DEPTHWISE CHANNEL-ASYMMETRIC ATTENTION: INVESTIGATING CAPACITY, SCALING, AND USES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision transformers embed image patches by summing over input channels, which assumes every channel measures the same quantity at the same point. This fails for instruments whose channels are separate sensors, such as calorimeter detectors, where electromagnetic and hadronic channels are physically distinct. DepthViT is an architecture which attends along the channel axis instead of the token axis. We extend this model for more general image tasks by restoring the resulting loss of spatial communication with sparse hierarchical attention pooling (HAP) blocks and a structure-preserving head. At a matched 22M budget on the HLS4ML Large Hadron Collider jet benchmark, DepthViT2 reaches 75.34% top-1 against 73.98% for a channel-symmetric ViT-Small, winning every class and using fewer FLOPs; on CIFAR-100 it leads 62.76% to 52.92%. The advantage is budget-dependent: across a ladder from 164K to 22M parameters the comparison changes sign, losing at 164K (-1.56 points), tying at 1M, and winning at 5.4M and 22M (+1.69, +1.36). We trace this to parameter allocation, since the channel operator scales quadratically in the capacity knob k while the feed-forward block is linear, and confirm it with a same-scaffold control showing a parameter-matched symmetric operator matches accuracy, so asymmetry is the cheapest configuration rather than the most accurate one. A scale-adjusted positional encoding further lets one trained model accept many input shapes, beating a fixed-crop ViT that discards an average of 44% of each image. Finally, matched FLOPs coexist with a 4x wall-clock cost that a fused Triton kernel removes at 71x on the isolated operator.

1 INTRODUCTION

A vision transformer begins by cutting an image into fixed-size square patches and mapping each patch to a vector, called a *token*, using a strided convolution that sums over the input channels (Dosovitskiy et al., 2021); attention then lets every token exchange information with every other. Summing over channels at the very first step is well justified for natural images, where the red, green and blue planes are three measurements of light at the same physical point.

Scientific instruments are often not like this. In a collider experiment the electromagnetic calorimeter (ECAL) and the hadronic calorimeter (HCAL) are physically distinct subdetectors, built from different materials, placed at different radii and read out through separate channels (CMS Collaboration, 2008).[1] An ECAL cell and the HCAL cell behind it therefore report different physical processes rather than two views of one quantity, and summing them discards which subdetector a deposit came from before the model has computed anything.

Preserving channel identity costs the spatial path. The natural response is to embed each channel with its own private filters and to run attention *along the channel axis*: at each spatial location, the model attends between the ECAL and HCAL entries of that location rather than between different locations. DepthViT (Authors, 2026) does this. The substitution is not free. In a standard transformer, attention is the only operator that moves information between different spatial positions: the feed-forward block acts on each token separately, so without token attention two patches at opposite

[1] The cited detector paper describes this construction; the inference we draw from it about patch embeddings is our own.

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.

corners of an image never influence one another inside the encoder. Moving attention onto the channel axis therefore removes that path entirely. A channel-asymmetric model is not a transformer with one operator swapped; it has given something up and needs it restored before its defining mechanism can be judged. DepthViT2 restores spatial communication sparsely, through interleaved hierarchical attention pooling (HAP) blocks that exchange information between neighbouring windows of patches, and reads out through a head that preserves the channel structure instead of flattening it, both selected on ImageNet-1K before any physics work.

The question. Reported efficiency gains for architectures of this kind are typically measured against a standard-configuration transformer, a comparison at unequal budget that credits a small channel-asymmetric model for being small rather than for being channel-asymmetric. We ask instead: *at equal parameter budget, does the complete channel-asymmetric design beat a channel-symmetric transformer?* Across a four-tier ladder on the HLS4ML LHC jet benchmark (Duarte et al., 2018; Pierini et al., 2020c;d;a;b; Moreno et al., 2020) the answer turns out to depend on the budget, and the gap does not merely narrow as the budget grows: it changes sign.

Related work. Most jet taggers operate on particle clouds requiring per-particle four-vectors (Qu & Gouskos, 2020; Qu et al., 2022), whereas image-based approaches treat calorimeter deposits as a multi-channel image (Pierini et al., 2020c; Moreno et al., 2020); HEP-JEPA (Bardhan et al., 2025) applies self-supervised joint-embedding pretraining to collider data. Efficiency work elsewhere keeps attention on the token axis and makes it cheaper (Liu et al., 2021; Wang et al., 2020; Mehta & Rastegari, 2022; Graham et al., 2021; Li et al., 2022), whereas channel-wise attention changes *what* attention is computed over. Feature-axis attention is not new: squeeze-and-excitation reweights channels with a learned gain but does not attend between them (Hu et al., 2018; Woo et al., 2018), and cross-covariance and restoration variants attend across hundreds of *latent* channels for computational reasons (El-Nouby et al., 2021; Zamir et al., 2022). Ours differs on both counts: the axis carries a few *physically distinct measurement streams*, each with a private embedding, producing the systems problem of Section 6.

Outline. Section 2 introduces the architecture; Section 3 the matched-budget results and a channel-treatment control; Section 4 traces where the parameters go; Section 5 tests generalisation and resolution independence; Section 6 measures runtime cost and a fused kernel; and Section 7–8 state limitations and conclude.

2 ARCHITECTURE

DepthViT2 carries information along two co-designed pathways: *physical-channel modelling* (private per-channel embeddings, channel-specific query/key/value projections, attention along the ECAL/HCAL axis) and *sparse spatial communication* (interleaved HAP blocks moving information between windows of patches while avoiding the $L \times L$ all-pairs term of token attention), read out by a structure-preserving head (Fig. 1). Against a standard transformer, the design substitutes early channel collapse plus dense spatial mixing for preserved channels plus sparse spatial mixing, without a flatten.

