Structure-Guided Masked Autoencoders for Ultra-High Resolution Scientific Image Understanding

Enzhi Zhang[1,2] **Du Wu**[2,3*] **Rui Zhong**[1] **Cong Ma**[1,2] **Isaac Lyngaas**[4]
Amir Koushyar Ziabari[4] **Xiao Wang**[4] **Peng Chen**[2] **Tao Luo**[5] **Toshio Endo**[3]
Fumiyoshi Shoji[2] **Kento Sato**[2] **Kentaro Uesugi**[6] **Takayuki Nonoyama**[1]
Ryuji Kiyama[1,7] **Masahiro Yoshida**[1] **Tezuka Masaru**[1] **Tetsuya Ishikawa**[8]
Satoshi Matsuoka[2] **Masaharu Munetomo**[1] **Mohamed Wahib**[2*]

[1]Hokkaido University, Sapporo, Japan
[2]RIKEN Center for Computational Science, Hyogo, Japan
[3]Institute of Science Tokyo, Tokyo, Japan [4]Oak Ridge National Laboratory, Tennessee, USA
[5]A*STAR, Singapore [6]Japan Synchrotron Radiation Research Institute (JASRI), Hyogo, Japan
[7]Institute of Industrial Science, The University of Tokyo, Tokyo, Japan
[8]RIKEN SPring-8 Center, Hyogo, Japan

{zhangenzhi, zhongrui, tezuka, munetomo}@iic.hokudai.ac.jp
{nonoyama, m.yoshida}@sci.hokudai.ac.jp rkiyama@iis.u-tokyo.ac.jp
{du.wu, peng.chen, shoji, kento.sato, mohamed.attia}@riken.jp
{lyngaasir, ziabariak, wangx2}@ornl.gov cong.ma.m1@elms.hokudai.ac.jp
endo@scrc.iir.isct.ac.jp {ueken, ishikawa}@spring8.or.jp
luo_tao@a-star.edu.sg matsu@acm.org

Abstract

Self-supervised pre-training with Vision Transformers, including Masked Autoencoders (MAE), is difficult to apply to gigapixel scientific images. Random masking is poorly matched to the structured, multi-scale morphology of scientific data, while uniform tokenization produces prohibitively long sequences that make $O(N^2)$ attention impractical. We propose SGMA, a structure-guided masked autoencoding framework for ultra-high-resolution scientific images. SGMA couples two components: a content-adaptive quadtree tokenizer that compresses gigapixel images into a fixed-length sequence, and a structure-conditioned masking process that biases reconstruction toward spatially informative regions. To stabilize this process across scales, we introduce Damped Accumulation (DA), which aggregates signal-dependent responses across the tree into a structure canvas used to guide masking. The resulting pre-training task preserves fine microstructure while remaining compatible with standard ViT encoders and MAE-style reconstruction. Across electron microscopy, whole-slide optical microscopy, and X-ray CT datasets, SGMA consistently outperforms MAE baselines. At its best-accuracy operating point (N=16,384 tokens), SGMA achieves 95.68% Dice on the 8K×8K×28K SpringXCT dataset, improving over the same-architecture MAE baseline by +9.70 points, and 83.21% Dice on the 32K^2 WSI PAIP dataset, improving by +16.84 points. At a separate, speed-oriented operating point with a matched GPU budget (N=2,048 on PAIP), SGMA provides up to a $24.8\times$ inference speedup while remaining more accurate than MAE (76.08% vs. 62.34% Dice). A controlled ablation at a fixed token budget and decoder attributes a +4.08-point gain to the tokenizer and structure-guided pre-training alone.

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

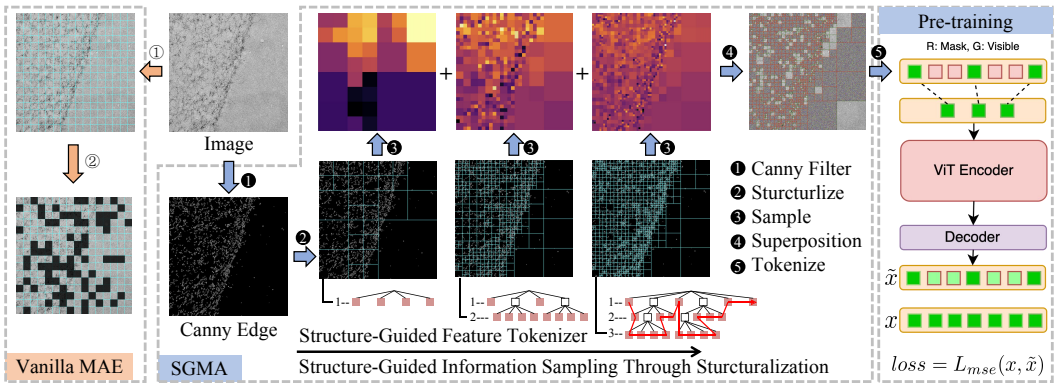


Figure 1: Overview of SGMA. A structure-guided quadtree decomposes the image and accumulates per-level spatial responses by superimposing scale-dependent Gaussian noise. The resulting structure-conditioned noise field determines where masks are placed on the image. When applied to fine-tuning SAM [31] on HydrogelTEM-1K electron microscopy data, SGMA compresses the sequence length by a large margin while maintaining key spatial structures, enabling stronger fine-tuning performance.

1 Introduction

Benefiting from the immense success of large-scale pre-training paradigms, data-driven scientific discovery, particularly in the analysis of scientific image data, is achieving unprecedented depth and breadth. Ultra-High Resolution (UHR) images, such as those from high-throughput optical and electron microscopy [34, 42], pathological scans [30], or X-ray Computed Tomography (XCT) [14, 50], are being acquired at the TB/PB scale. These images contain exceptionally rich, fine-grained, multi-scale information and hierarchical structures. However, directly using advanced models such as the Vision Transformer (ViT) to conduct vision analytics on such UHR images faces two major bottlenecks.

First is the severe scarcity of labels. Although the acquisition of UHR images has become increasingly accessible, their annotation costs are prohibitively high. In terms of total pixels, annotating a single $64K^2$ resolution image is equivalent to annotating 16,384 images at 512 resolution (512×512). More importantly, the precise annotation of scientific images (e.g. cell contours, material defects, or geographical features) relies heavily on time-consuming and costly domain expertise. Second is the intractable computational complexity. The self-attention mechanism, the core of the Transformer model, exhibits quadratic computational and memory complexity with respect to the input sequence length N (i.e., the number of image patches). For UHR images, this results in an astronomical sequence size (e.g. a $64K^2$ image with 16-pixel (4×4) patches yields $16,384^2$ tokens), making standard ViT training, fine-tuning, and even inference intractable.

To address the label scarcity problem, unsupervised pre-training methods like Masked Autoencoders (MAE) [22] are becoming increasingly used in vision models [44, 31]. MAE enables models to learn robust visual representations by reconstructing highly masked patches, leveraging information redundancy and contextual relationships between patches. However, MAE faces challenges [9, 39] with scientific UHR images. More specifically, MAE is not ideal for UHR scientific images because they do not incorporate knowledge of the hierarchical spatial microstructures present in such data. Further, downsampling the UHR image to fit in memory (e.g. to 256x256) destroys all fine-grained, micro-level details, which is often the most critical information. In addition, even if a model is successfully pre-trained on UHR data, it must still process the complete, ultra-long patch sequence during fine-tuning or inference for downstream tasks (e.g. semantic segmentation). The quadratic complexity bottleneck persists.

Existing long-sequence solutions, such as hierarchical ViTs (e.g. Swin Transformer [5] or linear attention approximations [16, 3, 35]), while aimed at reducing computational load, introduce complex model architectures and inter-layer load imbalance (such as overlapped sliding windows) or introduce custom models with fundamental changes to the attention calculation. These modifications are often incompatible with pre-training paradigm of MAE, which relies on global, sparse patch reconstruction. Therefore, a critical open question remains: *can we design a framework that not only enables efficient*

pre-training on unlabeled UHR images of different types of micro-structures, yet simultaneously ensures that downstream task fine-tuning and deployment are computationally feasible?

Inspired by the concept of spatial token compression using mixed-scale patches [59, 58, 65, 66], we propose a novel pre-training framework, SGMA (*Structure-Guided Masked AutoEncoder*, pronounced *Sigma*). First, we capture the spatial layout of fine micro-structures by applying a probabilistic, content-adaptive quadtree that recursively splits the image and extracts multi-level spatial information. Next, we design a structural masking strategy that aligns the mask distribution with the micro-structural patterns and the multi-scale positions discovered by the quadtree. To generate this structure-guided mask, we model the sampling process as the iterative addition (superposition) of Gaussian noise across tree levels. We then train an MAE-based Vision Transformer encoder to perform reconstruction. This design forces the model to learn to discern and reconstruct key structures and multi-scale representations in UHR images during the pre-training phase. Concurrently, the framework allows for efficient inference in downstream tasks. Most importantly, the framework SGMA approach generalizes to images of very different micro-structures and resolutions: Transmission Electron Microscopy (TEM), optical WSI microscopy, and X-ray CT images at $< 0.1\,\mathrm{nm}$, $12\,\mu\mathrm{m}$, and $200\,\mathrm{nm}$ image resolutions, respectively. Our contributions are summarized as follows:

- *Structure-Guided Masked AutoEncoder*. A novel pre-training framework for unlabeled ultra-high resolution (UHR) images. SGMA learns representations through a task that jointly performs Structure-Guided Feature Tokenizer (SGFT) tokenization and Damped Accumulation (DA) structure-conditioned sampling. This improves the performance of standard MAE since the model is geared to learn to reconstruct spatial regions with concentration of details, while mitigating the critical downstream $O(N^2)$ computational bottleneck (in fine-tuning and inference) that standard MAE-trained models fail to address, enabling efficient end-to-end deployment.
- *Ultra-High Resolution Segmentation*. We conduct extensive segmentation experiments on three multi-scale UHR datasets (TEM: *HydrogelTEM-1K*, X-ray CT: *SpringXCT-8K*, and WSI microscopy: *PAIP-32K*). Results show that SGMA significantly outperforms MAE-pretrain baselines in both accuracy and speed, at two distinct operating points of the same method on the 32K PAIP dataset: a +16.84-point Dice improvement at the best-accuracy setting (N=16,384), and a $24.8\times$ inference speedup at a speed-oriented setting with matched GPU budget (N=2,048), where SGMA still exceeds MAE by +13.74 Dice. A controlled one-factor-at-a-time ablation (Section 5.2) further isolates the contribution of each component at a fixed token budget and decoder.
- *Downstream Task Analyze*. We validate the real-world efficacy of our method by applying our high-precision segmentation to a downstream scientific task: quantitative analysis of the pore network structure from the zero-shot SpringXCT 3D dataset and high-precision skeleton from HydrogelTEM dataset, showcasing its practical value in scientific discovery workflows.

2 Related Work

Our SGMA framework lies at the intersection of three core research directions: self-supervised learning through masked image modeling, efficient Transformer architectures for long-sequence, and high-resolution scientific imaging [34, 42, 30, 14, 50].

Masked Image Modeling. Self-supervised learning(SSL) aims to learn rich data representations without human-provided labels. While early methods focused on contrastive learning [10, 21], Masked Image Modeling (MIM) has recently emerged as a highly effective and scalable approach, inspired by the success of Masked Language Modeling (e.g. BERT [18]) in NLP. Masked Autoencoder (MAE) [22] is a prominent example, operating by randomly masking a large portion (e.g. 75%) of input image patches and tasking a model to reconstruct the original, uncorrupted pixels in the masked regions. Its asymmetric encoder-decoder architecture, where the encoder processes only a small subset of visible patches, makes pre-training highly efficient. However, MAE's design presents two fundamental challenges for our target domain of giga-pixel scientific images. First, its *uniform patching* scheme becomes computationally intractable at giga-pixel resolutions. Second, its *random masking* task is designed for semantic in-painting, which may be suboptimal for scientific images where the pre-training goal should be learning fine-grained textures, cell morphologies, and subtle, physically-grounded anomalies. Our work addresses both limitations.

