AFFMAE: Scalable Vision Pre-Training for High-Resolution Microscopy Segmentation on Desktop Hardware

David Smerkous[1,3⋆], Zian (Andy) Wang[1,2⋆], and Behzad Najafian[1]

[1] Department of Laboratory Medicine & Pathology, University of Washington, Seattle WA, USA
[2] Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle WA, USA
[3] Electrical Engineering & Computer Science, Oregon State University, Corvallis, OR, USA

Abstract. Self-supervised pretraining has transformed computer vision by enabling data-efficient fine-tuning, yet high-resolution pretraining typically requires server-scale infrastructure, limiting custom in-domain training for many research laboratories. Masked Autoencoders (MAE) reduce computation by encoding only visible tokens, but combining MAE with hierarchical downsampling architectures has remained structurally challenging due to dense grid priors and mask-aware design compromises. We introduce **AFFMAE**, a masking-friendly hierarchical pretraining framework built on adaptive, off-grid token merging. AFFMAE removes dense-grid assumptions while preserving hierarchical scalability during pretraining and fine-tuning. To support this architecture, we developed numerically stable mixed-precision Triton kernels and a lightweight, point-based decoder that can be directly repurposed as a segmentation head. On high-resolution microscopy segmentation, AFFMAE matches MAE finetuning performance on foot process width estimation with ViT backbone at equal parameter counts while being 2x faster during pre-training and halving peak memory usage. Furthermore, AFFMAE achieves up to 5x throughput speedups fine-tuning at the 1024px resolution, providing high-resolution model training on desktop hardware. Code available at https://github.com/najafian-lab/affmae.

Keywords: Masked Autoencoders · Efficient Vision Transformers · Microscopy Segmentation

1 Introduction

Biomedical diagnostics accumulate large volumes of unlabeled high-resolution images (e.g., histology, electron microscopy, radiology), yet pretraining models remains out of reach for many groups. Two constraints dominate in practice: (i) *compute*, since high-resolution pretraining often assumes server-grade

⋆ Equal contribution

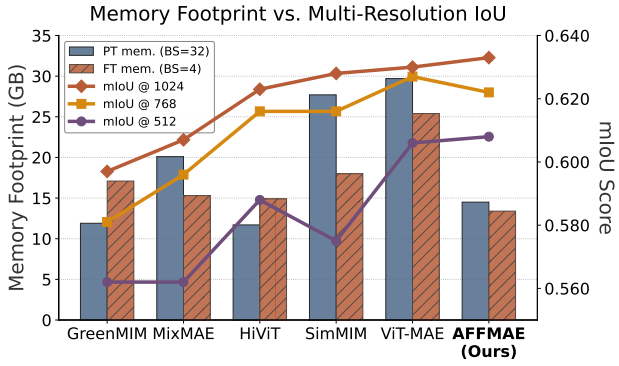


Fig. 1: Performance versus computational cost across architectures and resolutions. PT and FT means Pre-Training and Fine-Tuning, respectively. AFFMAE achieves ViT-level segmentation accuracy while requiring significantly smaller memory footprint.

multi-GPU setups, and (ii) *data governance*, where moving sensitive datasets to external cloud infrastructure can trigger lengthy HIPAA/IRB processes and institutional review. These frictions are especially pronounced in *novel* biomedical regimes—e.g., a newly studied cell phenotype, stain, scanner, or acquisition protocol—where no curated public dataset exists, but a lab may still have a large private archive. In such settings, prior work repeatedly finds that pretraining on the target domain improves transfer over generic ImageNet-style initialization [21, 35, 36]; more broadly, transfer improves as the pretraining distribution approaches the downstream domain [7]. This motivates *in-house* pretraining methods that are feasible on desktop-class GPUs for privacy-sensitive and institutionally constrained labs.

In this work we focus on kidney EM, where high-resolution, in-domain pretraining is particularly valuable and difficult to scale. Our primary focus are podocytes, which are post-mitotic cells that play a crucial role in maintaining plasma filtration in the kidneys, and most diseases causing end-stage kidney disease are related to podocyte injury. These cells form interdigitated foot processes that widen upon injury, leading to protein leakage into the urine; thus, foot process width (FPW) is a widely used biomarker of podocyte injury in clinical and research settings. EM segmentation for FPW measurement is particularly demanding because it requires both *fine detail* and *global context*: FPW depends on segmenting tiny filtration slits that lie along the surface of the glomerular basement membrane (GBM), and these structures are only reliably resolved at high input resolutions ($> 512 \times 512$) [17,27,33]. Many slit-like textures are locally ambiguous (e.g., within the capillary lumen), so correct classification depends on global glomerular geometry and whether a candidate lies on the GBM boundary; this is further amplified in foot process effacement, where slit morphology is degraded and local neighborhoods become insufficient to identify slits [27]. Importantly, these targets form long, thin boundaries, making them especially

sensitive to coarse grid-aligned downsampling that can dilute or erase narrow structures (Figure 5a), motivating architectures that can retain high spatial detail, while still scaling efficiently to high resolution for global context.

Masked image modeling (MIM), and in particular Masked Autoencoders (MAE), is attractive for resource-constrained in-house pretraining because the encoder runs only on visible tokens, reducing activation memory and FLOPs [13]. However, ViT-MAE can still be slow and memory-heavy at microscopy-scale resolutions on consumer GPUs, motivating hierarchical backbones that reduce token counts via downsampling. Unfortunately, most hierarchical ViTs (e.g., Swin) are tied to a dense lattice through windowing and patch merging, so discarding tokens breaks their core operations and often forces dense bookkeeping or mask-aware encoders, introducing pretrain–finetune mismatch and degraded downstream performance [15, 19, 32].

We address this gap with **AFFMAE**, a masking-friendly framework for *hierarchical* pretraining that is explicitly designed to make in-domain, in-house pretraining practical for consumer hardware. Building on AutoFocusFormer [38], AFFMAE performs adaptive, off-the-grid dynamic downsampling while keeping the encoder mask-agnostic. The key idea is to perform token merging and attention only over the *visible* tokens, preserving MIM efficiency while retaining the computational benefits of hierarchy without architectural differences between pretraining and finetuning beyond input token masking, and focusing computation on information-rich regions. Additionally, we provide efficient Flash cluster-attention Triton kernels that run on most modern desktop GPUs, not requiring CUDA [22], enabling base-model pretraining on consumer hardware. Empirically, AFFMAE matches ViT-MAE segmentation performance at the same parameter count while using 1/4th of the FLOPs, 1/2 the memory, and training 2× faster on an RTX 5090.