Channel-wise attention. Each channel $c \in \{1, \dots, C\}$ is embedded independently by a depthwise convolution, that is, a convolution whose filters are not shared across channels, with stride equal to the patch size p, so that token x_l has dimension Ck:

$$e_{l,c} = W_c * \mathrm{patch}_l(X_c) + b_c \in \mathbb{R}^k, \qquad x_l = \mathrm{concat}_{c=1}^{C}(e_{l,c}) \in \mathbb{R}^{Ck}. \tag{1}$$

Attention maps every element to a *query* q (what it looks for), a *key* ξ (what it offers) and a *value* v (what it passes on if selected); these three linear maps are the QKV projection, shared by all tokens in a standard transformer. Within each DepthViT2 block x_l is reshaped to $T_l \in \mathbb{R}^{C \times k}$ and the QKV projection is instead channel-specific,

$$[q_{l,c}, \xi_{l,c}, v_{l,c}] = T_{l,c,:} W_c^{qkv} + b_c^{qkv} \in \mathbb{R}^{3k}, \qquad W_c^{qkv} \in \mathbb{R}^{k \times 3k}, \tag{2}$$

so the ECAL and HCAL streams are read out by different weights. This gives per-channel scalars for each of $h = 1, \dots, k$ sub-feature components; each component acts as an independent *head*,

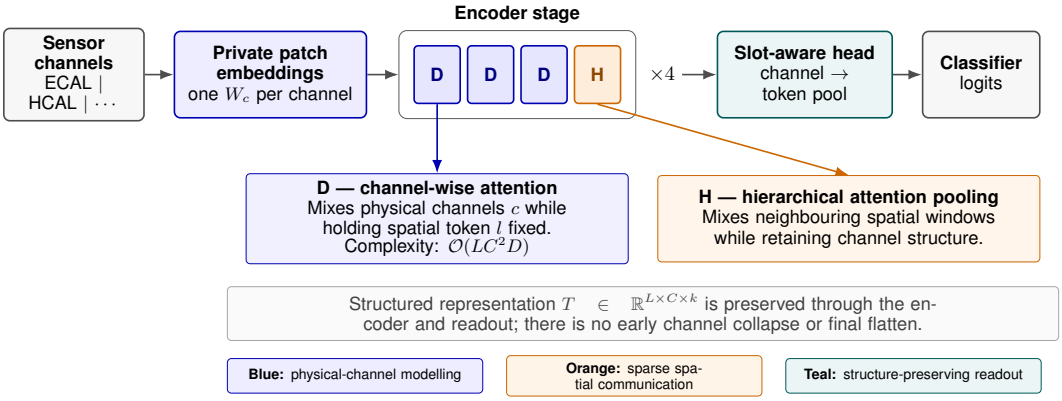

Figure 1: **DepthViT2 separates channel and spatial communication.** Private patch embeddings preserve physically distinct sensor channels. Each **D** block applies channel-wise attention independently at every spatial token; each interleaved **H** block supplies the cross-window spatial communication that channel attention omits. The $(\mathrm{D}-\mathrm{D}-\mathrm{D}-\mathrm{H}) \times 4$ encoder is followed by a slot-aware, structure-preserving readout rather than a flattening operation.

Table 1: Channel-wise attention versus standard self-attention. L is token count, C channel count, k the per-channel subspace width, D the token width. Row 3 is the difference most easily missed from the equations; row 1 is the source of the spatial-communication debt HAP repays.

	Standard self-attention	**Channel-wise attention**
Mixing axis	spatial tokens (L)	channels (C)
$Q/K/V$ projection	shared across tokens	private per channel (Eq. 2)
Softmax normalises over	source token (incoming)	destination channel (outgoing)
Score matrix	$L \times L$	$C \times C$, per head, per token
Attention complexity	$O(L^2D)$	$O(LC^2D)$
Heads	fixed count, shared width	k parallel, one per sub-feature

an attention computation run in parallel on its own slice of the feature vector. Attention is then computed between channels, separately per head:

$$A_l^h[c,c'] = \mathrm{softmax}_{c'}\left(\frac{q_{l,c,h}\,\xi_{l,c',h}}{\sqrt{k}}\right), \qquad O_{l,c',h} = \sum_c A_l^h[c,c']\,v_{l,c,h}, \tag{3}$$

where the softmax converts a vector of scores into non-negative weights summing to one. Here it normalises over the *destination* channel: each channel distributes its value with outgoing weights summing to one, rather than each destination gathering with incoming weights as in standard self-attention. Both are valid channel-mixing operators; we describe ours as implemented, with Table 1 stating the differences easy to miss from the equations. The output O_l passes through a linear map shared across channels and is flattened back to $\mathbb{R}^{Ck}$ before the residual connection. The capacity knob is k, which sets each channel's subspace width while leaving the $O(LC^2)$ structure and the spatial budget unchanged.

Row 5 matters beyond calorimetry: standard attention is quadratic in the number of patches and channel attention is linear, so doubling an image's resolution doubles the cost of a DepthViT2 encoder and quadruples that of a ViT of the same width.

Hierarchical attention pooling. Within the encoder, tokens at separate positions do not exchange information *at all* until a HAP block, a consequence Row 1 of Table 1 states but does not emphasise. Some intermediate spatial path is therefore needed, and the question is how much, since a dense token-attention path would restore exactly the quadratic cost the architecture avoids. Partitioning the grid X into $w \times w$ windows, the block first mixes within each window via learned sparse cyclic shifts, where $\mathrm{roll}_{(\delta y,\delta x)}$ translates the grid by $(\delta y, \delta x)$ with wraparound and $\odot$ is elementwise

multiplication,

$$\delta^{\text{micro}}_{i,j} = \sum_{(\delta y,\delta x)\in S_{\text{micro}}} \alpha^{\text{micro}}_{(\delta y,\delta x)} \odot \text{roll}_{(\delta y,\delta x)}(X_{i,j}), \qquad \bar{X}'_{i,j} = \bar{X}_{i,j} + \sum_{(\delta y,\delta x)\in S_{\text{macro}}} \alpha^{\text{macro}}_{(\delta y,\delta x)} \odot \text{roll}_{(\delta y,\delta x)}\big(\bar{X}\big)_{i,j}, \tag{4}$$

with S_{micro} the power-of-two shifts up to half the window size and α learned per-shift weights initialised to zero, so a freshly initialised HAP block is the identity. The updated window is mean-pooled to a summary $\bar{X}_{i,j}$, projected and broadcast back as a first cross-window correction; the summaries are then mixed across the window grid (second term, S_{macro} up to four windows in each direction) and broadcast back as a second. The total delta sums all three under a pre-norm residual, $X'' = X + \Delta + \text{MLP}(\text{LN}(X + \Delta))$, where LN is layer normalisation. Both too little and too much mixing are harmful, so the operative choice is how often HAP blocks are interleaved among the channel-attention blocks. We perform an ImageNet-1K sweep at fixed sixteen-block-equivalent parameter budget and evaluate the top 1% accuracy at epochs 30 and 40. The fully-depthwise model with no HAP blocks gives $49.0\%/56.3\%$. A pattern of ((D-D-D-H)$\times$4), where D is a Depthwise block and H is a HAP block, gives $\mathbf{52.3}\%/\mathbf{56.4}\%$. A pattern of ((D-D-H)$\times$6) gives $49.5\%/49.0\%$, and, finally, a pattern of 50% ((D-H)$\times$12) gives $45.5\%/47.2\%$. We find a 25% ratio gives the best early-training accuracy while 50% degrades it, and fully-depthwise catches up by epoch 40. The finding is therefore faster early learning together with a penalty for over-mixing, at an unchanged ceiling. We use 25% throughout.