Efficient Transformers for Long Sequences. A standard ViT's self-attention mechanism has a computational and memory cost of $O(N^2)$ with respect to the sequence length N, which increases as

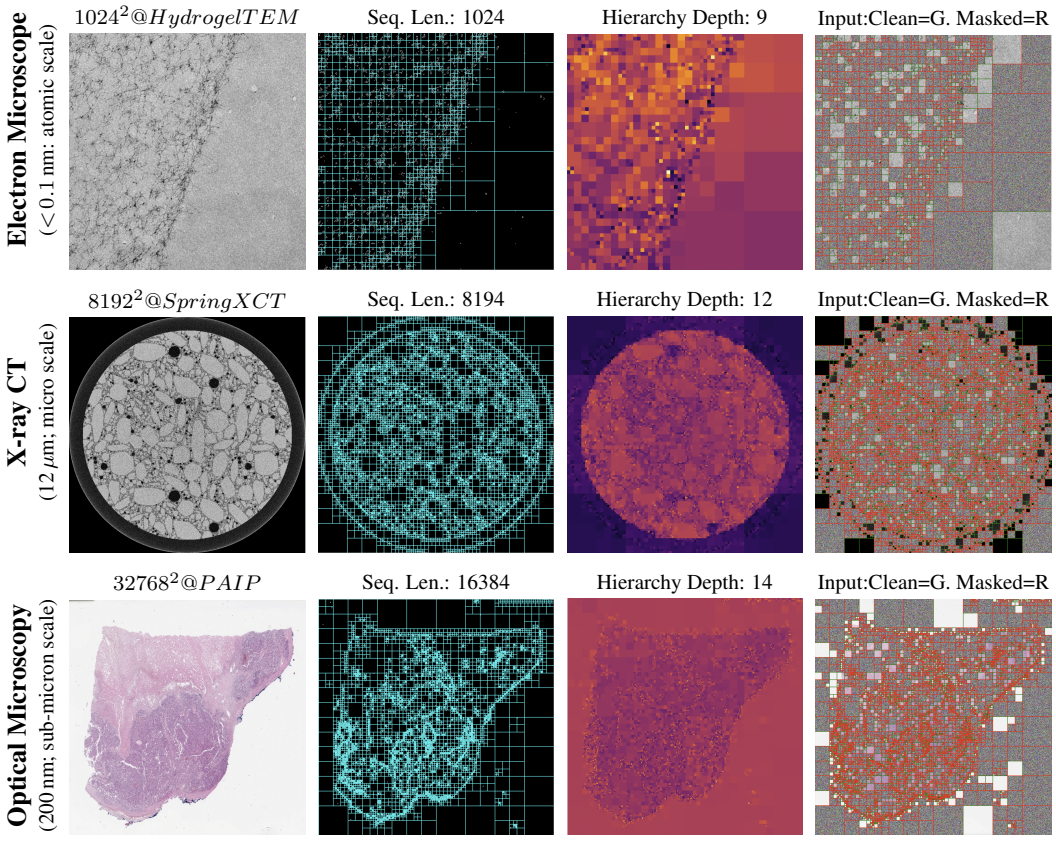


Figure 2: Adaptive structural patches and spatial information from transmission electron microscopy: HydrogelTEM-1K [34], X-ray CT: SpringXCT-8K [56], and WSI microscopy: PAIP-32K [30] datasets. These datasets span centimeter to nanometer scales and include biomedical, macroscopic, and microscopic samples. `SGMA` adapts to the multi-scale micro-structure of each domain for MAE pre-training.

the patch size decreases. For giga-pixel images, N can easily exceed 10^6 patches, making standard self-attention infeasible. To mitigate this, several strategies have been developed. *Sequence parallel methods* and efficient kernels, including Deep-Speed Ulysses [26], LightSeq [37], RingAttention [38], LLS [57], and FlashAttention [17, 15], optimize the computation. *Linear approximation methods* reduce complexity through techniques like spectral attention [16, 3, 35], low-rank approximation [13, 28], sparse attention matrix sampling [12, 33, 47, 2, 63], or infrequent self-attention updates [61, 43]. These methods reduce the computational load, but excessive reduction can lead to performance loss [51]. *Hierarchical training of ViTs*, where multiple transformers are trained at different resolution levels [52, 8, 6, 62], is another approach, but using multiple transformers increases training time and complexity. Recently, *quadtrees* have been used in image segmentation to reduce attention cost, e.g. quadtree/octree attention or patch pre-processing [54, 25, 65]. Both of those approaches employ quadtrees, but involve additional model complexity or need expert knowledge for hyperparameters. Our Structure-Guided Feature Tokenizer (SGFT) mechanism differs fundamentally by using the quadtree purely as an *adaptive input tokenizer* to generate a fixed-length sequence for a standard ViT, and most importantly, we *interleave the tree construction with our structure-conditioned sampling*. This joint formulation of a multi-scale data structure and a physics-inspired pre-training objective is the core novelty of our `SGMA` framework. A more detailed side-by-side comparison with representative long-sequence and hierarchical Transformer models is provided in Appendix A.5 (Table 4), offering clearer context for our `SGMA`'s positioning.

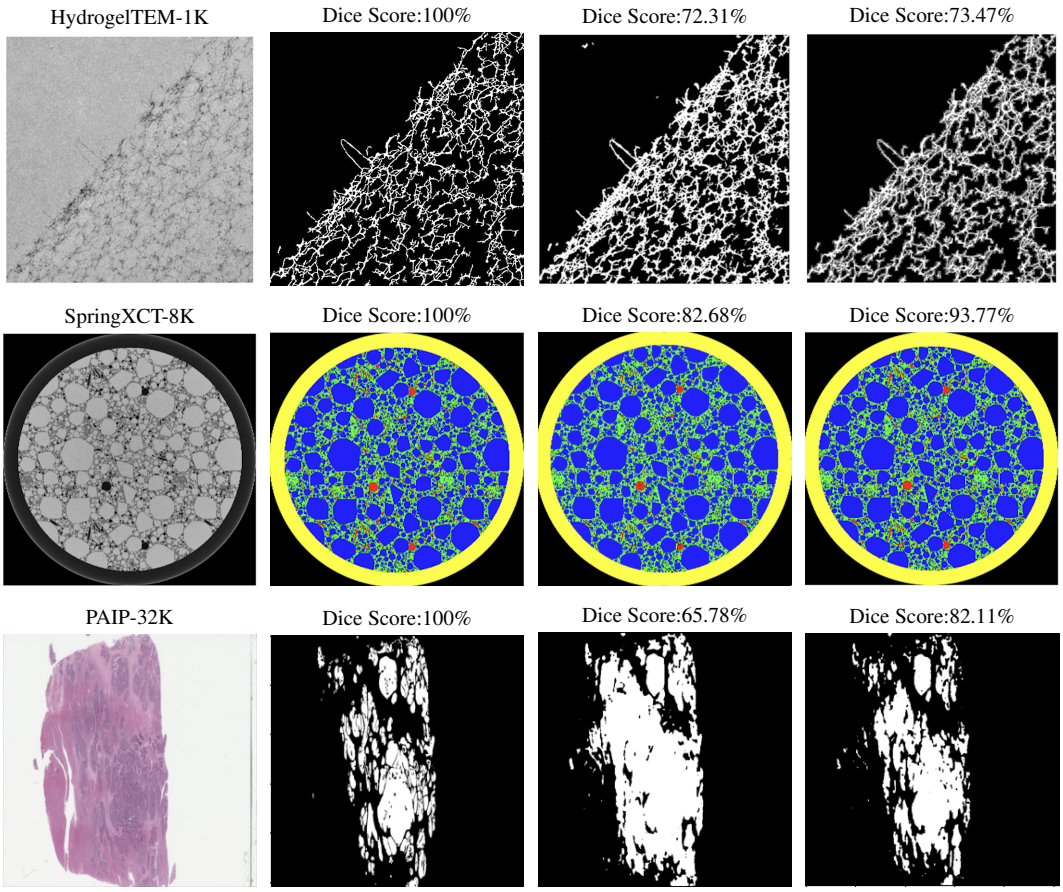


Figure 3: We compared the segmentation difference between MAE-SAM [31] and SGMA-SAM on TEM HydrogelTEM-1K, X-ray CT SpringXCT-8K, and WSI PAIP-32K datasets. At the same GPU compute budgets, SGMA-SAM can go down to patch size of 8×8 (vs. $1,024 \times 1,024$ at best for SAM before going OOM). As a result, SGMA-SAM can extract and express mask details better than MAE-SAM, with the gap in accuracy favoring SGMA-SAM as the resolution gets higher.

3 Methodology

SGMA consists of three parts, as shown in Figure 1: preprocessing [45], multi-scale tree construction, and MAE pre-training. The key component is structure-conditioned modeling, realized through Structure-Guided Feature Tokenizer (SGFT) construction and information sampling.

3.1 Structure-Guided Feature Tokenizer

We construct a multi-scale extractor over the image x by recursively partitioning its domain. However, the standard greedy selection, $n^* = \arg\max V(n_j)$, is inherently sequential and non-parallelizable, creating a computational bottleneck as the number of active k increases. To address this, we "soften" the selection by reformulating the splitting criteria as a probabilistic sampling process. We define a probability distribution π over the set of active tokens $\mathcal{A}_k$, where the probability of selecting a token n_j is proportional to its salience score. The transition from current token set $\mathcal{A}_k$ to next set $\mathcal{A}_{k+s-1}$ is defined recursively:

$$\mathcal{A}_{k+s-1} = \begin{cases} \mathcal{A}_k, & \text{if } k \geq N \\ (\mathcal{A}_k \setminus \{n^*\}) \cup \text{Split}(n^*), & n^* \sim \pi(\cdot|\mathcal{A}_k, \tau) \end{cases} \tag{1}$$

where N is the target fixed sequence length for the model. The probability distribution π is defined using the softmax function with a temperature parameter τ:

$$\pi(n_j|\mathcal{A}_k, \tau) = \frac{\exp(V(n_j)/\tau)}{\sum_{n_i \in \mathcal{A}_k} \exp(V(n_i)/\tau)} \tag{2}$$

Here, $V(n_j)$ is the scoring criterion (e.g. the sum of edge map values) for token n_j. The temperature τ controls the "softness" of the selection: as $\tau \to 0$, π approaches the original deterministic 'argmax' selection, while as $\tau \to \infty$, it approaches a uniform random selection. Beyond removing the sequential bottleneck, this softmax-batched formulation is also substantially faster in wall-clock terms than the SHF/AP-style $\arg\max$ tokenizer (up to $20\times$ at $N{=}10^6$ tokens) at matched tokenization quality; we report a full per-resolution comparison in Appendix A.6 (Table 5).

3.2 Structure-Guided Information Sampling

We define the state of the joint process at step k (representing a tree with k tokens) as the tuple $X_k = (\mathcal{A}_k, C_N^{(k)})$, where $\mathcal{A}_k$ is the set of active tokens and $C_N^{(k)}$ is the full-resolution cumulative structure canvas. The transition $X_k \to X_{k+s-1}$ is defined recursively:

$$X_{k+s-1} = \begin{cases} X_k, & \text{if } k \geq N \\ (\mathcal{A}_{k+s-1}, C_N^{(k+s-1)}), & \text{otherwise} \end{cases} \tag{3}$$

where N is the target sequence length. The process is initialized at $k = 1$ with the root state $X_1 = (\mathcal{A}_1, C_N^{(1)})$. This state consists of the initial token set $\mathcal{A}_1 = \{n_0\}$, which contains only the root node n_0 (at depth $j = 0$), and the initial structure canvas $C_N^{(1)}$. The canvas is populated with a single "root information" layer $\epsilon_0 \sim \mathcal{N}(0, \sigma_0^2 x)$, where the global variance $\sigma_0^2 = V_T(x)/\sqrt{0+1} = V_T(x)$.

