QUADIT: SPATIALLY ADAPTIVE DIFFUSION TRANSFORMERS FOR SCIENTIFIC IMAGE–MASK SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Building annotated datasets of high-resolution scientific images faces two bottlenecks: synthesizing large images is computationally expensive, while pixel-wise annotation is labor-intensive. We present `QuaDiT`, a framework that addresses both through a shared, spatially adaptive latent representation for image–mask synthesis. An edge-guided quadtree partitions each image into a fixed number of variable-size patches, allocating finer patches to structurally complex regions, while a shared autoencoder maps every patch into a common latent space. A diffusion transformer trained with rectified flow, which encodes each patch's position and scale in place of a fixed-grid positional embedding, generates these latent tokens, from which an image head and a semantic mask head decode paired outputs. The mask head is trained on simulator-provided labels with the image autoencoder frozen, eliminating the need for manual annotation. On concrete X-ray Computed Tomography images at a resolution of 4096×4096, `QuaDiT` represents each image with 1,501 tokens, compared with 4,096 for a dense latent DiT, and achieves global and patch FID scores of 22.64 and 38.51, compared with 46.84 and 73.73 for a dense DiT baseline trained with the same recipe. `QuaDiT` also provides $2.25\times$ higher training throughput and 54.6% lower peak memory. `QuaDiT` further scales to 8192×8192, where the dense baseline exceeds GPU memory, and achieves a global FID of 21.18. On labeled simulations, the mask head achieves test macro Dice scores of 0.968 and 0.996 under the two appearance conditions.

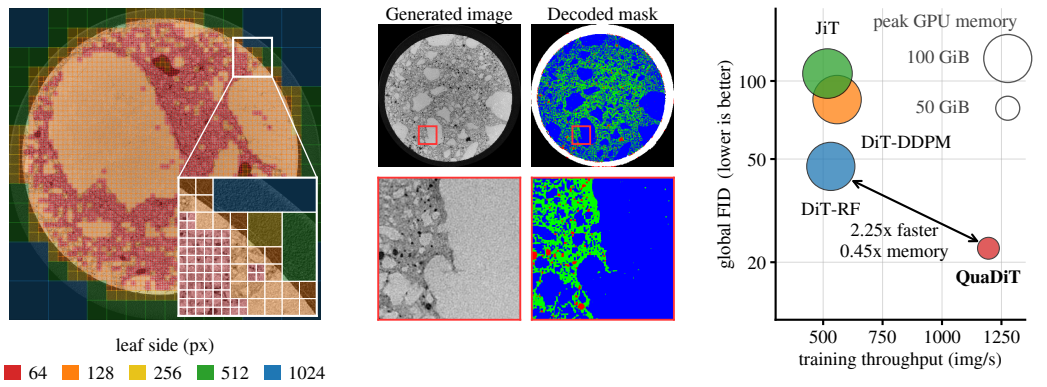
1 INTRODUCTION

High-resolution scientific images reveal structures whose interpretation requires both spatial detail and semantic information. In concrete X-ray Computed Tomography (CT), for example, aggregates, mortar, interfaces, and small voids coexist within a single field of view. Identifying and quantifying these structures requires images that preserve fine detail together with pixel-wise labels. Building such datasets faces two bottlenecks: synthesizing high-resolution images is computationally expensive, and densely annotating them requires substantial human effort. Increasing the supply of images alone leaves the annotation burden unresolved. This motivates a unified objective: efficiently synthesizing high-resolution scientific images together with their semantic masks.

Latent diffusion models reduce the cost of image synthesis by operating in a learned compressed representation (Rombach et al., 2022), and diffusion transformers provide a scalable architecture for modeling latent image patches (Peebles & Xie, 2023). However, a dense latent grid grows quadratically with image side, and the transformer processes every cell at the same granularity. In scientific images, local structure is unevenly distributed: homogeneous regions occupy large areas, whereas intricate boundaries and small features are concentrated in much smaller regions. This raises a representation question for high-resolution synthesis: how should a limited token budget be distributed across an image? Producing semantic masks introduces a related challenge. A generated image does not come with the labels required for supervised learning or semantic analysis, and manual annotation reinstates the original labor bottleneck. Simulators provide paired images and labels because the synthetic structures are known by construction, but their supervision must be transferred to the appearance and structures represented by a generator trained on real images.