Head and configuration. Channel attention maintains a structured token $T_l \in \mathbb{R}^{C\times k}$ throughout the encoder, which the usual readout, flattening every token to one long vector and averaging, discards in a single step. Ours instead pools the final grid in two attention stages, first over channels with m learned mixtures and then over the L tokens with rank r:

$$p_{l,\mu,:} = \sum_c \text{softmax}_c\Big(\tfrac{1}{\sqrt{k}}\langle q^{\text{depth}}_\mu, T_{l,c,:}\rangle\Big)\, T_{l,c,:}, \qquad z_\rho = \sum_l \text{softmax}_l\Big(\tfrac{1}{\sqrt{mk}}\langle q^{\text{tok}}_\rho, s_l\rangle\Big)\, s_l, \tag{5}$$

where q^{depth}_μ, q^{tok}_ρ are learned query vectors and $s_l \in \mathbb{R}^{mk}$ concatenates $p_{l,\mu,:}$ across mixtures, followed by a layer norm and a linear classifier. On ImageNet-1K this readout is worth 5.3 points over flatten-and-pool (39.7% baseline, 60 epochs), evidence the encoder produces structure a naive head destroys. For all jet experiments we use patch size 10 on 100×100 images ($L = 100$), HAP window 5, and the (D-D-D-H)$\times$4 pattern; tiers 1–3 fix twelve depth-wise and four HAP blocks and vary only k, tier 4 also varies depth.

3 JET CLASSIFICATION AT MATCHED BUDGET

With the architecture fixed, we now evaluate it under matched-budget comparisons on the HLS4ML LHC jet benchmark.

Dataset and protocol. We evaluate on the HLS4ML LHC jet dataset (Pierini et al., 2020c;d;a;b; Moreno et al., 2020), a five-class classification benchmark (gluon, light quark, W, Z, top) on two-channel ECAL/HCAL images of size 100×100: the paper's central domain because the two channels are separate subdetectors, not colour planes. Four splits of increasing particle multiplicity (30p–150p) hold $\sim$610K–640K samples each; images are $\log(1 + x)$-transformed and per-split standardised (App. A). All models train with SGD (lr 0.1, momentum 0.9), cosine schedule, $5{,}000$ warmup steps, batch 512, bf16, four A100s, 90 epochs. We report top-1 accuracy and macro AUC, the area under the receiver-operating-characteristic curve averaged equally over classes. **All matched-budget comparisons use the 150p split.**

The four-tier ladder. Table 2 is the central result. At each budget we train DepthViT2 and a channel-symmetric transformer matched to it under an identical protocol: ViT-Tiny shrunk in width at 164K and 1M (D=32, D=82), native ViT-Tiny at 5.4M, ViT-Small at 22M.

The ladder produces a sign change rather than a narrowing gap: a gap shrinking toward zero is consistent with both models converging to one ceiling, a gap crossing zero is not. The ordering is budget-dependent, so any claim that "channel-asymmetric attention is more parameter-efficient" is under-specified without naming a budget; the $32.8\times$ parameter and $34.3\times$ FLOP ratios a 164K DepthViT2 enjoys over *native* ViT-Tiny, reported in prior work, describe an unequal comparison, while at equal budget the two 164K models are identical to three significant figures in FLOPs.

Table 2: Matched-budget comparison, 150p split, 90 epochs, identical protocol. Δ is DepthViT2 minus baseline (single-seed). The sign of Δ top-1 changes between 164K and 5.4M: the central finding. Counts are exact (module counting; `calflops`). The right block gives three-seed means (42, 123, 1234) where available; per-seed values in App. B.

				Single-seed				Three-seed mean	
Tier	Model	Params	FLOPs	Top-1	AUC	ΔT1	ΔAUC	Top-1	AUC
164K	DepthViT2 (k=4)	164,417	0.032G	70.14	.9175	-1.56	$-.0063$		
	ViT-Tiny (shrunk)	162,373	0.032G	**71.70**	**.9238**				
1M	DepthViT2 (k=23)	1,008,226	0.197G	72.31	.9270	$+0.01$	$+.0010$	72.37±.41	.9271±.0019
	ViT-Tiny (shrunk)	1,006,473	0.200G	72.30	.9260			72.30±.20	.9259±.0005
5.4M	DepthViT2 (k=90)	5,389,691	1.039G	**74.99**	**.9361**	$+1.69$	$+.0063$	75.12±.13	.9367±.0008
	ViT-Tiny (native)	5,397,893	1.083G	73.30	.9298			73.30±.11	.9298±.0009
22M	DepthViT2 (L=18, k=196)	22,046,777	4.2945G	**75.12**	**.9366**	$+0.98$	$+.0040$	**75.34±.20**	**.9374±.0008**
	ViT-Small	22,012,613	4.3113G	74.14	.9326			73.98±.14	.9319±.0006

Table 3: Same-scaffold symmetric control at 22M (150p, 90 epochs). Only the channel treatment differs. `asym` is the tier-4 model (three-seed mean); `chansum` and `symmix` are single-seed.

Arm	Params	FLOPs	Chan. MMACs	Perm.-equiv. err.	Top-1 / Macro AUC
`asym` (DepthViT2)	22,046,777	4.2945G	556.01	6.7×10^{-1}	75.34±0.20% / 0.9374
`chansum`	22,085,977	4.3023G	556.01	6.7×10^{-1}	75.09% / 0.9360
`symmix`	22,046,777	4.7023G	761.34	0.0	75.53% / 0.9382

The 22M tier. This tier gives the strongest single result: DepthViT2 and ViT-Small agree to 0.16% on parameters and 0.39% on FLOPs, with DepthViT2 the *lower* on FLOPs, so no advantage can be attributed to more compute or capacity. The win holds on every class (gluon $+0.0018$, quark $+0.0028$, W $+0.0055$, Z $+0.0060$, top $+0.0038$; App. B), ruling out a macro-average driven by one easy class, and the train/validation gap is smaller (1.1 against 3.7). Across DepthViT2's capacity progression the largest gains accrue to W and Z ($+0.0301$, $+0.0292$ from k=4 to k=90), twice gluon and quark, interpretation offered rather than demonstrated since several quantities grow with k together.

Seed variation and controls. Multi-seed results (Table 2, right) confirm both crossings: at 22M the margin widens to $+1.36$ points with non-overlapping bands, and the 1M tie holds under replication (0.07 points, well inside either arm's spread). A matched ResNet-9 control and cross-split behaviour are reported in App. C.