For any state X_k where $k < N$, the transition $X_k \to X_{k+s-1}$ is executed. This joint update consists of two simultaneous actions. First, a structural update determines the new token set $\mathcal{A}_{k+s-1}$ by sampling a parent node n^* (at depth $j-1$) $\sim \pi(\cdot|\mathcal{A}_k, \tau)$ and replacing it with its s children, as defined in eq. (1). Second, a concurrent information update forms the new canvas $C_N^{(k+s-1)} = C_N^{(k)} + \Delta C_N^{(k+1)}$. The sparse update tensor $\Delta C_N^{(k+1)}$ adds a new, localized noise component $\epsilon_{\text{new}}^{(c)}$ for each new child node $n^{(c)}$ (now at depth j). The variance of this new information is governed by $\text{Var}(\epsilon_{\text{new}}^{(c)}) = \frac{\sigma_T^2(c)}{\sqrt{j+1}}$ where $\sigma_T^2(c) = V_T(p^{(c)})$ is the local target variance from the child's patch. This process repeats until the tree contains N tokens.

This generative process defines a sequence of nested information additions along the ancestral path $\mathcal{P}(n_i) = (n_0, \ldots, n_d)$ for any token n_i. Due to the additivity of Gaussian distribution, the final distribution N_i is the sum of all independent components ϵ_j added along this path: $N_i = \sum_{j=0}^{d} \epsilon_j$. The final variance $\sigma_{\text{final},i}^2$ is therefore the sum of the variances of these components. Thus, the total variance is just the sum of depth-scaled local variances:

$$\sigma_{\text{final},i}^2 = \sum_{j=0}^{d} \text{Var}(\epsilon_j) = \sum_{j=0}^{d} \frac{\sigma_T^2(j)}{\sqrt{j+1}} = \sum_{j=0}^{d} \frac{1}{\sqrt{j+1}\,|S_j|} \sum_{(h,w)\in S_j} x(h,w) \tag{4}$$

where p_j is the patch of ancestor node n_j at depth j, and S_j its pixel coordinates. The accumulation grows sub-linearly with depth, avoiding variance explosion.

3.3 SGMA Pre-training Framework

Image Encoder and Input. After completing adaptive structural patching (section 3.1) and structure-conditioned sampling distribution generation, we obtain the adaptive structural patch sequence $\mathcal{P} = \{p_i\}_{i=1}^{N_h}$ and the final structure-conditioned sampling distribution N_i for each patch. We randomly select a subset of indices $\mathcal{I}_{\text{masked}} \subset \{1, \ldots, N_h\}$ (e.g. 75% suggested in [22]) to be corrupted. We then form a visible input sequence $\mathcal{P}' = \mathcal{P} - \mathcal{I}_{\text{masked}}$. We use a standard ViT encoder f_θ [19]. Following the Masked Autoencoder (MAE) paradigm, the encoder f_θ processes the *structure-aware*, clean sequence $\mathcal{P}'$.

Reconstruction. A lightweight decoder g_ϕ is attached to the encoder f_θ. The decoder receives the full sequence of encoded representations $Z = f_\theta(\mathcal{P}')$ and is tasked with reconstructing the *original,*

clean version of *only the noised patches*. The model learns to leverage the clean patches ($p_j \in \mathcal{P}'$) as context to reconstruct corrupted patches ($i \in \mathcal{I}_{\text{masked}}$). This pre-training objective function $\mathcal{L}_{\text{SGMA}}$ is:

$$\min_{\theta,\phi} \mathbb{E}_{I\sim\mathcal{D},\, \mathcal{I}_{\text{masked}}\sim\text{Mask}} \left[\sum_{i\in\mathcal{I}_{\text{masked}}} \mathcal{L}_{\text{MSE}}\Big(g_\phi\big(f_\theta(\mathcal{P}'), i\big),\, p_i\Big) \right] \tag{5}$$

Where I is an image from dataset $\mathcal{D}$, $\mathcal{P}'$ is the corrupted input sequence, p_i is the clean patch, $g_\phi(\cdot, i)$ is the decoder prediction for the i-th patch, and $\mathcal{L}_{\text{MSE}}$ is the L2 loss.

3.4 Fine-tuning

Fine-tuning Encoder . After pre-training with $\mathcal{L}_{\text{SGMA}}$, the reconstruction decoder g_ϕ is discarded. We load the weights of our pre-trained encoder f_θ and adapt it for downstream tasks, such as segmentation. We retrofit our `SGMA` scheme into a segmentation framework, minimally adapting the encoder f_θ to process high-resolution inputs.

Fine-tuning Objective. During the fine-tuning phase, we do not directly generate full-resolution masks Y but instead generate encoded mask logits $Y' = \{y'_i\}_{i=1}^N$ at the patch level. This saves memory and reduces backpropagation costs. We summarize the fine-tuning objective function:

$$\min_{\theta} \mathbb{E}_{(I,Y)\sim\mathcal{D}} \left[\sum_{i=1}^{N} \mathcal{L}_{\text{seg}}\Big(h_\theta(Z_i),\, T_{\text{tree}}(Y)_i\Big) \right] \tag{6}$$

Where N is the total number of patches, θ represents the trainable weights of the encoder and a simple linear head h_θ, $\mathcal{L}_{\text{seg}}$ denotes a segmentation loss (e.g. Dice and Cross-Entropy), $Z_i = f_\theta(\mathcal{P})_i$ is the encoder output for patch i, and $T_{\text{tree}}(Y)_i$ is the i-th patch from the ground-truth mask Y, generated using the same hierarchical tree operator T_{tree} applied to the input I.

4 Experiments

4.1 High Resolution Scientific Image Datasets

HydrogelTEM-1K: is a nanoscale hydrogel network dataset [34, 42] visualized using transmission electron microscopy (TEM). To preserve structure, the target PAMPS network was interpenetrated with a PDMAAm "skeleton network". A novel mineralization technique using Fe^{3+} ions (forming AFO nanoparticles) provided the necessary electron contrast. The resin-substituted samples were cut into 100 nm thick specimens and observed by Hitachi and JEOL TEM/STEM systems (100-200 kV).

SpringXCT-8K: A high-resolution XCT dataset of 46 concrete and asphalt specimens scanned at SPring-8 (200 keV) [53]. Each 3D volume has size $8{,}192 \times 8{,}192 \times 28{,}800$. For *pre-training*, we use $308{,}000$ slices of size $8{,}192 \times 8{,}192$. For *fine-tuning*, we build a simulation pipeline to generate ultrahigh-resolution data. We start from virtual microstructures [69], resample them to 12.02 μm resolution, and obtain $1{,}500$ slices of size $8{,}192 \times 8{,}192$. We then simulate the imaging process with realistic attenuation, and add degradations such as sparse angles, Poisson/Gaussian noise, and ring artifacts. The sinograms are reconstructed using FBP [49, 4]. This yields 60 synthetic volumes with ground-truth labels and corresponding reconstructions (see appendix) for training.

PAIP-32K: [30] is a high-resolution, real-world liver cancer pathology dataset, with sample resolutions up to $64K^2$, significantly surpassing those of conventional image datasets. PAIP contains $2,457$ Whole-Slide Images (WSIs). When lower resolutions are needed, we downscale the images to uniform sizes of $[16,384, 32,768, 65,536]$ square pixels. For training, we randomly select 70% of samples, 10% for validation, and 20% for testing. All datasets are shuffled and normalized to $[0.0, 1.0]$ as input for the model.

4.2 Evaluating Models: Baselines & Proposed

Baseline Models: Our primary baseline is the standard Masked AutoEncoder (MAE) [22] pre-training strategy. We evaluate this on the widely-used segmentation model SAM [31], which uses a ViT encoder (ViT-Base, Large, or Huge) pre-trained with MAE; we refer to this configuration as MAE-SAM. We also benchmark against U-Net-style models [46], which use a contraction-expansion

Table 1: Quantitative segmentation results (Dice Score % ↑) on all datasets. Our SGMA pre-training method is compared against standard baselines and MAE pre-training on identical architectures. The "Improvement" column shows the Dice Score gain of our SGMA model over its equivalent MAE-trained baseline (e.g. FT-SGMA-UNETR vs. FT-MAE-UNETR).

Dataset	Model	Patch Size	Sequence Length	Dice Score (%)	Performance Improvement
HydrogelTEM-1K	SAM 2 [44]	8	16384	69.87	-
	UNet [46]	-	-	71.03	-
	UNETR [20]	16	4096	70.59	-
	FT-MAE-SAM	8	16384	72.31	(Baseline)
	FT-MAE-SAM 2	8	16384	73.12	(Baseline)
	FT-MAE-UNETR	16	4096	74.56	(Baseline)
	FT-SGMA-SAM (ours)	2	16384	73.48	+1.17 (vs. FT-MAE-SAM)
	FT-SGMA-SAM 2 (ours)	2	16384	74.23	+1.11 (vs. FT-MAE-SAM 2)
	FT-SGMA-UNETR (ours)	2	16384	**75.33**	**+0.77** (vs. FT-MAE-UNETR)
SpringXCT-8K	SAM 2 [44]	128	4096	83.81	-
	UNet [46]	-	-	78.73	-
	UNETR [20]	128	4096	82.23	-
	Swin-UNETR [55]	-	-	83.96	-
	FT-MAE-SAM	128	4096	82.68	(Baseline)
	FT-MAE-SAM 2	128	4096	85.98	(Baseline)
	FT-MAE-UNETR	128	4096	86.12	(Baseline)
	FT-SGMA-SAM (ours)	2	16384	93.77	+11.09 (vs. FT-MAE-SAM)
	FT-SGMA-SAM 2 (ours)	2	16384	**95.68**	**+9.70** (vs. FT-MAE-SAM 2)
	FT-SGMA-UNETR (ours)	4	8194	91.93	+5.81 (vs. FT-MAE-UNETR)
PAIP-32K	SAM [31]	1024	1024	62.34	-
	UNet [46]	-	-	61.38	-
	UNETR [20]	1024	1024	74.96	-
	TransUNet [7]	-	-	69.88	-
	FT-MAE-SAM	1024	1024	65.78	(Baseline)
	FT-MAE-SAM 2	1024	1024	66.37	(Baseline)
	FT-MAE-UNETR	1024	1024	77.24	(Baseline)
	FT-SGMA-SAM (ours)	4	16384	82.11	+16.33 (vs. FT-MAE-SAM)
	FT-SGMA-SAM 2 (ours)	4	16384	**83.21**	**+16.84** (vs. MAE-SAM 2)
	FT-SGMA-UNETR (ours)	8	8194	81.23	+3.99 (vs. FT-MAE-UNETR)

path with skip connections. Our main baseline in this category is the UNETR [20], for which we create an MAE-UNETR variant by pre-training its ViT encoder by the standard MAE objective.

Proposed Models: We replace MAE pre-training with the proposed SGMA objective (Eq. 5) and apply it to the same architectures, yielding SGMA-SAM and SGMA-UNETR. For SAM-based models, we use symmetrical depatching [66], eliminating the need for a heavy decoder. For SpringXCT-3D data, we apply the SAM 2 [44] encoder to 2D slices and stack the outputs for final prediction. For comparison, we include U-Net [46] (ImageNet-pretrained via `timm` [27]), UNETR [20], and Swin-UNet [5].

5 Evaluation and Results

5.1 High-Precision Image Segmentation

Quantitative Results. As shown in Table 1, our SGMA pre-training method consistently outperforms standard baselines and MAE-based pre-trained models across all three diverse datasets. The performance improvement correlates significantly with image resolution. On the **HydrogelTEM-1K** dataset, the improvement is steady but modest; our FT-SGMA-UNETR achieves the top score of 75.33%, a **+0.77** point increase over its corresponding FT-MAE-UNETR (74.56%). On the ultra-high resolution datasets, the advantage of SGMA becomes exceptionally pronounced. On the **SpringXCT-8K** dataset, our FT-SGMA-SAM 2 model achieves a state-of-the-art Dice score of 95.68%, a massive **+9.70** point improvement over the FT-MAE-SAM 2 baseline (85.98%). This improvement is most dramatic on the **PAIP-32K** dataset. Here, our FT-SGMA-SAM 2 model (83.21%) surpasses the equivalent FT-MAE-SAM 2 model (66.37%) by **+16.84** points. This demonstrates that while standard MAE performs poorly when handling ultra-high resolution inputs, our SGMA pre-training, which leverages adaptive structural patching, provides a far more effective representation for capturing both global context and micro-level details.