2 Related Works

Self-supervised pretraining objectives and trade-offs. Self-supervised learning for vision spans contrastive objectives (e.g., SimCLR, MoCo) [5, 14], multimodal contrastive objectives (e.g., CLIP) [24], and teacher–student / self-distillation approaches (e.g., BYOL, DINO, I-JEPA) [1,3,12]. These families produce strong transferable representations but often rely on multi-view pipelines, large effective batch sizes (or queues/memory banks), and/or momentum teachers, increasing memory, compute, and engineering overhead. In contrast, **masked image modeling (MIM)** methods [2,31] learn from a within-image reconstruction signal, and **Masked Autoencoders (MAE)** [13] discard masked tokens and run the encoder only on visible tokens, making encoder-side cost scale with the visible token count and enabling practical in-domain pretraining on desktop-class hardware.

Hierarchical Transformers for efficient high-resolution vision. A major challenge for high-resolution imagery is that standard Vision Transformers [9] incur quadratic attention cost in the number of tokens [28]. Hierarchical Trans-

formers mitigate this by progressively reducing token counts through downsampling mechanisms such as patch merging and local attention (e.g., Swin [20], PVT [29], and Multiscale ViTs [10]), substantially improving scaling to high-resolution inputs. However, these designs typically assume a *dense, rigid lattice* of tokens to support window partitioning and grid-aligned merging.

Masked pretraining for hierarchical Transformers. Adapting MIM to hierarchical backbones is structurally non-trivial because many hierarchical operations expect dense token layouts. SimMIM [31] works well with hierarchical backbones but typically retains a dense token stream with mask tokens, departing from MAE's efficiency premise of discarding masked tokens in the encoder; SwinMAE [32] similarly must reconcile window computation with masking. More recent approaches target MAE-style token dropping in hierarchical settings—MixMAE [19] mixes visible patches across images, GreenMIM [15] packs sparse visible tokens into dense windows during pre-training, and HiViT [34] serializes masked units via lightweight token mixing before a chosen main stage—but they remain coupled to dense-grid assumptions and/or introduce stage- or masking-specific interfaces, contributing to pretrain–finetune discrepancies and weaker performance on small, thin structures compared to ViT-style baselines in our setting.

3 Methods

3.1 AutoFocusFormer

AutoFocusFormer (AFF) [38] is a hierarchical backbone that replaces rigid, grid-aligned downsampling (e.g., stride-2 pooling or patch merging) with adaptive downsampling over an irregular set of token locations. Starting from a dense patch embedding, AFF maintains for each token both a feature vector and a 2D coordinate, and performs local computation by building *balanced neighborhoods* over these coordinates: token locations are grouped into equal-size clusters, via a lightweight space-filling-curve procedure, to preserve some linear ordering for attention and faster KNN grouping. AFF neighborhood construction enables local self-attention for a fixed neighborhood size while remaining agnostic to the underlying grid. Within each neighborhood, attention uses relative position derived directly from the 2D token coordinates, allowing computation to follow the token set as it becomes increasingly irregular [38].

After several local-attention blocks, AFF downsamples using a learnable *neighborhood merging* module. An importance score is learned for each token via a one-layer MLP, and a user-specified downsampling rate $d_s \in (0, 1]$ controls how many tokens are carried forward to the next stage (e.g., d_s=0.25 retains 25% of tokens). The top-ranked tokens are kept as *anchors*, and the remaining tokens are merged into nearby anchors through a differentiable, location-aware aggregation operator that transfers both features and spatial support, yielding a smaller token set with updated coordinates and features. By supporting arbitrary retention ratios (rather than fixed factors like $1/2, 1/4, 1/8$), AFF can

preserve high token density in information-rich regions while aggressively merging texture-less areas (Figure 6b). Crucially, unlike standard grid-based pooling that uniformly downsamples the spatial resolution of the entire feature map, this irregular downsampling reduces the overall token count while strictly retaining the original, high-resolution spatial coordinates around critical structures; we refer the reader to the original AFF paper for full algorithmic details and additional visualizations [38].

3.2 AFFMAE

To maximize pre-training efficiency, our framework preserves the asymmetric design of Masked Autoencoders [13]: we do not introduce mask tokens or modify the encoder downsampling.

To better exploit off-the-grid multi-scale encoder features, we use a point-based deformable cross-attention decoder [37]. Decoder queries are learnable mask tokens with absolute positional encodings. For each query with feature $\mathbf{f}$ and reference location $\mathbf{r} \in \mathbb{R}^2$, we predict an offset

$$\Delta = g(\mathbf{f}), \qquad \mathbf{q} = \mathbf{r} + \Delta,$$

where $g(\cdot)$ is a linear projection. At each encoder stage ℓ, we gather the K nearest visible tokens to $\mathbf{q}$ (KNN in 2D coordinate space) to form a *virtual sampled token*. Unlike bilinear sampling used in point-based decoders such as Mask2Former [6], AFF [38] aggregates neighbors with inverse-distance weighting:

$$d_i = \|\mathbf{q} - \mathbf{x}_i\|_2 + \varepsilon, \qquad \tilde{w}_i = d_i^{-p}, \qquad w_i = \frac{\tilde{w}_i}{\sum_{j=1}^{K} \tilde{w}_j},$$

where p controls the spatial concentration of the interpolation. Directly computing the inverse-power weights can be numerically fragile under mixed precision: for small distances, d_i^{-p} can become very large, whereas for more distant neighbors it can underflow toward zero, producing saturated or unstable normalizations. To improve numerical stability, we replace the inverse-power kernel with an exponential distance kernel,

$$\tilde{w}_i = \exp(-p\, d_i), \qquad w_i = \frac{\exp(z_i - z_{\max})}{\sum_{j=1}^{K} \exp(z_j - z_{\max})} = \mathrm{softmax}_i(-p\, d_i),$$

where $z_i = -p d_i$ and $z_{\max} = \max_k z_k$. This yields the standard softmax form, allowing the weights to be evaluated with max-logit subtraction for stable mixed-precision computation while retaining an explicit locality parameter p. The resulting virtual sampled token is

$$\hat{\mathbf{f}}^{(\ell)}(\mathbf{q}) = \sum_{i=1}^{K} \mathrm{softmax}_i(-p\, d_i)\ \mathbf{f}_i^{(\ell)}.$$

