A Smaller Transformer in Your Transformer

Dhananjay Tomar[1,2,*]
dhananjt@ifi.uio.no
Marius Aasan[2,3,*]
mariuaas@uio.no
Andreas Kleppe[1,2,3]
andrekle@ifi.uio.no
Adín Ramírez Rivera[2,3]
adinr@uio.no

[1] Institute for Cancer Genetics and Informatics
Oslo University Hospital
Oslo, Norway

[2] Department of Informatics
University of Oslo
Oslo, Norway

[3] SFI Visual Intelligence
UiT The Arctic University of Norway
Tromsø, Norway

Abstract

Recent findings indicate that Vision Transformers settle into locally similar computational phases, implying a level of depthwise computational redundancy. However, existing methods to exploit this redundancy either fail to reduce inference compute or severely degrade model expressivity. In this work, we formalise a unified view of block redundancy that decouples the geometry from specific surrogate interventions. We then introduce Transformer-Within-Transformer (TWT), a post-hoc method that fuses contiguous groups of redundant layers into a single learned surrogate layer. TWT reduces parameter count and inference compute while remaining competitive with original models using half the depth on natural images, and in several downstream histopathology settings, TWT matches or even improves on the original baseline.

1 Introduction

Vision Transformers (ViTs) [27, 33, 37] have a particular architectural austerity; after the initial token embedding, each layer acts on the same token space, updating the representation through residual attention and feed-forward networks. Beyond its elegance [30], this architectural homogeneity exposes a central mode of investigation: how do layers organise to perform distinct computational tasks?

Recent discoveries [9, 15] show that ViTs tend to settle into depthwise-contiguous computational phases with a high degree of inter-layer similarity, implying a form of *computational redundancy*. While concurrent research agrees on the identification of the symptom, the methods used to exploit local depthwise redundancy vary from local recurrence [15], which lowers parameter counts but maintains the overall compute, to non-mixing approximations [9, 19]—which reduce compute but incur a more prominent drop in performance.

Code available at: `https://github.com/dsb-ifi/TWT`.
*Equal contribution.

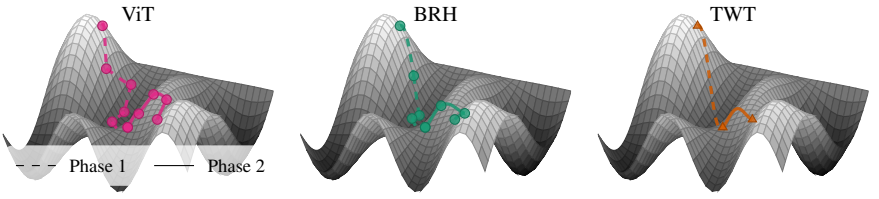


Figure 1: Schematic evolution of block approximations in representation space. Dashed and solid paths indicate two contiguous computational phases. (Left) A standard ViT follows jagged, uneven steps through the phases encoded by its successive blocks. (Middle) The block-recurrent approximation replaces each phase by repeated applications of a shared block, producing smoother, regular steps. (Right) Our TWT fuses each phase into a single learned surrogate transformation, bypassing the intermediate block trajectory while preserving its endpoint.

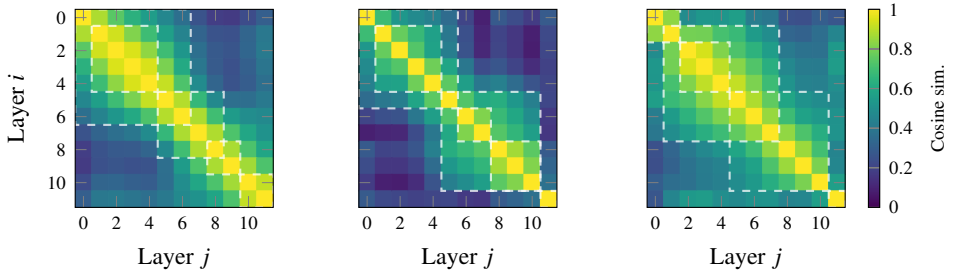
Our work focuses on providing a more unified view of the phenomenon of *block redundancy* in ViTs and proposes a method that better exploits redundancy with minimal performance loss. Our contribution is threefold. (i) We provide a formalisation of block redundancy in ViTs where recurrence and linear approximation appear as two sides of the same coin: both approximate local phases of computation whose constituent layers exhibit high functional similarity. (ii) We apply this paradigm with *Transformer-Within-Transformer* (TWT), a post-hoc method that fuses redundant blocks into a single surrogate layer, preserving the core ViT architecture. (iii) We show that TWT yields parameter reductions comparable to recurrent surrogates [15] while reducing executed inference computation, without incurring the performance degradation of simple linear surrogates. Notably, in histopathology foundation models, TWT improves downstream performance relative to the original model, and can even outperform larger baselines.

1.1 Note on Terminology and Nomenclature

The term *block* is used somewhat indiscriminately, denoting both the elementary computational unit of a ViT and, in discussions of depthwise redundancy, a contiguous group of such units. This quickly devolves into confusing references to *blocks of blocks*. To avoid ontological gymnastics, we elect to call the elementary ViT unit a *layer*: a multi-head attention operator and a feed-forward network. We reserve *block* to mean a contiguous sequence of layers, usually grouped by a homogeneity criterion, such as cosine similarity.

2 Block Redundancy in Vision Transformers

To formalise the procedural representational flow in ViTs, we distinguish between theoretical *phases* of computation and the empirical *blocks* of layers learned by a network. A *phase* is an ideal, theoretical stage of computation that executes a specific set of operations on the input. In practice, deep neural networks learn an approximation of these phases by distributing the computation across several learned layers. Evidence of these underlying phases emerges as *block patterns* in the network—contiguous groups of layers whose outputs exhibit high functional similarity. Previous work [4, 18] investigates the representational flow through these blocks, observing that updates tend to periodically decelerate, and some claim this implies an underlying recurrent program [15]. Our objective is to understand this



(a) DeiT-III B/16 **(b)** DINOv2 B/14 **(c)** DINOv3 B/16

Figure 2: Pairwise cosine similarity between block outputs from ImageNet-1k reveals consistent block-diagonal structure across architectures and training paradigms, both for self-supervised (DINOv2, DINOv3) and supervised (DeiT-III) models. Contiguous groups of blocks produce redundant representations, which can be leveraged to produce more effective models. Possible blocks are shown in white.

representational flow by characterising and exploiting inherent redundancies in empirical block structures that capture distinct computational phases of existing models.

We posit that the contiguous blocks observed in ViTs are geometrically redundant pathways navigating a noisy manifold, and their iterative behaviour is an artefact of the optimisation landscape rather than a strict computational necessity (cf. Fig. 1). Consequently, these multi-step iterative blocks can be approximated by surrogate models that reflect these ideal phases. Our claim is that *the particular choice of surrogate layer* is the distinguishing factor between different approaches, and is the target of our formalisation of block redundancy.