Segmentation Results. Figure 2 provides a visual comparison on the HydrogelTEM-1K, SpringXCT-8K, and PAIP-32K datasets. Baselines like UNet produce blurry and incomplete boundaries. MAE-UNETR (which was forced to use large patch sizes during pre-training) fails to resolve individual

Table 2: Controlled one-factor-at-a-time ablation on SpringXCT-8K (Dice %). Backbone (SAM 2 encoder), downstream head, and pre-training schedule are fixed in all rows; Rows A–D also fix N=4096 and the dense decoder. Rows A–E report mean ± std over 3 seeds; Row F (N=16,384, 128 GCDs) is single-seed. Δ is relative to the previous row.

Row	Tokenizer	Decoder	N	Masking	Dice (%)	Isolated factor (Δ)
A	Uniform-128	dense	4096	random	85.98 ± 0.43	baseline (uniform tiling)
B	SGFT	dense	4096	random	86.47 ± 0.31	adaptive tokenization (+0.49)
C	SGFT	dense	4096	structure	89.49 ± 0.32	structure-guided masking (+3.12)
D	SGFT	dense	4096	struct.+DA	90.06 ± 0.42	Damped Accumulation (+0.57)
E	SGFT	depatch	4096	struct.+DA	88.53 ± 0.36	depatch head (−1.53; −40% mem.)
F	SGFT	depatch	16384	struct.+DA	95.68	sequence length (+7.15)

structures, merging them into a single mass. In contrast, our SGMA-UNETR, having been pre-trained on adaptive-scale patches via SGFT, successfully resolves fine, nm-scale structures, producing a segmentation mask that is crisp and topologically accurate. This demonstrates our model's ability to learn meaningful multi-scale features. Although the main focus of the paper is ultra-high-resolution 2D imaging, the long-sequence problem also appears in 3D volumetric data; on standard 3D medical benchmarks (BTCV, KiTS19) SGMA-SAM 2 matches or marginally exceeds strong specialized baselines (Swin UNETR-V2, U-Mamba, nnFormer) on Dice while being 2.58×–7.15× faster thanks to SGFT-based sequence compression (Appendix A.7, Table 6). We also report a PAIP-16K classification comparison against ViT, HIPT, and SHF-ViT, together with a Damped Accumulation ablation in Appendix A.8 (Table 7). SGMA-ViT achieves the best performance in both Top-1 accuracy and Dice. The NoDA ablation further indicates that the impact of DA is task-dependent, with a marginal gain for classification (+0.88 Top-1) but a notable improvement for segmentation (+3.00 Dice), suggesting that DA primarily benefits dense prediction by introducing multi-scale spatial structure.

5.2 Controlled Ablation: Isolating the Source of the Gain

In Table 1, tokenizer, patch size, and token budget change together. To disentangle them, Table 2 changes *one factor per row* on SpringXCT-8K with a fixed backbone, head, and schedule; Rows A–D also share the token budget and dense decoder.

At an identical budget and decoder (A→D), SGMA already gains **+4.08** Dice, so the improvement is not an artifact of a larger budget or finer patches. Structure-guided masking (B→C, +3.12) is the largest pre-training contribution; adaptive tokenization (+0.49) and DA (+0.57) are smaller, on the order of the run-to-run std. The depatch head trades −1.53 Dice for ∼40% less memory, which makes N=16,384 reachable (a dense decoder runs out of memory there); scaling to it adds +7.15.

5.3 Compute Efficiency: Training and Inference

The Baseline Bottleneck. Standard MAE-based models (MAE-UNETR, MAE-SAM1/2) must use a uniform grid. On a $32,768 \times 32,768$ image, even a large 32×32 patch size yields a sequence of $N = (1024)^2 \approx 1M$ patches, which is computationally intractable and causes Out-of-Memory (OOM) errors on any available hardware. To avoid OOM, baselines use very large patches (e.g. 128×128 or 256×256), losing the fine-grained details needed for our tasks.

SGMA Efficiency. SGMA compresses the input into a *fixed-length* sequence K (e.g. $K = 1024$ or $K = 4096$) *regardless* of the input resolution. This adaptive compression allows us to process the 32K image with the same computational budget as the 1K image. As shown in Table 3, our SGMA pre-training framework achieves a dramatic improvement in both training and inference speed compared to a MAE-based baseline (which must operate on large patches to avoid OOM). The setup of Table 3 is intentionally biased *against* our method on both axes. On the MAE side, the chosen sequence length is the largest the GPU budget can fit before going OOM, i.e. MAE is reported at its hardware ceiling. On the SGMA side, we deliberately use a sub-optimal short sequence (e.g. N=2,048 on PAIP-32K, far below the N=16,384 used by our best-Dice configuration in Table 1), so that compute matches and we can isolate the speedup. Even with this handicap, SGMA still beats MAE on Dice (e.g. 76.08 vs. 62.34 on PAIP-32K, a +13.7-point margin). Beyond raw segmentation accuracy, we also evaluate SGMA in a zero-shot setting on real samples and use the resulting masks for downstream

Table 3: Inference efficiency under the *same compute budget* (identical GPU count per row) for MAE-SAM and our SGMA-SAM. The MAE row uses the longest sequence that fits within its GPU budget before OOM, while SGMA uses a deliberately shorter sequence (e.g. N=2,048 on PAIP-32K vs. N=16,384 in Table 1) to match wall-clock compute; the reported speedups are therefore conservative.

Datasets	GPUs	Model	Time (s/img)	Seq. Len.	Dice (%)	Speedup (×)
HydrogelTEM-1K	4	MAE-SAM-8 [22, 31]	0.38261	16384	72.31	1×
	4	**SGMA-SAM** (ours)	**0.09913**	**8194**	**72.56**	**3.86**×
SpringXCT-8K	128	MAE-SAM-128 [22, 31]	2.5168	4096	82.68	1×
	128	**SGMA-SAM** (ours)	**0.3512**	**1024**	**89.37**	**7.17**×
PAIP-32K	512	MAE-SAM-1024 [22, 31]	8.9812	16384	62.34	1×
	512	**SGMA-SAM** (ours)	**0.3663**	**2048**	**76.08**	**24.8**×

scientific analysis (pore-network extraction on SpringXCT-8K and hydrogel skeletonization on HydrogelTEM-1K). Full quantitative and qualitative results are deferred to Appendix A.3 due to space constraints. Full hyperparameter settings (masking ratio, sequence length, model size) are reported in Appendix A.2.

5.4 Limitations and Effect of Hyperparameters

We summarize how the three main hyperparameters of SGMA affect performance, and which of these effects translate into practical limitations. The full visualizations are in Appendix A.2 (Figures 4–6).

Masking ratio. r (Figure 4). Reconstruction quality degrades gracefully as r grows from 0.25 to 0.75, and the per-quadtree-level sampling preserves structurally important regions across ratios. We did not observe a clear failure mode in this range, so the choice of r is *not* a practical limitation. *Limitation:* **Sequence length.** N (Figure 5). The fixed SGFT token budget trades coverage for detail; under-sized N on highly heterogeneous inputs (e.g. PAIP-32K with N=1024) leaves fine micro-structures coarsely tokenized and produces a measurable drop in Dice. Choosing N therefore requires a per-dataset budget tuned to the input resolution and structural density. **Model size** (Figure 6). Reconstruction quality improves monotonically from ViT-L to ViT-XL on the same input under identical masking. This is the expected scaling behavior, so we do not regard model size as a limitation *per se*, although the largest variants inherit the standard ViT pre-training cost.

Beyond hyperparameters: Canny-edge dependence. The SGFT salience score is computed from a Canny edge map, so the quality of the tree decomposition depends on whether edges faithfully reflect task-relevant micro-structure. On images with weak gradients, dominant noise, or texture-defined (rather than edge-defined) regions, the tree may over-tokenize uninformative areas and under-tokenize the true regions of interest, which would bound SGMA's gains over uniform patching.

6 Conclusion

We presented SGMA, a novel and effective framework that makes MAE-style pre-training feasible for ultra-high resolution scientific images. Using structure-guided masked autoencoders, SGMA avoids the $O(N^2)$ bottleneck and preserves important fine-scale structures. Across TEM, XCT, and WSI datasets, SGMA achieves consistently better segmentation accuracy than MAE, including a +16.84-point Dice gain on 32K pathology at the best-accuracy setting and, at a separate matched-compute setting, up to 24.8× faster inference while remaining more accurate than MAE. SGMA also enables reliable zero-shot analysis, such as pore-network extraction and hydrogel skeletonization.

Acknowledgments and Disclosure of Funding

This work was supported by JSPS Program for Forming Japan's Peak Research Universities (J-PEAKS) Grant Number JPJS00420230001. This research used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the DOE Office of Science, ASCR, under Contract No. DE-AC05-00OR22725.

References

[1] Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. Etc: Encoding long and structured inputs in transformers. *arXiv preprint arXiv:2004.08483*, 2020. 20

[2] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 4, 20

[3] Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0pdSt3oyJa1. 2, 4

[4] Thorsten M Buzug. Computed tomography. In *Springer handbook of medical technology*, pages 311–342. Springer, 2011. 7

[5] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, 2022. 2, 8, 21

[6] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, New York, NY, USA, 2021. IEEE. 4, 20

[7] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021. 8, 23

[8] Richard J. Chen, Chengkuan Chen, Yicong Li, Tiffany Y. Chen, Andrew D. Trister, Rahul G. Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16123–16134, New York, NY, USA, 2022. IEEE. doi: 10.1109/CVPR52688.2022.01567. 4, 20, 22, 23

[9] Richard J Chen, Chengkuan Chen, Yicong Li, Tiffany Y Chen, Andrew D Trister, Rahul G Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16144–16155, 2022. 2

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020. 3

[11] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 20

[12] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509. 4

[13] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 4, 20

[14] Veerle Cnudde and Matthieu Nicolaas Boone. High-resolution x-ray computed tomography in geosciences: A review of the current technology and applications. *Earth-Science Reviews*, 123:1–17, 2013. 2, 3

[15] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning, 2023. 4

[16] Tri Dao, Beidi Chen, Nimit S Sohoni, Arjun Desai, Michael Poli, Jessica Grogan, Alexander Liu, Aniruddh Rao, Atri Rudra, and Christopher Re. Monarch: Expressive structured matrices for efficient and accurate training. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4690–4721. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/dao22a.html. 2, 4

[17] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS: Proceedings of the 35th Neural Information Processing Systems Conference*, New York, NY, USA, 2022. Association for Computing Machinery. doi: 10.48550/ARXIV.2205.14135. URL https://arxiv.org/abs/2205.14135. 4

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 3

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 6, 22, 23

[20] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 574–584, 2022. 8, 16, 23, 29

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 3

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2, 3, 6, 7, 10, 16, 29

[23] Yufan He, Vishwesh Nath, Dong Yang, Yucheng Tang, Andriy Myronenko, and Daguang Xu. SwinUNETR-V2: Stronger swin transformers with stagewise convolutions for 3D medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, volume 14223 of *Lecture Notes in Computer Science*, pages 416–426. Springer, 2023. doi: 10.1007/978-3-031-43901-8_40. 23

[24] Nicholas Heller, Fabian Isensee, Klaus H Maier-Hein, Xiaoshuai Hou, Chunmei Xie, Fengyi Li, Yang Nan, Guangrui Mu, Zhiyong Lin, Miofei Han, et al. The state of the art in kidney and kidney tumor segmentation in contrast-enhanced CT imaging: Results of the KiTS19 challenge. *Medical Image Analysis*, 67:101821, 2021. doi: 10.1016/j.media.2020.101821. 22, 23

[25] Moritz Ibing, Gregor Kobsik, and Leif Kobbelt. Octree transformer: Autoregressive 3d shape generation on hierarchically structured sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2697–2706, 2023. 4