Our central idea is a shared, spatially adaptive latent representation that serves as the interface between these tasks. We introduce `QuaDiT` (Figure 1), which implements this idea with three compo-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Adaptive layout (8K) (b) Paired synthesis (4K) (c) Quality vs. cost (4K)

Figure 1: `QuaDiT` at a glance. (a) An edge-guided quadtree covers an 8192^2 CT slice with 6,004 variable-size leaves, colored by side length: 64-pixel leaves follow aggregate boundaries, and large leaves cover background. A dense factor-8 latent grid would need 16,384 tokens. (b) One generated 4K latent sequence of 1,501 tokens is decoded by an image head and a mask head into a paired image and semantic mask; the red box marks a native 512×512 region. (c) At 4K, `QuaDiT` reaches lower global FID than dense latent DiT (DiT-RF, DiT-DDPM) and pixel-space JiT baselines at $2.25\times$ the training throughput of DiT-RF and less than half its peak memory (bubble size). At 8K, the dense DiT-XL/8 exceeds GPU memory.

nents. An edge-guided quadtree partitions each image into a fixed number of variable-size patches, allocating finer patches to regions with more local structure. A shared autoencoder maps patches of every size into a common latent space, so that the whole image becomes a fixed-length token sequence. A diffusion transformer, trained with rectified flow on real images, generates these tokens. Because the leaves lie on no fixed grid, it encodes each token's position and scale from the layout in place of a grid positional embedding; at sampling time, layouts are drawn from training images. An image head and a semantic mask head then decode the same generated features into paired outputs. The mask head is trained on simulator-provided pixel labels with the image autoencoder frozen, which supplies semantic supervision without manual pixel-wise annotation.

We evaluate `QuaDiT` on concrete X-ray CT images at resolutions up to 8192×8192. At 4096×4096, it uses 63% fewer tokens than a dense latent DiT and reaches global FID 22.64 versus 46.84, with $2.25\times$ higher training throughput; at 8192×8192, where the dense DiT exceeds GPU memory, it reaches global FID 21.18. The mask head reaches test macro Dice of 0.968 and 0.996 on labeled concrete simulations, and `QuaDiT` again attains the lowest global FID on a second dataset of hydrogel networks. Our contributions are threefold:

- A budgeted spatially adaptive representation: an edge-guided quadtree with a fixed token budget and a shared autoencoder for variable-size patches, which turns an image of any resolution into a fixed-length sequence of uniform latent tokens, reducing training cost and extending generation to 8K.
- A geometry-aware diffusion transformer that generates in this space with rectified flow, encoding each leaf's position and scale in place of a fixed-grid positional embedding, which reaches lower FID than dense latent and pixel-space baselines on concrete CT.
- Paired image–mask synthesis through a semantic head on the shared decoder features, trained on simulator-provided labels without manual annotation and evaluated on paired synthetic data.

2 Related Work

Diffusion transformers and adaptive patching. Latent diffusion (Rombach et al., 2022) and diffusion transformers (Peebles & Xie, 2023) model images as a dense grid of latent patches, and rectified flow (Liu et al., 2023) provides the training objective we adopt. Pixel-space transform-

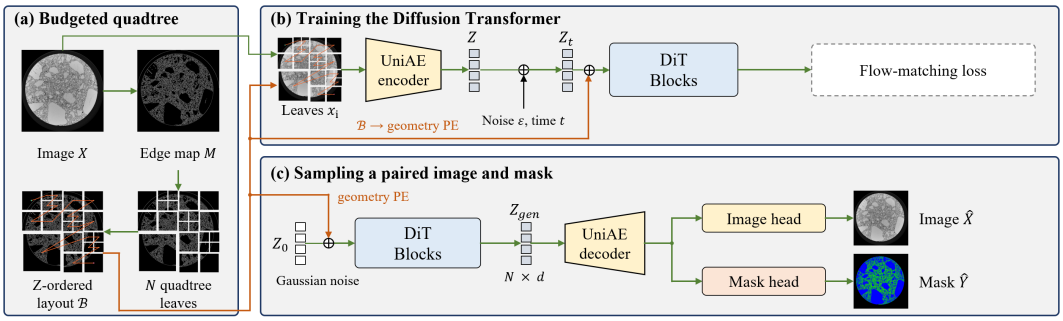


Figure 2: Overview of QuaDiT. (a) An edge-guided quadtree produces a fixed-budget, Z-ordered layout $\mathcal{B}$. (b) The DiT, with geometry PE from $\mathcal{B}$, is trained on the UniAE leaf tokens (Eq. 4). (c) Sampling draws a layout from the training images: the DiT maps Gaussian noise to latent tokens, and the UniAE decoder turns them into shared features read by an image head and a mask head.