ViT operators. At its core, a transformer F_Θ can be decomposed as a set of transformations

$$F_\Theta = f_{\theta_L} \circ \cdots \circ f_{\theta_1}, \tag{1}$$

where $f_{\theta_\ell} = f(\cdot;\theta_\ell)$ denotes a unique parametrisation for each layer ℓ, and the full model parametrisation is given by $F_\Theta = F(\cdot;\Theta)$ with $\Theta = (\theta_1,\ldots,\theta_L)$. Alternatively, Eq. (1) can instead be formulated as contiguous block segments

$$F_\Theta = F_{\Theta_m} \circ \cdots \circ F_{\Theta_1}, \tag{2}$$

where each subnetwork F_{Θ_j} for $1 \le j \le m$ contains layers with similar internal dynamics, i.e., some form of *block redundancy*. We begin by separating the phenomenon from the methods used to exploit it.

Block redundancy. Let $B_j = [s_j, e_j]$ be a contiguous block of length $k_j = e_j - s_j + 1$, where $1 \le s_j < e_j \le L$. We say that block B_j is ε-functionally-redundant if the layers within it implement highly similar transformations on the states visited by the model, such that

$$\max_{a,b \in B_j} \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left[d\Big(f_{\theta_a}\big(h_{a-1}(x)\big), f_{\theta_b}\big(h_{b-1}(x)\big) \Big) \right] \le \varepsilon, \tag{3}$$

where $h_\ell(x) = (f_{\theta_\ell} \circ \cdots \circ f_{\theta_1})(x)$ denotes the intermediate hidden state after layer ℓ, d is a representation-space discrepancy measure, and $\mathcal{D}$ is a data distribution. In simple terms, a phase is block-redundant when all layers behave similarly across depth. Notably, Jacobs

et al. [15] provide an analysis under $d(x,y) = 1 - \cos(x,y)$. Empirical observations of the layer-to-layer distance similarity matrix

$$S_{a,b} = \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[d\Big(f_{\theta_a}\big(h_{a-1}(x)\big), f_{\theta_b}\big(h_{b-1}(x)\big)\Big)\right] \tag{4}$$

reveal distinct block-diagonal structures. These structures define *blocks* B_j which approximate the ideal *phases* of computation, where the intra-block representational similarity remains exceptionally high, as shown in Fig. 2. This implies that F_Θ slowly integrates information over time to reach the final representations.

From redundancy to surrogate models. Once a block has been identified as functionally redundant, we can form a stronger and more constructive query. Can an entire block be approximated by a simpler surrogate? Let

$$F_{\Theta_j} = f_{\theta_{e_j}} \circ \cdots \circ f_{\theta_{s_j}} \tag{5}$$

denote the subnetwork implemented in block B_j with parameters $\Theta_j = (\theta_{s_j}, \ldots \theta_{e_j})$. We then seek a *surrogate mapping* g_j such that

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[d\Big(F_{\Theta_j}\big(h_{s_j-1}(x)\big), g_j\big(h_{s_j-1}(x)\big)\Big)\right] \leq \varepsilon. \tag{6}$$

The key distinction from previous approaches [2, 15] and our proposed TWT lies not in whether such a surrogate exists, but rather on the underlying assumptions motivating the choice of g_j.

2.1 Recurrent and Non-Mixing Surrogates

The Block Recurrent Hypothesis. Jacobs et al. [15] operationalise block redundancy through a specific structural choice for g_j, and asks whether the surrogate can be chosen to be a repeated application of a single parameter-tied operator. Under this view, one writes

$$g_j = f_{\theta_j^*}^{k_j} = \underbrace{f_{\theta_j^*} \circ \cdots \circ f_{\theta_j^*}}_{k_j \text{ times}} = f^{k_j}(\cdot;\theta_j^*), \tag{7}$$

such that the same parametrisation θ_j^* is reused k_j times inside block j. Their proposed Raptor method optimises the parametrisation θ_j^* via

$$\theta_j^* = \arg\min_{\phi} \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \left\| F_{\Theta_j}\big(h_{s_j-1}(x)\big) - f_\phi^{k_j}\big(h_{s_j-1}(x)\big)\right\|_2^2, \tag{8}$$

which fine-tunes an existing model via a recurrent objective, noting that Raptor performs this optimisation in two separate training stages.

While Raptor demonstrates that contiguous phases can be approximated by shared-weight layers, we argue that recurrence is an overly specific interpretation of a broader geometric phenomenon. In a residual architecture, small functional updates along a locally flat computational phase naturally induce block-diagonal similarity. A sequence of independent layers traversing such a phase can therefore yield a similar signature as a recurrent operator, without implementing repeated dynamics or requiring identical weights. From this perspective, recurrence is a useful parameter-sharing regulariser imposed on redundant phases, but not the fundamental mechanism generating them. And because it still executes all k_j transformations, computations at inference remain largely intact.

Non-mixing Surrogates. TOAST [2] takes the opposite approach to BRH. Rather than imposing recurrence, it bypasses redundant phases entirely by fitting a closed-form linear map between their endpoints. Similarly, NOSE [19] replaces specific ViT layers exclusively with feed-forward networks, adding more expressivity at the cost of more compute. While both reduce parameters and raw compute without retraining, this strategy also limits the expressivity of the surrogate g_j.

A key property of attention operators is their capability for *dynamic token-mixing*, which provides a high degree of expressivity [5, 28]. A linear map can somewhat preserve proximity, but it cannot meaningfully reproduce token interactions between layers in identified blocks. While TOAST and NOSE demonstrate that entire phases can be fused using strictly less-expressive operators, they produce models that cannot retain the expressivity of their original components.

2.2 Transformer-Within-Transformer

TWT looks to resolve the tension between surrogate expressivity and inference efficiency. We hypothesise that if a contiguous block $B_j = [s_j, e_j]$ approximates a single ideal computational phase, the intermediate representations, $h_{s_j}(x), \ldots, h_{e_j-1}(x)$, are not strict necessities, but transitional micro-steps across a locally flat manifold. Therefore, the surrogate g_j can be chosen as a single, non-recurrent operator f_{ϕ_j} that computes the phase directly via

$$h_{e_j}(x) \approx f_{\phi_j}\big(h_{s_j-1}(x)\big). \tag{9}$$

Crucially, to retain dynamic token-mixing, we restrict f_{ϕ_j} to the standard architecture of a ViT layer: a multi-head attention mechanism and a feed-forward network. In replacing a redundant block with exactly one layer, TWT fuses the computational phase into a single step, circumventing the iterative inference cost retained by BRH [15].

This formalisation fundamentally alters the algorithmic interpretation of ViT depth. It implies that deep networks do not strictly require k_j distinct attention passes to incrementally route information within a phase. Instead, a single, optimally parameterised attention and feed-forward pass possesses sufficient representational capacity to execute the entire spatial and channel-wise reorientation required for that ideal phase.

3 Transformer-Within-Transformer Surrogate Discovery

To operationalise block redundancy, we propose a pruning and distillation framework that condenses contiguous blocks of learned operators into a single algorithmic step reflecting the ideal phase. Unlike the BRH, which requires a surrogate operator $g_j = f_{\theta_j^*}^{k_j}$ to iteratively unroll k_j times to mimic a target block B_j, our method approximates a block's computational phase with a single application of f_{ϕ_j} without recurrent iteration, as formalised in Eq. (9). We achieve this through a three-step process: (1) dynamic block discovery, (2) optimal candidate initialisation, and (3) distillation via single-stage deep supervision. We depict this process and contrast it against the literature in Fig. 3.