Our decoder comprises four stages, directly mirroring the four hierarchical stages of the encoder. At each stage, deformable cross-attention is applied

against the corresponding stage-wise virtual tokens, followed by deformable self-attention. Following the final cross-attention stage, the dense token representations are processed through a standard LayerNorm and a linear reconstruction head to predict the original pixel values.

Because this decoder natively restores hierarchical spatial structures back on a dense grid, it is sufficient to be directly repurposed for downstream segmentation tasks with competitive performance as FPN-based segmentation decoders as shown in Section 4. This avoids the complexity of adapting and tuning separate, heavy task-specific heads (e.g., UperNet or Mask2Former) during fine-tuning.

3.3 Flash-style Cluster Attention and Fast Decoder KNN

While AFF reduces FLOPs by operating on sparse neighborhoods, the original implementation does not translate into proportional wall-clock speedups: cluster attention explicitly materializes attention scores (computing $QK^\top$ and storing the score matrix before multiplying by V), which is memory-bound and can be numerically unstable under mixed precision. We therefore re-implement local cluster attention in Triton following the FlashAttention principle [8]: we *stream* attention computation over tiles, fuse the neighborhood attention steps, and accumulate the softmax normalization in fp32 while writing outputs in fp16. Concretely, for each neighborhood n we compute

$$\mathrm{FlashNbhdAttn}(Q,K,V) = \mathrm{softmax}\big(Q_n[K_n \;\; K_{\mathrm{blank}}]^\top + B_n\big)\,[V_n \;\; V_{\mathrm{blank}}],$$

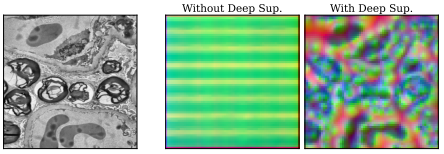
without materializing the full score matrix, where $(K_{\mathrm{blank}}, V_{\mathrm{blank}})$ are learned blank tokens used to regularize attention norms in texture-less regions as in [38]. This reduces memory traffic, improves throughput, and improves half-precision stability via a numerically stable streaming softmax with fp32 accumulators.

In the decoder, a major overhead comes from repeatedly retrieving the K nearest visible tokens for each continuous query location $\mathbf{q}$ when constructing virtual sampled tokens. Instead of performing a separate KNN search for every query, requiring a full pass over all visible token locations, we exploit the fact that token coordinates originate from a dense $H \times W$ patch grid (e.g., $64 \times 64 = 4096$ locations for a 512^2 image with patch size 8). We precompute a compact $H \times W \times K$ lookup table over this grid, where each entry stores the indices and 2D coordinates of the nearest visible-token candidates. At runtime, each continuous query location is quantized to its nearest grid cell, and neighbor retrieval becomes a table gather followed by a small fixed-size distance computation. After table construction, this makes KNN retrieval effectively $O(1)$ per query and allows the distance computation and softmax weighting to be fused into the same Triton kernel used for virtual token construction. This cache-based lookup reduces decoder overhead and results in a 1.5× forward-time speedup and a 1.4× backward-time speedup in the point decoder (Supp. 1.1).

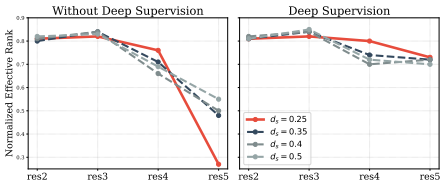
3.4 Deep Supervision

When adapting AFF to MIM-style pre-training with aggressive downsampling, we empirically observe a severe degradation in the representational quality of the

deepest encoder stages. Principal Component Analysis (PCA) visualizations of the latent tokens at the final, sparsest stages reveal a loss of semantic structure. Without intervention, the features degenerate into uniform, grid-like patterns, indicating that the tokens are primarily representing their positional embeddings rather than the underlying image semantics (Figure 2a).



(a) PCA projections of the learned representations. Features were extracted immediately after cross-attention with the respective encoder stages, allowing us to visualize a dense spatial grid instead of scattered visible tokens. The green PCA circles correspond with encoder downsampled token locations.



(b) Normalized Effective Rank. The evolution of representation rank across layers, computed from the encoder tokens.

Fig. 2: Analysis of representational collapse and the effect of Deep Supervision.

To mitigate this, we employ Deep Supervision [18]. We attach auxiliary reconstruction heads at the end of each intermediate decoder stage. By forcing the network to reconstruct the masked input directly from these intermediate tokens before they have cross-attended to the dense tokens at the shallowest encoder stage, we inject a strong, localized learning signal that encourages the retention of semantic information in the sparser stages.

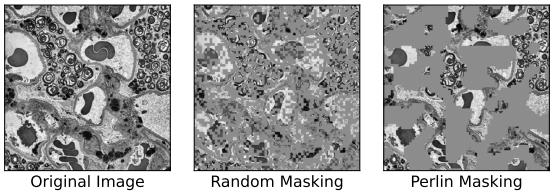
We validate the use of Deep Supervision by monitoring the Normalized Effective Rank [25] $\hat{R}$ of the feature space. For a batch of tokens at a specific downsampling stage with singular values σ_i, $\hat{R}$ is the exponential of the spectral entropy normalized by $\min(N, D)$:

$$\hat{R} = \frac{1}{\min(N, D)} \exp\left(-\sum p_i \log p_i\right), \quad p_i = \frac{\sigma_i}{\sum \sigma_j} \tag{1}$$

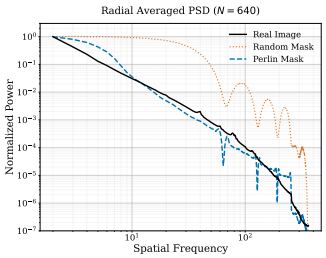
where N and D represent the number of tokens at that stage and the embedding dimension, respectively. Figure 2b visualizes the evolution of $\hat{R}$ across encoder stages. Without Deep Supervision, as the downsampling rate becomes more aggressive, the effective rank plummets at deeper stages; a phenomenon that remains severe even at moderate downsampling rates. Conversely, with Deep Supervision, the network successfully maintains a high effective rank ($\hat{R} > 0.7$) across all stages, largely independent of the downsampling rate.

