-stream hierarchical backbones on classification, detection, and segmentation, indicating that convolution, once equipped with a dynamic global kernel, constitutes a competitive foundation.

1 INTRODUCTION

A token mixer is expected to provide three properties at once: a *global* receptive field, so that any location can influence any other; *input-dependent* mixing, so that interactions follow image content rather than being fixed a priori; and *low* cost, so that the operator remains practical at the resolutions dense prediction requires. Convolution, the classical choice, provides translation equivariance and linear cost but neither of the first two: its kernel is *local* and *static*. Self-attention (Vaswani et al., 2017; Dosovitskiy et al., 2021) removes both limitations at once and has displaced convolution as the default mixer, at a cost quadratic in the number of tokens, which is prohibitive at high resolution.

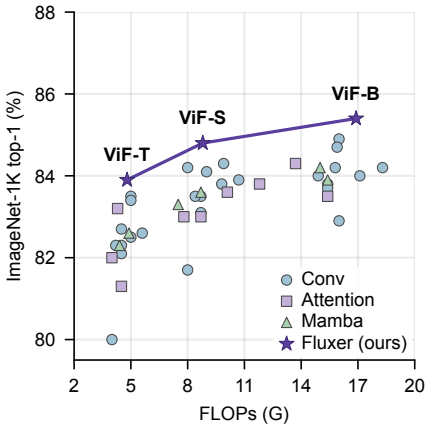

Figure 1: **Accuracy–compute trade-off on ImageNet-1K.** We plot top-1 accuracy against FLOPs at 224×224 for the 45 backbones of Table 1, which comprise every row of that table and no other model. ViF attains the most favorable trade-off at every scale, and the margin persists as the models grow. No baseline matches ViF-T below 8.0G FLOPs, 1.7 times its budget, and none reaches ViF-B at any budget shown, although more heavily optimized architectures outside this set do.

Each sub-quadratic family recovers only part of what attention provides (Section 2). The frequency-domain family comes closest: an element-wise product of spectra realizes a convolution whose kernel spans the entire image at $O(N \log N)$, but the filter is fixed after training. Conditioning the filter on the input supplies the remaining property, which AFF (Huang et al., 2023) and Orchid (Karami & Ghodsi, 2024) accomplish in different ways, and the resulting design space is organized along two axes: the domain in which the filter is generated, pointwise in frequency from the spectrum as in AFF, or

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.

pointwise in space from the signal and then transformed; and whether the kernel is rendered invariant to translation, as in Orchid, or left equivariant so that it can encode position. Orchid's choice on the second axis is deliberate. On the first, a linear generator cannot distinguish the two domains, and which domain the filter should be generated in has not been examined.

The question is substantive, because once the generator is nonlinear the two constructions are no longer interchangeable, in a sense that Section 2 makes precise. We introduce Fluxer, a Dynamic Global Convolution (DGC) that generates its kernel in the spatial domain and leaves it equivariant: the kernel is a direct projection of the input feature map, refined only by a nonlinearity and a normalization, transformed, multiplied against the spectrum of the signal, and transformed back.[1] By the convolution theorem the result is exactly a circular convolution whose kernel spans the entire image and is *generated on the fly*: global, dynamic, and $O(N \log N)$, without ever materializing an $N \times N$ affinity matrix. The mixing is a bilinear interaction between two nonlinear views of a single feature map, and because the kernel retains its phase it encodes position as well as content.

Section 3 states three separations, with proofs in the appendix. No *single* static spectral filter, under any parameterization, approximates even the simplest dynamic instance better than the zero operator, a degree argument that says nothing about deep static stacks. A stack of local convolutions at fixed resolution requires depth $\Omega(\sqrt{N}/r)$ to match the global aggregation performed by one Fluxer layer. Finally, a spatially generated filter is not a pointwise function of the spectrum, so it is not contained in the frequency-pointwise class. A fourth result relates the operator to attention: at any fixed input it is a circulant member of the dynamic all-to-all family, the corner of attention occupied by a purely relative-position score.

The third separation is the one our experiments bear on most directly. Retraining the backbone with only its kernel branch replaced compares the generators directly: a frequency-pointwise generator recovers only 0.1 point of the 0.7-point gap between a static filter and Fluxer on ImageNet-1K, and one that discards the phase stays 0.4 below Fluxer. Instantiated as a standard four-stage hierarchical backbone, Vision Fluxer (ViF) lies on the accuracy–compute frontier of Fig. 1 and outperforms representative Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), and Vision Mambas (ViMs) on ImageNet-1K, COCO, and ADE20K at matched parameter and FLOPs budgets.

2 Background

Global token mixing. Self-attention (Vaswani et al., 2017; Dosovitskiy et al., 2021) relates every pair of tokens with input-dependent weights at quadratic cost, and each cheaper alternative forgoes part of that: windowed and local attention (Liu et al., 2021; Hassani et al., 2023) bound the receptive field, linearized attention (Katharopoulos et al., 2020) discards the softmax and with it some spatial selectivity, state-space models (Gu & Dao, 2024; Zhu et al., 2024; Liu et al., 2024) attain linear time only by traversing the two-dimensional grid as a causal sequence, and large-kernel convolutions (Liu et al., 2023; Chen et al., 2024; Wang et al., 2023) widen the field but remain local. Several of these do achieve global, input-conditioned mixing below quadratic cost; what none retains alongside it is a native two-dimensional convolutional structure, a single mixing operation defined on the grid the image occupies rather than on a scan order or a window. Recovering that structure at $O(N \log N)$ is what motivates the frequency domain.

Where the filter comes from. By the convolution theorem, one element-wise product in the spectral domain realizes a full-resolution two-dimensional global convolution at $O(N \log N)$. The constructions built on this idea differ in a single respect: what the filter is a function of. For GFNet (Rao et al., 2021), FNet (Lee-Thorp et al., 2022), and the Fourier Neural Operator (FNO) (Li et al., 2021b), nothing: the filter is learned once and fixed. AFNO (Guibas et al., 2022), the adaptive variant of the FNO, stands apart: it never multiplies its mask back against the spectrum it was computed from, so the layer is not a convolution with a generated kernel at all, and is not even circulant at a fixed input. AFF (Huang et al., 2023) restores that multiplication, which is precisely what makes the layer a convolution with a kernel as large as the feature map, but its mask reads one frequency at a time, $\mathbf{H}(\xi) = g_\xi(\hat{\mathbf{u}}(\xi))$, so no coefficient of the filter couples two frequencies. Orchid (Karami

[1] The name is for the kernel, which is in *flux*: it is synthesized anew from every input rather than held fixed in a parameter tensor, so what convolves the signal changes with the signal. It is also a finite impulse response, with no poles to place and no recursion to unroll; one projection and a single transform produce it.

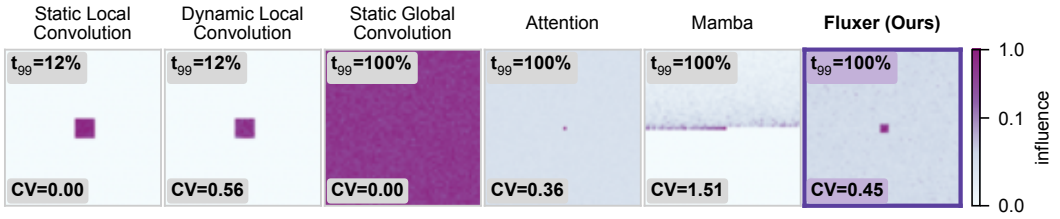

Figure 2: **Intrinsic single-layer receptive fields of six token mixers.** Reach and adaptivity are independent axes: making a local kernel dynamic does not enlarge its reach, and a scan makes that reach directional. Each mixer is randomly initialized and probed identically, with the center output back-propagated to the input and the per-pixel gradient magnitude plotted on a $\gamma = 0.25$ scale so that the weak global fields remain visible. t_{99} denotes the smallest centered square, as a fraction of the image width, that holds 99% of the influence mass, and CV denotes the coefficient of variation of the map over 72 inputs, which is exactly zero for an input-independent operator. The scan orders the two-dimensional grid causally, so the 49% of tokens after the center in raster order have *exactly* zero influence, which a centered-square statistic cannot register: the t_{99} reported for the scan therefore overstates its reach.

& Ghodsi, 2024) deliberately passes the squared magnitude of the transform: the kernel is thereby shift-*invariant*, so it captures what the image contains but not where. Our kernel branch is a channel projection, a nonlinearity, and a normalization, all pointwise in space and involving no spatial convolution, so the kernel at displacement $\mathbf{r}$ *is* the feature map at $\mathbf{r}$, recoded: the layer correlates the image against a pointwise recoding of itself. In the frequency domain this is AFF's bottleneck reversed, every frequency entering every coefficient (Eq. (A1)), and the phase survives, so the kernel can encode position. The domain in which the filter is generated, and whether it is left equivariant to translation, are the two axes; Propositions 2 and 4 establish that the classes differ, and Fig. 5 compares generators on either side, neither proposition indicating which side prevails. Generating convolution weights from the input predates this line of work (Jia et al., 2016; Yang et al., 2019; Chen et al., 2020; Li et al., 2021a), but those kernels are $K \times K$ and applied on the grid. The nearest relative among the methods absent from our tables is DFFormer (Tatsunami & Taki, 2024), whose filter is a softmax-weighted sum of learned frequency-domain filters with weights computed from the spatial mean of the input, so that every coefficient reads one global summary rather than its own frequency or the full map; the remainder vary the transform, take the product in space, or are ancestors of this construction (Yao et al., 2022; Patro & Agneeswaran, 2023; Yang et al., 2022; Guo et al., 2023; Poli et al., 2023; Chi et al., 2020).

3 Formulation

The operator. Given $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ with $N = HW$ tokens, a single linear projection produces a signal branch $\mathbf{u}$, a kernel branch $\mathbf{c}$, and a gate branch $\mathbf{z}$, each in $\mathbb{R}^{H\times W\times D}$ with $D = eC$. The branch refinement is a matter of architecture (Section 4); all long-range interaction takes place below.

Let $\mathcal{F}$ denote the orthonormal two-dimensional real fast Fourier transform (FFT), taken on the $H \times W$ grid rather than on a flattened sequence so that the filter it induces is a genuine two-dimensional circular one, and $\mathcal{F}^{-1}$ its inverse. Fluxer convolves the signal branch with the kernel branch, which by the convolution theorem is an element-wise product of spectra:

$$\mathbf{y} \;=\; \mathbf{u} \circledast \mathbf{c} \;=\; \mathcal{F}^{-1}\big(\mathcal{F}(\mathbf{c}) \odot \mathcal{F}(\mathbf{u})\big), \tag{1}$$

with $\odot$ element-wise, $\mathbf{H} := \mathcal{F}(\mathbf{c})$ the induced spectral filter, and $\hat{\mathbf{v}} := \mathcal{F}(\mathbf{v})$ below. Since $\mathcal{F}$ is unitary, $\circledast$ denotes throughout the *unitarily normalized* circular convolution defined by the right-hand side, which is $N^{-1/2}$ times the textbook one. The gate branch then modulates the result, and a projection returns it to C channels, $\mathbf{Y} = \mathbf{W}_o\,\mathrm{LN}(\mathbf{y} \odot \mathbf{z})$. The implementation adds a learnable per-channel $\mathbf{s}$ to $\mathbf{H}$, so that the layer computes $\mathbf{u} \circledast \mathbf{c} + \mathbf{s} \odot \mathbf{u}$ (Listing 1); Eq. (1) omits this term because a static filter represents it exactly. Making $\mathbf{c}$ input-independent recovers such a filter (Proposition 1), and the contrast reduces to a single symbol: a $\hat{\mathbf{w}}$ fixed after training where Eq. (1) places $\hat{\mathbf{c}}$.

Complexity. Evaluating Eq. (1) costs $O(NCD + ND\log N)$ in time and $O(ND)$ in memory, neither term quadratic in the token count (Proposition 6). Both terms matter, and at recognition resolutions the first dominates: what the operator spends is the projection producing its three branches rather than the transform mixing them, which is why the wall-clock measurements of Fig. 6 do not follow the asymptotics. Figure 3 identifies where the transform begins to pay off.

Why dynamic. A static filter is the input-independent limit of Eq. (1), and our first theorem shows that this generalization is strict in the strongest sense: no static filter approximates even the simplest dynamic instance better than the zero operator does.

Theorem 1 (Dynamic mixing is not approximable by static filters)**.** *Let $\mathcal{S}$ be the class of static spectral filters, that is, maps $\mathbf{u} \mapsto \mathbf{w} \circledast \mathbf{u}$ with an input-independent kernel $\mathbf{w}$, and let $T_1(\mathbf{u}) = \mathbf{u} \circledast \mathbf{u}$ be the operator that Eq. (1) realizes when $\mathbf{c} = \mathbf{u}$. Then for the operator seminorm $\|T\|_B = \sup_{\|\mathbf{u}\| \leqslant 1} \|T(\mathbf{u})\|$,*

$$\inf_{T \in \mathcal{S}} \|T - T_1\|_B = \|T_1\|_B > 0, \qquad (2)$$

attained at $T = 0$. No static filter, under any kernel, any parameter count, and any spectral or implicit parameterization, approximates T_1 better than the zero operator.

Remark 1 (The implemented layer)**.** The layer computes $T_1'(\mathbf{u}) = \mathbf{u} \circledast \mathbf{u} + \mathbf{s} \odot \mathbf{u}$ rather than T_1. The additional term is itself a static filter, a convolution with a scaled delta, so writing a candidate as $\mathbf{w} = \mathbf{s}\delta + \mathbf{w}'$ leaves the infimum at $\|T_1\|_B$, now attained at the pass-through rather than at zero: the static class absorbs that term exactly and none of the bilinear one. AFF and Orchid carry the same term. The bound does not survive depth, and its odd-symmetry argument does not cover AFNO; both qualifications are stated in Remark 3.

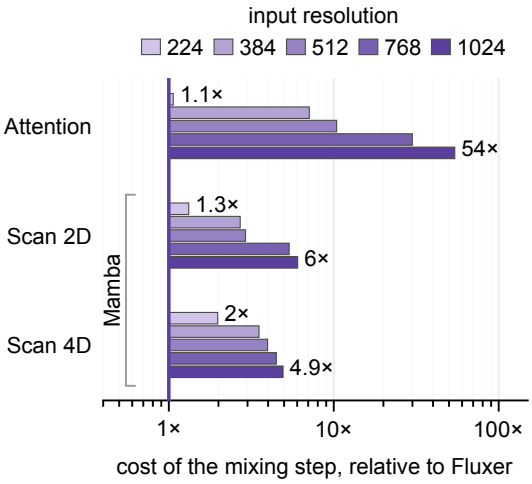

Figure 3: **Cost of the mixing step relative to ours, at five input resolutions.** Each operator shown is slower than ours at every resolution on this node, and the margin widens with resolution; at 224 the order depends on the node (Section E). Each is timed at width 96 and batch 1 on one H100, in its own process, as the median of nine repeats, with the projections every mixer pays kept outside the timed region; transforms run in `float32`, scans and attention in `bfloat16`. Table A7 reports every figure, including the single scan and memory.