3.1 Block Discovery via Max-Min Dynamic Programming

Our first objective is to identify contiguous subsets of layers that form ε-functionally-redundant blocks. Let F_Θ be a trained Vision Transformer with L layers. We compute

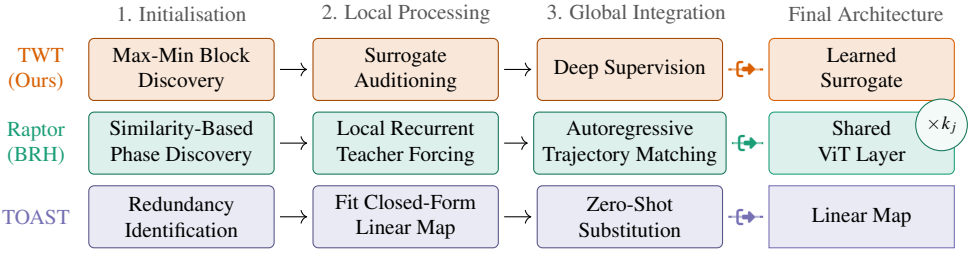


Figure 3: Pipeline and architectural comparison of phase compression methods. TWT collapses each phase into one learned ViT surrogate using auditioning and single-stage deep supervision. Raptor (BRH) [13] recurrently reuses a shared ViT layer and therefore retains the original inference cost, whereas TOAST [1] fits a closed-form linear map without dynamic token mixing.

the discrepancy matrix $S \in \mathbb{R}^{L\times L}$, where $S_{a,b}$ is the expected discrepancy d of the token representations $h_a(x)$ and $h_b(x)$ over a representative calibration set $\mathcal{D}$, as introduced in Section 2. In practice, we let d be the cosine distance between intermediate activations.

We frame block discovery as a partitioning problem over depth. We seek a set of contiguous blocks $\mathcal{P} = \{B_1, \ldots, B_m\}$ that cover the entire network, where each $B_j = [s_j, e_j]$ and $s_{j+1} = e_j + 1$. A valid partition must satisfy a maximum intra-block discrepancy threshold ε, ensuring that the boundary discrepancy satisfies $S_{s_j,e_j} \leq \varepsilon$ for all j.

To prevent catastrophic degradation, we employ a min-max dynamic programming approach. We first minimise the total number of blocks m, effectively maximising compression. To break ties among equally minimal partitions, we select the partition that minimises the worst-case discrepancy among its blocks. Formally, we optimise

$$\min_{\mathcal{P}} \left(m, \max_{j\in\{1\ldots m\}} S_{s_j,e_j} \right) \quad \text{s.t.} \quad S_{s_j,e_j} \leq \varepsilon, \forall j. \tag{10}$$

This yields a deterministic merge plan where each block B_j corresponds to a single ideal phase to be captured by a surrogate operator. We provide a sensitivity analysis in Supplementary Appendix D showing how varying the threshold ε controls the granularity of this partition.

3.2 Macro-Step Initialisation via Auditioning

Instead of initialising a surrogate operator f_{ϕ_j} randomly, we exploit the parameters the teacher has already learned. For a given block $B_j = [s_j, e_j]$, we generate a pool of candidate operators. This pool, denoted as $\mathcal{C}_j = \{f_{\theta_\ell} : \ell \in B_j\} \cup \{\bar{f}_j\}$, includes each individual layer from the teacher's block, as well as an averaged operator $\bar{f}_j$ constructed by averaging the weight matrices of all layers in the span (while keeping normalisation parameters isolated). We evaluate each candidate on a small calibration batch to minimise the local mapping error

$$f^*_{\phi_j} = \arg\min_{f\in\mathcal{C}_j} \mathbb{E}_{x\sim\mathcal{D}} \left[\left\| f(h_{s_j-1}(x)) - h_{e_j}(x) \right\|_2^2 \right]. \tag{11}$$

This *auditioning* process identifies the operator best naturally positioned to execute the macro-step, significantly stabilising early training dynamics.

We provide an ablation study in Supplementary Appendix C showing that the auditioned candidate generally outperforms the worst candidate, supporting the value of auditioning for downstream transfer.

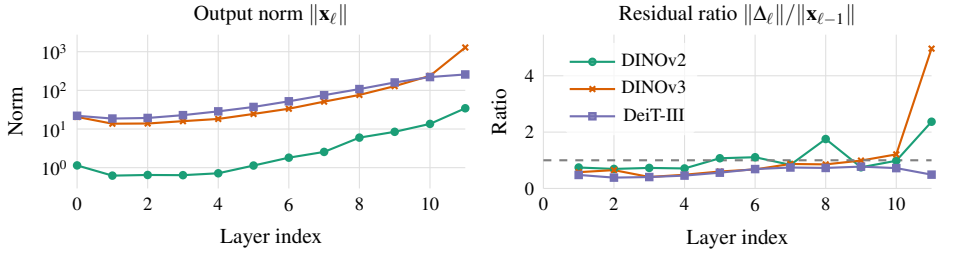


Figure 4: Norm growth across layers (patch tokens, averaged over 50k ImageNet samples). *Left:* output norms grow exponentially in later layers, with DINOv3 exhibiting a 63× increase from the first to the last layer. *Right:* the residual-to-output ratio exceeds 1 in late layers for the self-supervised models, meaning the residual addition is larger than the accumulated representation. This scale imbalance motivates introducing learnable LayerScale parameters to stabilise convergence during distillation.

3.3 Distillation via Deep Supervision

With the collapsed architecture initialised, we distil the teacher into the student without recurrent rollouts or linear surrogates. Instead of the dual-stage training procedure used in Raptor [15], we stitch the K surrogate blocks together into a unified student model F_Φ and optimise it end-to-end.

Let $\mathcal{B} = \{(s_j, e_j)\}_{j=1}^{K}$ denote the merge plan, where student block j replaces teacher blocks $s_j, \ldots, e_j$. For an input image x, let $h_j(x;\Phi)$ be the output of student block j, and let $h_{e_j}(x;\Theta)$ be the output of the last teacher block in the corresponding teacher segment. Let $F_\Phi(x)$ and $F_\Theta(x)$ denote the final pre-classification backbone features of the student and teacher, respectively. To gradually introduce deeper supervision terms, we use a staggered cosine schedule over normalised training time $\tau \in [0,1]$:

$$\omega_j(\tau) = \begin{cases} \frac{1-\cos\left(\pi\tau/\alpha_j\right)}{2}, & 0 \le \tau < \alpha_j, \\ 1, & \alpha_j \le \tau, \end{cases} \tag{12}$$

where $\alpha_j \in (0,1]$ is the activation time assigned to block j, with shallower blocks activated earlier and deeper blocks later. The training objective is

$$\mathcal{L}(x,\tau) = \sum_{j=1}^{K} \omega_j(\tau)\,\mathrm{MSE}\big(h_j(x;\Phi), h_{e_j}(x;\Theta)\big) + \omega_{K+1}(\tau)\,\mathrm{MSE}(F_\Phi(x), F_\Theta(x)). \tag{13}$$

Thus, the student is supervised both at intermediate block outputs and at the final representation, with deeper losses introduced progressively over training. All student parameters are optimised jointly using AdamW.