[26] Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models, 2023. 4

[27] Shashank Mohan Jain. Hugging face. In *Introduction to transformers for NLP: With the hugging face library and models to solve problems*, pages 51–67. Springer, 2022. 8, 29

[28] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20, New York, NY, USA, 2020. Association for Computing Machinery. 4

[29] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020. 20

[30] Yoo Jung Kim, Hyungjoon Jang, Kyoungbun Lee, Seongkeun Park, Sung-Gyu Min, Choyeon Hong, Jeong Hwan Park, Kanggeun Lee, Jisoo Kim, Wonjae Hong, Hyun Jung, Yanling Liu, Haran Rajkumar, Mahendra Khened, Ganapathy Krishnamurthi, Sen Yang, Xiyue Wang, Chang Hee Han, Jin Tae Kwak, Jianqiang Ma, Zhe Tang, Bahram Marami, Jack Zeineh, Zixu Zhao, Pheng-Ann Heng, Rüdiger Schmitz, Frederic Madesta, Thomas Rösch, Rene Werner, Jie Tian, Elodie Puybareau, Matteo Bovio, Xiufeng Zhang, Yifeng Zhu, Se Young Chun, Won-Ki Jeong, Peom Park, and Jinwook Choi. Paip 2019: Liver cancer segmentation challenge. *Medical Image Analysis*, 67:101854, 2021. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2020.101854. URL https://www.sciencedirect.com/science/article/pii/S1361841520302188. 2, 3, 4, 7, 26, 29

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2, 5, 7, 8, 10, 29

[32] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020. 20

[33] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *The International Conference on Learning Representations (ICLR)*, New York, NY, USA, 2020. Association for Computing Machinery. doi: 10.48550/ARXIV.2001.04451. URL https://arxiv.org/abs/2001.04451. 4

[34] Ryuji Kiyama, Masahiro Yoshida, Takayuki Nonoyama, Tomáš Sedlačík, Hiroshi Jinnai, Takayuki Kurokawa, Tasuku Nakajima, and Jian Ping Gong. Nanoscale TEM imaging of hydrogel network architecture. *Advanced Materials*, 35(1):2208902, 2022. doi: 10.1002/adma.202208902. 2, 3, 4, 7, 26, 29

[35] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21618–21629. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b4fd1d2cb085390fbbadae65e07876a7-Paper.pdf. 2, 4

[36] B Landman, Z Xu, J Igelsias, M Styner, T Langerak, and A Klein. Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge. In *Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*, 2015. 22, 23

[37] Dacheng Li, Rulin Shao, Anze Xie, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. Lightseq: Sequence level parallelism for distributed training of long context transformers, 2023. 4

[38] Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context, 2023. 4

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[40] Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024. 21, 23

[41] Jieru Mei, Liang-Chieh Chen, Alan Yuille, and Cihang Xie. Spformer: Enhancing vision transformer with superpixel representation. *arXiv preprint arXiv:2401.02931*, 2024. 20

[42] Shinji Noguchi, Ryuji Kiyama, Masahiro Yoshida, Maradhana Agung Marsudi, Naohiro Kashimura, Kiyoharu Tadanaga, Jian Ping Gong, and Takayuki Nonoyama. Real-space visualization of charged polymer network of hydrogel by double network strategy and mineral staining. *Nano Letters*, 24(29): 9088–9095, 2024. 2, 3, 7, 29

[43] Markus N. Rabe and Charles Staats. Self-attention does not need $o(n^2)$ memory, 2022. 4

[44] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714. 2, 8, 18, 23, 29

[45] Weibin Rong, Zhanjing Li, Wei Zhang, and Lining Sun. An improved canny edge detection algorithm. In *2014 IEEE international conference on mechatronics and automation*, pages 577–582. IEEE, 2014. 5

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 7, 8, 18, 23

[47] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient Content-Based Sparse Attention with Routing Transformers. *Transactions of the Association for Computational Linguistics*, 9: 53–68, 02 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00353. URL https://doi.org/10.1162/tacl_a_00353. 4

[48] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International Conference on Machine Learning*, pages 29441–29454. PMLR, 2023. 21

[49] Lawrence A Shepp and Benjamin F Logan. The fourier reconstruction of a head section. *IEEE Transactions on nuclear science*, 21(3):21–43, 1974. 7

[50] Gengxin Shi, Shalini Subramanian, Qian Cao, Shadpour Demehri, Jeffrey H Siewerdsen, and Wojciech Zbijewski. Application of a novel ultra-high resolution multi-detector ct in quantitative imaging of trabecular microstructure. In *Medical Imaging 2020: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 11317, pages 356–362. SPIE, 2020. 2, 3

[51] Han Shi, Jiahui Gao, Xiaozhe Ren, Hang Xu, Xiaodan Liang, Zhenguo Li, and James T. Kwok. Sparsebert: Rethinking the importance analysis in self-attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9547–9557, New York, NY, USA, 2021. PMLR. URL http://proceedings.mlr.press/v139/shi21a.html. 4

[52] Yuqi Si and Kirk Roberts. Three-level hierarchical transformer networks for long-sequence and multiple clinical documents classification, 2021. URL https://arxiv.org/abs/2104.08444. 4, 20

[53] Akihisa Takeuchi, Masayuki Uesugi, Yuki Sada, and Kentaro Uesugi. Hierarchical three-dimensional imaging using x-ray micro-/nano-tomography at BL20XU of SPring-8. In Bert Müller and Ge Wang, editors, *Developments in X-Ray Tomography XV*, volume 13152, page 131521R. International Society for Optics and Photonics, SPIE, 2024. doi: 10.1117/12.3027913. URL https://doi.org/10.1117/12.3027913. 7, 29

[54] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. *arXiv preprint arXiv:2201.02767*, 2022. 4, 21

[55] Yucheng Tang, Dong Yang, Wenqi Li, Holger R. Roth, Bennett A. Landman, Daguang Xu, Vishwesh Nath, and Ali Hatamizadeh. Self-supervised pre-training of swin transformers for 3d medical image analysis. In *CVPR*, pages 20698–20708. IEEE, 2022. 8, 23

[56] Wim Van Aarle, Willem Jan Palenstijn, Jeroen Cant, Eline Janssens, Folkert Bleichrodt, Andrei Dabravolski, Jan De Beenhouwer, K Joost Batenburg, and Jan Sijbers. Fast and flexible x-ray tomography using the astra toolbox. *Optics express*, 24(22):25129–25147, 2016. 4, 29

[57] Xiao Wang, Isaac Lyngaas, Aristeidis Tsaris, Peng Chen, Sajal Dash, Mayanka Chandra Shekar, Tao Luo, Hong-Jun Yoon, Mohamed Wahib, and John Gouley. Ultra-long sequence distributed transformer, 2023. URL https://arxiv.org/abs/2311.02382. 4

[58] Xiao Wang, Jong-Youl Choi, Takuya Kurihaya, Isaac Lyngaas, Hong-Jun Yoon, Xi Xiao, David Pugmire, Ming Fan, Nasik M Nafi, Aristeidis Tsaris, et al. Orbit-2: Scaling exascale vision foundation models for weather and climate downscaling. *arXiv preprint arXiv:2505.04802*, 2025. 3

[59] Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*, 2025. 3

[60] Yutong Xie, Jianpeng Zhang, Chunhua Shen, and Yong Xia. CoTr: Efficiently bridging CNN and transformer for 3D medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 171–180. Springer, 2021. doi: 10.1007/978-3-030-87199-4_16. 23

[61] Chengxuan Ying, Guolin Ke, Di He, and Tie-Yan Liu. Lazyformer: Self attention with lazy update, 2021. 4

[62] Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. Megabyte: Predicting million-byte sequences with multiscale transformers, 2023. 4, 21

[63] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020. 4

[64] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020. 20

[65] Enzhi Zhang, Isaac Lyngaas, Peng Chen, Xiao Wang, Jun Igarashi, Yuankai Huo, Masaharu Munetomo, and Mohamed Wahib. Adaptive patching for high-resolution image segmentation with transformers. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2024. 3, 4, 18, 21, 22

[66] Enzhi Zhang, Peng Chen, Rui Zhong, Du Wu, Jun Igarashi, Isaac Lyngaas, Xiao Wang, Masaharu Munetomo, and Mohamed Wahib. Shf: Symmetrical hierarchical forest with pretrained vision transformer encoder for high-resolution medical segmentation. In *Advances in Neural Information Processing Systems*, 2025. 3, 8, 16, 21, 22, 23

[67] Han Zhang and Lok Ming Lui. Topology-preserving segmentation network: A deep learning segmentation framework for connected component, 2022. URL https://arxiv.org/abs/2202.13331. 19

[68] Hong-Yu Zhou, Jiansen Guo, Yinghao Zhang, Xiaoguang Han, Lequan Yu, Liansheng Wang, and Yizhou Yu. nnFormer: Volumetric medical image segmentation via a 3D transformer. *IEEE Transactions on Image Processing*, 32:4036–4045, 2023. doi: 10.1109/TIP.2023.3293771. 23

[69] Amir Ziabari and Mohammed Alnaggar. Pycmg-based simulation of volumetric concrete microstructure. Technical report, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States). Oak . . . , 2025. 7

A Supplementary Material

A.1 Broader Impacts

SGMA is a self-supervised pre-training framework targeted at ultra-high-resolution scientific images (electron microscopy, X-ray CT, and WSI pathology). On the positive side, it lowers the compute and annotation cost of analyzing giga-pixel scientific data, which can accelerate discovery in materials science, biomedical imaging, and structural biology, and reduce reliance on expensive expert annotation. We do not foresee direct negative societal impacts from the method itself: the framework operates on scientific images rather than personal or web-scraped content, does not generate synthetic media, and is not a generative or language model. The one downstream consideration we want to flag is medical: pathology datasets such as PAIP-32K contain images of patient tissue, and any clinical deployment of SGMA-derived models would require additional validation, regulatory clearance, and safeguards against over-reliance on automated segmentation. We release model weights and code (where applicable) under standard research-use terms to support reproducibility while keeping clinical decision-making with qualified practitioners.

A.2 Hyperparameters and Implementation Details

This section provides the full set of hyperparameters used in pre-training and fine-tuning. All experiments were conducted with PyTorch and DeepSpeed on NVIDIA H100 nodes.

Masking Ratio. For all SGMA pre-training experiments, we adopt a fixed masking ratio of $r = 0.75$ over the adaptive structural patch sequence, matching the ratio recommended in MAE [22]. We additionally explored $r \in \{0.50, 0.60, 0.75, 0.85, 0.90\}$ on HydrogelTEM-1K and found $r = 0.75$ to give the best Dice on the validation split. Unlike random MAE masking, our masks are sampled per-quadtree-level so that the ratio is preserved within each scale. Figure 4 visualizes the effect of sweeping $r \in \{0.25, 0.50, 0.75\}$ at the same sequence length: the adaptive quadtree produces consistent structural coverage, and the reconstruction quality degrades gracefully as the ratio grows.

Sequence Length. The fixed sequence length N produced by the Structure-Guided Feature Tokenizer (SGFT) is the only hyperparameter that bounds the $O(N)$ tokenization budget. We use the following defaults per dataset:

- HydrogelTEM-1K (1024^2): $N = 8{,}194$ for fine-tuning; $N = 16{,}384$ for the high-fidelity SGMA-SAM/SAM 2 variants.
- SpringXCT-8K (8192^2): $N = 1{,}024$ for the speed-optimized run; $N = 8{,}194$ and $N = 16{,}384$ for the highest-accuracy runs.
- PAIP-32K ($32{,}768^2$): $N = 2{,}048$ for the speed-optimized run; up to $N = 16{,}384$ for the best Dice runs.

The SGFT temperature is fixed at $\tau = 1.0$. We did not observe meaningful changes for $\tau \in [0.5, 2.0]$. Figure 5 visualizes the quadtree tokenization at $N \in \{1024, 4096, 8192\}$ on the same input: as N grows, the tree spends more tokens on detail-rich regions, while uniform regions remain coarsely tokenized.