3.5 Perlin Masking

At high resolutions, random masking becomes increasingly ineffective. As the relative patch size decreases, the model can interpolate missing information from



(a) Illustration of our Perlin Masking strategy with a masking ratio of 0.5 and a patch size of 8.



(b) Spectral analysis of masking strategies.

Fig. 3: Visualization and spectral evaluation of the proposed masking approach.

immediate neighbors without capturing the global structure of biological features. We therefore seek a masking strategy that removes contiguous regions analogous to biological structures, so minimizing reconstruction loss requires learning global context and structural coherence. To achieve this, we utilize Perlin noise [23] to generate masks (Figure 3a).

To justify our masking strategy, we analyze the spectral statistics of the generated masks compared to the statistics of the EM dataset. Natural images, including biological microscopy, exhibit a power spectral density (PSD) that follows a power law decay $S(f) \propto 1/f^{\alpha}$ (typically $\alpha \approx 2$) [11, 26]. We compute the radially averaged PSD for the input images and the binary masks generated by both random and Perlin strategies.

Figure 3b presents the results of the spectral analysis. As expected, the original EM images exhibit the characteristic $1/f$ decay typical of natural signals. Standard random masking disrupts these statistics, introducing a high-frequency plateau indicative of white noise artifacts, alongside a significant spectral gap at mid-to-low frequencies. In contrast, the spectral profile of Perlin masking closely tracks both the slope and magnitude of the real biological signal, with barely any gaps over the entire frequency range.

4 Experiments

4.1 Experimental Settings

Datasets. We pre-train on 187,270 unlabeled glomerulus EM images taken between (7500-15000x) and fine-tune on the Foot Process Width (FPW) dataset (570 train, 235 test images) to segment the Podocyte Glomerular Basement Membrane Interface (PGBMI), Foot Process Filtration Slits (Slits), and background. To evaluate the transferability of AFFMAE and ensure our architectural improvements generalize, we additionally fine-tune on public datasets such as Lucchi++ FIB-SEM mitochondria dataset [4]. These evaluations on public microscopy benchmarks are provided in the supplementary material.

Implementation Details. We evaluate our approach against other efficient masked modeling baselines [15, 19, 31, 34], the standard MAE [13] with a ViT [9]

backbone, and nnUNetV2 [16]. During pre-training, all backbones are scaled to a ∼65M parameter budget. Models are pre-trained for 400 epochs on a single NVIDIA RTX 5090 at 512×512 resolution and an effective batch size of 256. We fine-tune end-to-end for 400 epochs at 512, 768, and 1024 resolutions using a class-weighted combined BCE and Dice loss on the FPW dataset. All MIM models adopt the UperNet decoder [13, 30] matched to the AFFMAE decoder's parameter count (∼10M) for downstream segmentation. FPW performance is evaluated using mean Intersection over Union (mIoU). Full hyperparameter configurations are detailed in the supplementary material.

4.2 Ablation Studies

We ablate key components of our framework—downsampling rate, masking ratio, masking strategy, and deep supervision—using a lightweight 23M parameter AFFMAE variant pre-trained for 300 epochs.

Downsampling Rate. In Table 1a, we analyze the impact of the adaptive downsampling rate. As expected, higher token retention (0.5) yields the highest mIoU (0.6009). However, the performance drop-off at 0.4 is minimal while the required computation decreases by 16%, representing a sweet spot for efficiency. Aggressive downsampling (0.25) leads to a noticeable performance degradation (0.5729), likely due to the lack of tokens to represent spatial details required for segmenting small structures like filtration slits.

Masking Ratio. Table 1b investigates the pre-training masking ratio. Performance peaks at 50%, but drops off noticeably at 75% (0.5739). This contrasts with the standard recommendation for MAE of 75%. We hypothesize this is due to the fine-grained nature of biological structures (e.g., thin basement membranes), where aggressive masking ($> 50\%$) makes it significantly more difficult for the model to learn detailed, high-frequency features during pre-training.

Deep Supervision. We verify the necessity of Deep Supervision in sparse architectures in Table 1c. At $d_s = 0.4$, removing Deep Supervision causes a sharp performance drop to 0.5734. Re-introducing it recovers performance to 0.5908 (+3%). This aligns with our observation in Section 3, confirming that auxiliary losses are critical for preventing feature collapse in deep, sparse layers. Another reasonable approach to resolving feature collapse is to enforce reconstruction solely on the deepest encoder tokens. However, as shown ("Stage 4 Recon."), this yields poor downstream performance. We believe this is due to the lack of the necessary spatial granularity for tokens at the deepest stage to guide dense semantic learning on their own, especially when fine-tuned for segmentation tasks.

Masking Strategy. Table 1d compares standard Random masking against our Perlin masking. Random masking yields 0.5947, while Perlin masking improves this to 0.6009, aligning with our expectations that Perlin Masking allows the model to learn more useful structural information than a Random masking strategy.

Table 1: Ablation Studies. We ablate key components of the AFFMAE framework. Default settings (unless otherwise specified) are $d_s = 0.4$, 0.5 masking ratio, Perlin masking, and deep supervision enabled.

(a) Downsampling Rate.

d_s	GFLOPs	mIoU
0.50	36	**0.6009**
0.40	30	0.5908
0.35	28	0.5753
0.25	23	0.5729

(b) Mask Ratio.

Ratio	mIoU
0.35	0.5853
0.50	**0.5908**
0.75	0.5739

(c) Deep Sup.