3.1 Same-scaffold symmetric control

The baselines above lack HAP and the head, so that comparison leaves open whether asymmetry itself produces the win. We hold the scaffold fixed and change *only* the channel treatment at 22M: **chansum** replaces the private per-channel embedding with a channel-summing one (`groups`=$C \to 1$), testing *physical* asymmetry; **symmix** replaces the operator with a permutation-equivariant map of matched parameter count at C=2, testing *private-per-channel* treatment against symmetric mixing of the same budget. `symmix` is verified numerically equivariant (error 0.0 against 6.7×10^{-1}) while costing 9.5% more FLOPs, reported honestly since it runs against DepthViT2 (App. D).

Table 3 is a careful null: turning off physical asymmetry leaves accuracy essentially unchanged (75.09% against $75.34 \pm 0.20\%$), and replacing the operator entirely does not hurt either, with `symmix`'s 75.53% sitting *inside* the asymmetric seed band. At 22M, channel asymmetry *per se* is not what separates DepthViT2 from a channel-symmetric transformer. The honest reading is that this is evidence about *efficiency* rather than mechanism: the asymmetric operator is the cheapest of the three in FLOPs (8.7% below `symmix`, also below ViT-Small) at equal accuracy, so it remains the efficient default, but the control removes any claim that asymmetry *drives* the win; the scaffold and allocation of Section 4 do. This is a single-tier control, and whether the null holds where the mechanism is barely funded is untested.

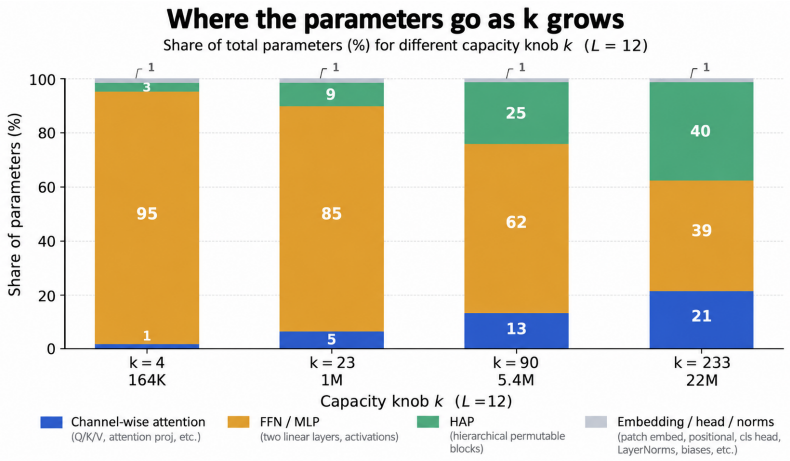

Figure 2: Parameter share by component as k grows at fixed depth $L = 12$. Channel-wise attention rises from about 1% to about 21%, the opposite of the dilution intuition. **The k=233 bar is hypothetical** (the width-only route to 22M at L=12), not the trained configuration, which is L=18, k=196 at 22% share.

4 Funding the mechanism: where the parameters go

Even once the design is viable, the operator that gives the architecture its name may be too small to matter: at our smallest configuration channel attention holds roughly one part in a hundred of the budget. That is a property of the scaling rule we adopt, in which k is the capacity knob and the feed-forward width is held fixed, not of channel attention as such; this section makes the rule explicit and shows it explains the crossover of Section 3.

The intuition one starts with is that the channel mechanism gets *diluted* as the model grows; tracing the actual parameter tensors shows the opposite, with channel-attention's share *rising* from 1.0% to 20.8% as k climbs from 4 to 233 (Fig. 2), a race between growth rates. The projection of Eq. (2) costs $3Ck^2$ per block, *quadratic* in k since both the input and output widths grow with k; the feed-forward block projects from width Ck into a *fixed* $\mathrm{mlp_dim}$=768 and back, costing $2 \cdot Ck \cdot 768$, *linear* in k; HAP is quadratic with a steeper coefficient still, operating on window summaries of width Ck. Summing these counts over twelve depth-wise and four HAP blocks, plus embedding, head and normalisation terms, gives a total exactly quadratic in k for tiers 1–3:

$$P(k) = 244k^2 + 37823k + 9221. \tag{6}$$

Equation (6) is included for three reasons. First, it checks the bookkeeping: it reproduces every tier-1–3 count to the exact integer ($P(4) = 164{,}417$, $P(23) = 1{,}008{,}226$, $P(90) = 5{,}389{,}691$), so the decomposition behind Fig. 2 is verified rather than asserted. Second, it is the instrument that builds the ladder: being a quadratic it inverts by the quadratic formula, so a target budget P gives the k that hits it, which is how every DepthViT2 arm in Table 2 lands within 0.2% of its baseline. Third, its coefficients carry the interpretation: 244 belongs to the channel mechanism and its spatial counterpart, 37823 to the pinned feed-forward block and the embedding, so the mechanism's share is $\Theta(k^2)/\Theta(k^2 + k)$ and rises *by construction* of the scaling rule, not by accident of a configuration.

The contrast with a standard ViT is instructive: there, one width D sets attention and feed-forward together ($4D^2$ and $8D^2$ per block), so attention holds a fixed one third of the parameters at every size, and a ViT cannot be scaled to change that fraction, whereas DepthViT2, pinning one width and growing the other, can, the freedom that makes the crossover possible. On compute, channel attention is linear rather than quadratic in L (Table 1), letting the matched-parameter arms of Table 2 also come in *below* their baselines on FLOPs and making the many-token regime of Section 5 affordable.

At k=4 the model is 95.3% feed-forward: a 164K feed-forward network with a channel-attention operator attached, consistent with its losing there. The mechanism receives a double-digit share only

above roughly 5M, coinciding with where Δ top-1 changes sign, an explanation consistent with the evidence rather than an isolated causal claim, since a controlled test would hold mechanism share fixed while varying budget, which the configuration space does not cleanly permit. The account also dictates tier 4: reaching 22M on k alone needs $k \approx 233$ at twelve layers, where HAP balloons to nearly 40% of parameters. A depth sweep at fixed budget shows channel-attention's share invariant to depth (21–22%), trading HAP against feed-forward only (App. C); L=18, k=196 is the shallowest depth pulling HAP below co-dominance (28.1%). Tiers 1–3 are single-knob scaling on k; tier 4 steps off that axis, and we report the departure.