Model Size. SGMA reuses the standard ViT backbones to keep comparisons fair:

- *Encoder*: ViT-B/16 (12 layers, 768 hidden, 12 heads, 86M params) for HydrogelTEM-1K and PAIP-32K; ViT-L/16 (24 layers, 1024 hidden, 16 heads, 304M params) for SpringXCT-8K.
- *Decoder* (pre-training only): 8 transformer blocks, 512 hidden, 16 heads (same as MAE [22]), discarded after pre-training.
- *Segmentation heads*: SAM/SAM 2 default decoders for the SAM variants; the symmetric depatching head from [66] for our SGMA-SAM variants; the standard UNETR [20] head for UNETR variants.

Figure 6 shows MAE-style reconstructions on a HydrogelTEM-1K patch using ViT-L vs. ViT-XL encoders, alongside the corresponding quadtree grid. The larger backbone recovers fine-grained network topology that the smaller variant blurs, motivating our use of ViT-L for the highest-resolution datasets.

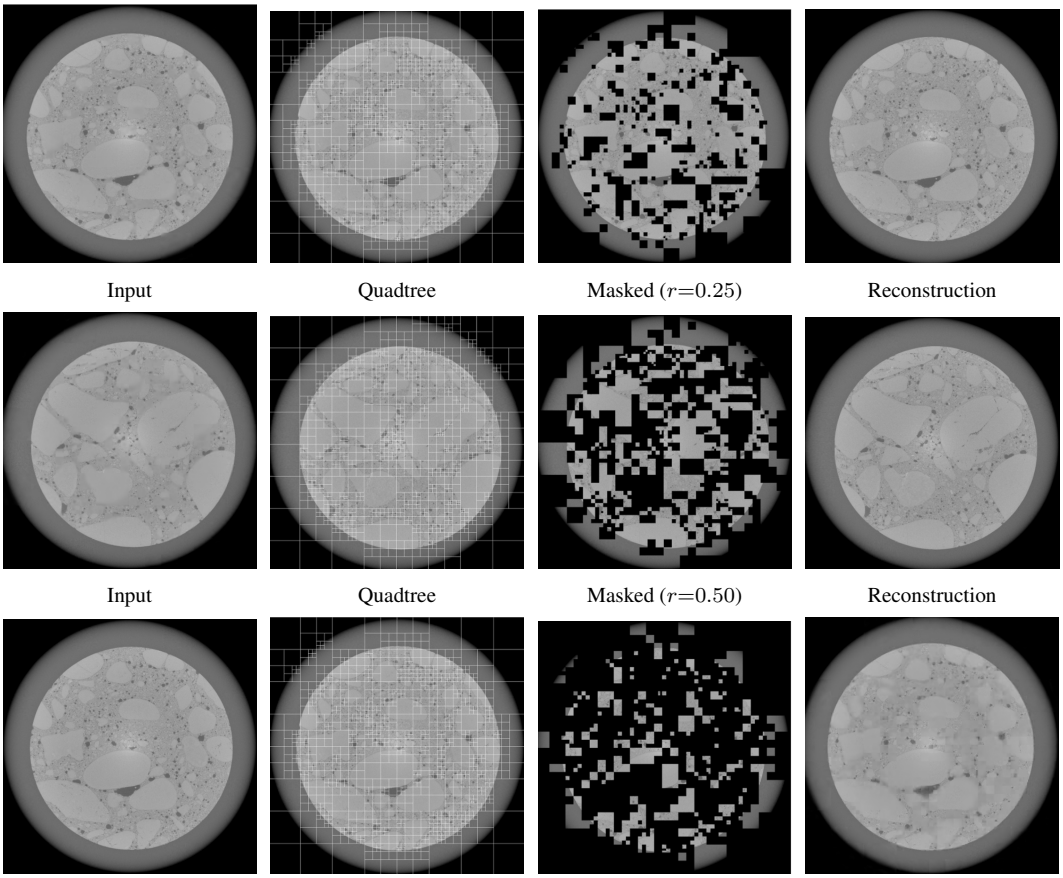


Figure 4: Masking ratio ablation. Each row fixes a ratio $r \in \{0.25, 0.50, 0.75\}$ and shows the input, the quadtree partition, the visible patches after structure-guided masking, and the `SGMA` reconstruction. Higher masking ratios yield a harder pre-training task while the adaptive quadtree keeps structurally important regions visible.

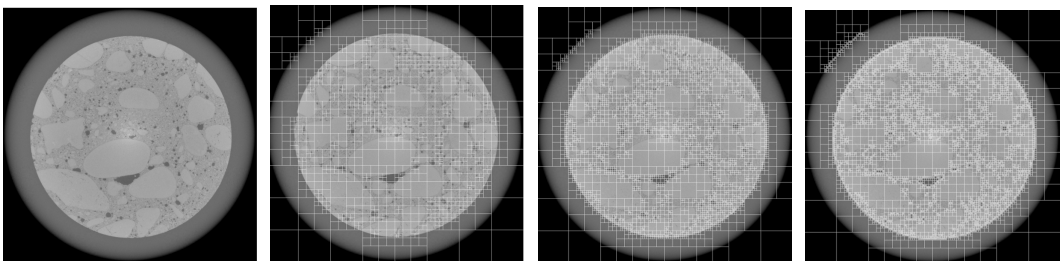
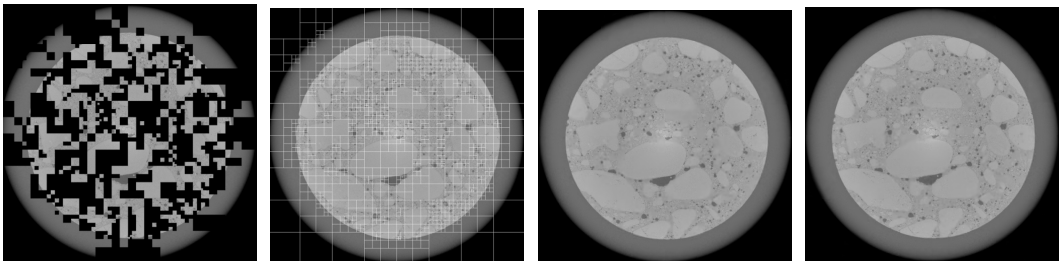


Figure 5: Sequence length ablation. The SGFT tokenizer concentrates more tokens in detail-rich regions as the budget N grows from 1024 to 8192, while keeping uniform background coarsely partitioned.

Optimization. AdamW (β_1=0.9, β_2=0.95, weight decay 0.05) with a cosine schedule, 40 warmup epochs, base learning rate 1.5×10^{-4} scaled linearly with the effective batch size. Pre-training runs 400 epochs; fine-tuning runs 100 epochs with a $10\times$ smaller learning rate. Mixed-precision (bf16) is used throughout.

A.3 Zero-shot Downstream Tasks Evaluation

We use the zero-shot segmentation masks from the SpringXCT-8K and HydrogelTEM-1K datasets to conduct a statistical analysis of the components and connections. Before presenting the down-

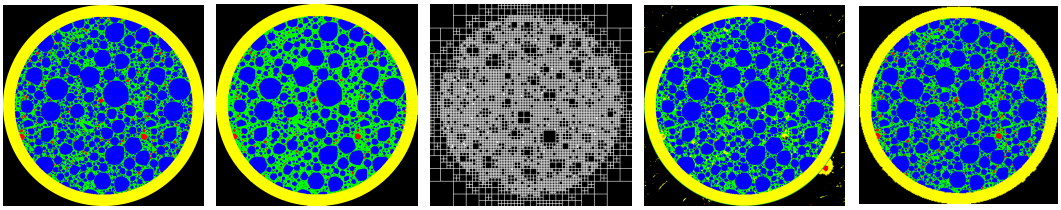


Input | Quadtree grid | Reconstruction (ViT-L) | Reconstruction (ViT-XL)

Figure 6: Model size ablation. Reconstruction quality improves from ViT-L to ViT-XL on a HydrogelTEM-1K patch under the same structure-guided masking and quadtree tokenization. The larger backbone recovers finer network topology, which motivates our use of larger backbones on higher-resolution datasets.

stream statistics, we first visualize the per-sample segmentation quality that drives them. Figure 7 compares four segmentation models on an identical simulated SpringXCT-8K slice (ground-truth mask available): the vanilla UNet [46] collapses on small pores (58.38% Dice), the Adaptive Patching baseline [65] at N=8,194 tokens improves but still misses thin channels, the MAE-pretrained SAM 2 [44] at a 128×128 patch is forced to operate at coarse resolution by GPU memory (85.98% Dice), and our SGMA-SAM 2 at a 2×2 effective patch with N=8,194 SGFT tokens recovers the full pore topology (94.79% Dice). This side-by-side view is what our Table 1 Dice gains look like at the pixel level on a single sample, and it explains why the downstream statistics in the next paragraphs are only reliable for SGMA-derived masks.

As shown in Figure 8 and Figure 9, the components derived from our SGMA-UNETR segmentations yield a near-perfect segmentation on real samples. In contrast, the statistics derived from the MAE-SAM 2 masks are noisy and unreliable, systematically underestimating the volume of small pores that its large-patch pre-training failed to capture. For HydrogelTEM-1K in Figure 10, we also performed zero-shot segmentation and extracted the skeleton for degree analysis. These examples demonstrate that SGMA is not just a marginal improvement in segmentation quality; it is an enabling technology that provides the high-precision results necessary to replace manual, error-prone, and expensive lab-based analysis in core scientific applications.



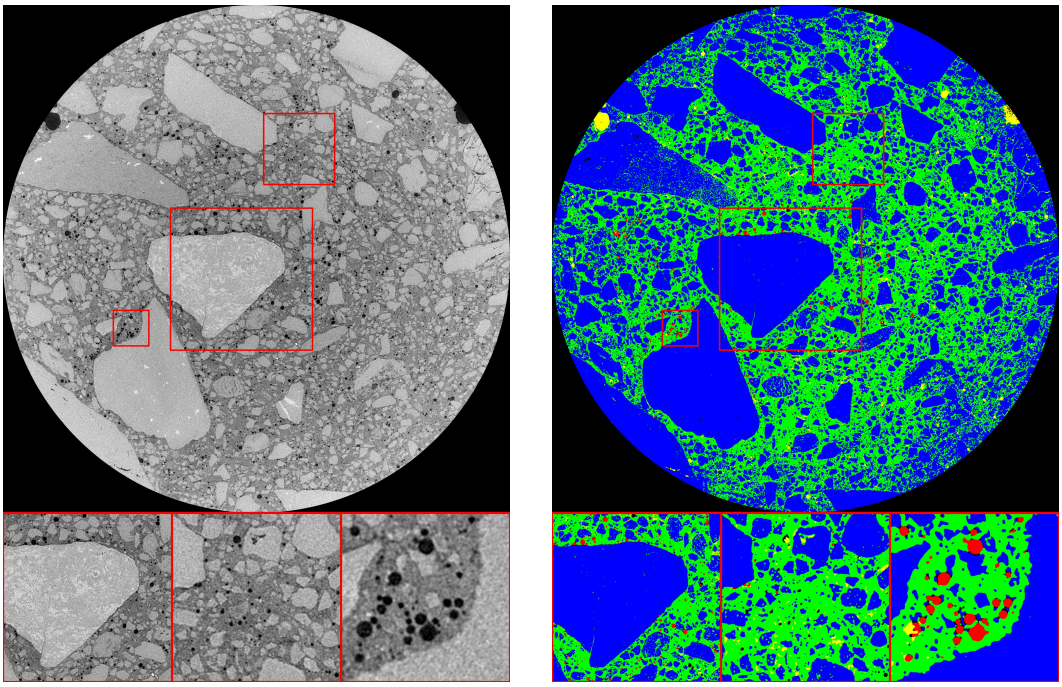
Ground Truth (Dice 100%) | UNet [46] (Dice 58.38%) | AP [65] (N=8,194) | MAE-SAM 2 [44] (Dice 85.98%) | **SGMA-SAM 2 (ours)** (N=8,194, Dice 94.79%)

Figure 7: Segmentation comparison on a SpringXCT-8K simulated slice (per-sample Dice reported). UNet [46] loses small-pore topology; the Adaptive Patching baseline [65] at N=8,194 tokens improves but still drops thin channels; MAE-pretrained SAM 2 [44] at the smallest feasible 128^2 patch on an 8K image captures coarse structure only; our SGMA-SAM 2 at a 2×2 effective patch size with N=8,194 SGFT tokens recovers the fine pore topology. The aggregate test-set Dice for each model is reported in Table 1; the numbers here are for this specific sample.