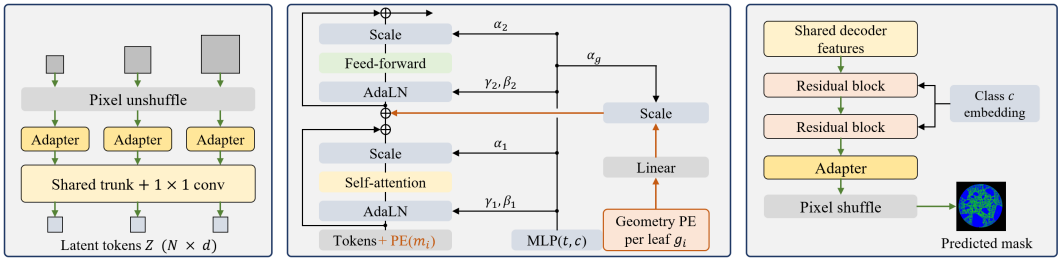
ers such as JiT (Li & He, 2026) and PixelDiT (Yu et al., 2026) remove the autoencoder, while RAE (Zheng et al., 2026) and REPA (Yu et al., 2025) bring pretrained visual representations into the latent space or the training objective; all keep a uniform grid. Content-dependent patching instead allocates tokens by local complexity: the edge-guided quadtree of Zhang et al. (2024) and the entropy-based subdivision of APT (Choudhury et al., 2026) produce image-dependent numbers of patches and embed multiple patch sizes into a common dimension for recognition and dense prediction. QuaDiT uses a prescribed leaf budget so that every image becomes a fixed-length sequence, which the generator models as a whole.

Adaptive generative tokenization. QDM (Yang et al., 2026) uses a quadtree of the low-quality input to select regions for sparse diffusion refinement in super-resolution, with a coarse global stream supplying context. GPSToken (Zhang et al., 2025) represents regions as learnable 2D Gaussians with texture features, rendered by splatting, and generates in two stages: a layout, then texture and geometric residuals. Both couple spatially adaptive tokens with generation. QuaDiT uses non-overlapping square leaves that a shared codec decodes at their native extent, and reads paired image and semantic outputs from the same decoder features.

Simulator supervision and paired outputs. Annotated image synthesis has been approached by conditioning generation on semantic maps (Park et al., 2019; Zhang et al., 2023), by learning joint image–mask distributions (Mao et al., 2025; García-Torres et al., 2026), and by predicting labels from generative features (Zhang et al., 2021; Baranchuk et al., 2022; Wu et al., 2023). Promptable segmentation models such as SAM (Kirillov et al., 2023) and its microscopy adaptation μSAM (Archit et al., 2025) label existing images and can assist annotation of scientific data. QuaDiT produces the mask together with the image: a semantic head reads the decoder features of the image head, so both outputs share one spatial layout. Simulators supply paired images and labels by construction (Shrivastava et al., 2017; Billot et al., 2023); concrete microstructure simulation (Ziabari & Alnaggar, 2025) provides the pairs that supervise our mask head, while the generator learns appearance from real data.

3 Method

QuaDiT synthesizes a high-resolution image and its semantic prediction from a shared sequence of spatially adaptive latent tokens. The framework (Figure 2) consists of an edge-guided quadtree, an autoencoder for variable-size image patches, a geometry-aware DiT that generates latent tokens, and a semantic head that reuses the image-decoder features. The autoencoder is trained first; the DiT and the semantic head are then trained in its fixed latent space.



(a) UniAE encoder (b) DiT block with geometry PE (c) Mask head

Figure 3: `QuaDiT` component architectures. (a) Pixel unshuffle and size-specific adapters bring leaves of different sizes to a common shape, and a shared trunk maps them to latent tokens; the decoder mirrors this design. (b) Geometry PE replaces DiT's grid PE: $\mathrm{PE}(m_i)$ is added to the input, and each block adds g_i via a gated residual. (c) The mask head reads the shared decoder features. Its residual blocks are modulated by the class embedding through scale, shift, and gate, as in AdaLN (Peebles & Xie, 2023), and a size-specific adapter with pixel shuffle restores the native patch size.