Why global. Theorem 1 separates dynamic from static by degree. The second separation is between global and local, and turns on reach instead: a single layer that already sees the entire feature map, against a stack that must grow before it can reach across one.

Theorem 2 (One global layer replaces $\Omega(S/r)$ local ones)**.** *On an $S \times S$ feature map (so that $N = S^2$ tokens), let $M(\mathbf{u}) = \mathrm{mean}(\mathbf{u})\mathbf{1}$, and let $\mathcal{A}_{\mathrm{loc}}(r, L)$ be the class of maps built from at most L convolutions of kernel width r at a* fixed *resolution, interleaved with arbitrary pointwise operations. A single Fluxer layer realizes M exactly, whereas every $G \in \mathcal{A}_{\mathrm{loc}}(r, L)$ with $L(r-1)/2 \leqslant S/8$ satisfies $\|M - G\|_B \geqslant 1/12$. Approximating M within $1/12$ therefore requires depth $L > \frac{S}{4(r-1)}$, that is, $\Omega(\sqrt{N}/r)$, within the local class.*

Remark 2 (What Theorem 2 does and does not settle)**.** The fixed resolution is what the bound is about, and the absence of downsampling from $\mathcal{A}_{\mathrm{loc}}$ is what makes it a statement about kernel width rather than about depth alone. A hierarchical backbone doubles the reach of a fixed-width kernel at every downsampling, covering the map in $O(\log S)$ layers, and the convolutional baselines of Table 1 are all of that kind. Theorem 2 separates a stack at one resolution, which is the setting in which a single mixer layer is compared, rather than an entire backbone.

Section A proves both theorems and collects the remaining analysis: an input-independent kernel reduces Eq. (1) to a static spectral filter (Proposition 1); a spatially generated filter is not a pointwise function of the spectrum, so the two constructions of Section 2 form different operator classes (Proposition 2); at a fixed input the layer is a circulant member of the dynamic all-to-all family, the corner of attention occupied by a purely relative-position score (Proposition 3 and Corollary 1); and the layer trades exact translation equivariance for a kernel that can encode position, is spectrally smoothed by the preceding depthwise convolution, and is Lipschitz on bounded sets (Propositions 4 and 5 and Observation 1). Of the three separations, the experiments bear most directly on the domain in which the filter is generated, then on static against dynamic; the local-depth bound is the most idealized, its target being a global average that a squeeze-and-excitation path realizes in one step, and we treat it accordingly.

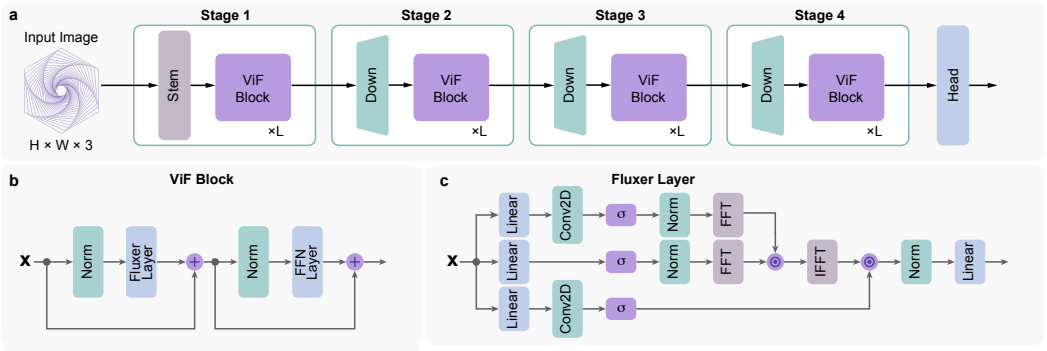

Figure 4: **Architecture of the ViF backbone.** ViF is an otherwise standard four-stage hierarchical backbone whose only non-standard component is the token mixer. *a* An overlapped convolutional stem reduces the $H \times W \times 3$ input to $H/4 \times W/4 \times C$, after which four stages each stack the number of ViF blocks marked $\times L$ in the figure, with the per-stage counts given in Table A10; stride-2 overlapped downsampling separates consecutive stages, halving the resolution and doubling the width at every transition. *b* The ViF block places the Fluxer layer and a gated MLP in pre-normalized residual branches. *c* Within the Fluxer layer, a single linear projection yields the signal $\mathbf{u}$ (linear, depthwise convolution, SiLU, normalization), the kernel $\mathbf{c}$ (linear, SiLU, normalization), and the gate $\mathbf{z}$ (linear, depthwise convolution, SiLU), with the nonlinearity marked σ in the figure. The signal and kernel spectra are multiplied element-wise, realizing Eq. (1), and the gate modulates the inverse transform before the output projection. The kernel is the one branch carrying no spatial convolution, a choice we argue for below.

4 Architecture

Overview. ViF follows the four-stage hierarchical design (Fig. 4), consistent with established vision backbones (Liu et al., 2021; 2022; 2024). An input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ is first mapped by an overlapped convolutional stem into a $\frac{H}{4} \times \frac{W}{4} \times C$ feature map, which then passes through four successive stages. Each stage stacks multiple ViF blocks and is followed by a stride-2 downsampling layer, except for the final stage, so that the resolution is progressively halved and the channel width doubled, yielding a feature pyramid at strides $\{4, 8, 16, 32\}$. A task-specific head processes the final feature map for downstream applications. The three variants ViF-T/S/B scale depth and width to comparable parameter and FLOPs budgets (Table A10), with further details in Section C.

Backbone design. The ViF block is the fundamental building unit, composed of the Fluxer operator and a gated convolutional MLP, each wrapped in a pre-normalized residual branch (He et al., 2016). The operator performs the global, dynamic token mixing of Eq. (1), while the MLP facilitates cross-channel information flow, consistent with the feed-forward blocks of classical ViTs. A depthwise Conditional Position Encoding (CPE) precedes both the operator and the MLP in order to inject local inductive bias (Guo et al., 2022), so that a block applies, each within its own residual branch, a CPE, the pre-normalized Fluxer operator, a second CPE, and the pre-normalized MLP. Because the operator already supplies a global, position-aware receptive field, neither absolute positional embeddings nor relative-position biases are required. The stem and every inter-stage transition use overlapped 3×3 stride-2 convolutions rather than the non-overlapped ones of hierarchical ViTs (Liu et al., 2021) and ViMs (Liu et al., 2024), which preserves spatial detail at each reduction in resolution (Hassani et al., 2023; Wu et al., 2021); the transition itself is an inverted bottleneck (Section C).

The three branches. The projection of Section 3 is followed by lightweight local refinement, and how that refinement is distributed across the branches is a design decision rather than an implementation detail. The signal branch carries a depthwise 3×3 convolution, a SiLU and a normalization and the gate branch the same convolution and nonlinearity without one, whereas the kernel branch carries the SiLU and the normalization but *no* spatial convolution, since the depthwise convolution on the signal branch already provides the same refinement of the effective filter. Two consequences follow. First, the kernel remains a *direct projection* of the input, so that the mixing is a bilinear interaction between two nonlinear views of a single feature map, which keeps the operator lightweight and allows the kernel to inherit the structure of the input. Second, the one depthwise convolution that remains also reduces the high-frequency energy entering the transform at the coarsest stage we measured, though not at the finer ones (Fig. A1). The exact per-branch structure is given in Section C.

Table 1: **Image classification on ImageNet-1K at 224×224 resolution.** ViF attains the best top-1 accuracy among the backbones compared, at all three scales. Within each scale, backbones are ordered by publication year and alphabetically within a year; a backbone is listed only when its authors report all three scales, so that every family appears at every scale. Type names the token mixer rather than the network, which is why ViF is listed as a convolution: a dynamic global convolution is still one.

Method	Year	Type	# P (M)	# F (G)	Top-1 (%)
Tiny					
RegNetY-4G	2020	Conv	21	4.0	80.0
Swin-T	2021	Attn	28	4.5	81.3
ConvNeXt-T	2022	Conv	29	4.5	82.1
FocalNet-T	2022	Conv	29	4.5	82.3
PVTv2-B2	2022	Attn	25	4.0	82.0
InternImage-T	2023	Conv	30	5.0	83.5
NAT-T	2023	Attn	28	4.3	83.2
SLaK-T	2023	Conv	24	5.0	82.5
InceptionNeXt-T	2024	Conv	28	4.2	82.3
MogaNet-S	2024	Conv	25	5.0	83.4
PeLK-T	2024	Conv	29	5.6	82.6
VMamba-T	2024	Mamba	30	4.9	82.6
MambaOut-T	2025	Conv	27	4.5	82.7
MambaVision-T	2025	Mamba	32	4.4	82.3
ViF-T	–	Conv	29	4.8	**83.9**
Small					
RegNetY-8G	2020	Conv	39	8.0	81.7
Swin-S	2021	Attn	50	8.7	83.0
ConvNeXt-S	2022	Conv	50	8.7	83.1
FocalNet-S	2022	Conv	50	8.7	83.5
PVTv2-B4	2022	Attn	63	10.1	83.6
InternImage-S	2023	Conv	50	8.0	84.2
NAT-S	2023	Attn	51	7.8	83.0
SLaK-S	2023	Conv	55	9.8	83.8
InceptionNeXt-S	2024	Conv	49	8.4	83.5
MogaNet-B	2024	Conv	44	9.9	84.3
PeLK-S	2024	Conv	50	10.7	83.9
VMamba-S	2024	Mamba	50	8.7	83.6
MambaOut-S	2025	Conv	48	9.0	84.1
MambaVision-S	2025	Mamba	50	7.5	83.3
ViF-S	–	Conv	51	8.8	**84.8**
Base					
RegNetY-16G	2020	Conv	84	16.0	82.9
Swin-B	2021	Attn	88	15.4	83.5
ConvNeXt-B	2022	Conv	89	15.4	83.8
FocalNet-B	2022	Conv	89	15.4	83.7
PVTv2-B5	2022	Attn	82	11.8	83.8
InternImage-B	2023	Conv	97	16.0	84.9
NAT-B	2023	Attn	90	13.7	84.3
SLaK-B	2023	Conv	95	17.1	84.0
InceptionNeXt-B	2024	Conv	87	14.9	84.0
MogaNet-L	2024	Conv	83	15.9	84.7
PeLK-B	2024	Conv	89	18.3	84.2
VMamba-B	2024	Mamba	89	15.4	83.9
MambaOut-B	2025	Conv	85	15.8	84.2
MambaVision-B	2025	Mamba	98	15.0	84.2
ViF-B	–	Conv	99	16.9	**85.4**

5 Experiments

We evaluate ViF as a general-purpose backbone on image classification, object detection and instance segmentation, and semantic segmentation, using the three variants described in Table A10.

5.1 Image Classification

Settings. We train on ImageNet-1K (Deng et al., 2009) following the protocol of prior work (Liu et al., 2021; 2024), with full details in Section B. We compare against three families, CNNs, ViTs, and ViMs, with the individual backbones and their sources listed in Table 1. A backbone enters the comparison only if its authors report all three scales, so that no family is present at one size and absent at another.

Results. As reported in Table 1, ViF-T/S/B reach $83.9/84.8/85.4\%$ top-1 accuracy. ViF-T surpasses Swin-T by 2.6 points and VMamba-T by 1.3 points, and the larger variants retain that lead over comparable backbones at matched parameter and FLOPs budgets. The static spectral filter that Theorem 1 concerns is evaluated within our own architecture rather than across papers: the same backbone retrained with the filter made input-independent reaches 83.2 at the tiny scale (Fig. 5), 0.7 below ViF-T. A DGC is therefore a competitive backbone operator across scales.

5.2 Dense Prediction

Settings. We fine-tune ImageNet-pretrained ViF-T/S/B as backbones in Mask R-CNN (He et al., 2017) on COCO (Lin et al., 2014) under the standard $1\times$ and $3\times$ schedules (Liu et al., 2021) using MMDetection (Chen et al., 2019), and in UPerNet (Xiao et al., 2018) on ADE20K (Zhou et al., 2019) using MMSegmentation (MMSegmentation Contributors, 2020), reporting box and mask AP for the former and single- and multi-scale mIoU for the latter; details are in Section B.

Results. ViF improves box and mask AP over comparable backbones at similar computational cost under both schedules (Table 2), the sole exception being that ViF-S ties MogaNet-B on box AP under the $3\times$ schedule while still leading it on mask AP, and it attains the best single- and multi-scale mIoU at every scale. The dynamic global receptive field of the operator therefore benefits localization and dense labeling as well as recognition. The detector budgets are close but not matched: ViF-T costs 287G against 262–271G for the baselines, six to ten percent more, and the 0.6 box AP over the strongest of them should be read against that difference.

Table 2: **Object detection and instance segmentation on COCO, and semantic segmentation on ADE20K.** ViF leads on box AP, mask AP, and mIoU at every scale and under every protocol, tying MogaNet-B on box AP at the small scale under 3×. Detection uses Mask R-CNN (He et al., 2017) with FLOPs at 800 × 1280; segmentation uses UPerNet (Xiao et al., 2018) for 160K iterations with FLOPs at 512 × 2048. On detection, SS and MS refer to the scale augmentation of the 1× and 3× schedules; on segmentation, to single- and multi-scale testing. Entries the source papers do not report are marked "–"; the construction of the baseline set and the sources of its numbers are given in Section E.