5 Image size-invariance and generalisation to standard image sets

CIFAR-100. A natural worry is that the design is a calorimeter-specific artefact. We repeat the matched comparison on CIFAR-100 at ~ 5.3M, deliberately adversarial since the three channels are ordinary colour planes, exactly where Section 2 argues channel-axis attention is *not* obviously motivated. DepthViT2 (k=61, 5,323,575 params, 0.6454G) leads a matched ViT (D=192, 5,380,132, 0.6935G) by $\mathbf{9.84}$ points top-1 ($62.76 \pm 0.18\%$ vs $52.92 \pm 0.32\%$) and 0.0150 macro AUC, at 1.06% fewer parameters and 6.9% fewer FLOPs, over three seeds. The comparison does not *invert* outside calorimetry, and that it survives with ordinary colour-plane channels dovetails with the symmetric control in pointing at scaffold and allocation rather than physical asymmetry.

Scale-adjusted positional encoding. Every result so far fixes a single square resolution. A trained ViT cannot accept a differently-shaped input without surgery: token attention is $O(L^2)$, so changing L changes cost quadratically, and the positional embedding is a table indexed by position in one particular grid. DepthViT2 removes the first by construction, since channel-axis attention is computed per location at cost linear in L; the second we remove explicitly.

Let an image of size $H \times W$ be cut with patch size p into a grid of $R = H/p$ rows and $S = W/p$ columns, giving $L = RS$ tokens. The original implementation indexed by rank in its own grid,

$$\mathrm{idx}_{\text{naive}}(r, s) = rS + s, \tag{7}$$

which depends on S, so images of different widths assign one table entry to different locations and vertical adjacency is silently scrambled: a one-row step is 28 tokens in the widest bucket of Table 4 and 21 in the tallest. We instead index into a fixed maximum grid of side G,

$$\mathrm{idx}_{\text{2d}}(r, s) = rG + s, \qquad E \in \mathbb{R}^{G^2 \times Ck}, \qquad r < G,\ s < G, \tag{8}$$

so a location maps to the same entry regardless of image shape, and any patch-aligned grid up to $G \times G$ is a sub-block of one shared table ($G = 30$, the 480×480 bound at patch 16); only the sub-block for the current input is read, and a square input reproduces the original indexing exactly (App. D).

Resolution study: how the buckets are built. Both jet images (100×100) and CIFAR-100 (32×32) are fixed-size, so we test on Imagewoof, a ten-class ImageNet subset of dog breeds, at full native resolution. The two arms are matched at ~ 5.5M parameters, trained 90 epochs on three seeds under the protocol of Section 3. We histogram the training set's aspect ratio and bin it into five patch-16-aligned buckets (Table 4) whose token counts all fall in a narrow 576–600 band, so compute is matched by token count rather than image area. Each image is resized to its nearest bucket with *no crop*; a custom batch sampler keeps each batch single-bucket, and all five shapes share one positional table (Eq. 8). ViT-384, the fixed-grid baseline, centre-crops to 384×384=576 tokens, the bottom of the same band.

The fixed-square arm discards an average of 44% of each image and upscales the rest by an average of $1.20\times$; the bucketed arm keeps essentially all of it.

Operating at native aspect ratio gives a 1.50-point top-1 advantage (Table 5) and 0.0045 macro AUC while DepthViT2 is the *smaller* model by 0.86%, so the win cannot be read as extra capacity. Four qualifications keep this honest: the ViT arm's seed spread is $\sim 4\times$ DepthViT2's; Top-5 is essentially tied, marginally favouring ViT; the per-class picture softens under replication, seed 42 alone showing $8/10$ but seeds 123/1234 each $5/10$ (pooled $18/30$, App. E); and DepthViT2's training accuracy is

Table 4: Left: the five aspect buckets, patch-16 aligned, with grid and token count. Right: the preprocessing tax, as means over the training set. *Retained area* is the fraction of the original image surviving to the model; *scale*> 1 means upscaled, interpolated pixels.

Bucket	$H \times W$	Grid $R \times S$	Tokens
landscape (3:2)	480×320	30×20	600
landscape (4:3)	448×336	28×21	588
square (1:1)	384×384	24×24	576
portrait (3:4)	336×448	21×28	588
portrait (2:3)	320×480	20×30	600

Arm	Retained area	Scale
DepthViT2 (bucketed)	$\sim 100\%$	$0.906\times$
ViT-384 (fixed crop)	55.8%	$1.20\times$

Table 5: Imagewoof resolution study, ~ 5.5M tier, 90 epochs, three seeds, mean $\pm$ s.d. The bucketed arm processes each image at native aspect ratio with no crop; the ViT arm pays the fixed-square crop/upscale tax.

Model	Params	Top-1	Macro AUC	Top-5
DepthViT2 (bucketed, k=62)	5,551,220	$\mathbf{57.40 \pm 0.29}\%$	$\mathbf{0.9093 \pm 0.0014}$	$93.40 \pm 0.14\%$
ViT-384 (fixed, D=192)	5,599,306	$55.90 \pm 1.14\%$	0.9048 ± 0.0031	$\mathbf{93.87 \pm 0.29}\%$

far higher than ViT's (72.9% against 56.8%), consistent with more usable signal per image but confounded by an augmentation asymmetry, since the ViT arm's random-resized-crop supplies implicit augmentation the bucket pipeline does not. Beyond the numbers, one trained DepthViT2 accepts five input shapes and benefits from native-resolution processing, which a fixed-grid ViT cannot do; these runs also reproduce the systems signature of Section 6 (~ 3.5–$4\times$ slower).

6 Executing the mechanism: wall-clock cost and a fused kernel

At 22M, FLOP parity does not translate into wall-clock parity: once channel attention holds a fifth of the parameters it also dominates runtime, for reasons of operator *shape* rather than arithmetic volume. DepthViT2-22M trained at ~ 13.2 min/epoch against ViT-Small's 3.3, with GPU utilisation 91–95%, ruling out data-loading stalls. Channel attention operates over only C=2 channels, independently at every location and each of k=196 heads ($\sim 19{,}600$ tiny 2×2 products per block across 18 blocks), executed as an unfused `matmul`→`softmax`→`einsum` that never dispatches to `F.scaled_dot_product_attention`, since fused kernels are built for large-N token attention, precisely the shape channel attention lacks. High utilisation here means enormous numbers of tiny, memory-bandwidth-bound operations. Component attribution (Table 10, App. F) quantifies this: in eager mode channel attention consumes 83.9% of wall-clock for 26.0% of the arithmetic, and compiling moves that ratio only from 3.23 to 3.27, evidence that the disproportion is structural rather than an artifact of the `torch.compile` path. The symmetric control cross-checks it, since `symmix` does $\sim 37\%$ *more* channel-module arithmetic yet trains faster (~ 9 hours against $\sim$ 13+) by dispatching as a few larger batchable operations. We therefore implemented channel attention as a fused Triton kernel specialized to C=2: queries are $(C, 1)$ and keys $(1, C)$, so the score matrix is a rank-one outer product, the 2×2 attention collapses to elementwise operations over a tile of k heads, and one launch replaces the three-op sequence with no intermediate in global memory. The eager path runs at 2.6 GB/s against a $\sim 1{,}555$ GB/s device peak, direct evidence for the memory-bandwidth-bound diagnosis, while the fused kernel reaches 183.0 GB/s: a $\mathbf{71}\times$ speedup where `torch.compile` reaches $2.4\times$, so the penalty is implementation-shaped, not intrinsic to the arithmetic. The kernel is validated on the *isolated* operator, is not integrated into the training path, and no accuracy number here was produced with it.