3.1 Budgeted spatially adaptive representation

Let $X \in \mathbb{R}^{C_{\text{in}} \times H \times H}$ be an image with C_{in} channels, for example $C_{\text{in}} = 1$ for grayscale and $C_{\text{in}} = 3$ for RGB data. A quadtree partitions its domain into N non-overlapping square leaves, $\mathcal{B} = \{b_i\}_{i=1}^{N}$, where $b_i = (u_i, v_i, u_i + s_i, v_i + s_i)$ specifies a patch of side length s_i. The leaves cover the image exactly and are ordered by Z-order. Uppercase symbols denote whole images and lowercase symbols their leaves: we write $x_i = P_{b_i} X$ for the patch extracted at b_i and $\mathcal{A}_{\mathcal{B}}$ for the inverse operation, which assembles patches at their bounding boxes.

Our partitioning builds on the edge-guided quadtree of Zhang et al. (2024), adapted to produce exactly N spatial tokens with leaf sides in $[s_{\min}, s_{\max}]$, without dropping or padding leaves. Algorithm 1 in Appendix A gives the construction, using edge mass $q(b; M) = \sum_{p \in b} M_p$ as the default split priority. Dataset-specific region constraints and edge-detection settings are listed in Appendix B.

3.2 UniAE: a uniform-shape autoencoder for variable-size patches

Leaf side lengths differ by a factor of up to $s_{\max}/s_{\min}$, yet every leaf must map to one token of dimension d. We design UniAE, a uniform-shape autoencoder, so that quadtree leaves of every size share the same latent shape (Figure 3a). Size-specific layers appear only at its input and output; a single shared network operates in between.

Pixel unshuffle. For an entry resolution $e \le s_{\min}$, pixel unshuffle (Shi et al., 2016) rearranges a patch of size $s_i \times s_i$ into $C_{\text{in}}(s_i/e)^2$ channels on an $e \times e$ grid. A size-specific convolutional adapter then projects these channels to a common width W. We use $e = 32$ and $W = 64$ throughout this paper.

Shared trunk. After the adapters, every leaf already has the same shape $W \times e \times e$ regardless of its size. Leaves of all sizes are concatenated and pass through a single residual encoder trunk in one batch, which reduces the grid from $e \times e$ to $r \times r$.

Spatial latent. A 1×1 convolution maps the trunk output to d/r^2 channels on the $r \times r$ grid, and flattening this map gives the token z_i. Writing E_ϕ for the encoder, the image is represented by the sequence

$$z_i = E_\phi(x_i; s_i) \in \mathbb{R}^d, \qquad Z = [z_1, \ldots, z_N]^\top \in \mathbb{R}^{N \times d}. \tag{1}$$

Every leaf contributes one token of the same dimension, so its side length determines the local spatial compression rate.

Decoder. The decoder mirrors the encoder: a shared trunk gives a feature map $F_\phi(z_i) \in \mathbb{R}^{W \times e \times e}$, and a size-specific head $D^{\text{img}}_{\phi, s_i}$ with pixel shuffle restores the native patch,

$$\widehat{x}_i = D^{\text{img}}_{\phi, s_i}\big(F_\phi(z_i)\big), \qquad \widehat{X} = \mathcal{A}_{\mathcal{B}}\big(\{\widehat{x}_i\}_{i=1}^{N}\big). \tag{2}$$

The latent shape used for each dataset is given in Appendix B.

We train UniAE using a pixel-weighted reconstruction objective,

$$\mathcal{L}_{\text{ae}} = \mathbb{E}_X \left[\frac{1}{H^2} \sum_{i=1}^{N} \|\widehat{x}_i - x_i\|_1 \right]. \tag{3}$$

Summing native-pixel errors before normalization gives each image pixel equal weight despite the different leaf sizes. Once trained, UniAE defines a fixed latent space for both generative modeling and semantic decoding.