Layer Scaling. While most pre-trained models use LayerScale [35] during initial optimisation, these parameters are not always included in pre-trained checkpoints. Our experiments indicate that several pre-trained models have a marked increase in norms for later layers, particularly in natural image models, as seen in Fig. 4. To improve convergence, we explicitly introduce LayerScale, initialised as identity to allow the model to more easily adapt norms to dropped layers in block fusion. We find that a single scalar parameter typically suffices to improve convergence during fitting. These are fused with existing parameters in the final model weights.

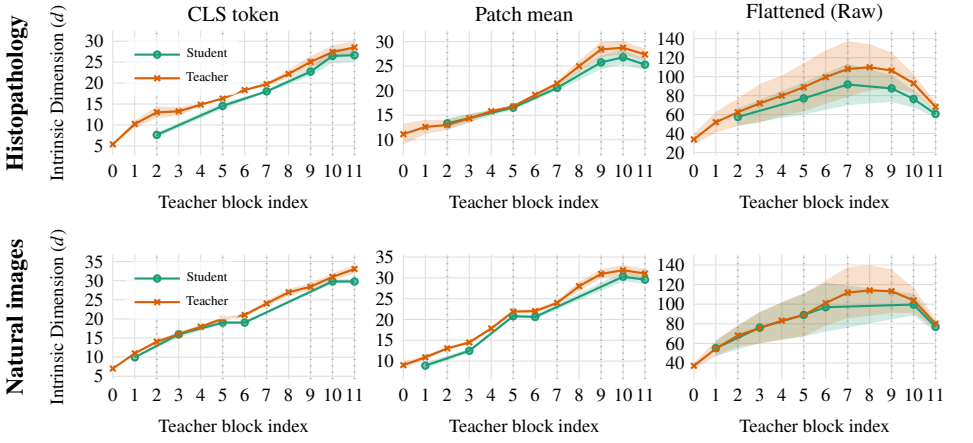


Figure 5: Intrinsic dimension (ID) trajectory comparison. ID (d) estimated via the Two-NN estimator across H0-mini (top) and DINOv2 (bottom). We compare the IDs of: the global class (CLS) token (left), the spatially averaged patch tokens (centre), and the raw, unpooled sequence of all tokens flattened into a single vector (right). In the histopathology setting, the student closely tracks the teacher's manifold trajectory while exhibiting markedly lower variance. This regularisation effect is not present in natural images, where student and teacher variance remain comparable, suggesting that the variance reduction is specific to the domain-specialised distillation regime. Shaded bands denote ± 1 standard deviation across random seeds and data augmentations.

4 Geometric Evidence of Block Collapse

The formulation in Section 2.2 posits that a single learned surrogate layer can approximate a contiguous redundant block. To test this hypothesis, we examine how collapse alters the representation space. If intermediate layers within a block primarily refine a shared computational phase rather than introducing a new representational stage, the pruned student network should preserve the teacher's coarse trajectory at block boundaries. We quantify this geometric alignment by evaluating the intrinsic dimension (ID) of the token representations across depth using the Two-NN estimator [8].

We compare an unpruned teacher model using the H0-mini [10] backbone for histopathology images and DINOv2 [27] for natural images against the corresponding pruned student networks. To capture the full scope of the representational flow, we track the ID of the global class token, the spatially averaged patch tokens, and the raw, unpooled flattened patch token sequence.

4.1 Intrinsic Dimension and Smoothing

The raw, flattened token representation reveals geometric changes that are less apparent after pooling. Across block boundaries, the student follows the same broad ID trajectory as the unpruned teacher, indicating that the surrogate layers preserve the coarse representational progression of the original network. At the same time, the student consistently has a lower ID in the histopathology setting, particularly for the flattened patch-token representation in the rightmost panel of Fig. 5. This pattern suggests that block collapse removes part of the high-dimensional variation introduced by the teacher's intermediate layers while retaining the endpoint geometry.

Table 1: Intrinsic Dimension (ID) and Standard Deviation (STD) of Flattened Tokens at Block Boundaries (Overall Combined: Clean + Augmented). The unpruned teacher is H0-mini, while the block-collapsed student is the pruned H0-mini obtained by replacing the discovered blocks with 6 surrogate layers. The student network shows consistently lower ID and reduced variance, indicating lower sensitivity to the sampled augmentations.

Block Boundary (Layer Index)	Unpruned Teacher ID ($\mu \pm \sigma$)	Block-Collapsed Student ID ($\mu \pm \sigma$)	STD Reduction (%)
2	70.1 ± 21.4	58.3 ± 11.2	**47.6%**
5	89.4 ± 26.1	77.1 ± 14.8	**43.2%**
7	109.8 ± 29.5	91.2 ± 17.5	**40.6%**
9	106.5 ± 27.2	87.4 ± 16.9	**37.8%**
10	92.3 ± 24.8	76.5 ± 14.1	**43.1%**
11	68.4 ± 12.5	60.2 ± 6.3	**49.6%**

4.2 Variance Reduction Under Stochastic Augmentation

We next evaluate whether the lower ID observed in the student is accompanied by greater stability under input perturbations. For each input, we apply stochastic spatial and colour augmentations and measure the standard deviation of the resulting token representations at block boundaries. The shaded regions in Fig. 5 represent the variance across these augmentation passes, showing larger variation for the unpruned teacher at several intermediate boundaries. We quantify this instability in Table 1.

As shown in Table 1, the block-collapsed student demonstrates a substantial reduction in representational standard deviation—ranging from 37.8% to 49.6% across all boundaries compared to the teacher network. This indicates that the collapsed student is less sensitive to augmentations at each measured boundary. Since the student still follows the teacher's coarse ID trajectory, the lower variance suggests that the surrogate layers preserve the phase boundaries while reducing augmentation-sensitive variation along the intermediate path.

This can be interpreted as evidence for the claim in Section 3. Intermediate layers within a redundant block are not contributing distinct computational stages, but instead accumulate stochastic variation as they traverse a locally flat region of the representation manifold. By removing these transitional steps, TWT can provide a more compact, stable representation with less sensitivity to perturbations by augmentation. The downstream results in Tables 2 and 3 confirm that this source of variation carries little-to-no task-relevant signal and is likely noise introduced by over-provisioned depth.

5 Experiments

We evaluate TWT across two distinct visual domains: natural images and histopathology Whole-Slide Image (WSI) classification. Our primary objective is to demonstrate that collapsing redundant blocks not only reduces parameter counts and computation but also does so with little to no performance loss. We compare TWT against Raptor [15] via BRH and two state-of-the-art depth pruning methods: WDPruning [40] and NOSE [19], where each method is applied to a baseline model (reported as such in each result for reference).

Table 2: Histopathology MIL results for H0-mini. We report QWK (×100) for prostate grading and accuracy for breast metastasis detection. Prostate MIL heads train on PANDA and test on PANDA/TCGA-PRAD; metastasis MIL heads train on CAMELYON17 and test on CAMELYON17/CAMELYON16. Deltas compare against the uncompressed backbone with the same MIL head.