Method	mIoU
None	0.5734
Stage 4 Recon.	0.4353
Deep Sup.	**0.5908**

(d) Masking Strategy. $d_s = 0.5$

Strategy	mIoU
Random	0.5947
Perlin	**0.6009**

4.3 Main Results

Having established our architectural configuration, we evaluate AFFMAE against the dense MAE and other efficient MIM frameworks. We first highlight the importance of in-domain pre-training by comparing the downstream mIoU (at 512×512) between the MAE baseline trained from scratch, initialized with official MAE ImageNet-1K weights (ViT-Base, 86M parameters, 1600 epochs), and pre-trained on our in-domain EM dataset:

Scratch	ImageNet-1K	**In-Domain (EM)**
0.476	0.494	**0.606**

Even though standard ImageNet weights provide a boost over random initialization, we note that in-domain pre-training contributes a massive performance improvement. We found that even when comparing against the larger model (86M vs 66M), much larger dataset (1.28M vs 187k), and 4x the epochs (1600ep vs 400ep) the MAE ImageNet-1k pretrained ViT model underperformed the smaller in-domain pre-trained ViT model. Showing the need for efficient pre-training methods for niche domains. All subsequent models in our evaluations are pre-trained on the EM dataset.

Pre-Training Efficiency and Scalability. As shown in Table 3, AFFMAE achieves a pre-training throughput of 151 images/sec, representing a 2x speedup over MAE (76 images/sec) and 1.4x over SimMIM (111 images/sec), at a computational footprint of just 58.7 GFLOPs. More critically, AFFMAE reduces peak pre-training memory to 14.5 GB, a 2x reduction compared to the 29.7 GB and 27.7 GB required by MAE and SimMIM at 32 batch size, respectively. AFFMAE enables efficient model pre-training on a single consumer-grade RTX 5090, whereas dense baselines typically require server-scale infrastructure to be trained at reasonable speeds. Furthermore, natural image models are often pre-trained at lower resolutions and fine-tuned at higher ones, medical imaging relies on high-frequency structures that are destroyed by image downscaling. Pre-training directly at high resolutions is therefore desirable. We reduce the batch size to 16 for this scalability test, as both baselines go Out-of-Memory (OOM)

Table 2: Main downstream fine-tuning results on the FPW dataset. Results are reported as mean ± std across 4 random seeds. Throughput (Img/s) and memory are measured with batch size 4 to ensure OOM-free evaluation across resolutions. Foot Process Width (FPW) MAE denotes the mean per-image pixel error $|\text{fpw}_{\text{pred}} - \text{fpw}_{\text{GT}}|$ scaled to 1024×1024. Bold denotes the best result and underline denotes the second-best result among non-ablation methods at the same resolution.

Method	Res.	FT Img/s	Mem.	Slits IoU (↑)	mIoU (↑)	FPW MAE (↓)
ViT-MAE	512	37	6.7 GB	$\underline{.447 \pm .008}$	$\underline{.606 \pm .004}$	21.18 ± 1.22
	768	11	14.5 GB	$\underline{.488 \pm .003}$	$\mathbf{.627 \pm .001}$	19.69 ± 2.00
	1024	4	25.4 GB	$.494 \pm .009$	$\underline{.630 \pm .005}$	$\underline{19.26 \pm 1.12}$
SimMIM	512	82	4.8 GB	$.414 \pm .002$	$.575 \pm .007$	$\mathbf{19.87 \pm 1.30}$
	768	37	10.4 GB	$.478 \pm .005$	$.616 \pm .002$	$\mathbf{17.58 \pm 1.43}$
	1024	21	18.0 GB	$\underline{.506 \pm .003}$	$.628 \pm .002$	19.70 ± 0.59
HiViT	512	81	**4.1 GB**	$.416 \pm .004$	$.588 \pm .004$	24.93 ± 1.34
	768	36	$\underline{8.6\text{ GB}}$	$.471 \pm .003$	$.616 \pm .004$	22.49 ± 0.23
	1024	18	$\underline{14.9\text{ GB}}$	$.480 \pm .008$	$.623 \pm .003$	19.54 ± 0.86
GreenMIM	512	42	4.7 GB	$.415 \pm .005$	$.562 \pm .005$	31.45 ± 2.00
	768	22	9.8 GB	$.450 \pm .005$	$.581 \pm .004$	20.49 ± 2.43
	1024	12	17.1 GB	$.480 \pm .002$	$.597 \pm .002$	21.69 ± 2.13
MixMAE	512	82	$\underline{4.3\text{ GB}}$	$.399 \pm .001$	$.562 \pm .002$	28.10 ± 1.20
	768	38	8.7 GB	$.459 \pm .002$	$.596 \pm .001$	21.24 ± 1.55
	1024	21	15.3 GB	$.473 \pm .004$	$.607 \pm .003$	20.45 ± 0.65
nnU-Net V2	1024	–	22 GB	$.444 \pm .030$	$.583 \pm .020$	31.60 ± 2.09
AFFMAE (Ours)	512	82	6.2 GB	$\mathbf{.459 \pm .002}$	$\mathbf{.608 \pm .002}$	$\underline{21.16 \pm 1.07}$
	768	36	**7.9 GB**	$\mathbf{.490 \pm .005}$	$\underline{.622 \pm .003}$	$\underline{18.61 \pm 2.11}$
	1024	20	**13.4 GB**	$\mathbf{.514 \pm .006}$	$\mathbf{.633 \pm .002}$	$\mathbf{15.97 \pm 0.92}$
Ablation ViT-MAE Point Dec.	512	18	15.1 GB	$.478 \pm .006$	$.615 \pm .004$	19.31 ± 0.79
	768	–	OOM	–	–	–
	1024	–	OOM	–	–	–

past the 512 resolution at a batch size of 32. Even under this load, MAE and SimMIM fail beyond the 640×640 resolution on a 32GB GPU, while AFFMAE scales, maintaining latency and consuming 22.3 GB of VRAM at 896 (Figure 4).