7 Limitations

Isolating channel asymmetry. The matched baselines omit HAP and the head: asymmetry *without* spatial support is structurally incomplete rather than a meaningful control, and the symmetric control that qualifies this is evaluated at only one tier. **Fixed feed-forward width.** mlp_dim=768 is pinned across tiers and load-bearing for the allocation account, so every parameter-share claim is specific to

this scaling rule; a width-coupled schedule is the cleanest untested route to falsifying it. **Replication and scope.** The 1M, 5.4M and 22M comparisons use three seeds per arm, the 164K tier is single-seed, and the ladder is evaluated on one benchmark, above 164K only on the 150p split; CIFAR-100 and Imagewoof extend to two domains but neither reproduces the full crossover. **Wall-clock.** The $4\times$ gap is diagnosed and a remedy demonstrated on the isolated operator, but the fused kernel is not integrated into training.

8 Conclusion

The original DepthViT targeted a narrow problem, anomaly detection in single-pixel readouts, where attending purely along depth was the right bias (Authors, 2026). The tasks here are broader, involving several sensors *and* long-range spatial relationships, which a depth-only model cannot represent; the upgrades studied here (HAP, a structure-preserving head, and the scale-adjusted positional encoding) carry the same idea into that class, gaining an encoder linear rather than quadratic in patch count and one set of weights that accepts many input shapes at native resolution. At equal parameter budget the answer depends on the budget: the comparison changes sign across the ladder, trailing by 1.56 points at 164K, tying at 1M, and leading by 1.69 and 1.36 at 5.4M and 22M, matched on parameters *and* FLOPs. This traces to allocation: the mechanism grows quadratically in k against a linear rest, holding 1% of the budget where it loses and over 20% where it wins, and a same-scaffold control shows a symmetric operator matches accuracy, leaving asymmetry the cheapest rather than the most accurate configuration. The advantage persists outside calorimetry ($+9.84$ on CIFAR-100), resolution independence adds 1.50 points on Imagewoof, and the $4\times$ wall-clock gap is real but not intrinsic. More broadly, a defining operator does not by itself characterise an inductive bias, and such asymmetries should be evaluated at budgets where they are actually funded.

AI use statement

Generative AI tools (Claude, Anthropic) were used in this project's research-engineering workflow and in drafting this manuscript. Claude Code was used for experiment setup (SLURM scripts, trainer/config scaffolding, data-loading pipelines including the bucketed loader of Section 5, and smoke tests); Claude (chat) drafted prose whose content and framing were specified by the authors and produced early code for auxiliary analysis scripts. We did *not* use generative AI to generate experimental results, write core scientific claims, or select which results to report: all numbers came from the authors' own runs, and every accuracy, AUC, parameter, FLOP and timing figure was independently verified by instantiating the model and counting tensors. We reviewed all AI-assisted code and prose before inclusion and take responsibility for the final content of this work.

Ethics statement

This work complies with the ICLR Code of Ethics and uses only public simulated or standard research data: the HLS4ML LHC jet dataset (Pierini et al., 2020c;d;a;b), ImageNet-1K/Imagewoof, and CIFAR-100. No human subjects or sensitive attributes are involved; the jet data are Monte Carlo simulation with no real-world observations of people. We release no model or dataset with appreciable misuse potential and declare no conflicts of interest. The central claim includes a negative result at small scale: the architecture *loses* at the smallest budget tested, reported as such rather than restricted to budgets where it wins.

Reproducibility statement

Every quantitative claim is traceable to a specific artefact. Parameter/FLOP counts (Table 2) come from counting instantiated modules and `calflops`, not estimation; the training protocol (Section 3) is identical across every comparison, with per-split statistics in App. A. The closed-form parameter model (Eq. 6) reproduces every count exactly. Symmetric-control and positional-indexing details are in App. D–D, per-seed/per-class results in App. B and App. E, and systems methodology in App. F. The supplementary material includes an anonymised repository ZIP with source code, configurations, SLURM scripts, evaluation outputs and systems benchmarks.

REFERENCES

Anonymous Authors. Continual learning via ensemble-based depth-wise masked autoencoders for data quality monitoring in high-energy physics. *arXiv preprint*, 2026.

Jai Bardhan, Radhikesh Agrawal, Abhiram Tilak, Cyrin Neeraj, and Subhadip Mitra. Hep-jepa: A foundation model for collider physics using joint embedding predictive architecture. *arXiv preprint arXiv:2502.03933*, 2025. URL https://arxiv.org/abs/2502.03933.

CMS Collaboration. The cms experiment at the cern lhc. *Journal of Instrumentation*, 3(08): S08004, 2008. doi: 10.1088/1748-0221/3/08/S08004. URL https://doi.org/10.1088/1748-0221/3/08/S08004.

Evan Coleman, Marat Freytsis, Andreas Hinzmann, Meenakshi Narain, Jesse Thaler, Nhan Tran, and Caterina Vernieri. The importance of calorimetry for highly-boosted jet substructure. *Journal of Instrumentation*, 13(01):T01003, 2018. doi: 10.1088/1748-0221/13/01/T01003. URL https://arxiv.org/abs/1709.08705.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.11929.

Javier Duarte, Song Han, Philip Harris, Sergo Jindariani, Edward Kreinar, Benjamin Kreis, Jennifer Ngadiuba, Maurizio Pierini, Ryan Rivera, Nhan Tran, and Zhenbin Wu. Fast inference of deep neural networks in fpgas for particle physics. *Journal of Instrumentation*, 13(07):P07027, 2018. doi: 10.1088/1748-0221/13/07/P07027. URL https://arxiv.org/abs/1804.06913.

Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. XCiT: Cross-covariance image transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021. URL https://arxiv.org/abs/2106.09681.

Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12259–12269, 2021. URL https://arxiv.org/abs/2104.01136.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html.

Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 12934–12949, 2022. URL https://arxiv.org/abs/2206.01191.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021. URL https://arxiv.org/abs/2103.14030.

Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2110.02178.

Vinicius Mikuni and Florencia Canelli. Point cloud transformers applied to collider physics. *Machine Learning: Science and Technology*, 2(3):035027, 2021. doi: 10.1088/2632-2153/ac07f6.

Eric A. Moreno, Olmo Cerri, Javier M. Duarte, Harvey B. Newman, Thong Q. Nguyen, Avikar Periwal, Maurizio Pierini, Aidana Serikova, Maria Spiropulu, and Jean-Roch Vlimant. JEDI-net: a jet identification algorithm based on interaction networks. *European Physical Journal C*, 80(1): 58, 2020. doi: 10.1140/epjc/s10052-020-7608-4. URL https://arxiv.org/abs/1908.05318.

Maurizio Pierini, Javier Mauricio Duarte, Nhan Tran, and Marat Freytsis. HLS4ML LHC jet dataset (100 particles), January 2020a. URL https://doi.org/10.5281/zenodo.3602254.

Maurizio Pierini, Javier Mauricio Duarte, Nhan Tran, and Marat Freytsis. HLS4ML LHC jet dataset (150 particles), January 2020b. URL https://doi.org/10.5281/zenodo.3602260.

Maurizio Pierini, Javier Mauricio Duarte, Nhan Tran, and Marat Freytsis. HLS4ML LHC jet dataset (30 particles), January 2020c. URL https://doi.org/10.5281/zenodo.3601436.

Maurizio Pierini, Javier Mauricio Duarte, Nhan Tran, and Marat Freytsis. HLS4ML LHC jet dataset (50 particles), January 2020d. URL https://doi.org/10.5281/zenodo.3601443.

Huilin Qu and Loukas Gouskos. Particlenet: Jet tagging via particle clouds. *Physical Review D*, 101 (5):056019, 2020. doi: 10.1103/PhysRevD.101.056019. URL https://arxiv.org/abs/1902.08570.

Huilin Qu, Congqiao Li, and Sitian Qian. Particle transformer for jet tagging. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18281–18292, 2022. URL https://arxiv.org/abs/2202.03772.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. URL https://arxiv.org/abs/2006.04768.

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018. URL https://arxiv.org/abs/1807.06521.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5728–5739, 2022. doi: 10.1109/CVPR52688.2022.00564. URL https://arxiv.org/abs/2111.09881.

A Data processing and per-split statistics

The HLS4ML jet files are HDF5 shards of $10{,}000$ samples. We use `jetImageECAL` and `jetImageHCAL` as the two input channels; a third key, `jetImage`, is bitwise identical to `jetImageHCAL` (max abs. difference 0.0 across the training set), so treating the data as three-channel would feed a duplicated channel. Labels come from the one-hot block at columns 53–57. The data originate from simulated $\sqrt{s} = 13$ TeV proton-proton collisions with the initiating parton or boson at $p_T \approx 1$ TeV; jets are reconstructed with anti-k_T at $R = 0.8$ and energies smeared to approximate CMS-like resolution rather than produced with a full detector simulation (Coleman et al., 2018). Images are $\sim 99.8\%$ (HCAL) and 99.9999% (ECAL) zero and heavy-tailed (max ~ 1200), so we apply $\log(1 + x)$ before z-scoring each channel per split. After normalisation HCAL has s.d. close to 1.0; ECAL retains large maximum z-scores (~ 390) because its s.d. is small (~ 0.008) when almost every pixel is zero, a consequence of sparsity re-normalised internally by each block's layer norm.

B Per-class and per-seed results

The 5.4M seed-42 pair is measured from the training log; seeds 123/1234 were evaluated through `eval_roc_flops.py`, and the two paths agree to hundredths at the other tiers.

Table 6: Per-split channel statistics after the $\log(1 + x)$ transform.

Split	Train samples	ECAL mean	ECAL std	HCAL mean	HCAL std
30p	610,000	1.801×10^{-5}	0.00832015	0.00541123	0.13826324
50p	610,000	1.923×10^{-5}	0.00844423	0.00672488	0.14590938
100p	640,000	1.958×10^{-5}	0.00847558	0.00728792	0.14803880
150p	620,000	1.954×10^{-5}	0.00846553	0.00729897	0.14805554

Table 7: Left: per-class AUC at 150p (g = gluon, q = light quark). Right: individual seed results, 90-epoch protocol; seeds differ only in the random seed and output paths.

Tier	Model	g	q	W	Z	top
164K	DViT (k=4)	.9198	.9044	.9145	.9019	.9470
1M	DViT (k=23)	.9275	.9114	.9302	.9151	.9507
5.4M	DViT (k=90)	.9318	.9181	.9446	.9311	.9547
5.4M	ViT-Tiny	.9276	.9137	.9357	.9216	.9506
22M	DepthViT2	**.9323**	**.9180**	**.9449**	**.9324**	**.9553**
22M	ViT-Small	.9305	.9152	.9394	.9264	.9515

Tier	Seed	Top-1	AUC
1M DViT	42	72.31%	.9270
	123	72.00%	.9290
	1234	72.80%	.9252
1M ViT-T	42	72.30%	.9260
	123	72.50%	.9263
	1234	72.11%	.9253
5.4M DViT	42	74.99%	.9361
	123	75.24%	.9376
	1234	75.14%	.9364
22M DViT	42	75.12%	.9366
	123	75.50%	.9381
	1234	75.40%	.9375
22M ViT-S	42	74.14%	.9326
	123	73.94%	.9315
	1234	73.86%	.9316

C CROSS-SPLIT BEHAVIOUR AND DEPTH SWEEP

A ResNet-9 matched to 164,671 parameters (within 0.15% of DepthViT2's 164,417) reaches 60.40% top-1 under the identical protocol, 9.74 points below DepthViT2 and 11.30 below matched ViT-Tiny, so both attention architectures substantially outperform convolution at this budget, though this speaks only to the attention-versus-convolution axis. Across multiplicity splits the 164K model's gap to *native* ViT-Tiny narrows monotonically from 9.63 points at 30p to 3.16 at 150p.

Table 8: DepthViT2 (k=4) against *native* ViT-Tiny (5,397,893 params) across all four splits, 90 epochs. **Unequal-budget**, reported only to characterise substructure sensitivity.