					Prostate (QWK)		Breast/Lymph (Acc)	
Agg.	Method	Layers	Params (M)	FLOPs (G)	PANDA Test	TCGA-PRAD	CAM17 Test	CAM16
ABMIL	Baseline	12	85.74	23.56	93.77 ± 0.16	69.14 ± 1.22	88.35 ± 0.72	97.31 ± 0.23
	NOSE	12	57.36	14.91	85.40 ± 0.26 −8.38	42.56 ± 2.99 −26.58	89.24 ± 0.98 +0.89	81.30 ± 1.60 −16.01
	WDPruning	5	42.40	9.37	93.44 ± 0.26 −0.33	64.65 ± 1.59 −4.49	91.08 ± 1.30 +2.73	91.09 ± 2.73 −6.22
		4	35.31	7.52	93.65 ± 0.21 −0.13	55.52 ± 0.28 −13.62	91.59 ± 0.48 +3.24	90.26 ± 0.75 −7.05
	Raptor	5	36.30	23.57	94.23 ± 0.22 +0.45	69.30 ± 0.59 +0.16	89.28 ± 0.92 +0.93	96.94 ± 0.22 −0.36
		4	29.17	23.57	93.97 ± 0.13 +0.20	69.38 ± 0.37 +0.25	89.32 ± 0.68 +0.97	96.94 ± 0.34 −0.36
	TWT	5	36.11	9.36	93.79 ± 0.25 +0.02	70.08 ± 1.08 +0.94	90.46 ± 0.43 +2.12	96.73 ± 0.39 −0.57
		4	29.02	7.51	93.71 ± 0.28 −0.06	69.69 ± 0.55 +0.55	89.70 ± 1.75 +1.36	95.49 ± 0.70 −1.81
TransMIL	Baseline	12	85.74	23.56	94.57 ± 0.41	56.02 ± 6.21	89.11 ± 0.71	96.27 ± 0.98
	NOSE	12	57.36	14.91	89.93 ± 0.57 −4.64	49.15 ± 2.22 −6.87	88.65 ± 1.43 −0.46	81.23 ± 0.93 −15.04
	WDPruning	5	42.40	9.37	94.62 ± 0.30 +0.05	51.29 ± 7.10 −4.73	90.61 ± 1.31 +1.50	90.65 ± 1.27 −5.62
		4	35.31	7.52	94.66 ± 0.50 +0.09	38.19 ± 3.98 −17.83	89.97 ± 1.32 +0.86	88.20 ± 1.93 −8.07
	Raptor	5	36.30	23.57	94.62 ± 0.59 +0.04	57.22 ± 5.63 +1.20	90.84 ± 0.64 +1.73	96.84 ± 0.50 +0.57
		4	29.17	23.57	94.64 ± 0.39 +0.07	57.73 ± 8.90 +1.71	90.04 ± 0.50 +0.93	96.01 ± 1.41 −0.26
	TWT	5	36.11	9.36	94.49 ± 0.67 −0.08	61.89 ± 3.56 +5.87	90.13 ± 0.71 +1.02	95.29 ± 0.67 −0.98
		4	29.02	7.51	94.29 ± 0.59 −0.28	59.11 ± 4.13 +3.09	90.93 ± 0.86 +1.82	95.23 ± 0.59 −1.04

5.1 Histopathology Models

Histopathology models process gigapixel whole-slide images (WSIs), where inference cost scales with the number of retained tissue tiles per slide. We, therefore, evaluate whether TWT preserves slide-level performance after reducing the depth of two domain-specific DINOv2-style backbones: H0-mini [10] and Hibou-B [26].

We use PANDA training tiles for histopathology phase discovery and backbone compression. We then freeze each pruned or unpruned backbone and precompute tile embeddings for all downstream cohorts. Each WSI forms a bag of foreground tile embeddings, and we train slide-level MIL heads on these frozen bags. We evaluate two MIL aggregators, ABMIL [14] and TransMIL [31], to test whether the compressed features remain useful across different slide-level pooling mechanisms.

For prostate cancer grading, we train MIL heads on PANDA [3] using the split from Song et al. [34]. We select checkpoints by PANDA validation QWK and report QWK (×100) on PANDA test and TCGA-PRAD [42]. For breast metastasis detection, we train MIL heads on CAMELYON17 [21] using corrected binary labels from Ling et al. [20]. We select checkpoints by the 5-epoch moving average of CAMELYON17 validation accuracy and report slide-level accuracy on CAMELYON17 test and CAMELYON16 [2]. We provide tiling, tissue filtering, label processing, and optimisation details in the supplementary material.

Pruning Implementation Details. We train pruned backbones using AdamW [23], a cosine learning-rate decay [22], and gradient clipping at norm 1.0. We use a peak learning rate of 3×10^{-4}. We employ colour jitter and Gaussian blur data augmentations during distillation. Notably, we omit both the auxiliary LayerScale parameters and the staggered cosine loss schedule for these histopathology models. Unlike natural image backbones, they do not exhibit late-layer norm explosion and optimise stably without these additions. We scale training budgets by student size: for H0-mini, we train NOSE and WDPruning for 2 epochs, Raptor for 3 epochs total (1 in Stage-1, 2 in Stage-2), and TWT for 2 epochs (depth 4) or 1.5 epochs (depth 5). For Hibou-B, we use the same hyperparameters as for H0-mini across all methods, with 2 epochs for TWT at both depths. We also experimented with extended training budgets on Hibou-B (up to 5 epochs for NOSE and WDPruning, and 3 epochs for

Table 3: Histopathology MIL results for Hibou-B. We report QWK ($\times 100$) for prostate grading and accuracy for breast metastasis detection. Prostate MIL heads train on PANDA and test on PANDA/TCGA-PRAD; metastasis MIL heads train on CAMELYON17 and test on CAMELYON17/CAMELYON16. Deltas compare against the uncompressed backbone with the same MIL head.