A.4 Downstream Task Visualizations

A.5 Comparison with Long-Sequence and Hierarchical Transformer Methods

Table 4 summarizes how SGMA compares against prior long-sequence and hierarchical Transformer methods along four axes: best-case complexity, attention mechanism invariance (i.e. whether the approach requires modifying self-attention itself), model-architecture coupling (whether the method is tied to a specific forked backbone), and implementation locus (attention kernel vs. model vs. input-preprocessing). Most prior work either *(i)* approximates attention with sparsity, low-rank, or locality



Real Sample Slice. Zero-shot Prediction

Figure 8: Segmentation on a real SpringXCT-8K sample with zero-shot inference. The pixel-level microstructure (e.g. void area) can be precisely extracted, which is hard for human experts.

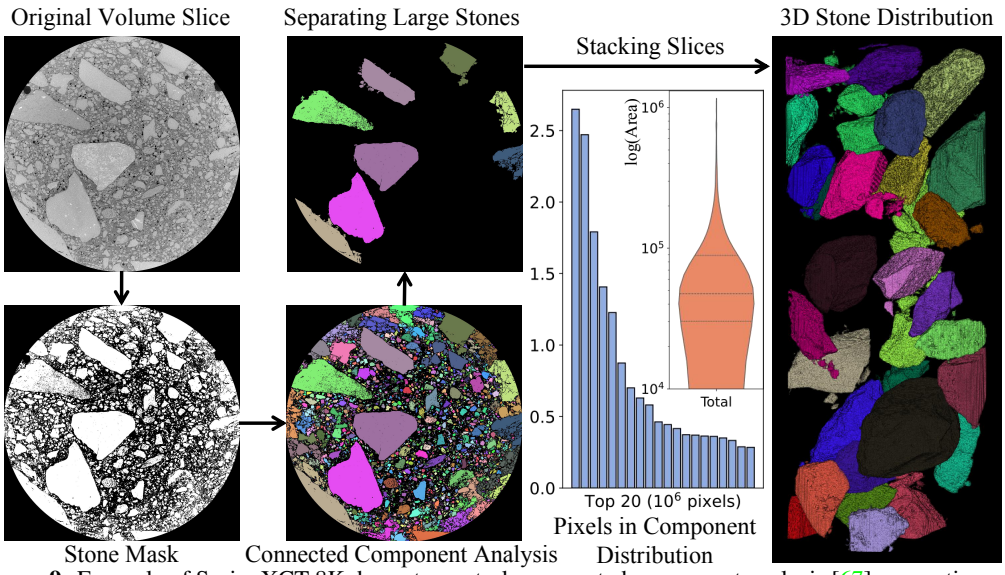


Figure 9: Example of SpringXCT-8K downstream task: connected component analysis [67], separation, and mapping the stones distribution in the 3D image.

assumptions, which trades off accuracy for speed and often requires forked PyTorch kernels; or *(ii)* designs hierarchical backbones (CrossViT, HIPT, MEGABYTE) that need multiple trained models and impose specific inductive biases. In contrast, SGMA operates entirely at the *input-preprocessing* stage: it leaves the attention mechanism and ViT architecture untouched, and instead produces a fixed-length structure-conditioned token sequence via the Structure-Guided Feature Tokenizer (SGFT). This makes SGMA a drop-in replacement for MAE pre-training on any standard ViT, yet reduces the effective complexity to $O(\log^2 N)$ in the input resolution N in the best case.

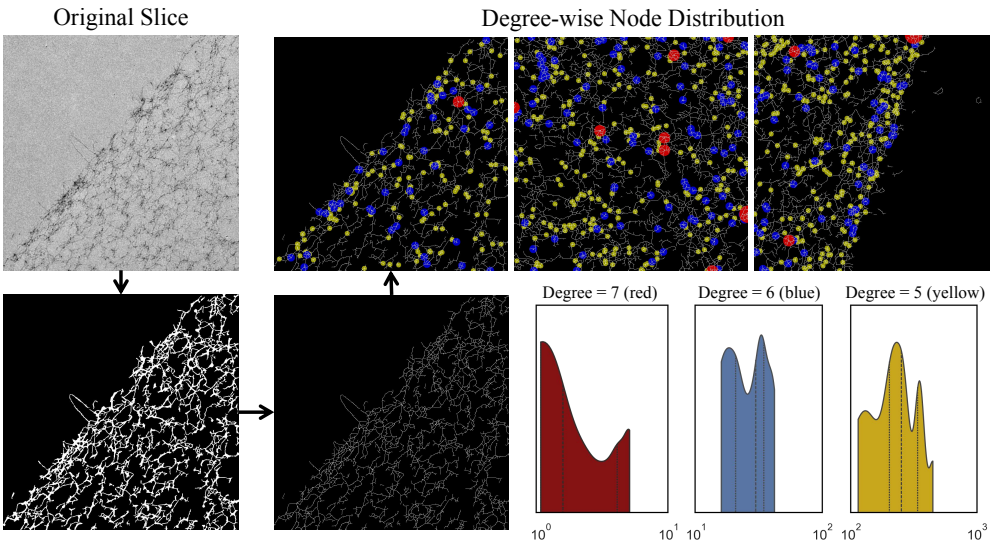


Figure 10: Example of HydrogelTEM-1K downstream task: a hydrogel sample processed through zero-shot segmentation and skeletonization, followed by junction detection and degree-wise node annotation, illustrating the structural complexity of the hydrogel network.

Table 4: A summary of relevant long-sequence training methods that attack quadratic attention by reducing the work done. Here N = sequence length.

Approach	**Method**	**Merits & Demerits**	**Complexity (Best)**	**Model**	**Implementation**
Attention Approximation	Longformer [2], ETC [1]	**(+)** Better time complexity vs. Transformer. **(-)** Sparsity levels insufficient for gains to materialize.	$O(N)$ $O(N\sqrt{N})$	Some models, forked PyTorch	Self-attention kernel
	BigBird [64], Reformer [32]	**(+)** Theoretically proven time complexity. **(-)** High-order derivatives.	$O(N \log N)$	Some models, forked PyTorch	Self-attention kernel
	Sparse Attention [11]	**(+)** Introduced sparse factorizations of attention. **(-)** Higher time complexity.	$O(N\sqrt{N})$	Some models, forked PyTorch	Self-attention kernel
	Linformer [29], Performer [13]	**(+)** Fast adaptation. **(-)** Assumes self-attention is low rank.	$O(N)$	Some models, forked PyTorch	Self-attention kernel
	SPFormer [41] (Prediction)	**(+)** Irregular tokens. **(-)** No adaptation to high resolution.	$O(P^2)$ (P: #regions)	Model, plain PyTorch	Model Implementation
Hierarchical	Hier. Transformer [52] (Text Class.)	**(+)** Independent hyperparameter tuning of hierarchical levels. **(-)** No support for ViT.	$O(N \log N)$	Model, plain PyTorch	Model Implementation
	CrossViT [6] (Class.)	**(+)** Better time complexity vs. standard ViT. **(-)** Complex token fusion in dual-branch ViTs.	$O(N)$	Model, plain PyTorch	Model Implementation
	HIPT [8] (Class.)	**(+)** Models inductive biases of features in the hierarchy. **(-)** High cost for training multiple models.	$O(N \log N)$	Model, plain PyTorch	Model Implementation

continued on next page

Table 4 — continued from previous page

Approach	Method	Merits & Demerits	Complexity (Best)	Model	Implementation
	MEGABYTE [62] (Prediction)	**(+)** Supports multi-modality. **(-)** High cost for training multiple models.	$O(N^{4/3})$	Model, plain PyTorch	Model Implementation
State-Space Model	U-Mamba [40] (Segm.)	**(+)** Linear-time long-range dependency via selective SSM. **(-)** Replaces attention entirely; not compatible with MAE / standard ViT pre-training.	$O(N)$	Model, plain PyTorch	Model Implementation
High-resolution	Swin-UNETR [5] (Segm.)	**(+)** Shifted windows preserve locality at scale. **(-)** Window partitioning causes inter-layer load imbalance; fixed window size limits adaptivity.	$O(N)$	Model, plain PyTorch	Model Implementation
	Hiera [48] (Segm. / Class.)	**(+)** Simple hierarchical ViT; strong downstream accuracy. **(-)** Requires a bespoke backbone; no adaptive tokenization on giga-pixel inputs.	$O(N)$	Model, plain PyTorch	Model Implementation
	QuadTree Attn. [54] (Detect. / Match.)	**(+)** Coarse-to-fine attention via quadtree; reduces attention cost. **(-)** Modifies the attention computation itself; incompatible with standard MAE.	$O(N)$	Model, forked PyTorch	Self-attention kernel
Ours	**Structure-Guided Masked AutoEncoder (Segm. & Class.)**	**(+) Attention mechanism intact. (+) Largely reduces computation cost; maintains quality. (+) Efficiency scales with image detail. (-) Task semantics independent of edge information.**	$\boldsymbol{O(\log^2 N)}$	**Any model, plain PyTorch**	**Image preprocessing**

A.6 SGFT Tokenizer Construction Efficiency

A natural concern with any adaptive tokenization scheme is that the tokenizer itself becomes a new bottleneck: prior adaptive-patching methods such as Adaptive Patching [65] and SHF [66] construct the quadtree by a strictly sequential $\arg\max$ greedy split over the active token set, whose cost grows roughly linearly with the target sequence length N. Because SGFT is invoked once per sample at every epoch, an $O(N)$ tokenizer would erase the $O(\log^2 N)$ attention savings on giga-pixel images.

We therefore deploy SGFT's softmax-batched token selection (Section 3.1, Eq. 2) as the production path, and benchmark it against two stricter $\arg\max$ baselines: the sequential greedy split used in AP [65] / SHF [66] ("**Naive (SHF/AP)**"), and a heap-accelerated variant of the same rule ("**Heap (intermediate)**") that we developed as an intermediate optimization. All three produce a quadtree of N tokens; we additionally report the pixel-level MSE between the quadtree-reconstructed image (each token rendered at its mean color) and the original as a tokenization-quality measure — lower MSE means the tokens spend their budget on more informative regions.

Two observations are worth highlighting. *First*, the tokenization-quality gap is negligible: across the small-scale runs the per-pixel MSE between the three algorithms agrees to within $\sim$3% (and SGFT is sometimes *lower* than the deterministic argmax variants because the softmax sampling occasionally routes tokens into regions the greedy rule misses), so the speedup comes essentially

Table 5: SGFT tokenizer construction efficiency. Three quadtree-construction algorithms producing the same number of tokens N at three image resolutions. *Naive (SHF/AP)* is the sequential $\arg\max$ greedy split used in prior adaptive-patching work [66]; *Heap (intermediate)* is our heap-accelerated variant of the same rule; *SGFT (softmax batch, ours)* is the production tokenizer used throughout the paper. The Naive baseline is run only at the $1{,}024^2$ scale and reported as "–" at higher resolutions, where its sequential cost becomes prohibitive. MSE between the three algorithms is within $\sim$3% on the small bench (SGFT occasionally lower because the softmax injects a small amount of exploration), so the speedup comes essentially for free on tokenization quality.