Downstream Finetuning Performance. Table 2 details the end-to-end fine-tuning performance across multiple input resolutions. Figure 5a presents a qualitative comparison between models. AFFMAE achieves segmentation mIoU on par with the dense MAE baseline at all resolutions while drastically reducing computational overhead. At the 1024×1024 resolution, AFFMAE reaches 0.633 mIoU, closely matching MAE's 0.630 mIoU. In contrast, other efficient MAE baselines such as HiViT and SimMiM plateaus at 0.623 mIoU. This performance parity with MAE is achieved at a fraction of the hardware cost. For upstream pre-training, AFFMAE achieves a throughput of 151 images/s, which is more

Table 3: Pre-training throughput (images/sec) FLOPs, total time, and memory at the 512×512 resolution with a shared batch size of 32.

Method	GFLOPs	Mem. (GB)	Img/s
ViT-MAE	274.5	29.7	76
MixMAE	126.3	20.1	146
SimMIM	119.3	27.7	111
HiViT	96.7	11.7	318
GreenMIM	71.3	11.9	76
AFFMAE (Ours)	**58.7**	**14.5**	**151**

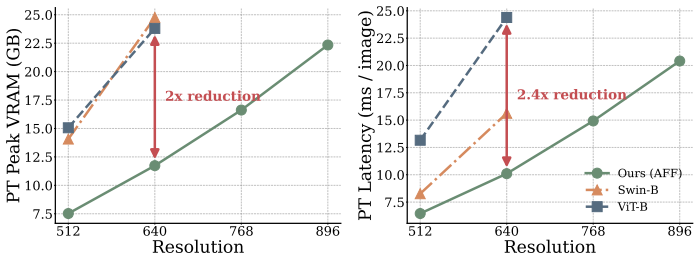
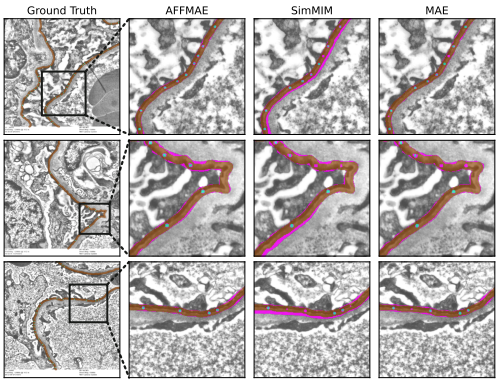


Fig. 4: Pre-training computational scaling (batch-size 16) with input resolution. "PT" means Pre-Training

than 2x faster than MAE (76 images/s), 1.4x faster than SimMIM (111 images/s), while matching or exceeding ViT finetuned mIoU. During downstream fine-tuning at the 1024 resolution, AFFMAE yields a training throughput of 20 images/sec compared to MAE's 4 images/sec (5x speedup) while nearly halving peak memory consumption from 25.4 GB to 13.4 GB. We evaluate AFFMAE against various public datasets in the supplementary materials and observe similar competitive performance at a fraction of the computation.

Segmenting Fine-Grained Structures. A critical failure point for hierarchical models like Swin is their uniform downsampling on a rigid grid. As shown in Table 2, hierarchical models like MixMAE, SimMiM, HiViT, and GreenMiM struggle heavily on the small Filtration Slits class, yielding only $.399 - .415$ IoU compared to MAE (.447 IoU) and AFFMAE (.459 IoU) at 512px. Due to Swin's grid-based patch merging uniformly halving the spatial resolution at each stage, the localized features of these thin slits are irreparably diluted into neighboring patches. Conversely, AFFMAE merges tokens dynamically based on information density rather than a fixed lattice, allowing it to retain high token density around informative regions such as the slits at its sparsest stages (see Figure 6b). This allows AFFMAE to outperform the MAE baseline on the slits class across all resolutions, yielding a Slits IoU of 0.514 at the 1024 resolution. Additional segmentation results are included in the supplementary materials.



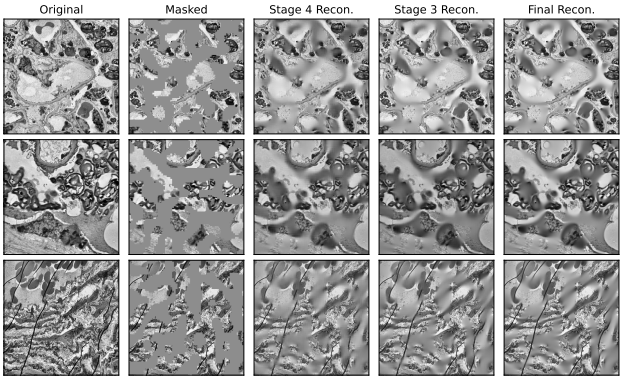
(a) Qualitative segmentation comparison on the FPW dataset with magnified predictions from AFFMAE, SimMIM, and MAE. Dark orange is the PGBMI class, teal is the slits class and segmentation errors are highlighted in pink.

Fig. 5: Qualitative evaluation of downstream segmentation.

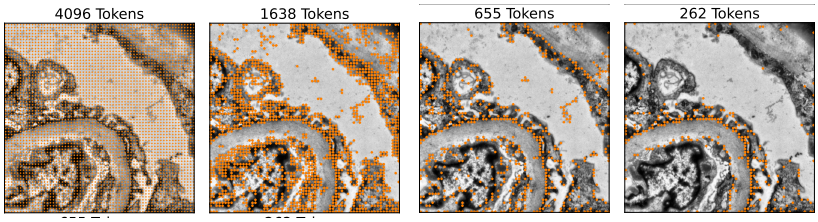
5 Discussion and Limitations

Explainability. Beyond enabling efficient high-resolution training, AFF offers another form of interpretability through image-dependent token locations. In conventional hierarchical backbones, downsampling is grid-aligned and deterministic, so token coordinates are fixed and carry little explicit information about the image. In AFF, token merging is adaptive: tokens are retained and concentrated in regions of high information content, and their locations change with the input. As a result, the learned token trajectories can be visually inspected, and potentially used as downstream task-relevant signals. For FPW segmentation, Figure 6b shows that by the deepest stage, tokens concentrate along the GBM boundary and largely disappear from the visually similar parietal membrane side, providing qualitative evidence that the model allocates token capacity to structures that drive slit detection. This suggests a promising direction for downstream measurement tasks that directly leverage token locations, such as GBM width or slit spacing, alongside dense predictions in an end-to-end framework.