					Prostate (QWK)		Breast/Lymph (Acc)	
Agg.	Method	Layers	Params (M)	FLOPs (G)	PANDA Test	TCGA-PRAD	CAM17 Test	CAM16
ABMIL	Baseline	12	85.74	23.56	93.70 ± 0.24	66.27 ± 1.11	90.25 ± 0.87	90.52 ± 2.48
	NOSE	12	57.36	14.91	86.49 ± 0.21 −7.21	51.02 ± 2.15 −15.26	90.47 ± 0.84 +0.21	83.94 ± 0.99 −6.58
	WDPruning	4	35.31	7.94	93.64 ± 0.15 −0.07	66.55 ± 0.89 +0.28	89.66 ± 1.58 −0.59	73.99 ± 6.77 −16.53
		3	28.22	5.99	92.96 ± 0.46 −0.75	64.66 ± 0.98 −1.61	88.14 ± 0.00 −2.11	65.54 ± 0.00 −24.98
	Raptor	4	29.17	23.57	93.79 ± 0.33 +0.09	67.13 ± 0.65 +0.86	90.97 ± 1.06 +0.72	94.82 ± 1.56 +4.30
		3	22.04	23.57	93.57 ± 0.29 −0.14	66.47 ± 0.42 +0.19	88.69 ± 1.59 −1.56	95.60 ± 0.55 +5.08
	TWT	4	29.02	7.93	93.27 ± 0.32 −0.44	67.77 ± 0.67 +1.50	90.25 ± 1.03 +0.00	96.16 ± 0.28 +5.64
		3	21.93	5.98	93.30 ± 0.21 −0.41	65.49 ± 0.60 −0.79	89.66 ± 0.92 −0.59	93.63 ± 1.43 +3.11
TransMIL	Baseline	12	85.74	23.56	94.13 ± 0.44	57.32 ± 7.11	89.32 ± 0.65	92.80 ± 2.99
	NOSE	12	57.36	14.91	88.82 ± 0.36 −5.31	53.88 ± 2.05 −3.44	89.19 ± 0.75 −0.13	83.99 ± 1.44 −8.81
	WDPruning	4	35.31	7.94	94.36 ± 0.35 +0.23	45.27 ± 5.86 −12.04	90.04 ± 0.81 +0.72	79.48 ± 3.69 −13.32
		3	28.22	5.99	93.38 ± 0.51 −0.75	51.66 ± 3.77 −5.66	87.97 ± 2.27 −1.35	67.31 ± 9.44 −25.49
	Raptor	4	29.17	23.57	94.15 ± 0.51 +0.02	59.49 ± 6.96 +2.18	88.64 ± 1.39 −0.68	74.30 ± 9.00 −18.50
		3	22.04	23.57	94.67 ± 0.18 +0.54	58.59 ± 8.49 +1.27	89.02 ± 0.66 −0.30	91.55 ± 5.29 −1.25
	TWT	4	29.02	7.93	94.65 ± 0.29 +0.52	55.47 ± 2.70 −1.85	91.02 ± 1.12 +1.70	93.26 ± 2.36 +0.46
		3	21.93	5.98	94.02 ± 0.07 −0.11	54.58 ± 3.12 −2.74	90.42 ± 0.98 +1.10	91.09 ± 2.84 −1.71

Raptor); however, these extended runs yielded similar downstream outcomes, confirming that the performance bottlenecks are not due to insufficient training time.

H0-mini results. Table 2 shows that TWT preserves prostate grading performance while reducing the active backbone size by more than half. The five-layer TWT model uses 36.11M parameters instead of 85.74M and matches the uncompressed model on PANDA for both ABMIL and TransMIL. It also improves TCGA-PRAD QWK for both MIL heads, with gains of $+0.94$ and $+5.87$, respectively. The four-layer model further reduces the active parameter count to 29.02M and remains close to the baseline on PANDA while still improving TCGA-PRAD. On CAMELYON17, TWT improves accuracy across both MIL heads and both depths. CAMELYON16 shows a small drop relative to the baseline, but the pruned models remain within roughly two percentage points while using about one third of the active backbone parameters.

Hibou-B results. Table 3 evaluates whether the same compression behaviour transfers to a second histopathology foundation model. TWT again provides a strong parameter–performance trade-off: the four-layer model uses 29.02M active parameters compared with 85.74M for the full backbone, while remaining close to the baseline on PANDA and improving ABMIL performance on TCGA-PRAD. On CAMELYON, TWT is particularly strong for the external CAMELYON16 evaluation, indicating that the collapsed backbone can retain transferable features across tissue type and dataset shift. Some competing pruned baselines, such as WDPruning and NOSE, are less stable in this setting, especially at aggressive compression levels, but our TWT achieves similar parameter-performance trade-offs while fundamentally reducing inference compute.

5.2 Natural Image Models

Our experiments with natural images closely follow the pruning methodology from Section 5.1, with few exceptions. For natural images, we use a total of 8 epochs across all methods, and while training stabilises quite early, we see a more dramatic effect from using

Table 4: Natural image classification results for base-capacity backbones. We report top-1 accuracy at 224×224. Deltas compare against the uncompressed backbone of the same model.

Model	Method	Layers	Params (M)	FLOPs (G)	ImageNet-1k Val	ReaL	v2
DEiT-III B/16	Baseline	12	86.6	17.81	83.1	87.7	71.9
	NOSE	12	70.1	14.06 −3.75	81.5 −1.6	86.2 −1.5	70.4 −1.5
	WDPruning	6	45.0	8.92 −8.89	82.7 −0.4	87.1 −0.6	71.1 −0.8
		5	38.2	7.43 −10.38	82.1 −1.0	86.8 −0.9	70.8 −1.1
	Raptor	6	39.4	17.81 +0.00	82.9 −0.2	87.3 −0.4	71.5 −0.4
		5	32.5	17.81 +0.00	82.2 −0.9	86.9 −0.8	70.9 −1.0
	TWT	6	39.0	8.91 −8.90	82.8 −0.3	87.5 −0.2	71.4 −0.5
		5	32.1	7.42 −10.39	82.1 −1.0	86.9 −0.8	71.0 −0.9
DINOv2 B/14	Baseline	12	86.6	23.42	84.6	88.5	74.9
	NOSE	12	70.1	18.38 −5.04	82.8 −1.8	87.0 −1.5	72.1 −2.8
	WDPruning	6	45.0	11.72 −11.70	83.5 −1.1	87.5 −1.0	73.6 −1.3
		5	38.2	9.77 −13.65	82.4 −2.2	86.4 −2.1	73.4 −1.5
	Raptor	6	39.4	23.42 +0.00	84.0 −0.6	87.8 −0.7	74.6 −0.3
		5	32.5	23.42 +0.00	83.4 −1.2	87.2 −1.3	74.0 −0.9
	TWT	6	39.0	11.71 −11.71	84.2 −0.4	88.1 −0.4	74.6 −0.3
		5	32.1	9.76 −13.66	82.4 −2.2	86.9 −1.6	73.8 −1.1
DINOv3 B/16	Baseline	12	86.6	17.82	85.2	89.3	75.0
	NOSE	12	70.1	14.07 −3.75	83.2 −2.0	87.5 −1.8	73.2 −1.8
	WDPruning	6	45.0	8.93 −8.89	84.1 −1.1	88.4 −0.9	74.3 −0.7
		5	38.2	7.44 −10.38	83.3 −1.9	87.6 −1.7	73.6 −1.4
	Raptor	6	39.4	17.82 +0.00	84.6 −0.6	88.6 −0.7	74.6 −0.4
		5	32.5	17.82 +0.00	83.7 −1.5	88.1 −1.2	74.0 −1.0
	TWT	6	39.0	8.92 −8.90	84.4 −0.8	88.6 −0.7	74.3 −0.7
		5	32.1	7.43 −10.39	83.4 −1.8	88.0 −1.3	73.9 −1.1

a staggered cosine schedule for the loss function than in histopathology. Our experiments focus on ImageNet-1k [7] for classification and ADE20k [41] for segmentation. Evaluation protocols follow the respective baselines [16, 33, 36].