	Top-1 accuracy		Macro AUC	
Split	DepthViT2	ViT-Tiny	DepthViT2	ViT-Tiny
30p	58.43%	68.06%	0.8549	0.9029
50p	64.01%	69.37%	0.8896	0.9129
100p	68.62%	72.80%	0.9109	0.9286
150p	70.14%	73.30%	0.9175	0.9298

The macro AUC gap narrows from 0.048 to 0.012. Training to 90 epochs is necessary because splits converge at different rates: extending from 30 epochs leaves 50p and 100p nearly unchanged (+1.0/+0.9) but improves 150p by 7.9 points. A depth sweep at fixed 22M budget with four HAP blocks confirms channel-attention's share stays flat at 21–22% regardless of depth, with depth trading HAP against the feed-forward block and not against the mechanism; $L = 18$ is the shallowest depth pulling HAP below 30%. For external context, particle-constituent methods report higher discrimination on this benchmark (per-class AUC up to ~ 0.98 for a point-cloud transformer (Mikuni & Canelli, 2021)) using four-vectors rather than calorimeter images; our macro AUC range (0.9175–0.9374) sits below that while operating on the image representation, which is orthogonal to this paper's controlled matched-budget question.

D SYMMETRIC-CONTROL CONSTRUCTION AND POSITIONAL INDEXING

Both control arms are post-construction module swaps, leaving the rest of the model and the training loop untouched. `chansum` rebuilds `conv_proj` with `groups=1`; the two-way channel axis survives but stops carrying detector identity. `symmix` replaces `CrossDepthMultiheadSelfAttention` with a permutation-equivariant `ChannelSymmetricMixing`: it keeps the tensor shapes $(C, k, 3k)$ and $(C, 3k)$ and reinterprets them, row 0 as a self map and row 1 as a map on the channel-pooled mean plus a gain. A permutation-equivariant linear map on the channel axis has exactly two free matrices, and $C{=}2$, so the budget matches to the integer; the module raises for $C \neq 2$ by design. Because both modules share identical state-dict key names and shapes, a mismatched checkpoint would load into the wrong architecture with zero missing or unexpected keys; we add a persistent `_symmetric_marker` buffer as a tripwire and verify with a 3×3 cross-load matrix that all six mismatched loads are rejected and all three matched loads succeed. Gradient coverage was confirmed so the module is DDP-safe with no unused parameters.

The resolution study of Section 5 requires the positional scheme of Eq. (8), consistent across grids of different width and height. The original indexing, Eq. (7), placed each token by its rank within the current image's own grid, which silently scrambles vertical adjacency when images of different widths are compared: token i sits under a different spatial neighbour depending on the row stride. We add `_make_pos_idx_2d`, indexing every token into a fixed maximum grid ($G = 30$, the 480×480 patch-16 bound), so a spatial location maps to the same positional entry regardless of image shape. It is gated behind a `pos_embed_2d` flag defaulting to `False`, so every fixed-square result in this paper is bit-identical to the original indexing. Two DDP fixes were required: the custom bucket batch sampler owns rank-splitting, so Lightning's default `DistributedSampler` is disabled for that arm, and a redundant manual sampler in the fixed-resolution loader that double-sharded with Lightning's wrapper was removed.

E IMAGEWOOF PER-CLASS RESULTS

Table 9: Per-class validation AUC, seeds 123 and 1234. Bold marks the winning arm per cell. DepthViT2 wins 10 of the 20 cells; adding seed 42's $8/10$ gives the pooled $18/30$ quoted in Table 5.

	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9
seed 123 (DepthViT2 macro .9102, ViT .9073)										
DepthViT2	**.8927**	.9366	.9004	.9015	**.9105**	.8989	**.8909**	**.9168**	**.9662**	.8879
ViT-384	.8752	**.9377**	**.9112**	**.9047**	.9035	**.9007**	.8676	.9059	.9583	**.9085**
seed 1234 (DepthViT2 macro .9073, ViT .9067)										
DepthViT2	**.8910**	.9383	.9028	**.9070**	.8987	.8871	**.8750**	**.9201**	**.9632**	.8902
ViT-384	.8805	**.9411**	**.9098**	.9018	**.9036**	**.8923**	.8734	.9055	.9540	**.9055**

The DepthViT2 edge concentrates in the macro average and the harder classes rather than being uniform across every breed, which is why the honest headline is a Top-1 and macro-AUC win with tight variance, not an every-class domination. The preprocessing tax quantified in Table 4 is computed as a mean over the training set, per image, before any augmentation.

F SYSTEMS MEASUREMENT METHODOLOGY

Two standalone scripts produce Table 10; neither requires the dataset or a trained checkpoint, as inputs are synthetic tensors of the training shape. The first times each component class in isolation, forward and forward-plus-backward, multiplies by block count (18, 18, 4), computes analytic forward FLOPs, and reports share of wall-clock, share of FLOPs and their ratio. It additionally builds the full tier-4 model and asserts the parameter count equals 22,046,777 before proceeding (a guard against silent configuration drift), then wraps three optimisation steps in `torch.profiler`, reporting top operations by self CUDA time and CUDA kernel invocations per step, which quantifies launch overhead directly. All measurements are on a single A100-SXM4-40GB in bf16 at batch 32, median of repeated trials. The second benchmarks the fused Triton implementation of Eq. (3) against the unfused reference, running a parity gate first: both are evaluated on identical fp32 inputs

Table 10: Left: wall-clock versus FLOP attribution by component, tier-4 (L=18, k=196), A100-SXM4-40GB, bf16, batch 32; ratio is share of wall-clock over share of FLOPs. Right: fused-kernel benchmark on the isolated operator, forward only, shape $(32, 100, 2, 196)$; parity verified before timing (fp32 max abs. error 4.8×10^{-7}).

	Eager		Compiled		
Component	Time	Ratio	Time	FLOPs	Ratio
Chan.-attn	83.9%	**3.23**×	85.1%	26.0%	**3.27**×
Feed-fwd	6.3%	0.12×	10.4%	50.7%	0.21×
HAP	9.8%	0.42×	4.5%	23.3%	0.19×

Implementation	Time/op	Bandwidth
Eager (unfused)	3.895 ms	2.6 GB/s
`torch.compile`	1.626 ms	N/A
Triton (fused)	**0.055** ms	183.0 GB/s
A100 HBM2e peak	N/A	$\sim$ 1,555 GB/s

and if relative error exceeds 10^{-5} the script exits without printing any timing, so a numerically incorrect kernel cannot yield a reported speedup. Parity is re-checked in bf16 so dtype-induced error is visible separately. Timing is the median of 100 iterations per path, and the script computes minimum required memory traffic (three reads, one write) and achieved bandwidth, the direct evidence for the memory-bandwidth-bound diagnosis. Extrapolated across 18 blocks the operator cost falls from 70.1 ms to 0.99 ms per forward pass.