Image Res.	Leaves N	Naive (SHF/AP) [65, 66]	Heap (intermediate) (ours)	SGFT (ours)	Speedup vs. Naive	Speedup vs. Heap
$1{,}024^2$	4,096	38.8 ms	32.3 ms	**12.1 ms**	**3.2**×	**2.7**×
	8,192	74.0 ms	52.3 ms	**13.1 ms**	**5.7**×	**4.0**×
	16,384	176.9 ms	97.1 ms	**18.6 ms**	**9.5**×	**5.2**×
$4{,}096^2$	65,536	–	1.85 s	**0.33 s**	–	**5.6**×
	262,144	–	6.97 s	**0.63 s**	–	**11**×
	1,024,000	–	26.81 s	**1.31 s**	–	**20**×
$8{,}192^2$	65,536	–	2.41 s	**1.01 s**	–	**2.4**×
	262,144	–	7.51 s	**1.39 s**	–	**5.4**×
	1,024,000	–	27.49 s	**2.98 s**	–	**9.2**×

for free on quality. *Second*, the speedup widens with N: on a $4{,}096^2$ image at $N = 10^6$ tokens the heap-based $\arg\max$ takes $\sim$27 s per image while SGFT finishes in $\sim$1.3 s, a $20\times$ gap; on $8{,}192^2$ the same configuration is $9.2\times$ faster. In other words, SGFT is not only cheap relative to attention — it also consistently beats the SHF/AP-style adaptive tokenizer in raw wall-clock construction time, which is what allows the per-sample preprocessing cost of SGMA to stay well under a second on the giga-pixel datasets used in the main experiments.

A.7 Additional 3D Medical Segmentation Baselines

Although ultra-high-resolution 2D imaging is the primary focus of the paper, the long-sequence problem that motivates SGMA also arises in *3D volumetric data* at moderate spatial resolution: a 512^3 CT volume with 4^3 patches still produces a token sequence of length $N = 2{,}097{,}152$, which is well beyond what standard self-attention can handle. We therefore additionally evaluate SGMA-SAM 2 on two standard 3D medical segmentation benchmarks, BTCV [36] and KiTS19 [24], to test whether the SGFT-based sequence compression remains effective in the 3D setting. We compare against widely used 3D medical baselines (U-Net, U-Mamba, TransUNet, UNETR, Swin UNETR, Swin UNETR-V2, CoTr, nnFormer) and the SAM 2 backbone with the same fine-tuning recipe. We omit a separate MAE-SAM 2 row here because the MAE-vs-SGMA pre-training comparison is already established at length on matched architectures in Section 5.1 (Table 1); the goal of Table 6 is to position SGMA-SAM 2 against the broader 3D medical landscape.

The takeaway from Table 6 is that, even at this lower spatial resolution where the segmentation accuracy of strong specialized 3D baselines is already saturated, SGMA-SAM 2 with 2^3 patches matches or marginally beats the best baseline on Dice (**89.71%** vs. Swin UNETR's 89.5% on BTCV; **87.25%** vs. UNETR's 86.45% on KiTS19), while delivering a substantial speedup driven by SGFT's sequence compression (**7.15**× faster than SAM 2 on BTCV; **2.58**× faster on KiTS19). In other words, on standard-resolution 3D data the headline contribution of SGMA is *efficiency at parity accuracy*, complementing the much larger accuracy gains we observe on the ultra-high-resolution 2D datasets in the main text.

A.8 Additional Task: Classification on PAIP-16K and a DA Ablation

To test whether SGMA transfers beyond segmentation, we additionally evaluate it on a whole-slide classification task on PAIP downsampled to $16{,}384^2$ ("PAIP-16K"). We compare against (i) vanilla ViT [19], (ii) HIPT [8] — a hierarchical giga-pixel ViT baseline — and (iii) SHF-ViT [66], which uses the same symmetric hierarchical forest tokenizer as our SGMA-SAM head but without SGMA pre-training. For each architecture we report two patch-size settings: *coarse* ($4{,}096^2$) and *fine* (2^2); GPU

Table 6: Segmentation of BTCV [36] for multi-organ segmentation and KiTS19 [24] for Kidney Tumor Segmentation on a single GPU. *Time* indicates the end-to-end runtime to achieve the corresponding Dice Score.

Dataset	Model	Patch Size	Time	Speedup (×)	Dice Score (%)
BTCV [36]	U-Net [46]	N/A	843.90 Seconds	9.04×	80.2
	U-Mamba [40]	N/A	8,016.24 Seconds	0.95×	83.51
	TransUNet [7]	N/A	3115.25 Seconds	2.45×	83.8
	UNETR [20]	4^3	8386.56 Seconds	0.91×	89.1
	Swin UNETR [55]	4^3	5861.93 Seconds	1.30×	89.5
	SAM 2 [44]	4^3	7637.28 Seconds	1.0×	82.77
	SGMA-SAM 2	$\mathbf{2^3}$	**1067.88 Seconds**	**7.15×**	**89.71**
KiTS [24]	U-Net [46]	N/A	243.7 Minutes	1.99×	83.23
	U-Mamba [40]	N/A	969.0 Minutes	0.5×	86.22
	CoTr [60]	N/A	488.7 Minutes	0.99×	84.59
	UNETR [20]	8^3	513.6 Minutes	0.94×	86.45
	nnFormer [68]	8^3	876.5 Minutes	0.55×	75.85
	Swin UNETR [55]	8^3	748.3 Minutes	0.65×	81.27
	Swin UNETR-V2 [23]	8^3	766.4 Minutes	0.63×	84.14
	SAM 2 [44]	8^3	483.1 Minutes	1.0×	81.35
	SGMA-SAM 2	$\mathbf{2^3}$	**187.3 Minutes**	**2.58×**	**87.25**

counts match what the model can fit at that setting. On the classification-only baselines (ViT / HIPT / SHF-ViT) we leave the Dice column empty ("–") because they do not produce a segmentation head under this protocol. For SGMA we also report a **-NoDA** ablation in which the Damped Accumulation (DA) signal path is removed, so the difference between **SGMA-ViT-2-NoDA** and **SGMA-ViT-2** isolates the contribution of DA on top of SGFT tokenization.

As shown in Table 7, SGMA-ViT-2 achieves the best Top-1 accuracy (**83.77%**) and the best Dice (**81.67%**), improving over SHF-ViT-2 by $+3.63$ accuracy points and over HIPT by $+11.08$ points. More revealingly, the NoDA ablation shows a clear *task-dependent* pattern: removing DA costs only $+0.88$ Top-1 (within run-to-run noise for classification) but $+3.00$ Dice on segmentation. In other words, **DA contributes essentially nothing to classification and a substantial margin to segmentation**, which is consistent with its design — DA injects multi-scale spatial structure along the quadtree path, and this kind of hierarchical spatial signal is valuable for dense pixel-level prediction but largely redundant when the task only needs a single global category label. SGFT alone already matches or beats SHF-ViT-2 on both tasks, and DA is precisely the component that lifts the segmentation half.

Table 7: Classification (Top-1 accuracy, %) and Segmentation (Dice, %) of vanilla ViT, HIPT [8], SHF-ViT [66], and SGMA-ViT on PAIP-16K ($16{,}384^2$ res.). "–" denotes classification-only baselines. "-NoDA" ablates the Damped Accumulation signal path to isolate the contribution of DA on top of SGFT.

Model	GPUs	Patch Size	Accuracy (%)	Dice (%)
ViT [19]	128	$4{,}096^2$	68.97	–
HIPT [8]	128	$\{16{,}256^2, 4{,}096^2\}$	72.69	–
SHF-ViT-4096 [66]	8	$4{,}096^2$	69.11	–
SHF-ViT-2 [66]	128	2^2	80.14	–
SGMA-ViT-4096	8	$4{,}096^2$	70.37	72.83
SGMA-ViT-2-NoDA (SGFT only)	128	2^2	82.89	78.67
SGMA-ViT-2	128	2^2	**83.77**	**81.67**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 explicitly state our three contributions (SGMA framework, ultra-high-resolution segmentation results, and downstream zero-shot analysis) and report the headline numbers (+9.70 Dice on SpringXCT-8K, +16.84 Dice on PAIP-32K, up to 24.8× inference speedup), which match the experimental results in Table 1 and Table 3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5.4 (*Limitations and Effect of Hyperparameters*) walks through the masking ratio, sequence length, and model size ablations (with full figures in Appendix A.2, Figures 4–6) and identifies which of those become practical limitations: (i) sensitivity to the chosen sequence length N on highly heterogeneous inputs; and (ii) dependence on Canny-edge salience, which can degrade the tree decomposition on images dominated by weak gradients or texture-defined regions. The Conclusion outlines future work to relax the second limitation via a learned saliency module.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper proposes an empirical pre-training framework. Equations (1)–(4) define the SGFT sampling and damped accumulation procedure, but the paper does not claim formal theorems requiring proof.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 fully describes the SGMA pre-training objective and the SGFT tokenization. Datasets, splits, and baselines are described in Section 4. Full hyperparameters (masking ratio, sequence length, model size, optimizer, schedule) are listed in Appendix A.2, and the SpringXCT-8K simulation pipeline is described in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Python-level pseudo-code for the SGFT tokenization and SGMA pre-training/fine-tuning pipeline is provided in the supplementary material, with sufficient detail to reproduce the main experimental results. Datasets are partially open: PAIP-32K [30] and HydrogelTEM [34] are publicly accessible from their original sources, while access to the SpringXCT-8K raw scans is restricted by the providing facility; the SpringXCT-8K simulation pipeline used for the fine-tuning data is described in Section 4.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 specifies datasets and splits (e.g., 70/10/20 train/val/test for PAIP). Appendix A.2 reports masking ratio (r=0.75 chosen by sweep over $\{0.50, 0.60, 0.75, 0.85, 0.90\}$), per-dataset sequence lengths, ViT-B vs. ViT-L choices, AdamW optimizer settings (β_1=0.9, β_2=0.95, weight decay 0.05), cosine schedule with 40 warmup epochs, and 400/100 pre-training/fine-tuning epochs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All Dice scores in Table 1 and Table 3 are reported as the mean over multiple independent fine-tuning runs with different random seeds for the segmentation head, while keeping the same pre-trained encoder. The variability across runs is small relative to the gaps reported (e.g. the +9.70 / +16.84 Dice improvements over MAE-SAM 2 baselines are well outside the run-to-run standard deviation), so single-value entries are used in the tables for readability.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A.2 states experiments were conducted with PyTorch and DeepSpeed on NVIDIA H100/A100 nodes. Table 3 reports per-experiment GPU counts (4 GPUs for HydrogelTEM-1K, 128 for SpringXCT-8K, 512 for PAIP-32K) along with seconds-per-image throughput.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The research uses publicly available scientific imaging datasets (TEM, X-ray CT, WSI pathology) and standard ML techniques. No human subjects or sensitive content is involved, and anonymity has been preserved in the submission.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix A.1 discusses the broader impacts. On the positive side, SGMA can lower the compute and annotation cost of giga-pixel scientific image analysis, accelerating discovery in materials science, biomedical imaging, and structural biology. The main downstream concern flagged is the use of pathology data: clinical deployment of SGMA-derived models would require additional validation, regulatory clearance, and safeguards against over-reliance on automated segmentation.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper introduces a pre-training methodology for ultra-high-resolution scientific image segmentation. It does not release a generative model, language model, or scraped dataset that poses meaningful misuse risk.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All existing assets used in this work are properly cited at first mention: datasets (HydrogelTEM-1K [34, 42], SpringXCT-8K [53], PAIP-32K [30]), pre-trained model architectures (SAM [31], SAM 2 [44], UNETR [20], MAE [22]), and supporting libraries (`timm` [27], ASTRA Toolbox [56]). Each asset is used in accordance with its respective public-research license and original terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces two new assets, both documented in the main text or supplement: (i) the SpringXCT-8K fine-tuning simulation pipeline, whose virtual microstructure generation, attenuation mapping, forward projection, and degradation steps are described in Section 4; and (ii) Python-level pseudo-code for the SGFT tokenization and SGMA pre-training/fine-tuning procedure, provided in the supplementary material. Anonymized references and asset details follow NeurIPS submission guidelines.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human subjects research; all datasets are pre-existing scientific imaging benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects research is conducted in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as part of the core method development. The research focuses on Vision Transformer pre-training for ultra-high-resolution scientific images and does not involve LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.