Classification. Table 4 shows that TWT achieves competitive accuracy at a fraction of the inference cost. At six layers, TWT matches or closely tracks Raptor across all three backbones while operating at roughly half the FLOPs; on DINOv2 B/14, the six-layer TWT model outperforms Raptor on all folds despite requiring only 11.71 GFLOPs versus 23.42 GFLOPs. Interestingly, Raptor [15] provides marginal accuracy improvements on B/16 models, notably with no reduction in compute. With fewer tokens, the attention matrix is smaller (196×196 vs. 256×256), so each recurrent application achieves more complete global mixing relative to the total information content. A recurrent layer could have a smoother optimisation target simply because the token-mixing space has a lower dimensionality.

Compared to WDPruning, which operates at a near-identical compute budget, TWT consistently matches or improves accuracy, suggesting that the phase-aware surrogate initialisation and deep supervision yield a more effective model than uniform depth pruning. At five layers, all methods degrade more noticeably, reflecting compression that likely goes beyond the redundant phases into functionally distinct computation. Nevertheless, TWT remains within approximately one point of the baseline across all three tested models.

In contrast to the histopathology results (Tables 2 and 3), no method improves over the uncompressed baseline on natural images. This is consistent with our observations on smoothing from Section 4.1; ImageNet backbones were trained on this distribution, so the

Table 5: Semantic segmentation results on ADE20k. Deltas compare against the uncompressed baseline.

Model	Method	Layers	FLOPs (G)		Size	mIoU	
DINOv2	Baseline	12	147.4		518 × 518	47.3	
	Raptor	6	147.4	+0.00	518 × 518	44.2	−3.1
	TWT	6	75.9	−71.5	518 × 518	42.9	−4.4
DINOv3	Baseline	12	107.0		512 × 512	54.9	
	Raptor	6	107.0	+0.00	512 × 512	49.1	−5.8
	TWT	6	53.6	−53.4	512 × 512	47.4	−7.5

redundant phases are already well-tuned and collapsing them can at best preserve performance. In histopathology, the backbone operates out-of-distribution relative to its pre-training data, and the intermediate iterative steps accumulate domain-irrelevant noise that block collapse actively removes.

Dense Tasks. In addition to instance-level classification, we evaluate semantic segmentation on ADE20k, comparing against baseline DINO-family models and Raptor [15]. Table 5 shows that dense prediction is sensitive to block approximation even when the full recurrent execution cost is retained. Raptor preserves the original network's iterative computation but still incurs a modest drop in mIoU. In comparison, TWT reduces inference compute by approximately 50%, at the cost of an additional 1.3 and 1.7 mIoU on DINOv2 and DINOv3, respectively.

This result is consistent with the distinction made by Jacobs et al. [15] between patch-token and instance-level dynamics. Dense prediction depends directly on the spatial token field, whereas classification can remain robust when the instance representation is preserved. A recurrent surrogate may therefore retain local patch-token evolution more faithfully because it maintains iterative token mixing. The additional mIoU loss therefore reflects a trade-off between preserving patch-level trajectory structure and eliminating recurrent inference cost, suggesting that dense prediction may require less aggressive block collapse. We discuss this trade-off further in Section 7.1.

6 Related Work

Depth as Recurrent Dynamics. Existing work conceptualises deep networks as continuous dynamical systems where representations naturally cluster into meta-stable attractors as they propagate through depth [11, 17, 38]. BRH exploits this simplicity bias by approximating contiguous blocks with a single parameter-tied layer applied recurrently [15], operating on the assumption that multi-step iterative execution is strictly necessary. In contrast, we argue that this multi-step behaviour is merely an artefact of the optimisation landscape. Instead of unrolling a recurrent operator, TWT suggests that these redundant blocks approximate a single, ideal phase of computation that maps inputs directly to the terminal representation in a fused step.

Layer Relevance and Structural Pruning. To handle block redundancy, current methods often identify "ineffective" layers and excise them using structural pruning [9, 12, 29], accuracy-grounded relevance metrics [13], or structural linearisation of consecutive blocks [1, 24, 32] to salvage downstream performance. Rather than simply dropping layers based on

heuristics or generic proxies, we frame this multi-step processing explicitly as executable redundancy through the lens of TWT. This theoretically justifies replacing an entire contiguous sequence of layers with a single step instead of selectively pruning isolated parts.

Trajectory Distillation and Block Collapse. Advanced model compression techniques move beyond simple logit matching by distilling internal hidden-state trajectories (using metrics like CKA or value-relations) [6, 39] or by iteratively substituting full layers with retrained compact blocks [25]. Building upon this, we introduce "block collapse" as a localised distillation mechanism guided by TWT. By anchoring the start and end of a phase through deep supervision, we replace the entire redundant block with a single, non-recurrent surrogate operator. This forces the network to bypass intermediate representational dawdling, actively denoising the manifold and directly yielding inference-time compute savings.

7 Conclusion

In this work, we investigated the phenomenon of block redundancy in Vision Transformers. While recent frameworks like BRH interpret highly similar contiguous layers as evidence of underlying recurrence, our analyses demonstrate that these iterative micro-steps are not strictly computationally necessary. Our intrinsic dimension and variance analyses further suggest that, in the histopathology setting, intermediate steps within redundant phases can accumulate augmentation-sensitive variation without altering the coarse representational trajectory.

To resolve this inefficiency, we introduced Transformer-Within-Transformer (TWT). By dynamically identifying redundant phases and collapsing them into single, learned surrogate layers, TWT replaces intermediate iterations with a single macro-step. Unlike linear approximations that sacrifice token-mixing expressivity, or block-recurrent methods that still execute costly iterations at inference, our approach tracks the coarse geometric trajectory while reducing both parameter count and inference compute. Across natural-image models, TWT preserves competitive downstream performance at substantially reduced compute, while in several histopathology settings it matches or improves over the uncompressed baseline.

7.1 Limitations and Further Work

While TWT provides a general methodology for exploiting block redundancy in ViTs, it is not to be interpreted as a universal panacea. As discussed in Tables 2 to 4, performance improvement over the uncompressed baseline appears only in specific settings, notably the histopathology models in our experiments. This does not imply that a performance increase should be expected in the general case. Dense prediction remains more sensitive to block collapse than instance-level classification. Our segmentation results suggest that patch-level tasks benefit from preserving more of the intermediate token trajectory, and may therefore require less aggressive collapse or task-aware distillation.

The partitioning scheme in Section 3.1 remains an approximation. Better block discovery improves performance, motivating future work on submodular optimisation or optimal transport for this combinatorial problem.

Finally, we focus on post-hoc exploitation of block redundancy. Explaining how such redundancy arises in vision models, and how it might be mitigated during pre-training, remain important directions for future work.

Acknowledgments

This work was funded by the Research Council of Norway through Visual Intelligence, Centre for Research-based Innovation (309439), and by the South-Eastern Norway Regional Health Authority (2024039). The computations were performed on resources provided by Sigma2 (NN8104K) — the National Infrastructure for High-Performance Computing and Data Storage in Norway. We acknowledge Sigma2 for access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through Sigma2, Norway.

References

[1] Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *Inter. Conf. Learn. Represent. (ICLR)*, 2024.

[2] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017. doi: 10.3410/f.732283043.793567347.