Method	Detection						Segmentation			
	# F (G)	# P (M)	1× SS		3× MS		# F (G)	# P (M)	SS	MS
			AP^b ↑	AP^m ↑	AP^b ↑	AP^m ↑			mIoU↑	mIoU↑
Swin-T	267	48	42.7	39.3	46.0	41.6	945	60	44.5	45.8
ConvNeXt-T	262	48	44.2	40.1	46.2	41.7	939	60	46.0	46.7
FocalNet-T	268	49	46.1	41.5	48.0	42.9	949	61	46.8	47.8
InternImage-T	270	49	47.2	42.5	49.1	43.7	944	59	47.9	48.1
MogaNet-S	272	45	46.7	42.2	48.5	43.1	–	–	–	–
NAT-T	–	–	–	–	–	–	934	58	47.1	48.4
VMamba-T	271	50	47.3	42.7	48.8	43.7	949	62	48.0	48.8
MambaOut-T	262	43	45.1	41.0	–	–	938	54	47.4	48.6
ViF-T	287	49	**47.9**	**43.1**	**49.2**	**43.9**	963	60	**49.2**	**49.8**
Swin-S	354	69	44.8	40.9	48.2	43.2	1038	81	47.6	49.5
ConvNeXt-S	348	70	45.4	41.8	47.9	42.9	1027	82	48.7	49.6
FocalNet-S	365	72	48.3	43.1	49.3	43.8	1044	84	49.1	50.1
InternImage-S	340	69	47.8	43.3	49.7	44.5	1017	80	50.1	50.9
MogaNet-B	373	63	47.9	43.2	50.3	44.4	–	–	–	–
NAT-S	–	–	–	–	–	–	1071	82	48.0	49.5
VMamba-S	384	70	48.7	43.7	49.9	44.2	1038	82	50.6	51.2
MambaOut-S	354	65	47.4	42.7	–	–	1032	76	49.5	50.6
ViF-S	375	71	**49.5**	**44.3**	**50.3**	**44.6**	1054	83	**50.8**	**51.4**
Swin-B	496	107	46.9	42.3	48.6	43.3	1188	121	48.1	49.7
ConvNeXt-B	486	108	47.0	42.7	48.5	43.5	1170	122	49.1	49.9
FocalNet-B	507	111	49.0	43.5	49.8	44.1	1192	126	50.5	51.4
InternImage-B	501	115	48.8	44.0	50.3	44.8	1185	128	50.8	51.3
MogaNet-L	495	102	49.4	44.1	50.5	44.5	–	–	–	–
NAT-B	–	–	–	–	–	–	1137	123	48.5	49.7
VMamba-B	485	108	49.2	43.9	–	–	1170	122	51.0	51.6
MambaOut-B	495	100	47.4	43.0	–	–	1178	112	49.6	51.0
ViF-B	536	117	**50.7**	**45.2**	**51.5**	**45.5**	1220	129	**51.2**	**52.0**

5.3 Analysis

Receptive field. Figure 2 visualizes the intrinsic receptive field of each mixer as the input-gradient support of a single layer. A static depthwise convolution reaches only its footprint ($t_{99} = 12\%$), and generating that kernel per position from the input (Li et al., 2021a) leaves the reach exactly where it was, so input dependence alone confers no globality; a unidirectional scan (Gu & Dao, 2024) carries state arbitrarily far but orders the grid causally, so the 49% of tokens after the center exert no influence on it. Only the static spectral filter, attention, and Fluxer see the whole image, and of these only the latter two weight it dynamically. The probe is taken at initialization, which is the reach the operator has by construction; measured on trained networks instead (Luo et al., 2016), ViF-T, Swin-T, and ConvNeXt-T reach t_{99} of 99.6, 100.0, and 99.6 percent and are indistinguishable at every quantile we examined. Depth and downsampling give any backbone an effectively global field, which is why the comparison must be made one layer at a time.

Component ablation. The left panel of Fig. 5 retrains ViF-T from scratch with one change at a time, holding everything outside the kernel branch fixed: stem, downsampling, position encodings, gated MLP, depths, widths, and recipe. A learned input-independent filter, the static limit of Proposition 1, reaches 83.2. Making it input-dependent *in the frequency domain*, as a mask computed pointwise from the spectrum after Huang et al. (2023), reaches 83.3; *in space* but shift-*invariant*, after Karami & Ghodsi (2024), 83.5; in space and left equivariant, 83.9. The ordering follows the distinction Propositions 2 and 4 draw, but each substitute also changes the input, capacity, and nonlinearity of its generator, so the gaps compare generators rather than isolate either axis: the frequency-pointwise one recovers 0.1 point of the 0.7-point gap, and the phase-discarding one stays 0.4 below ours. The frequency-domain variant is not a static filter in disguise, since it multiplies its mask back against the spectrum it reads.

The component removals are smaller by comparison: the two depthwise convolutions cost 0.2 and 0.3 and the gate branch 0.4, so the accuracy does not originate in the local operations feeding the

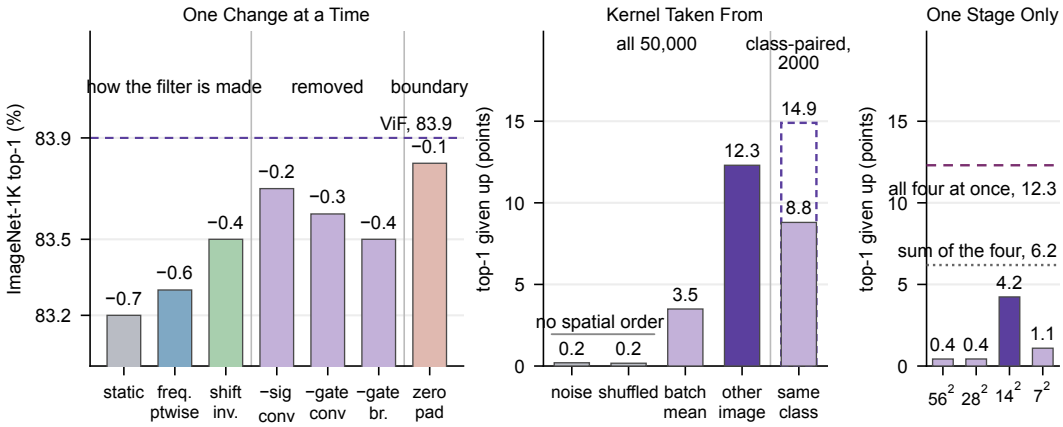

Figure 5: **Kernel-branch ablations and substitutions for ViF-T on ImageNet-1K.** A frequency-pointwise generator recovers 0.1 point of the 0.7-point gap between a static filter and ours, and a phase-discarding one 0.3 point. *Left:* the model is retrained from scratch with one change at a time, first to how the per-image filter is produced, then to which component is removed, then to whether the convolution wraps; the dashed line is the full model. *Middle* and *right:* the trained model is evaluated with the kernel branch redirected and the signal and gate branches taken from the evaluated image, first with substitutes ordered by the structure they carry, then with the derangement confined to one stage. The middle panel reports points given up against the control of each evaluation set, all 50,000 images left of the divider and a 2000-image class-paired subset right of it, where a same-class kernel (filled) sits inside a random one (dashed). Exact figures are in Table A6.

spectral step. The static variant still reaches 83.2, above ConvNeXt-T (82.1) and VMamba-T (82.6), so the surrounding architecture is competitive in its own right. The last bar addresses the boundary condition: retrained with the convolution made linear rather than circular, with both branches zero-padded to twice the grid and the result cropped back so that nothing wraps at any layer, the model reaches 83.8, so the margin is not an artifact of the wrap-around.

Is the input dependence used? The kernel does vary with the input: over 72 validation images the coefficient of variation of its magnitude is 0.39, against exactly 0 for a static filter (Fig. A2 and Section F). That holds by construction, however, and does not establish that the network relies on the variation. We therefore keep the signal and gate branches of one image and take the kernel branch from elsewhere, in every operator at once; Fig. 5 orders the substitutes by the structure each carries, and matching every image with its own kernel reproduces the reported top-1, which validates the probe. A kernel whose spatial arrangement is destroyed is nearly free: Gaussian noise matched to the kernel's own per-image, per-channel moments, and the kernel's own values spatially permuted, each cost two tenths of a point. The pass-through of Remark 1 may explain why, since the signal reaches the output whether or not the convolution it is mixed with is meaningful.

What is expensive is a kernel that is structured and wrong, and the cost rises with how wrong it is: the batch mean costs 3.5 and a kernel from a random other image 12.3, while on a class-paired subset a kernel from a same-class partner costs 8.8 against 14.9 for a random one, so the kernel carries class-level structure in addition to what is specific to the image. That 12.3 measures interference from a wrong kernel, not the value of the right one. Deranging one stage at a time localizes the effect: the two high-resolution stages give up less than half a point each and the 14×14 stage alone 4.2, and the per-stage costs sum to far less than deranging everything, so the mechanism compounds across depth. Retraining around the mechanism, in the left panel, costs a fraction of that: a design can be inexpensive to retrain around yet costly to remove, and this one is both.

Efficiency. Figure 3 times the mixing step alone, namely the attention product, the selective scan, and the transforms, with the projections every mixer pays kept outside the timed region, because at matched d_{model} those projections determine the ordering. What remains is the mechanism itself: over the last fourfold increase in tokens, FLASHATTENTION grows $15.8\times$ and the scans 3.8 to $6.4\times$, against $3.0\times$ for Fluxer. We lead FLASHATTENTION by $1.1\times$ at 224 and $54\times$ at 1024; the bidirectional scan of Zhu et al. (2024) by $1.3\times$ and $6\times$; and the four-directional SS2D of Liu et al. (2024) by $2\times$ and $4.9\times$. Only a *single* raster scan, which no ViM runs, is cheaper, and only up to about 320^2. At 224, where ours is launch-bound, a second node reversed the first two (Section E). A static spectral filter runs at about 0.7 of our latency from 512 upward, which is the price of our second forward transform. We claim no advantage in memory (Section E).

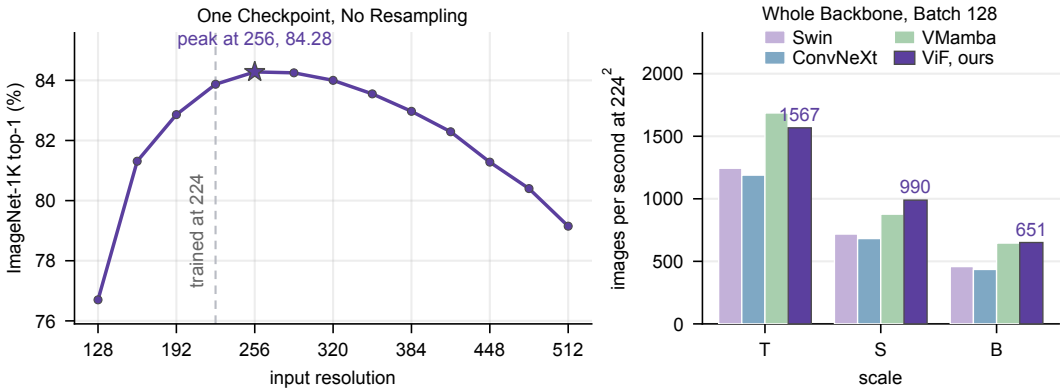

Figure 6: **Resolution transfer and whole-backbone throughput.** The 224 checkpoint runs at every resolution from 128 to 512 without a parameter being changed, and at its training resolution it outruns ConvNeXt and Swin at every scale. *Left:* ImageNet-1K top-1 for ViF-T evaluated at every multiple of 32, the total stride of the backbone, with no fine-tuning and no resampling of any weight; the maximum lies above the training resolution for the reason given in the text. *Right:* whole-backbone throughput at $3 \times 224 \times 224$ with batch 128, where every bar is our own measurement on one H100 under a single protocol, so that the four families are directly comparable. Exact figures and the timing protocol are given in Tables A8 and A9.

Matched FLOPs do not imply matched wall-clock time in either direction, so we time all four backbones ourselves under a single protocol (Table A9). ViF reaches 1567, 990, and 651 images per second at the three scales, ahead of ConvNeXt and Swin by 26 to 50 percent throughout; VMamba is 8 percent faster at the tiny scale, and ViF is 13 and 1 percent faster at the two larger ones. This is the comparison to trust over the layer-level one, since it is the only setting in which the two are matched on parameters and FLOPs, and where a scan is faster than ViF we say so. What the operator spends at this size is the projection producing its three branches rather than the transform, and the advantage of the transform is asymptotic in the token count, which is fixed here; the crossover against VMamba between the tiny and small scales comes from width and depth.

Transfer to an unseen resolution. The kernel is generated from whatever feature map arrives and transformed on the grid that map defines, so nothing in the operator is tied to the training resolution, whereas a static filter with one free parameter per grid frequency requires a resizing rule that changes the operator itself (Section E). The left panel of Fig. 6 evaluates ViF-T at every multiple of 32 from 128 to 512 with the 224 checkpoint unchanged. Accuracy rises to 84.28 at 256 from the 83.87 at which it was trained, and falls to 76.70 at 128 and 79.15 at 512; the maximum above the training resolution is the train–test discrepancy of Touvron et al. (2019) and belongs to the protocol, whereas what belongs to the operator is that the curve exists at all without a weight being resampled. A fully convolutional backbone is resolution-agnostic for the same reason, so the property separates Fluxer from static spectral filters and absolute position embeddings, but not from convolution.

6 Conclusion

We have asked in which domain an input-conditioned spectral filter should be generated, a question that a linear generator renders moot and a nonlinear one makes substantive, and answered it with Fluxer, which generates its kernel in space as a pointwise projection of the input and applies it in frequency. The construction is a convolution in the strict sense, circulant at every fixed input, yet its kernel is global, generated on the fly, and free of the $N \times N$ affinity matrix that attention pays for. Three separations place it against static spectral filters, local convolution, and frequency-pointwise masks, and the experiments bear most on the third: in the same backbone, a frequency-pointwise generator recovers 0.1 point of the gap between a static filter and Fluxer, and a phase-discarding one stays 0.4 below it.

As a backbone, ViF attains the most favorable accuracy–compute trade-off among the single-stream hierarchical CNNs, ViTs, and ViMs compared, and its checkpoint runs at other resolutions without a weight being resampled. What convolution lacked, on this evidence, was not its structure but its kernel; given a dynamic global one, it remains a competitive foundation for visual representation at the scales we examine, and whether the margin persists under larger-scale pre-training is the question we leave open.

AI USE STATEMENT

Large Language Models (LLMs) were used during this project as coding and writing assistants, including to help draft and refactor experiment and analysis scripts, revise prose, and check the consistency of the manuscript with its tables and figures. The empirical results reported in this paper are based on experiment runs conducted and reviewed by the authors, with tables and figures generated primarily from recorded experimental outputs. Citations and LLM-assisted content were reviewed by the authors. The research questions, experimental design, evaluation choices, and interpretation of the results were determined by the authors, who take responsibility for the final content of the paper.

ETHICS STATEMENT

All authors have read and adhere to the ICLR Code of Ethics. This work involves no human subjects and collects no new data. ImageNet-1K, COCO, and ADE20K are public benchmarks, each used under the terms its maintainers set for research use; like most photographic collections they contain images of people, and the only dataset images reproduced here, in Figs. A2 to A4, are shown to illustrate the model and not to identify anyone. Fluxer is a general-purpose backbone primitive, so a model built on it inherits whatever biases its training data encode, and nothing in the operator mitigates them. Detection and segmentation, two of the tasks it is evaluated on, are also building blocks of surveillance systems; the operator adds no capability specific to identifying people, but nothing in it prevents that use either. The authors declare no conflict of interest, and sources of funding will be acknowledged in the de-anonymized version.

REPRODUCIBILITY STATEMENT

Section 3 and Eq. (1) define the operator, and Listing 1 gives its forward pass. Section 4 and Section C specify the backbone, its three variants, and the three kernel-branch substitutes of Fig. 5, with the layer-by-layer configuration in Tables A5 and A10. Section B and Table A4 give the training and fine-tuning recipes for all three tasks, each on a public dataset with its standard splits. Section A gives complete proofs of the two theorems and of every proposition, with their assumptions stated. The timing protocols are stated in the captions of Tables A7 and A9 and discussed in Section E, and Section D tabulates every value that Figs. 3, 5 and 6 summarize. All source code will be released upon acceptance.