3.3 Geometry-aware latent diffusion

`QuaDiT`'s generative model learns a distribution over latent sequences placed on a layout $\mathcal{B}$, whose leaf geometry takes the place of the fixed token grid of a standard DiT, conditioned on an image-level label c when the dataset provides one. We adopt the rectified flow formulation (Liu et al., 2023; Lipman et al., 2023) and train the model with its velocity-matching loss. With $\epsilon \sim \mathcal{N}(0, I)$ and $Z_t = (1-t)\epsilon + tZ$,

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{Z,\epsilon,t} \left[\frac{1}{Nd} \|v_\theta(Z_t, t; \mathcal{B}, c) - (Z - \epsilon)\|_F^2 \right]. \tag{4}$$

A standard diffusion transformer (DiT; Peebles & Xie, 2023) parameterizes v_θ, with one output vector per leaf. Its fixed-grid 2D positional embedding is undefined for leaves of different sizes, and sequence indices alone do not specify a leaf's location or extent. `QuaDiT` therefore replaces it with a geometry positional embedding (geometry PE). For leaf center m_i and depth $\ell_i = \log_2(H/s_i)$,

$$g_i = \text{LN}\left(\text{Emb}(\ell_i) + \text{PE}(m_i) + \text{MLP}(\bar{b}_i)\right), \tag{5}$$

where $\bar{b}_i$ contains normalized bounding-box coordinates and normalized log widths and heights. The sinusoidal encoding $\text{PE}(m_i)$ of the leaf center is added to the projected input token, where DiT adds its grid embedding, and every block adds g_i to the token stream through a gated residual branch between attention and feed-forward (Figure 3b), much as rotary embeddings act in every attention layer (Su et al., 2024). The layout thus enters v_θ only as token positions, as the grid does in DiT; t and c are the only conditions.

At sampling time, $\mathcal{B}$ is the layout of a training image of class c, drawn at random from the layouts computed for training. We integrate

$$\frac{\mathrm{d}Z_t}{\mathrm{d}t} = v_\theta(Z_t, t; \mathcal{B}, c), \qquad Z_0 \sim \mathcal{N}(0, I), \qquad Z_{\text{gen}} = Z_1. \tag{6}$$

The layout is held fixed during integration; only its bounding boxes are used, not the pixels or latents of its source image, and the model generates the image content within this geometry.

3.4 Semantic decoding and paired synthesis

We attach a trainable semantic head to the frozen decoder feature map $F_\phi(z_i)$ (Figure 3c). The head contains two residual blocks followed by a size-specific convolutional adapter and pixel shuffle, producing K class logits at each native patch pixel. The head is conditioned on the same image-level class label c as the DiT. As with adaLN-Zero in a DiT block (Peebles & Xie, 2023), the class embedding produces a scale, a shift, and a zero-initialized gate for each residual block; without class labels this conditioning is omitted. The semantic logits and their assembled prediction are

$$\Lambda_i = D^{\text{mask}}_{\psi,s_i}\big(F_\phi(z_i); c\big), \qquad \widehat{Y} = \mathcal{A}_{\mathcal{B}}\left(\{\arg\max_k \Lambda_{i,k}\}_{i=1}^{N}\right). \tag{7}$$

The image-level class c is distinct from the K pixel classes. The head uses no encoder skip features from a source image, so it can operate directly on generated latents.

We train the head on simulator-generated image–mask pairs (Ziabari & Alnaggar, 2025). Because UniAE is trained on real images only, a label-preserving appearance transformation $\widetilde{X}^{\text{sim}} = T_c(X^{\text{sim}})$ first matches the simulator images to the real appearance of class c without changing their

geometry (Appendix B). The transformed images are partitioned with Algorithm 1 and encoded by UniAE, and labels are extracted with the same bounding boxes as the image patches. Let p_{qk} be the predicted probability for class k at pixel q, and y_{qk} its one-hot target. With Ω denoting the pixels of an image, we minimize

$$\mathcal{L}_{\text{mask}} = -\frac{1}{|\Omega|}\sum_{q\in\Omega}\sum_{k=1}^{K} y_{qk}\log p_{qk} + 1 - \frac{1}{K}\sum_{k=1}^{K}\frac{2\sum_{q\in\Omega} p_{qk}y_{qk} + \delta}{\sum_{q\in\Omega} p_{qk} + \sum_{q\in\Omega} y_{qk} + \delta}, \tag{8}$$

where δ is a numerical stabilizer. Cross-entropy and soft-Dice (Milletari et al., 2016) statistics are accumulated across the complete image rather than averaged over leaves. All classes and all image pixels participate in the training objective. Only ψ is updated, leaving the encoder, shared decoder features, and image output heads fixed.