[3] Wouter Bulten, Kimmo Kartasalo, Po-Hsuan Cameron Chen, Peter Ström, Hans Pinckaers, Kunal Nagpal, Yuannan Cai, David F Steiner, Hester Van Boven, Robert Vink, et al. Artificial intelligence for diagnosis and gleason grading of prostate cancer: the panda challenge. *Nat. Med.*, 28(1):154–163, 2022.

[4] Irene Cannistraci, Simone Antonelli, Emanuele Palumbo, Thomas M. Sutter, Emanuele Rodolà, Bastian Rieck, and Julia E Vogt. TOAST: Transformer optimization using adaptive and simple transformations. *Trans. Mach. Learn. Res.*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=fSwMCsBtTG.

[5] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *Inter. Conf. Learn. Represent. (ICLR)*, 2020. URL https://openreview.net/forum?id=HJlnC1rKPB.

[6] Sayantan Dasgupta and Trevor Cohn. Improving language model distillation through hidden state matching. In *Inter. Conf. Learn. Represent. (ICLR)*, 2025.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: a large-scale hierarchical image database. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 248–255. Ieee, 2009. doi: 10.1109/CVPR.2009.5206848.

[8] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Sci. Rep.*, 7(1): 12140, 2017.

[9] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *Inter. Conf. Learn. Represent. (ICLR)*, 2020.

[10] Alexandre Filiot, Nicolas Dop, Oussama Tchita, Auriane Riou, Rémy Dubois, Thomas Peeters, Daria Valter, Marin Scalbert, Charlie Saillard, Geneviève Robin, et al. Distilling foundation models for robust and efficient models in digital pathology. In *IEEE Inter. Conf. Med. Image Comput. Comput. Assist. Interv. (MICCAI)*, pages 162–172. Springer, 2025.

[11] Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2023.

[12] Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers. In *Inter. Conf. Learn. Represent. (ICLR)*, 2025.

[13] Cristian Hinostroza, Rodrigo Toro Icarte, Christ Devia, Andres Carvallo De Ferari, Eugenio Herrera-Berg, Denis Parra, and Jorge F Silva. Rethinking layer relevance in large language models beyond cosine similarity. In *Inter. Conf. Learn. Represent. (ICLR)*, 2026. URL https://openreview.net/forum?id=mRLnS8jQWt.

[14] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Inter. Conf. Mach. Learn. (ICML)*, pages 2127–2136. PMLR, 2018.

[15] Mozes Jacobs, Thomas Fel, Richard Hakim, Alessandra Brondetta, Demba E. Ba, and T. Anderson Keller. Block recurrent dynamics in vision transformers. In *Inter. Conf. Learn. Represent. (ICLR)*, 2026. URL https://openreview.net/forum?id=gH3HhnfWLC.

[16] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, Oriane Siméoni, Huy V. Vo, Patrick Labatut, and Piotr Bojanowski. DINOv2 meets text: a unified framework for image- and pixel-level vision-language alignment, 2024.

[17] Nikita Karagodin, Yury Polyanskiy, and Philippe Rigollet. Clustering in causal attention masking. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2024.

[18] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Inter. Conf. Mach. Learn. (ICML)*, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

[19] Sihao Lin, Pumeng Lyu, Dongrui Liu, Tao Tang, Xiaodan Liang, Andy Song, and Xiaojun Chang. MLP can be a good transformer learner. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 19489–19498, 2024.

[20] Xitong Ling, Yuanyuan Lei, Jiawen Li, Junru Cheng, Wenting Huang, Tian Guan, Jian Guan, and Yonghong He. Comprehensive benchmark dataset for pathological lymph node metastasis in breast cancer sections. *Scientific Data*, 12(1):1381, 2025. doi: 10.1038/s41597-025-05586-5.

[21] Geert Litjens, Peter Bandi, Babak Ehteshami Bejnordi, Oscar Geessink, Maschenka Balkenhol, Peter Bult, Altuna Halilovic, Meyke Hermsen, Rob Van de Loo, Rob Vogels, et al. 1399 h&e-stained sentinel lymph node sections of breast cancer patients: the camelyon dataset. *GigaScience*, 7(6):giy065, 2018. doi: 10.1093/gigascience/giy065.

[22] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Inter. Conf. Learn. Represent. (ICLR)*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[24] Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the structural pruning of large language models. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2023.

[25] Malthe Have Musaeus and Rob van der Goot. Iterative structured knowledge distillation: Optimizing language models through layer-by-layer distillation. In *Inter. Conf. Comput. Ling. (COLING)*, 2025.

[26] Dmitry Nechaev, Alexey Pchelnikov, and Ekaterina Ivanova. Hibou: A family of foundational vision transformers for pathology. *arXiv preprint arXiv:2406.05074*, 2024.

[27] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

[28] Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is turing complete. *J. Mach. Learn. Res.*, 2021.

[29] Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models. *Comput. Speech Lang.*, 2023.

[30] Avi Schwarzschild, Arjun Gupta, Amin Ghiasi, Micah Goldblum, and Tom Goldstein. The uncanny similarity of recurrence and depth. In *Inter. Conf. Learn. Represent. (ICLR)*, 2022. URL https://openreview.net/forum?id=3wNcr5nq56.

[31] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. TransMIL: Transformer based correlated multiple instance learning for whole slide image classification. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, volume 34, pages 2136–2147, 2021.

[32] Dmitriy Shopkhoev, Ammar Ali, Magauiya Zhussip, Valentin Malykh, Stamatios Lefkimmiatis, Nikos Komodakis, and Sergey Zagoruyko. ReplaceMe: Network simplification via depth pruning and transformer block linearization. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2025. URL https://openreview.net/forum?id=zEj1FSYCRn.

[33] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

[34] Andrew H Song, Richard J Chen, Tong Ding, Drew FK Williamson, Guillaume Jaume, and Faisal Mahmood. Morphological prototyping for unsupervised slide representation learning in computational pathology. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 11566–11578, 2024. doi: 10.1109/cvpr52733.2024.01099.

[35] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Herve Jegou. Going deeper with Image Transformers . In *IEEE Inter. Conf. Comput. Vis. (ICCV)*, pages 32–42, 2021. doi: 10.1109/ICCV48922.2021.00010. URL https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00010.

[36] Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: revenge of the ViT. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *European Conf. Comput. Vis. (ECCV)*, volume 13684 of *Lecture Notes in Computer Science*, pages 516–533. Springer, 2022. doi: 10.1007/978-3-031-20053-3_30.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Adv. Neural Inf. Process. Sys. (NeurIPS)*, volume 30, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[38] Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2016.

[39] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2020.

[40] Fang Yu, Kun Huang, Meng Wang, Yuan Cheng, Wei Chu, and Li Cui. Width & depth pruning for vision transformers. In *AAAI Conf. Artif. Intell. (AAAI)*, volume 36, pages 3143–3151, 2022.

[41] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20k dataset. *Inter. J. Comput. Vis.*, 127(3):302–321, 2019. doi: 10.1007/s11263-018-1140-0.

[42] Margarita L Zuley, Rose Jarosz, Bettina F Drake, Danielle Rancilio, Aleksandra Klim, Kimberly Rieger-Christ, and John Lemmerman. Radiology data from the cancer genome atlas prostate adenocarcinoma [tcga-prad] collection. *Cancer Imaging Arch*, 9(10.7937): K9, 2016.