REFERENCES

Honghao Chen, Xiangxiang Chu, Yongjian Ren, Xin Zhao, and Kaiqi Huang. PeLK: Parameter-efficient large kernel ConvNets with peripheral convolution. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 6

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6, A7

Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

Lu Chi, Borui Jiang, and Yadong Mu. Fast Fourier convolution. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. RandAugment: Practical automated data augmentation with a reduced search space. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020. A7

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and Memory-Efficient exact attention with IO-Awareness. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. A11

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6, A7

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 *Proceedings of International Conference on Learning Representations (ICLR)*, 2021. 1, 2

Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent Y. F. Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. A7, A9

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024. 2, 7, A6, A11

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive Fourier neural operators: Efficient token mixers for transformers. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2022. 2, A1

Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. CMT: Convolutional neural networks meet vision transformers. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *Computational Visual Media*, 9(4):733–752, 2023. 3

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 5, 6

Ali Hatamizadeh and Jan Kautz. MambaVision: A hybrid Mamba-transformer vision backbone. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 6

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 6, 7, A7

Zhipeng Huang, Zhizheng Zhang, Cuiling Lan, Zheng-Jun Zha, Yan Lu, and Baining Guo. Adaptive frequency filters as efficient global token mixers. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 7, A2, A4, A6, A9

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 3

Mahdi Karami and Ali Ghodsi. Orchid: Flexible and Data-Dependent convolution for sequence modeling. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2, 7, A4, A9

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of International Conference on Machine Learning (ICML)*, 2020. 2, A6

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: Human Language Technologies (NAACL-HLT)*, 2022. 2

Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021a. 3, 7

Siyuan Li, Zedong Wang, Zicheng Liu, Cheng Tan, Haitao Lin, Di Wu, Zhiyuan Chen, Jiangbin Zheng, and Stan Z Li. MogaNet: Multi-order gated aggregation network. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024. 6

Zongyi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021b. 2

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014. 6, A7

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Tommi Kärkkäinen, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More ConvNets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023. 2, 6, A6

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. VMamba: Visual state space model. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2, 5, 6, 8, A10, A11, A12

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 International Conference on Computer Vision (ICCV)*, 2021. 2, 5, 6, A7, A9

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5, 6

Meng Lou and Yizhou Yu. OverLoCK: An Overview-first-Look-Closely-next ConvNet with Context-Mixing dynamic kernels. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. A12, A13

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 7

MMSegmentation Contributors. MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 6, A7

Badri N. Patro and Vijay Srinivas Agneeswaran. Scattering vision transformer: Spectral mixing matters. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3

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 *Proceedings of International Conference on Machine Learning (ICML)*, 2023. 3

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global Filter Networks for image classification. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, A6, A9, A11

Yuki Tatsunami and Masato Taki. FFT-based dynamic token mixer for vision. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2024. 3

Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 9

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1, 2, A6, A11

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. PVT v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 6

Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, and Yu Qiao. InternImage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 6

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CvT: Introducing convolutions to vision transformers. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 5

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 6, 7, A7

Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. CondConv: Conditionally parameterized convolutions for efficient inference. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3

Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3, 6

Ting Yao, Yingwei Pan, Yehao Li, Chong-Wah Ngo, and Tao Mei. Wave-ViT: Unifying wavelet and transformers for visual representation learning. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2022. 3

Weihao Yu and Xinchao Wang. MambaOut: Do we really need Mamba for vision? In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 6

Weihao Yu, Pan Zhou, Shuicheng Yan, and Xinchao Wang. InceptionNeXt: When Inception meets ConvNeXt. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 6

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 *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. A7

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018. A7

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2020. A7

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision (IJCV)*, 127:302–321, 2019. 6, A7

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 *Proceedings of International Conference on Machine Learning (ICML)*, 2024. 2, 8, A10, A11

A THEORETICAL ANALYSIS

We prove the two theorems of Section 3 and collect the propositions it cites. Throughout, $\mathcal{F}$ denotes the orthonormal two-dimensional fast Fourier transform (FFT) of Section 3 and $\circledast$ the unitarily normalized circular convolution it induces; all operators act per channel, $N = HW$, and $\|T\|_B = \sup_{\|\mathbf{u}\| \leqslant 1} \|T(\mathbf{u})\|$. The theory uses the full unitary two-dimensional DFT on the $H \times W$ grid: token indices range over $\mathbb{Z}_H \times \mathbb{Z}_W$, a difference $i - j$ is taken modulo (H, W), a matrix is circulant when its entries depend on $i - j$ alone, and $\langle \xi, v \rangle = \xi_1 v_1 / H + \xi_2 v_2 / W$. The implementation stores the Hermitian half-spectrum returned by `rfft2`, which represents the same real operator.

Proof of Theorem 1. T_1 is positively homogeneous of degree 2, whereas every $T \in \mathcal{S}$ is linear and hence homogeneous of degree 1. If $T = T_1$, then $\alpha T(\mathbf{u}) = T(\alpha \mathbf{u}) = \alpha^2 T_1(\mathbf{u}) = \alpha^2 T(\mathbf{u})$ for all α, which forces $T \equiv 0$; but $\widehat{T_1(\mathbf{u})}(\xi) = \hat{\mathbf{u}}(\xi)^2 \not\equiv 0$ for $\mathbf{u} \neq 0$, so $T_1 \notin \mathcal{S}$. For the quantitative statement, fix $T \in \mathcal{S}$ and $\mathbf{u}$ in the unit ball, and write $\mathbf{a} = T(\mathbf{u})$ and $\mathbf{b} = T_1(\mathbf{u})$. Linearity gives $T(-\mathbf{u}) = -\mathbf{a}$, whereas $T_1(-\mathbf{u}) = \mathbf{b}$, so the errors at $\mathbf{u}$ and $-\mathbf{u}$ are $\|\mathbf{a} - \mathbf{b}\|$ and $\|\mathbf{a} + \mathbf{b}\|$, and the parallelogram identity $\|\mathbf{a} - \mathbf{b}\|^2 + \|\mathbf{a} + \mathbf{b}\|^2 = 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2 \geqslant 2\|\mathbf{b}\|^2$ gives $\max(\|\mathbf{a} - \mathbf{b}\|, \|\mathbf{a} + \mathbf{b}\|) \geqslant \|T_1(\mathbf{u})\|$. Both $\pm\mathbf{u}$ lie in the unit ball, so $\|T - T_1\|_B \geqslant \|T_1(\mathbf{u})\|$ for every unit-ball $\mathbf{u}$, whence $\|T - T_1\|_B \geqslant \|T_1\|_B$. Equality holds at $T = 0$, which gives Eq. (2). □

Remark 3 (Depth and scope of Theorem 1)**.** *Depth.* Stacking static filters with nonlinearities raises the degree but still cannot represent T_1, since such a stack is translation-equivariant at every depth, whereas a shift of the input displaces T_1 by twice that shift (Proposition 4). This is an obstruction to exact representation rather than an approximation bound, and we do not claim that Eq. (2) survives depth. *Scope.* The quantitative half of the proof uses $T(-\mathbf{u}) = -T(\mathbf{u})$, which linearity supplies and positive homogeneity alone does not. AFNO (Guibas et al., 2022) applies its ReLU to the real and imaginary parts separately and carries biases and a soft shrinkage, which leaves it neither odd nor homogeneous, so the theorem does not cover it.

Proof of Theorem 2. One layer suffices. Take the kernel branch to be spatially constant, $\mathbf{c} = \gamma \mathbf{1}$, which a zero projection with a per-channel bias produces, so that $\mathcal{F}(\mathbf{c}) = \gamma N^{1/2} \delta_0$ is supported at the zero frequency. Then $\mathcal{F}(\mathbf{y}) = \mathcal{F}(\mathbf{u}) \odot \mathbf{H}$ is supported at $\xi = 0$ with $\hat{\mathbf{y}}(0) = \gamma N^{1/2} \hat{\mathbf{u}}(0)$, and since $\hat{\mathbf{u}}(0) = N^{1/2} \mathrm{mean}(\mathbf{u})$, inverting gives the constant map $\mathbf{y} = \gamma N^{1/2} \mathrm{mean}(\mathbf{u}) \mathbf{1}$, which is M up to a scalar absorbed by $\mathbf{W}_o$.

Local stacks fail. A pointwise operation has dependency radius 0, a width-r convolution has radius $(r-1)/2$, and radii add under composition, so every $G \in \mathcal{A}_{\mathrm{loc}}(r, L)$ has the following property: if $\mathbf{u}$ and $\mathbf{u}'$ agree on the ball $B(i, R)$ with $R = L(r-1)/2$, then $G(\mathbf{u})_i = G(\mathbf{u}')_i$. Let $\varepsilon = N^{-1/2}$, let Ω be a half-grid slab of width $S/2$, and set $\mathbf{u}^{(1)} = \varepsilon \mathbf{1}$ and $\mathbf{u}^{(2)} = \varepsilon \mathbf{1}_\Omega$, which agree on Ω and satisfy $\|\mathbf{u}^{(1)}\| = 1$ and $\|\mathbf{u}^{(2)}\| \leqslant 1$. For every i in the deep set $D = \{i \in \Omega : B(i, R) \subseteq \Omega\}$ we have $G(\mathbf{u}^{(1)})_i = G(\mathbf{u}^{(2)})_i$, whereas $M(\mathbf{u}^{(1)})_i = \varepsilon$ and $M(\mathbf{u}^{(2)})_i = \varepsilon/2$; if both errors at i were below $\varepsilon/4$, the triangle inequality would contradict this gap of $\varepsilon/2$, so each $i \in D$ incurs an error of at least $\varepsilon/4$ on one of the two inputs. With $R \leqslant S/8$ the deep set has $|D| \geqslant S(S/2 - 2R) \geqslant S^2/4 = N/4$ positions, so by the pigeonhole principle one of the two inputs incurs an error of at least $\varepsilon/4$ at no fewer than $N/8$ positions, giving $\|M - G\|_B^2 \geqslant (N/8)(\varepsilon/4)^2 = 1/128$ and $\|M - G\|_B \geqslant 128^{-1/2} > 1/12$. An error below $1/12$ therefore forces $L(r-1)/2 > S/8$. Note that the radius R is a spatial quantity compared with the side length $S = \sqrt{N}$, whereas the counting and the normalization use the token count N. □

The two theorems isolate two distinct ingredients. Theorem 1 is a symmetry obstruction: static filters are translation-equivariant and linear, whereas the dynamic operator is neither. Theorem 2 is a light-cone obstruction, independent of the nonlinearity, normalization, gating, or channel count of the competing local class. We now record the limiting cases and the remaining structural properties.

Proposition 1 (Static spectral filters as a special case)**.** *If the kernel branch is input-independent, that is, $\mathbf{c}$ is a fixed feature map, then Fluxer reduces to a static global filter of the kind used by GFNet and the Fourier Neural Operator (FNO).*

Proof. With $\mathbf{c}$ independent of the input, $\mathbf{H} = \mathcal{F}(\mathbf{c})$ is a constant per-frequency tensor, so $\mathbf{u} \mapsto \mathcal{F}^{-1}(\mathcal{F}(\mathbf{u}) \odot \mathbf{H})$ is a linear map whose spectral weights are shared across all inputs, which is exactly a static spectral filter. □

Proposition 2 (A spatially generated filter is not a frequency-pointwise mask)**.** *Let $N \geqslant 3$, and let $\mathcal{M}_{\text{freq}}$ be the class of operators $\mathbf{u} \mapsto \mathcal{F}^{-1}\big(m(\hat{\mathbf{u}}) \odot \hat{\mathbf{u}}\big)$ whose mask is pointwise in frequency, that is, $m(\hat{\mathbf{u}})(\xi) = g_\xi\big(\hat{\mathbf{u}}(\xi)\big)$ for functions g_ξ that depend on no other frequency. Let $T(\mathbf{u}) = \mathcal{F}^{-1}\big(\hat{\mathbf{c}} \odot \hat{\mathbf{u}}\big)$ with $\mathbf{c} = \varphi(\mathbf{u})$ for a spatially pointwise φ whose power series has a nonzero quadratic coefficient. Then $T \notin \mathcal{M}_{\text{freq}}$.*

Proof. Take $\varphi(t) = t^2$; the argument is unchanged for any φ whose expansion has a nonzero quadratic coefficient. Since $\mathcal{F}$ is unitary, $\hat{\mathbf{c}} = \hat{\mathbf{u}} \circledast \hat{\mathbf{u}}$, that is,

$$\hat{\mathbf{c}}(\xi) = N^{-1/2} \textstyle\sum_\eta \hat{\mathbf{u}}(\eta)\, \hat{\mathbf{u}}(\xi - \eta). \tag{A1}$$

If T belonged to $\mathcal{M}_{\text{freq}}$, then $\hat{\mathbf{c}}(\xi)$ would be a function of $\hat{\mathbf{u}}(\xi)$ alone, so holding $\hat{\mathbf{u}}(\xi)$ fixed and varying $\hat{\mathbf{u}}$ elsewhere would leave it unchanged. Fix ξ and pick $\eta \notin \{0, \xi\}$, which exists because $N \geqslant 3$, and choose $\hat{\mathbf{u}}$ with $\hat{\mathbf{u}}(\xi - \eta) \neq 0$. Perturbing $\hat{\mathbf{u}}(\eta)$ alone leaves $\hat{\mathbf{u}}(\xi)$ fixed and changes the term $\hat{\mathbf{u}}(\eta)\hat{\mathbf{u}}(\xi - \eta)$ of Eq. (A1), hence changes $\hat{\mathbf{c}}(\xi)$; for a real signal the same holds when the conjugate pair $\hat{\mathbf{u}}(\pm\eta)$ is perturbed together. No such g_ξ exists. □

The proposition concerns a simplified core in which the kernel is a pointwise function of the signal it convolves. The implemented layer computes both branches from a common input, $\mathbf{u} = f(\mathbf{X})$ and $\mathbf{c} = g(\mathbf{X})$, and f contains a depthwise convolution, so $\mathbf{c}$ is not a function of $\mathbf{u}$ and the proposition does not transfer to the layer as stated. What carries over is the mechanism: g, a 1×1 channel projection followed by a SiLU and a per-token normalization, acts pointwise in space, so for generic weights each coefficient of $\hat{\mathbf{c}}$ depends on more than one frequency of $\hat{\mathbf{X}}$, whereas a frequency-pointwise mask reads one. Placing the implemented layer outside a frequency-pointwise class defined over $\mathbf{X}$ would need conditions on the vector-valued generator and the normalization that we do not establish, and we claim the separation for the core only.