Extension to volumetric imaging. Our work focuses on 2D TEM because it reflects the composition of our EM archive and the current clinical FPW measurement workflow, where 2D sections are far more abundant and routinely used for measurement. However, AFFMAE is not architecturally restricted to 2D. Its core operations—explicit token coordinates, balanced cluster attention, adaptive token merging, and point-based virtual sampling—extend naturally from image-plane tokens $\mathbf{x}_i \in \mathbb{R}^2$ to volumetric tokens $\mathbf{x}_i \in \mathbb{R}^3$. In a 3D setting, cluster-attention neighborhoods become local spatial clusters in voxel or physical coordinate space, anchor selection and token merging operate over volumetric neighborhoods, and decoder offsets/reference points become 3D queries that gather nearby visible tokens from multi-scale volumetric feature sets. This



(a) Qualitative reconstruction visualizations of AFFMAE from various stages of the encoder.



(b) Token merging locations learned by AFFMAE during fine-tuning.

Fig. 6: Qualitative reconstruction and token location visualizations.

extension is appealing for volumetric microscopy since token counts grow cubically with input side length n: for a cubic volume, dense global attention scales as $O(n^6)$ rather than the $O(n^4)$ scaling of 2D inputs. A 3D AFF-style encoder could therefore preserve token density near thin membranes, slits, organelle boundaries, or other sparse high-frequency structures while merging less informative regions. We view volumetric AFFMAE as a natural future direction, with open challenges in efficient balanced clustering, 3D neighborhood lookup, anisotropy-aware masking, and optimized kernels for volumetric token sets.

Limitations. A limitation of AFF is that its irregular token set complicates low-level efficiency compared to dense grid indexing and can increase memory traffic. Although AFF substantially reduces FLOPs, these reductions do not always translate into proportional wall-clock speedups. Grid-based models benefit from simple addressing and neighborhood lookups, whereas AFF relies on neighborhood queries, KNN operations, and gather/scatter patterns that can become bandwidth-limited. We mitigate this by caching neighborhood tables for encoder stages and for the decoder's virtual token sampling, where neighbor queries are invoked repeatedly. However, at very high resolutions, dense lookup tables can grow large enough to exceed cache capacity, making simple grid indexing competitive or faster in some regimes. Improving neighborhood lookup, space-filling strategies, caching, and decoder design will therefore be important for scaling

beyond ∼1200px inputs and for extending AFFMAE to volumetric data. Further kernel optimization also remains an opportunity to close the gap between theoretical FLOP savings and realized throughput.

6 Conclusion

We introduced AFFMAE, a masking-friendly hierarchical pretraining framework for high-resolution microscopy segmentation on desktop-class hardware. AFFMAE combines adaptive off-grid token merging, a mask-agnostic encoder, a point-based decoder, and efficient mixed-precision Triton kernels to retain MAE-style visible-token pretraining while scaling to high-resolution inputs. Across microscopy segmentation benchmarks, AFFMAE matches ViT-MAE performance, and outperforming Swin based methods with small segmentations, at comparable parameter counts while reducing pre-training time by up to 2×, high-resolution fine-tuning time by up to 5×, and peak memory by up to 2× compared to standard MAE. These gains make in-domain self-supervised pretraining practical for laboratories with large private archives but limited access to server-scale infrastructure. More broadly, AFFMAE suggests that learned adaptive token allocation can preserve fine biological structures without paying the full cost of dense global computation, providing a promising path toward scalable pretraining for high-resolution 2D, volumetric, and time-resolved biomedical imaging.

Acknowledgements

We thank Dr. Michael Mauer for generously providing access to an extensive electron microscopy archive used for pretraining in this study. We are grateful for his support in making this research possible.

References

1. Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., Ballas, N.: Self-supervised learning from images with a joint-embedding predictive architecture. arXiv preprint arXiv:2301.08243 (2023)
2. Bao, H., Dong, L., Wei, F.: Beit: BERT pre-training of image transformers. CoRR **abs/2106.08254** (2021), https://arxiv.org/abs/2106.08254
3. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. CoRR **abs/2104.14294** (2021), https://arxiv.org/abs/2104.14294
4. Casser, V., Kang, K., Pfister, H., Haehn, D.: Fast mitochondria detection for connectomics. In: Medical Imaging with Deep Learning. pp. 111–120. PMLR (2020)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 1597–1607. PMLR (13–18 Jul 2020), https://proceedings.mlr.press/v119/chen20j.html

6. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1290–1299 (2022)
7. Cui, Y., Song, Y., Sun, C., Howard, A., Belongie, S.: Large scale fine-grained categorization and domain-specific transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
8. Dao, T., Fu, D.Y., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness (2022), https://arxiv.org/abs/2205.14135
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. CoRR **abs/2010.11929** (2020), https://arxiv.org/abs/2010.11929
10. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. CoRR **abs/2104.11227** (2021), https://arxiv.org/abs/2104.11227
11. Field, D.J.: Relations between the statistics of natural images and the response properties of cortical cells. J. Opt. Soc. Am. A **4**(12), 2379–2394 (Dec 1987). https://doi.org/10.1364/JOSAA.4.002379, https://opg.optica.org/josaa/abstract.cfm?URI=josaa-4-12-2379
12. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., kavukcuoglu, k., Munos, R., Valko, M.: Bootstrap your own latent - a new approach to self-supervised learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 21271–21284. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf
13. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 15979–15988 (2022). https://doi.org/10.1109/CVPR52688.2022.01553
14. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
15. Huang, L., You, S., Zheng, M., Wang, F., Qian, C., Yamasaki, T.: Green hierarchical vision transformer for masked image modeling (2022), https://arxiv.org/abs/2205.13515
16. Isensee, F., Wald, T., Ulrich, C., Baumgartner, M., Roy, S., Maier-Hein, K., Jaeger, P.F.: nnu-net revisited: A call for rigorous validation in 3d medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 488–498. Springer (2024)
17. Laudon, A., Wang, Z., Zou, A., Sharma, R., Ji, J., Tan, W., Kim, C., Qian, Y., Ye, Q., Chen, H., et al.: Digital pathology assessment of kidney glomerular filtration barrier ultrastructure in an animal model of podocytopathy. Biology Methods and Protocols **10**(1), bpaf024 (2025)
18. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: Artificial intelligence and statistics. pp. 562–570. Pmlr (2015)
19. Liu, J., Huang, X., Zheng, J., Liu, Y., Li, H.: Mixmae: Mixed and masked autoencoder for efficient pretraining of hierarchical vision transformers. In: Proceedings