In the core, spatial generation is therefore not contained in the frequency-pointwise class. We prove containment in one direction only, which is the direction the paper uses, and make no claim about the reverse; the two classes should not be described as mutually non-nested on the strength of this proposition alone. The mechanism is visible in Eq. (A1): a mask formed after the transform reads one frequency at a time, whereas a kernel formed before it carries every frequency of the input into each coefficient of the filter. Were φ affine, the sum would collapse and the classes would coincide, which is why a linear generator cannot tell them apart. That direction of the statement is due to Huang et al. (2023), who use the commutation of a linear layer with the transform to argue that a frequency-domain linear stack is not input-adaptive; Proposition 2 is its converse reading. Once the generator is nonlinear the two constructions come apart, and which one a network occupies is a choice rather than an implementation detail.

Proposition 3 (Circulant mixing and its attention counterpart)**.** *A token-mixing layer whose mixing matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is shift-invariant (circulant), which is the shift-equivariant limit of a linearized self-attention layer, is a circular convolution, and Fluxer realizes such a layer with an input-dependent kernel.*

Proof. A circulant $\mathbf{A}$ with first row $\mathbf{a}$ satisfies $(\mathbf{A}\mathbf{v})_i = \sum_j \mathbf{a}_{i-j}\, \mathbf{v}_j = N^{1/2}(\mathbf{a} \circledast \mathbf{v})_i$ under the normalization of Section 3. The Fluxer mixing term $\mathbf{u} \circledast \mathbf{c}$ of Eq. (1) is therefore the action of a circulant mixing matrix whose defining row is the input-generated kernel $\mathbf{c}$; unlike a static filter, this row is regenerated per input, giving the input-dependent, all-to-all interaction of self-attention without materializing $\mathbf{A}$. □

Corollary 1 (Softmax attention)**.** *The equivalence carries over to softmax attention under shift invariance of the score function. If the query–key interaction depends only on the relative position of two tokens, $s_{ij} = \phi(i - j)$, then the attention matrix $\mathbf{A} = \mathrm{softmax}(s)$ is circulant, and by Proposition 3 the attention layer computes a circular convolution, now with the non-negative, row-normalized kernel* $\mathrm{softmax}(\phi)$.

Proof. Shift-invariant scores give $s_{ij} = \phi(i-j)$; row-wise softmax preserves this dependence on $i-j$, so $\mathbf{A}$ is circulant with first row $\mathbf{a} = \mathrm{softmax}(\phi)$, and Proposition 3 applies. □

This special case is the point of departure for Fluxer: shift-invariant softmax attention (Corollary 1) is a circular convolution whose kernel is *fixed by position* and *simplex-normalized*. Fluxer relaxes both constraints, generating its kernel $\mathbf{c}$ from the input and leaving it unnormalized, so it can be read as a shift-invariant self-attention whose kernel is made data-dependent. In this precise sense Fluxer recovers the global, position-aware mixing that a relative-position score gives attention while restoring the input dependence such a score removes. The correspondence stops there: a general attention layer weights each query differently, so its mixing matrix need not be circulant, whereas Fluxer applies one kernel at every location and is circulant for every input. Fluxer occupies the translation-structured corner of dynamic all-to-all mixing, not the whole of it.

Together, the propositions place Fluxer in the design space: it contains static spectral filters, it is not contained in the frequency-pointwise masks, and it meets attention at the translation-structured corner of the dynamic all-to-all family.

Proposition 4 (Convolutional structure and dynamic equivariance)**.** *For any fixed kernel* $\mathbf{c}$*, the signal map* $\mathbf{u} \mapsto \mathbf{u} \circledast \mathbf{c}$ *is a circular convolution and is therefore translation-equivariant:* $(\tau_v \mathbf{u}) \circledast \mathbf{c} = \tau_v(\mathbf{u} \circledast \mathbf{c})$ *for every cyclic shift* τ_v*. In its signal argument, Fluxer thus applies a genuine convolution. Because the kernel is generated from the* same *input, however, a shift of the input propagates to both branches: with* $\mathbf{u} = f(\mathbf{X})$ *and* $\mathbf{c} = g(\mathbf{X})$ *for translation-equivariant* f *and* g *(a depthwise convolution, a pointwise nonlinearity, and a per-token normalization, exactly equivariant under circular boundary conditions),* $\mathbf{X} \mapsto \tau_v \mathbf{X}$ *implies* $\mathbf{y} \mapsto \tau_{2v}\mathbf{y}$*, so the operator as a whole is* not *translation-equivariant. Exact equivariance is recovered only in the static limit of Proposition 1, where* $\mathbf{c}$ *is input-independent.*

Proof. By the Fourier shift theorem, $\mathcal{F}(\tau_v \mathbf{u}) = \mathcal{F}(\mathbf{u}) \odot e^{-2\pi i \langle \cdot, v \rangle}$, so $(\tau_v \mathbf{u}) \circledast \mathbf{c}$ has spectrum $\mathcal{F}(\mathbf{u}) \odot \mathcal{F}(\mathbf{c}) \odot e^{-2\pi i \langle \cdot, v \rangle} = \mathcal{F}\big(\tau_v(\mathbf{u} \circledast \mathbf{c})\big)$; when both branches shift, the two phase factors multiply and the output shifts by $2v$. □

The statement concerns Eq. (1). The implemented layer adds the static pass-through of Remark 1, whose own output shifts by v, so what it produces under a shift is the superposition $\tau_{2v}(\mathbf{u} \circledast \mathbf{c}) + \tau_v(\mathbf{s} \odot \mathbf{u})$ of two differently shifted terms, which is not a shift of $\mathbf{y}$ by any amount. Fluxer therefore trades exact translation equivariance for input adaptivity while retaining circulant, that is, convolutional, structure at every fixed input, which is the precise sense in which it is a *dynamic* convolution rather than a static one.

Proposition 4 assumes exact equivariance of the two generating maps, which holds under circular boundary conditions but not under the zero padding that every convolution in the network in fact uses. Table A2 measures the three displacements on the trained model under both. Under circular padding the proposition holds exactly: across 32768 measurements per condition, the cyclic cross-correlation peak against the unshifted feature map, taken per image and per channel with the spatial mean removed, places the signal and kernel branches at v and the convolution term at $2v$ in every case, at all four stages and at both shifts we applied. The sum behaves as Remark 1 requires, as a superposition of the two rather than as either displacement, and which displacement its peak follows changes with depth: the convolution wins the peak in 69% of measurements at 56×56 and in 3% at 7×7. That ordering is not the energy ordering. On the same images the pass-through carries 7.6% of the root-mean-square response at the first operator and 28% at the first operator of each of the two coarse stages, so where the peak follows the pass-through the convolution term is still more than twice its size. A convolution whose kernel spans the map is smooth, and a broad correlation peak can lose an argmax to a sharp one carrying less energy, but we have not established that this is the reason and do not join the two measurements further.

Under zero padding the same probe fails its own control at the two high-resolution stages, where the signal branch itself does not read v, and there the convolution term reads v rather than $2v$. We report the failure rather than only the column that works. A circular convolution spreads any stationary component of its argument over the whole map, so a border that a cyclic roll does not move, namely the zero padding and the seam such a roll opens in otherwise continuous content, enters the term as products of a moving branch with a stationary one, and those move by v. The kernel branch, which

carries no spatial convolution and is therefore a pointwise function of the layer input, is the least affected of the three, consistent with that reading. What we claim is the circular column, which is the hypothesis the proposition states; the zero-padded column is reported because it is the network as it runs.

The property is a design position rather than an oversight, and it is the axis on which Fluxer and Orchid differ. Karami & Ghodsi (2024) devote a section to preserving shift equivariance in a data-dependent convolution, observe that it suffices for the kernel to satisfy $h(\tau_v \mathbf{x}) = h(\mathbf{x})$, and construct such a kernel by suppressing the phase, in the released default by squaring the modulus of the transform, or by cross-correlation. That is the same statement as Proposition 4 used in the opposite direction: they make the kernel invariant so that the operator is equivariant, and we let the kernel be equivariant so that it can encode position, at the cost of an operator that moves at $2v$. A magnitude discards the phase, and the phase is where position resides, so a kernel built that way can express what an image contains but not where; ours expresses both. Which is the better trade is an empirical question that Proposition 4 does not settle. In our setting Fig. 5 answers it: built their way the same backbone reaches 83.5, and built ours 83.9, so for ImageNet-1K classification the position the phase carries is worth more than the equivariance it costs. We do not read that ordering as general, since on a task scored on translation consistency itself the invariant construction may well be the right one, and nowhere do we claim that Fluxer inherits the equivariance of the convolution it is built from.

Proposition 4 says the layer is not translation-equivariant and Table A2 confirms the mechanism, but neither says whether the network pays for it. Table A1 asks how often the predicted class survives a translation of the input. ViF-T keeps its prediction on 97.7% of images under a one-pixel cyclic roll and 96.4% under sixteen, against 96.0 and 94.4 for Swin-T and 97.2 and 94.6 for ConvNeXt-T; under a shift of the crop window the figures are 98.1 and 97.0, against 96.3 and 94.3, and 97.5 and 95.7. The operator whose layer moves at twice the input shift is therefore the most translation-consistent of the three, at every shift and under both kinds of translation. We do not read that as a benefit of the doubled displacement: ViF-T is also the most accurate of the three on this data (83.9 against 81.4 and 81.9), and a more accurate model is more likely to keep its answer under a perturbation, so the comparison is confounded. What it does establish is that the geometry of Proposition 4 is not paid for where it would matter.

A second consequence of the transform is that the convolution is circular, so content at one edge is mixed into the other. Replacing it with a linear convolution measures what that is worth, and the answer depends on how the kernel is read. Taken as displacements $[0, N)$ and zero-padded to twice the size, the linear convolution costs 16.2 points, but that removes more than the wrap-around: an uncentered whole-image kernel reaches only backwards, so about half the displacement pairs disappear with it. Read as displacements $[-N/2, N/2)$, the same kernel over the same support with only the genuinely wrapped terms removed, the cost is 2.0 points, 81.9 against 83.9. The circular boundary is not ours alone; Huang et al. (2023) pad circularly for the same reason, that the transform makes the convolution cyclic. Both figures are surgeries at inference on a network trained circularly, so they measure the cost of withdrawing a coupling the trained network had come to rely on rather than the value of the coupling itself. Retraining from scratch separates the two: with the convolution made linear throughout, signal and kernel zero-padded to twice the grid before each transform and the result cropped back so that nothing wraps at any layer, the model reaches 83.8 against 83.9 (Fig. 5). The operator therefore does not need the wrap-around; a network trained with it will use it, for about two points, and the remaining 14.2 measures how much the operator depends on reaching in both directions rather than what the boundary condition is worth.

Observation 1 (Spectral smoothness of the signal branch)**.** The depthwise convolution K_1 on the signal branch attenuates the aliasing-prone high-frequency band of the spectral step. In the linear idealization, where the branch is $\mathbf{u} = K_1 * \tilde{\mathbf{u}}$ with no intervening nonlinearity, the output satisfies per channel $\hat{\mathbf{y}}(\omega) = \widehat{K_1}(\omega)\, \hat{\tilde{\mathbf{u}}}(\omega)\, \hat{\mathbf{c}}(\omega)$, so for a smoothing K_1, whose response $|\widehat{K_1}|$ decreases with $|\omega|$, the energy above any cutoff is suppressed by $\sup_{|\omega| \geqslant \omega_0} |\widehat{K_1}(\omega)|^2$ relative to a broadband (identity) branch. With the SiLU that the branch applies after K_1, this factorization is only approximate: a pointwise nonlinearity moves energy across frequencies, so K_1 attenuates the high band rather than band-limiting it, and the residual is governed by the higher Taylor coefficients of the nonlinearity.

Figure A1 measures the effect on the trained ViF-T, bypassing K_1 at the probed layer alone. The attenuation is strongly resolution-dependent and monotone in depth: at the two high-resolution stages, removing K_1 leaves the fraction of mixed-output energy above 0.25 cycles/sample essentially un-

Table A1: **Translation consistency of predictions under cyclic rolls and crop-window shifts.** The operator whose layer moves at twice the input shift keeps its prediction under translation more often than either backbone that is equivariant by construction. Each entry is the percentage of the 50,000 validation images whose predicted class is unchanged when the input is translated by the stated number of pixels, for trained ViF-T against the torchvision ImageNet-1K weights of Swin-T and ConvNeXt-T. The upper block applies a cyclic roll, the operation Proposition 4 concerns; the lower block shifts the crop window, which is what a translation of the scene looks like and lets new content enter. The untranslated top-1 accuracy is given because a more accurate model is more likely to keep its answer under a perturbation, which confounds the comparison; we read the table as evidence that the geometry of Proposition 4 is not paid for at the level of predictions, not as evidence that it helps.

	Top-1	shift, in pixels				
Backbone	untranslated	1	2	4	8	16
cyclic roll						
ViF-T	83.87	97.7	97.1	96.5	96.3	96.4
Swin-T	81.35	96.0	95.1	94.6	94.1	94.4
ConvNeXt-T	81.89	97.2	96.5	96.2	95.3	94.6
shift of the crop window						
ViF-T	–	98.1	97.5	97.3	96.8	97.0
Swin-T	–	96.3	95.4	95.0	94.3	94.3
ConvNeXt-T	–	97.5	96.8	96.5	95.9	95.7

Table A2: **Displacement of the branches and of the mixed term under a cyclic roll of the input.** Under the circular boundary condition that Proposition 4 assumes, every one of the 32768 measurements per condition agrees with it: the two branches move by v and the convolution term by $2v$. Each entry is the percentage of measurements whose cyclic cross-correlation peak sits at the stated displacement, for an input rolled by $v = 32$ pixels, on trained ViF-T at the first operator of each stage; the peak is taken per image and per channel with the spatial mean removed, over 512 validation images and 64 channels. The upper block sets all convolutions to circular padding, the hypothesis of the proposition. The lower block is the network as it runs, with zero padding, where the probe fails its own control at the two high-resolution stages and nothing can be read from the convolution term there. The last row is the rate a uniformly drawn displacement would achieve.

Padding	Term	56^2	28^2	14^2	7^2
circular, the boundary condition of Proposition 4					
	signal $\mathbf{u}$, at v	100	100	100	100
	kernel $\mathbf{c}$, at v	100	100	100	100
	$\mathbf{u} \circledast \mathbf{c}$, at $2v$	100	100	100	100
	the sum, at $2v$	69	55	15	3
	the sum, at v	29	41	76	97
zeros, the network as it runs					
	signal $\mathbf{u}$, at v	8	20	54	85
	kernel $\mathbf{c}$, at v	100	95	76	89
	$\mathbf{u} \circledast \mathbf{c}$, at $2v$	5	11	21	59
	$\mathbf{u} \circledast \mathbf{c}$, at v	78	54	35	22
	a uniformly drawn displacement	1.8	3.6	7.1	14.3

changed ($0.7\times$ and $0.9\times$), whereas at the third it inflates that fraction by $6.4\times$. The smoothing therefore operates at the coarsest grid we measured, 14×14, rather than uniformly across the network, which is where the measured stages leave the least headroom below Nyquist. The measured ratios are finite rather than unbounded, as the nonlinear case predicts; the correspondence with the accuracy cost of removing K_1 (Fig. 5) is correlational.

Proposition 5 (Lipschitz stability)**.** *On the domain where both branch features are norm-bounded, $\|\mathbf{u}\|, \|\mathbf{c}\| \leqslant B$, as the layer normalization on each branch enforces, the bilinear map $(\mathbf{u}, \mathbf{c}) \mapsto \mathbf{y} = \mathcal{F}^{-1}(\mathcal{F}\mathbf{u} \odot \mathcal{F}\mathbf{c})$ is Lipschitz, and for any two inputs*

$$\|\mathbf{y} - \mathbf{y}'\| \;\leqslant\; \|\mathbf{u}\|\,\|\mathbf{c} - \mathbf{c}'\| + \|\mathbf{c}'\|\,\|\mathbf{u} - \mathbf{u}'\| \;\leqslant\; B\big(\|\mathbf{u} - \mathbf{u}'\| + \|\mathbf{c} - \mathbf{c}'\|\big). \tag{A2}$$

Proof. Because $\mathcal{F}$ is unitary, $\|\mathbf{p} \circledast \mathbf{q}\| = \|\hat{\mathbf{p}} \odot \hat{\mathbf{q}}\|_2 \leqslant \|\hat{\mathbf{p}}\|_\infty \|\hat{\mathbf{q}}\|_2 \leqslant \|\hat{\mathbf{p}}\|_2 \|\hat{\mathbf{q}}\|_2 = \|\mathbf{p}\|\,\|\mathbf{q}\|$, using $\|\hat{\mathbf{p}}\|_\infty \leqslant \|\hat{\mathbf{p}}\|_2$ and Parseval. Applying this to $\mathbf{y} - \mathbf{y}' = \mathcal{F}^{-1}\big(\hat{\mathbf{u}} \odot (\hat{\mathbf{c}} - \hat{\mathbf{c}}') + (\hat{\mathbf{u}} - \hat{\mathbf{u}}') \odot \hat{\mathbf{c}}'\big)$ gives the stated bound. □

Because the branch maps f and g are themselves Lipschitz on bounded sets, being a depthwise convolution, the SiLU, whose Lipschitz constant is about 1.1, and a layer normalization (Lipschitz away from zero variance), the operator is Lipschitz in the input feature map, so bounded input perturbations produce bounded output perturbations. The constant is not resolution-free, and the statement above understates that. Layer normalization is applied per token, so what it fixes is the

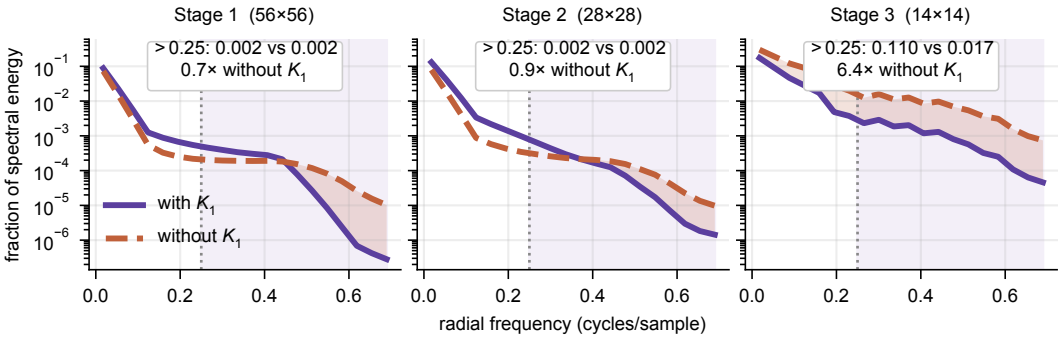

Figure A1: **Radial power spectrum of the spectral-mixing output with and without the signal-branch depthwise convolution K_1.** K_1 attenuates high-frequency energy at the coarsest stage measured and not at the fine stages. Spectra are measured on trained ViF-T over 16 ImageNet images at the first block of each of the first three stages, with K_1 applied (solid) and bypassed at that layer alone (dashed); the shaded band marks the region above 0.25 cycles/sample, and the annotations give the fraction of energy it holds. At 56×56 and 28×28 the two curves carry the same high-frequency mass, whereas at 14×14 bypassing K_1 inflates it by $6.4\times$.

norm of each token rather than of the map: with N tokens of D channels each normalized to $O(\sqrt{D})$, the global norm is $\|\mathbf{u}\| = \Theta(\sqrt{ND})$, and the admissible B grows with the token count. Proposition 5 therefore says that the operator is Lipschitz on any fixed grid, with a constant that grows with the grid, and not that it is stable uniformly across resolutions. Whether that is related to the accuracy falling away above 256 in Fig. 6 we have not established, and we do not claim it is.

Proposition 6 (Complexity)**.** *For an input with $N = HW$ tokens, C input channels, and $D = eC$ branch channels, the Fluxer operator runs in $O(NCD + ND \log N)$ time and $O(ND)$ memory. Neither term is $\Omega(N^2)$.*

Proof. The forward pass comprises (i) the dense projection $C \mapsto 3D$ that produces the three branches and the projection $D \mapsto C$ that closes the operator, costing $O(NCD)$ together; (ii) the depthwise refinements, $O(NDk^2) = O(ND)$ for a fixed $k \times k$ kernel; (iii) two forward and one inverse two-dimensional real FFTs, each $O(N \log N)$ per channel and hence $O(ND \log N)$ in total; and (iv) the element-wise spectral product and the gating, both $O(ND)$. Summing gives $O(NCD + ND \log N)$, and no intermediate tensor exceeds the $O(ND)$ feature map, so memory is $O(ND)$. □

The first term is the channel projections, which self-attention and linear attention carry in the same form, so it cancels in any comparison between mixers; what separates them is the second term and the absence of any N^2. Table A3 places this alongside other token mixers. Fluxer matches the near-linear cost and full-image receptive field of static spectral filters while adding the input dependence that otherwise only attention and state-space models provide, and unlike attention it avoids both the quadratic time and the $N \times N$ affinity matrix.

Table A3: **Asymptotic cost and properties of token mixers.** Costs are given for N tokens, C input channels, and D branch channels, with k the local kernel size. Two cost columns are given because a single column would force different rows into different accounting conventions: the first counts the mixing operation alone, and the second adds the channel projections that every row carries. Memory is quoted for the implementation each operator is run in, so self-attention is listed at the $O(ND)$ of a fused kernel; written out, it materializes the $N \times N$ scores and costs $O(N^2+ND)$, and Table A7 reports both. Several rows are global, input-dependent, and near-linear at once, so the column pattern is not what distinguishes Fluxer; what does is where the filter is generated, which no asymptotic column records.

Operator	Mixing	With projections	Memory	Global	Dynamic
Self-attention (Vaswani et al., 2017)	$O(N^2D)$	$O(NCD+N^2D)$	$O(ND)$	✓	✓
Linear attention (Katharopoulos et al., 2020)	$O(ND^2)$	$O(NCD+ND^2)$	$O(ND)$	✓	✓
State-space model (Gu & Dao, 2024)	$O(ND^2)$	$O(NCD+ND^2)$	$O(ND)$	scan	✓
Large-kernel conv. (Liu et al., 2023)	$O(NDk^2)$	$O(NCD+NDk^2)$	$O(ND)$	local	×
Static spectral (Rao et al., 2021)	$O(ND \log N)$	$O(NCD+ND \log N)$	$O(ND)$	✓	×
AFF (Huang et al., 2023)	$O(ND \log N)$	$O(NCD+ND \log N)$	$O(ND)$	✓	✓
Fluxer (ours)	$O(ND \log N)$	$O(NCD+ND \log N)$	$O(ND)$	✓	✓

B Dataset Details

ImageNet-1K for image classification. ImageNet-1K (Deng et al., 2009) comprises 1.28M training and 50K validation images over 1000 categories. We train under the same protocol as Swin Transformer (Liu et al., 2021): the AdamW optimizer, 300 epochs from scratch, a cosine learning-rate schedule with a 20-epoch linear warm-up, and a weight decay of 0.05. The peak learning rate is 10^{-3} for the total batch of 1024, warmed up linearly from 10^{-6} and decayed to 10^{-5}; gradients are clipped at norm 5, labels are smoothed by 0.1, inputs are resized with bicubic interpolation, and the random seed is 0. The stochastic-depth rate is $0.2/0.3/0.5$ for ViF-T/S/B. We adopt MESA (Du et al., 2022) to mitigate overfitting, at $1.0/1.5/2.0$ for the three variants; it is worth 0.25, 0.24, and 0.06 points, and Table A11 ablates it. Training also uses RandAugment (Cubuk et al., 2020), Mixup (Zhang et al., 2018), CutMix (Yun et al., 2019), and random erasing (Zhong et al., 2020).

COCO for object detection. COCO (Lin et al., 2014) comprises 118K training and 5K validation images over 80 object categories. We follow the standard $1\times$ and $3\times$ Mask R-CNN (He et al., 2017) settings of Swin Transformer (Liu et al., 2021), fine-tuning ImageNet-pretrained ViF-T/S/B as backbones with MMDetection (Chen et al., 2019). The $1\times$ schedule trains for 12 epochs with learning-rate decay at epochs 8 and 11, and the $3\times$ schedule for 36 epochs with decay at epochs 27 and 33, both with AdamW, weight decay 0.05, and batch size 16.

ADE20K for semantic segmentation. ADE20K (Zhou et al., 2019) comprises 20K training, 2K validation, and 3K test images over 150 semantic categories. We follow the standard UPerNet (Xiao et al., 2018) settings of Swin Transformer (Liu et al., 2021) with MMSegmentation (MMSegmentation Contributors, 2020), fine-tuning ImageNet-pretrained ViF-T/S/B for 160K iterations with AdamW, a polynomial learning-rate decay of power 0.9, weight decay 1×10^{-4}, and batch size 16, and report both single- and multi-scale testing.

Table A4 summarizes the training configuration for the three tasks.

Table A4: **Training configuration for the three tasks.** Classification is trained from scratch on ImageNet-1K, whereas detection and segmentation fine-tune the ImageNet-pretrained backbone. Settings follow the Swin protocol (Liu et al., 2021), including the classification batch size of 1024; "–" denotes not applicable.

Configuration	Classification	Detection	Segmentation
Framework	Swin codebase	MMDetection	MMSegmentation
Task head	–	Mask R-CNN	UPerNet
Optimizer	AdamW	AdamW	AdamW
Training length	300 epochs	12 / 36 epochs	160K iters
LR schedule	cosine	step	poly (0.9)
Warm-up	20 epochs	linear	linear
Weight decay	0.05	0.05	1×10^{-4}
Batch size	1024	16	16
Stoch. depth (T/S/B)	0.2 / 0.3 / 0.5	–	–
Augmentation	RandAug (m9, mstd 0.5), Mixup 0.8, CutMix 1.0, Rand. Erasing 0.25	–	–
Regularization	MESA	–	–
Weight EMA decay	0.9999	–	–
Label smoothing	0.1	–	–

C Architecture Details

ViF follows the four-stage hierarchical framework of Liu et al. (2021). An input image at 224×224 is first processed by a convolutional stem that downsamples it by $4\times$ into a 56×56 feature map: a 3×3 stride-2 convolution, a residual pair of 3×3 stride-1 convolutions, and a 3×3 stride-2 convolution followed by a 1×1 projection, each convolution with batch normalization and a ReLU nonlinearity. Between stages, a downsampling layer halves the resolution and doubles the channel width through an inverted bottleneck, namely a 1×1 expansion, a depthwise 3×3 stride-2 convolution, and a 1×1 projection. A ViF block applies, each with a residual connection, a depthwise Conditional Position Encoding (CPE), the Fluxer operator, a second depthwise CPE, and a gated convolutional MLP, which is a linear layer split into a gate and a value with a depthwise 3×3 convolution applied to the gate. Within the operator, a single linear layer projects the input into the signal, kernel, and

Table A5: **Layer-by-layer architecture of the three ViF variants.** Local denotes the depthwise convolution applied to the signal and gate branches, and Global mix denotes the spectral product of Eq. (1), whose width is the mixer ratio times the stage width; the linear layer preceding it produces all three branches at once. The FFN is gated, so its hidden width is $\frac{2}{3}$ of the nominal ratio in Table A10. Every convolution in the stem and in the downsampling layers is followed by batch normalization, and the two CPEs, the operator, and the FFN each sit in their own residual branch.

Layer	Output size	ViF-T	ViF-S	ViF-B
Stem	56×56	Conv 3×3 stride 2; residual pair of Conv 3×3 stride 1; Conv 3×3 stride 2; Conv $1 \times 1 \to C$		
		$C = 72$	$C = 84$	$C = 96$
Stage 1	56×56	ViF Blocks CPE 3×3 Linear $72 \to 216$ Local 3×3 Global mix 72 Linear $72 \to 72$ FFN $72 \to 384 \to 72$ $\times 2$	ViF Blocks CPE 3×3 Linear $84 \to 252$ Local 3×3 Global mix 84 Linear $84 \to 84$ FFN $84 \to 448 \to 84$ $\times 3$	ViF Blocks CPE 3×3 Linear $96 \to 576$ Local 3×3 Global mix 192 Linear $192 \to 96$ FFN $96 \to 512 \to 96$ $\times 3$
		Down sampling: Conv $1 \times 1 \to 4C$, DWConv 3×3 stride 2, Conv $1 \times 1 \to C$		
		$C = 144$	$C = 168$	$C = 192$
Stage 2	28×28	ViF Blocks CPE 3×3 Linear $144 \to 432$ Local 3×3 Global mix 144 Linear $144 \to 144$ FFN $144 \to 768 \to 144$ $\times 4$	ViF Blocks CPE 3×3 Linear $168 \to 504$ Local 3×3 Global mix 168 Linear $168 \to 168$ FFN $168 \to 896 \to 168$ $\times 5$	ViF Blocks CPE 3×3 Linear $192 \to 1152$ Local 3×3 Global mix 384 Linear $384 \to 192$ FFN $192 \to 1024 \to 192$ $\times 5$
		Down sampling: Conv $1 \times 1 \to 4C$, DWConv 3×3 stride 2, Conv $1 \times 1 \to C$		
		$C = 288$	$C = 336$	$C = 384$
Stage 3	14×14	ViF Blocks CPE 3×3 Linear $288 \to 864$ Local 3×3 Global mix 288 Linear $288 \to 288$ FFN $288 \to 768 \to 288$ $\times 8$	ViF Blocks CPE 3×3 Linear $336 \to 1008$ Local 3×3 Global mix 336 Linear $336 \to 336$ FFN $336 \to 896 \to 336$ $\times 12$	ViF Blocks CPE 3×3 Linear $384 \to 2304$ Local 3×3 Global mix 768 Linear $768 \to 384$ FFN $384 \to 1024 \to 384$ $\times 16$
		Down sampling: Conv $1 \times 1 \to 4C$, DWConv 3×3 stride 2, Conv $1 \times 1 \to C$		
		$C = 576$	$C = 672$	$C = 768$
Stage 4	7×7	ViF Blocks CPE 3×3 Linear $576 \to 1728$ Local 3×3 Global mix 576 Linear $576 \to 576$ FFN $576 \to 1536 \to 576$ $\times 4$	ViF Blocks CPE 3×3 Linear $672 \to 2016$ Local 3×3 Global mix 672 Linear $672 \to 672$ FFN $672 \to 1792 \to 672$ $\times 5$	ViF Blocks CPE 3×3 Linear $768 \to 4608$ Local 3×3 Global mix 1536 Linear $1536 \to 768$ FFN $768 \to 2048 \to 768$ $\times 5$
Head	1×1	Layer norm, global average pool, Linear $\to 1000$		

gate branches with per-channel mixer ratio e; a depthwise 3×3 convolution followed by SiLU refines the signal and gate branches, the kernel branch is refined by SiLU and normalization alone, and the spectral filtering of Eq. (1) performs the global mixing. We obtain ViF-T/S/B by scaling the embedding dimension and the number of blocks per stage, as summarized in Table A10; Table A5 gives the layer-by-layer specification.