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6252–6261 (2023)
20. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. CoRR **abs/2103.14030** (2021), https://arxiv.org/abs/2103.14030
21. Mei, X., Liu, Z., Robson, P.M., Marinelli, B., Huang, M., Doshi, A., Jacobi, A., Cao, C., Link, K.E., Yang, T., Wang, Y., Greenspan, H., Deyer, T., Fayad, Z.A., Yang, Y.: RadImageNet: An open radiologic deep learning research dataset for effective transfer learning. Radiol. Artif. Intell. **4**(5), e210315 (Sep 2022)
22. NVIDIA, Vingelmann, P., Fitzek, F.H.: Cuda, release: 13.1 (2026), https://developer.nvidia.com/cuda-toolkit
23. Perlin, K.: An image synthesizer. In: Computer Graphics, Vol. 19, No. 3. pp. 287–296 (1985)
24. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/radford21a.html
25. Roy, O., Vetterli, M.: The effective rank: A measure of effective dimensionality (01 2007)
26. Ruderman, D.L., Bialek, W.: Statistics of natural images: Scaling in the woods. Phys. Rev. Lett. **73**, 814–817 (Aug 1994). https://doi.org/10.1103/PhysRevLett.73.814, https://link.aps.org/doi/10.1103/PhysRevLett.73.814
27. Smerkous, D., Mauer, M., Tøndel, C., Svarstad, E., Gubler, M.C., Nelson, R.G., Oliveira, J.P., Sargolzaeiaval, F., Najafian, B.: Development of an automated estimation of foot process width using deep learning in kidney biopsies from patients with fabry, minimal change, and diabetic kidney diseases. Kidney Int. **105**(1), 165–176 (Jan 2024)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Neural Information Processing Systems (2017), https://api.semanticscholar.org/CorpusID:13756489
29. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 568–578 (October 2021)
30. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European conference on computer vision (ECCV). pp. 418–434 (2018)
31. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: A simple framework for masked image modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9653–9663 (June 2022)
32. Xu, Z., Dai, Y., Liu, F., Chen, W., Liu, Y., Shi, L., Liu, S., Zhou, Y.: Swin mae: Masked autoencoders for small datasets. Computers in biology and medicine **161**, 107037 (2023)
33. Yamashita, M., Piaseczna, N., Takahashi, A., Kiyozawa, D., Tatsumoto, N., Kaneko, S., Zurek, N., Gertych, A.: Ai-driven glomerular morphology quantification: a novel pipeline for assessing basement membrane thickness and podocyte

foot process effacement in kidney diseases. Computer Methods and Programs in Biomedicine **268**, 108842 (2025)
34. Yu, J., Li, S., Tan, L., Zhou, H., Li, Z., Li, J.: Hivit: Hierarchical attention-based transformer for multi-scale whole slide histopathological image classification. Expert Syst. Appl. **277**(C) (Jun 2025). https://doi.org/10.1016/j.eswa.2025.127164, https://doi.org/10.1016/j.eswa.2025.127164
35. Zhou, Y., Chia, M.A., Wagner, S.K., Ayhan, M.S., Williamson, D.J., Struyven, R.R., Liu, T., Xu, M., Lozano, M.G., Woodward-Court, P., Kihara, Y., Allen, N., Gallacher, J.E.J., Littlejohns, T., Aslam, T., Bishop, P., Black, G., Sergouniotis, P., Atan, D., Dick, A.D., Williams, C., Barman, S., Barrett, J.H., Mackie, S., Braithwaite, T., Carare, R.O., Ennis, S., Gibson, J., Lotery, A.J., Self, J., Chakravarthy, U., Hogg, R.E., Paterson, E., Woodside, J., Peto, T., Mckay, G., Mcguinness, B., Foster, P.J., Balaskas, K., Khawaja, A.P., Pontikos, N., Rahi, J.S., Lascaratos, G., Patel, P.J., Chan, M., Chua, S.Y.L., Day, A., Desai, P., Egan, C., Fruttiger, M., Garway-Heath, D.F., Hardcastle, A., Khaw, S.P.T., Moore, T., Sivaprasad, S., Strouthidis, N., Thomas, D., Tufail, A., Viswanathan, A.C., Dhillon, B., Macgillivray, T., Sudlow, C., Vitart, V., Doney, A., Trucco, E., Guggeinheim, J.A., Morgan, J.E., Hammond, C.J., Williams, K., Hysi, P., Harding, S.P., Zheng, Y., Luben, R., Luthert, P., Sun, Z., McKibbin, M., O'Sullivan, E., Oram, R., Weedon, M., Owen, C.G., Rudnicka, A.R., Sattar, N., Steel, D., Stratton, I., Tapp, R., Yates, M.M., Petzold, A., Madhusudhan, S., Altmann, A., Lee, A.Y., Topol, E.J., Denniston, A.K., Alexander, D.C., Keane, P.A., & Vision Consortium, U.B.E.: A foundation model for generalizable disease detection from retinal images. Nature **622**(7981), 156–163 (Oct 2023). https://doi.org/10.1038/s41586-023-06555-x, https://doi.org/10.1038/s41586-023-06555-x
36. Zhou, Z., Sodha, V., Pang, J., Gotway, M.B., Liang, J.: Models genesis. CoRR **abs/2004.07882** (2020), https://arxiv.org/abs/2004.07882
37. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
38. Ziwen, C., Patnaik, K., Zhai, S., Wan, A., Ren, Z., Schwing, A.G., Colburn, A., Fuxin, L.: Autofocusformer: Image segmentation off the grid. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18227–18236 (2023)