Operator. Listing 1 gives a PyTorch-style forward pass of the Fluxer operator, making the correspondence to Eq. (1) explicit: a single linear layer produces the signal, kernel, and gate branches; a depthwise convolution and SiLU refine the signal and gate branches, whereas the kernel branch is refined by SiLU and normalization only; the kernel branch becomes a per-frequency filter multiplied against the signal spectrum; and the inverse FFT returns the spatially mixed feature, which is gated and projected back to C channels. All FFTs use the orthonormal normalization, so that the operator is unitary up to the learned filter. The listing additionally applies a learnable per-channel residual (`skip`, initialized to identity), adding $\mathbf{s} \odot \mathbf{u}$ to the mixed output. It is not redundant with the block-level residual: that one bypasses the operator entirely, whereas this term passes through the gate and the output projection together with the convolution. Equation (1) states the convolution alone because that is the term the separations concern, and Remark 1 and Proposition 4 record the two places where the extra term changes what a statement says.

Listing 1: **PyTorch-style forward pass of the Fluxer operator.** `skip` is a learnable per-channel vector initialized to 1, and only the signal and gate branches receive a depthwise convolution.

```
def convolver_layer(x):
   # x: (B, H, W, C) -> (B, H, W, C)
   u, c, z = in_proj(x).chunk(3, dim=-1) # signal u, kernel c, gate z (B,H,W,e*C)
   u = u_norm(silu(dwconv3x3(u))) # signal: dwconv + SiLU + LN
```

```
    c = c_norm(silu(c)) # kernel: SiLU + LN (no conv)
    z = silu(dwconv3x3(z)) # gate: dwconv + SiLU
    uf = rfft2(u, dim=(1, 2), norm='ortho') # signal spectrum F(u)
    cf = rfft2(c, dim=(1, 2), norm='ortho') # input-generated filter F(c)
    H = cf + skip # filter H = F(c) + skip
    y = irfft2(uf * H, s=x.shape[1:3], dim=(1, 2), norm='ortho') # y = u (*) c (Eq. 1)
    y = out_norm(y * z) # gating z + LayerNorm
    return out_proj(y) # (B, H, W, C)
```

Variants. The per-stage configuration of the three variants is given in Table A10. All variants use two depthwise 3×3 CPEs per block, SiLU activations inside the operator, and GELU in the MLP; the mixer ratio is 1 for ViF-T/S and 2 for ViF-B, and the gated convolutional MLP uses a larger expansion at the two higher-resolution stages and a smaller one in the deeper stages. The resulting parameter and FLOPs counts at 224×224 are those reported in Table 1.

The three comparison variants of Fig. 5. Each changes the kernel branch and nothing else: the projection to three branches, the depthwise refinements on the signal and gate branches, the gate, the output normalization, and the output projection are as above, and the depths, widths, schedule, and recipe are those of ViF-T.

Static. The kernel branch is replaced by a learned per-frequency tensor $\mathbf{H} \in \mathbb{C}^{D \times H \times (\lfloor W/2 \rfloor + 1)}$, shared across inputs. This is the reduction of Proposition 1 and the operator of Rao et al. (2021).

Frequency-pointwise, after Huang et al. (2023). The kernel branch is removed and the filter is computed from the spectrum of the signal branch. Writing $\hat{\mathbf{u}}(\xi) \in \mathbb{C}^D$ for the spectrum at frequency ξ,

$$\mathbf{H}(\xi) \;=\; S_\lambda\big(\mathbf{W}_2\, \rho(\mathbf{W}_1 \hat{\mathbf{u}}(\xi) + \mathbf{b}_1) + \mathbf{b}_2\big), \tag{A3}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are complex and block diagonal in eight groups of $D/8$ channels, ρ is a ReLU, and S_λ is the soft shrinkage at $\lambda = 10^{-2}$ that their released layer carries over from the operator it was forked from; this is the mask of Huang et al. (2023), a group linear layer, a nonlinearity, and a second group linear layer. The filter multiplies the spectrum it was computed from, which is what makes the variant a dynamic convolution rather than a frequency-domain feed-forward stack. Every operation is applied one frequency at a time, so $\mathbf{H}(\xi)$ depends on no $\hat{\mathbf{u}}(\eta)$ with $\eta \neq \xi$, which is the property Proposition 2 concerns. The two mixings cost $D^2/2 + 4D$ parameters per operator, against the CD our kernel branch occupies in the shared projection.

Shift-invariant, after Karami & Ghodsi (2024). The kernel branch is kept but its phase is discarded. The branch is refined and transformed as in Fluxer, reduced to its squared magnitude $|\hat{\mathbf{c}}|^2$, and a depthwise convolution of width three along the frequency axis maps that to the real and imaginary parts of the filter. The kernel is then invariant to translations of the input and, by Proposition 4, the operator is equivariant. This is the default conditioning network of Orchid read on a two-dimensional grid rather than a sequence, with the non-causal branch taken, since an image has no direction.

D Measured Numbers

The probes summarized in Figs. 3, 5 and 6 are tabulated here.

E Notes on the Analysis

The paragraphs below are the full versions of the arguments the analysis states in brief.

Why one model carries three numbers. ViF-T appears as 83.9 in Tables 1 and A6, as 83.93 in Table A11, and as 83.87 in Table A8 and the left panel of Fig. 6. The first is the second rounded to the one decimal the comparison tables carry. The third is the same checkpoint read through the resolution sweep, whose evaluation transform is derived from the target size rather than fixed at the $256 \rightarrow 224$ resize-and-crop of the classification protocol; at 224 the two differ by 0.06. No comparison in the paper crosses the two: every delta we quote is taken within one protocol.

What the training recipe reaches. The recipe adds MESA (Du et al., 2022) to the protocol we otherwise share with Swin (Liu et al., 2021), and most numbers we compare against were published without it. Table A11 ablates it: without MESA the three variants reach 83.68, 84.60, and 85.35, against 83.93, 84.84, and 85.41 with it, so it is worth 0.25, 0.24, and 0.06 points, shrinking as the

Table A6: **Kernel-branch substitutions and retrained variants for ViF-T on ImageNet-1K.** Destroying the spatial arrangement of the kernel costs two tenths of a point, whereas replacing it with the intact kernel of another image costs 12.3; on a class-paired subset, a kernel from an image of the same class costs 8.8 against 14.9 for a random one. Each entry is top-1 accuracy with the signal and gate branches always taken from the evaluated image and the kernel branch redirected as stated, in every operator at once unless a stage is named. Each block carries its own matched control, which reproduces the accuracy of the checkpoint on the images that block is measured over; the class-paired block uses a 2000-image subset in which every image has a partner of its own class, so its rows are compared with each other and not with the rows measured on the full set. The final block is not a swap: it retrains the backbone from scratch with one thing changed and everything else held fixed.

	Kernel branch taken from	Top-1	Δ
matched	the same image (control)	83.9	–
norm	Gaussian noise at the kernel's own moments	83.7	−0.2
shuffle	the kernel's own values, spatially permuted	83.7	−0.2
mean	the batch mean, no individual information	80.4	−3.5
shifted	the next image in the batch	71.7	−12.2
deranged	a random other image	71.6	−12.3
class-paired subset, 2000 *images*			
matched	the same image (control)	84.1	–
class	a different image of the same class	75.3	−8.8
deranged	a random other image	69.2	−14.9
deranged within one stage only			
stage 1	56 × 56	83.4	−0.4
stage 2	28 × 28	83.4	−0.4
stage 3	14 × 14	79.6	−4.2
stage 4	7 × 7	82.8	−1.1
retrained from scratch, one change at a time			
Fluxer	projected from the signal in space, left equivariant	83.9	–
static	a learned input-independent spectral filter	83.2	−0.7
freq. pointwise	a mask computed pointwise from the spectrum	83.3	−0.6
shift-invariant	generated in space, phase suppressed	83.5	−0.4
−signal conv	without the signal depthwise convolution	83.7	−0.2
−gate conv	without the gate depthwise convolution	83.6	−0.3
−gate branch	without the gate branch	83.5	−0.4
zero pad	linear convolution, nothing wraps	83.8	−0.1

model grows and of the same order as several of the architectural margins we report against papers that did not use it; it was set to 1.0, 1.5, and 2.0 at the three scales. The confound touches every comparison across papers, including the 2.6 points over Swin-T, the 1.3 over VMamba-T, and the gap between the static variant at 83.2 and a published ConvNeXt-T, and none of the comparisons within this paper, since every retrained row of Fig. 5 carries the same recipe on both arms and the swap conditions are read off one checkpoint at inference. We report the ablations as within-recipe differences and leave the cross-paper decomposition open.

Efficiency. ViF-T uses 29M parameters and 4.8G FLOPs against Swin-T (28M and 4.5G) and VMamba-T (30M and 4.9G), and improves top-1 by 2.6 and 1.3 points (Table 1). What that budget buys as the resolution grows is Fig. 3, read at batch 1 for the mixing step alone; the backbone figures closing this paragraph are a separate job at batch 8. The comparison that matters is against a fused attention kernel. Written out, the attention product needs 88.8 ms and 64 GB at 1024; through FLASHATTENTION, 11.2 ms and 49 MB. The memory advantage is therefore not ours to claim, since that kernel never builds the score matrix and uses a third of what Fluxer does (49 against 145 MB); what survives is time, by a factor of 54 (0.207 ms against 11.2). Measured at matched d_{model} with the projections included, the ordering is set by the projections rather than the mixing: at width 96 the bidirectional scan performs 3.3 times, and the static spectral filter half, the per-token multiply-accumulates ours performs, and the latencies come out at 3.5 and 0.75 times ours. Figure 3 therefore times the primitive alone. A *single* selective scan is the faster operation up to about 320^2, but a Vision Mamba (ViM) does not run one: Zhu et al. (2024) scan in two directions and Liu et al. (2024) in four, and against those we are ahead at every resolution of Table A7, by 1.3 and 2 times at 224 and by 6 and 4.9 at 1024. The reason is not the asymptotics, which favor the scan: over the last fourfold increase in tokens the scans grow 3.8 to 6.4×, the one- and two-directional ones more than linearly, whereas Fluxer grows 3.0×, less than linearly. A batch-1 scan at 96 channels does not fill an H100, and its sequential depth grows with the sequence, whereas three transform calls are bandwidth-bound and parallel throughout. The scan is given its own CUDA kernel and its native channel-last layout, so it is not charged the permute a channel-first backbone would owe, while the

Table A7: **Latency and peak memory of the mixing step alone, behind Fig. 3.** Each operator is timed at width 96 and batch 1 on one H100 with the projections every mixer pays kept outside the timed region, and every operator is handed tensors of the same shape, so nothing here compares projection widths. Each primitive runs on its standard kernel: `scaled_dot_product_attention` for attention, `selective_scan_fn` for the scan, and cuFFT for the transforms; the cross-scan and merge of SS2D, however, take the PyTorch path rather than the Triton kernel of Liu et al. (2024), which if anything overstates their cost. The fused attention row is FLASHATTENTION rather than whatever the dispatcher chose: forcing that backend in a separate run gives 11.021 ms at 1024 against 11.020 for the default and 30.669 for the memory-efficient backend, so the default is that kernel. The three scan rows are the same primitive run once, twice, and four times over the raster orders of Zhu et al. (2024) and Liu et al. (2024), with the transposes and flips inside the timed region; a ViM runs two or four directions, so those rows are the ones to read against ours. The static spectral row is the global filter of Rao et al. (2021), one learned complex weight per channel and grid frequency, which performs one forward transform against our two, the whole of the difference between the two rows. Each operator is measured in its own process, for the reason given in Section E, and every entry is the median of nine repeats of 100 iterations whose spread is under 3% throughout, apart from the static spectral filter at 224 and 384, at 4.6% and 13.5%. The transforms run in `float32`, cuFFT having no `bfloat16` path, whereas the scan and attention kernels run in `bfloat16`; the asymmetry is a cost to us. The backbone block is a separate job at batch 8 and is not comparable line by line with the blocks above.

Input resolution	224	384	512	768	1024
the mixing step alone, batch-1 latency (ms)					
Self-attention, written out (Vaswani et al., 2017)	0.284	2.203	4.810	26.73	88.80
Self-attention, FLASHATTENTION (Dao et al., 2022)	0.054	0.410	0.706	3.901	11.19
Selective scan, 1D raster (Gu & Dao, 2024)	0.032	0.076	0.096	0.251	0.616
the same, 2-directional (Zhu et al., 2024)	0.067	0.156	0.198	0.699	1.248
the same, 4-directional (SS2D) (Liu et al., 2024)	0.101	0.203	0.269	0.585	1.018
SS2D issued as four separate launches	0.142	0.324	0.434	1.474	2.593
Static spectral filter (Rao et al., 2021)	0.050	0.049	0.049	0.094	0.151
Fluxer (ours)	0.051	0.058	0.068	0.130	0.207
the mixing step alone, batch-1 peak memory (MB)					
Self-attention, written out (Vaswani et al., 2017)	186	1333	4137	20788	65604
Self-attention, FLASHATTENTION (Dao et al., 2022)	9	7	12	28	49
Selective scan, 1D raster (Gu & Dao, 2024)	3	9	16	36	64
the same, 2-directional (Zhu et al., 2024)	4	12	22	50	88
the same, 4-directional (SS2D) (Liu et al., 2024)	14	42	74	168	292
Static spectral filter (Rao et al., 2021)	7	21	36	84	145
Fluxer (ours)	7	21	36	84	145
ViF-T backbone, batch 8, measured separately					
throughput (img/s)	784	399	250	118	69
peak memory (MB)	254	427	689	1441	2489

Table A8: **Top-1 accuracy of the 224 checkpoint of ViF-T across input resolutions.** The model is evaluated at every multiple of 32 from 128 to 512 with the 224 checkpoint unchanged, with no fine-tuning and no resampling of any weight, on the full validation set; Δ is measured against the training resolution, which is shaded. Multiples of 32 are the admissible sizes because 32 is the total stride of the backbone.

Input size	Tokens at stride 4	Top-1	Δ
128	1024	76.70	−7.17
160	1600	81.31	−2.56
192	2304	82.86	−1.01
224	3136	83.87	– (train)
256	4096	84.28	+0.41
288	5184	84.25	+0.38
320	6400	84.00	+0.13
352	7744	83.55	−0.32
384	9216	82.97	−0.90
416	10816	82.29	−1.58
448	12544	81.28	−2.59
480	14400	80.40	−3.47
512	16384	79.15	−4.72

transposes and flips a multi-directional scan needs are inside its timed region because it cannot avoid them.

The table is measured one operator per process, which is not fastidiousness. A sweep that ran every operator at one size before moving to the next read up to 2.8 times high at 224^2, our own primitive at 0.102 ms, twice what an isolated process gives, because by the time a given primitive ran the allocator had been churned by the written-out attention and emptied between repeats; the large end was unaffected. Every entry therefore comes from an isolated process and carries the spread of its nine repeats, which is under 3% for all but two static-filter entries. The whole table was run a second time on another node of the same cluster. Every row agrees to within 1.2% at 768 and 1024, and the

Table A9: **Whole-backbone throughput on one H100 at** $3 \times 224 \times 224$**, batch** 128**.** Every row is our own measurement under one protocol, with official checkpoints and reference implementations, `bfloat16`, 20 warm-up and 50 timed iterations timed with CUDA events, the median of three passes, and no compilation or fusion beyond what each repository ships, so the four backbones are directly comparable and nothing here is quoted from another paper. The best figure at each scale is in bold.

		images per second
Backbone	Params	`bf16`
tiny scale		
ViF-T	29M	1567
Swin-T	28M	1244
ConvNeXt-T	29M	1189
VMamba-T	30M	**1686**
small scale		
ViF-S	51M	**990**
Swin-S	50M	718
ConvNeXt-S	50M	682
VMamba-S	50M	877
base scale		
ViF-B	99M	**651**
Swin-B	88M	458
ConvNeXt-B	89M	435
VMamba-B	89M	646

Table A10: **Configuration of the ViF variants.** Blocks and channels are per stage. The mixer ratio e is the per-channel expansion inside the Fluxer operator, and the FFN ratio is the inverted-bottleneck expansion of the gated MLP, larger at the two higher-resolution stages; stoch. depth is the maximum stochastic-depth rate.

Model	Blocks	Channels	FFN ratio	Mixer ratio	Stoch. depth
ViF-T	[2, 4, 8, 4]	[72, 144, 288, 576]	[8, 8, 4, 4]	1	0.2
ViF-S	[3, 5, 12, 5]	[84, 168, 336, 672]	[8, 8, 4, 4]	1	0.3
ViF-B	[3, 5, 16, 5]	[96, 192, 384, 768]	[8, 8, 4, 4]	2	0.5

attention and scan rows already from 384; the transforms are launch-bound below 768, and our own read up to 1.7 times slower on the second node, which puts it behind FLASHATTENTION and the bidirectional scan at 224 and moves the crossing with the single scan to about 440^2; the comparisons with FLASHATTENTION and both multi-directional scans keep their order from 384 upward. The scan rows carry a second correction of the same kind. SS2D issues its four directions as one launch over $4D$ channels rather than four launches, which we had not done; timed as Liu et al. (2024) run it, it costs 1.018 ms at 1024 against the 2.593 four launches cost, and our lead over it falls from 13× to 4.9×. The two forms compute the same thing, to 0 absolute error on the same parameters. None of this survives into the backbone, where at matched parameters and FLOPs VMamba-T is the faster of the two (Table A9); the mixing step is a small part of what a backbone spends at recognition resolution, which is the point Fig. 3 and Table A9 make together. The full backbone processes 1024×1024 inputs at 69 images per second in 2.5 GB.

Matched FLOPs do not imply matched wall-clock time, and the right panel of Fig. 6 places the operator against the three families it is compared with throughout. ViF reaches 1567, 990, and 651 images per second at the tiny, small, and base scales, against 1189, 682, and 435 for ConvNeXt, 1244, 718, and 458 for Swin, and 1686, 877, and 646 for VMamba. Every figure is our own, taken on one H100 under the protocol of Table A9, so the four are comparable, and none should be compared against numbers published on other hardware. ViF is ahead of ConvNeXt and Swin at all three scales and behind VMamba at the tiny scale only. What the operator spends at this size is not the transform but the projection producing three branches at the inner width, together with permutations and transforms that move far more memory than they do arithmetic, none of which occupies the matrix engines that a windowed attention or a depthwise convolution keeps busy. The advantage is asymptotic and is realized where the token count is large: Fig. 3 is what makes a 1024×1024 input tractable at all, and it is not a statement about throughput at 224.

The baseline set of Table 2. Detection and segmentation share all but one entry of their baseline set and are reported together for that reason: MogaNet is reported for detection only and NAT for segmentation only, and some papers omit the 3× schedule. The set is the intersection of the one used by Lou & Yu (2025), which is not itself a row for the reason given below, with Table 1, restricted to families reported at all three scales so that no family is represented by a single favorable entry; backbones are ordered and cited as in Table 1. Baseline numbers are quoted from that source except

Table A11: **Effect of MESA on ImageNet-1K top-1 accuracy.** MESA is worth a quarter of a point at the tiny and small scales and almost nothing at the base scale. Each variant is trained with and without MESA, with everything else in the recipe held fixed.

	w/o MESA	ViF	Δ
ViF-T	83.68	83.93	+0.25
ViF-S	84.60	84.84	+0.24
ViF-B	85.35	85.41	+0.06

for MambaOut, and for ConvNeXt-B and VMamba-B on segmentation, which it omits and which come from the original papers.

Why that source is not itself a row. Lou & Yu (2025) does not appear among the backbones we compare against, and the omission is deliberate. Every entry of Table 1 is a single-stream four-stage pyramid but stems, downsampling, feed-forward blocks, widths, depths, and recipes still differ between its rows, so the table compares complete systems at similar budgets rather than isolating mixers; OverLoCK is a branched design carrying three sub-networks and a top-down context path, which departs from that shared macro-architecture as well. Its accuracy is nonetheless worth stating rather than leaving to inference: 84.2 against our 83.9 at the tiny scale, at 33M parameters and 5.5G FLOPs against our 29M and 4.8G; level with ours at 84.8 at the small scale, with ours at the smaller budget; and 85.1 against our 85.4 at the base scale. A branched top-down architecture is therefore ahead of our single-stream design at the smallest scale we test, and the narrower question of what changes when only the kernel branch changes is the one Fig. 5 addresses.

Transfer to an unseen resolution. Nothing in the operator is tied to the resolution it was trained at: the kernel is generated from whatever feature map arrives and transformed on the grid that map defines, so a change of input size changes the size of $\mathbf{c}$ and nothing else. A static filter carrying one free parameter per grid frequency, GFNet and its relatives, cannot say the same, and the reason is a dependence on the parameterization rather than a fact about frequencies. If the image domain is fixed at $[0, 1]^2$, index k denotes k cycles per image on every grid, and refining the sampling does not change what the index means; what changes is the shape of the parameter tensor, since the number of representable indices is set by the grid. A free parameter per discrete frequency therefore has no value at the indices a finer grid adds, and continuing to run requires a rule, whether interpolation, zero-padding, or tiling, that is a modeling choice rather than a recovery of the trained operator. Spectral operators parameterized continuously in frequency, or by a fixed number of low-order modes, need no such rule; the tie is to the discrete parameterization, not to the spectral domain. The absolute position embeddings of a plain Vision Transformer (ViT) are tied to the grid for the analogous reason, which is why higher-resolution results for those models are reported after fine-tuning. Generating the filter before the transform, rather than learning it after, is what removes the tie.

F VISUALIZATION

This appendix collects the qualitative evidence. Figure A2 shows that the kernel is regenerated per input, which is the property Theorem 1 concerns, and Figs. A3 and A4 show what the resulting backbone does on the two dense-prediction benchmarks.

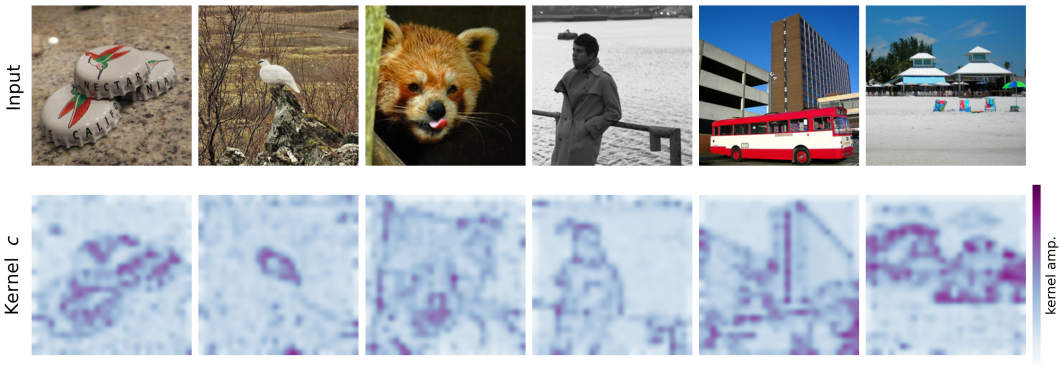

Figure A2: **Input-generated kernels of Fluxer for six images at a fixed layer.** The kernel $\mathbf{c}$ (bottom row) changes with the input (top row), whereas a static filter would produce an identical kernel for every column. Measured over 72 validation images, the coefficient of variation of the effective filter $|\mathbf{H}|$ across inputs is 0.39, against exactly 0 for a static filter.



Figure A3: **Object detection and instance segmentation on COCO with a ViF-T backbone.** Each row is one validation image under the three headings, and the prediction reports a confidence for every instance. The rows probe four difficulties: a crowd of small overlapping people, a cluttered interior spanning many categories at once, two animals whose stripes destroy the local cues that separate them, and a scene whose objects are a few pixels across.

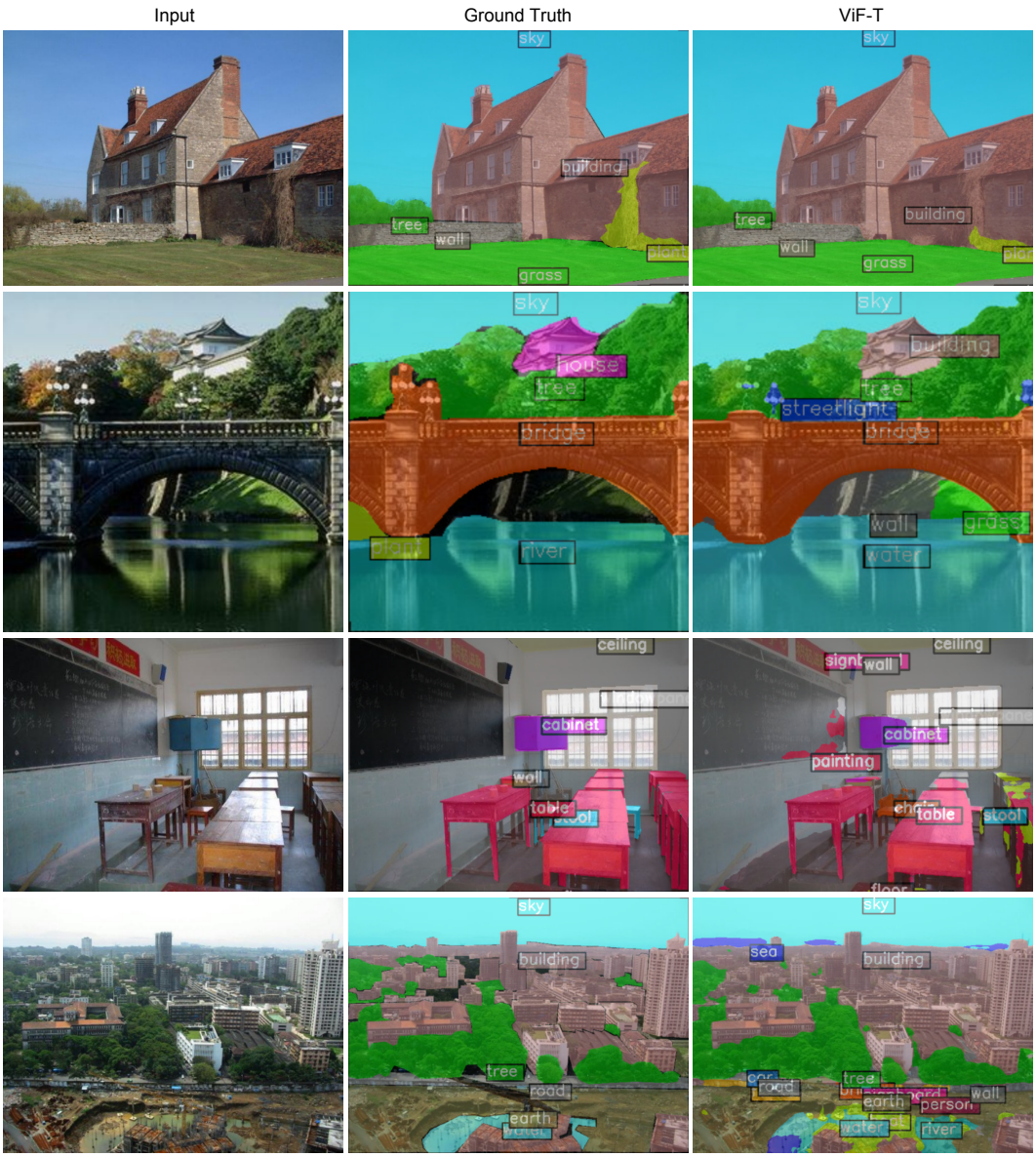

Figure A4: **Semantic segmentation on ADE20K with a ViF-T backbone.** Each row is one validation image under the three headings. The rows span an outdoor building, a bridge over water, a classroom, and a distant cityscape; the prediction recovers classes that the ground truth leaves unmarked, such as the streetlights on the bridge and the sea behind the city.