For paired synthesis, the decoder features of the generated latents Z_{gen} are computed once per leaf and read by both the image and mask heads, which receive the same label c as the DiT. Assembling both outputs with the same layout yields spatially aligned image–mask pairs without re-encoding the generated image.

4 Experiments

4.1 Setup

Datasets. S8C consists of grayscale X-ray CT slices of three concrete specimens (`No_001`, `No_014`, and `No_015`) scanned at SPring-8. The supplied slices are native 8192^2 (8K); we downsample them to 4096^2 (4K) and 2048^2 (2K). The split contains 53,432 training and 2,856 validation images, separated by case folder. The image-level condition is $c = 0$ for `No_001` and $c = 1$ for `No_014` and `No_015`; all S8C generators use the same image identities. S8C has no paired segmentation masks. S8D provides simulated concrete images with five-class labels from the concrete microstructure simulation pipeline of Ziabari & Alnaggar (2025). Its 4K image–label pairs train and quantitatively evaluate the CT mask head, with grayscale augmentation matched to the two S8C conditions while geometry and labels are preserved. Hydrogel contains 9,000 synthetic 1024^2 TEM-style images of hydrogel networks with paired binary masks and centerlines (8,550 training, 450 validation). `QuaDiT` uses 199 leaves per image, subdivided by centerline density, and no class conditioning. Appendices B, C, and H give the full details.

Models and training. All generative models are at DiT-XL scale (28 blocks, width 1152, 16 heads). `QuaDiT` uses the DiT-XL architecture, with its grid positional embedding replaced by the geometry encoding of Section 3.3 (Figure 3b). It is trained from scratch with rectified flow; sampling uses EMA weights, 50 uniform Euler steps, and guidance scale 1. The main S8C baseline is dense latent DiT-XL/8 on an SD-style (Rombach et al., 2022) factor-8, 16-channel autoencoder, trained as two separate systems: DiT-RF shares `QuaDiT`'s modern recipe (rectified flow, SwiGLU (Shazeer, 2020), Q/K normalization (Henry et al., 2020), and identical AdamW settings (Loshchilov & Hutter, 2019)), whereas DiT-DDPM follows the original DiT recipe (Peebles & Xie, 2023) with the DDPM objective (Ho et al., 2020). JiT-XL/64 (Li & He, 2026) operates directly on pixels without an autoencoder, and RAE (Zheng et al., 2026) is evaluated only for autoencoder reconstruction (Appendix F). The main S8C comparison uses 16 NVIDIA GB200 GPUs, global batch 256, and checkpoints at 100k optimizer updates. Autoencoders and mask heads are trained separately. Table 5 in Appendix C lists the complete recipes.

Metrics. For S8C, global FID (Heusel et al., 2017) compares 2,856 generated images with the 2,856 validation images. `QuaDiT` samples draw their layouts without replacement from training images, matching the validation set in number per specimen. Patch-FID uses 18 native 256×256 crops per image, or 51,408 patches per side, lying entirely within the specimen ROI of each image. Global FID captures full-frame appearance after resizing, while Patch-FID measures specimen-interior texture at native resolution. Grayscale images are repeated to RGB before feature extraction. KID (Bińkowski et al., 2018) uses the same Inception-v3 features (Szegedy et al., 2016), and precision and recall (Kynkäänniemi et al., 2019) on native patches use VGG16 (Simonyan & Zisserman, 2015) features with $k = 3$. Reconstruction FID (rFID) for autoencoders uses the same images and crops. Mask heads are evaluated by macro Dice against paired simulator labels.

Table 1: Generation on S8C and Hydrogel; lower is better. S8C rows use 100k updates and batch 256, Hydrogel rows 75k updates. S8C `QuaDiT` layouts come from training images. Patch metrics use native 256×256 crops. Samplers: Euler50 for `QuaDiT` and DiT-RF, DDPM250 for DiT-DDPM, Heun50 for JiT. JiT's token count excludes its 32 in-context class tokens.

				Global		Patch	
Dataset	Res.	Model	Tokens	FID ↓	KID$\times 10^3$ ↓	FID ↓	KID$\times 10^3$ ↓
S8C	2K	`QuaDiT`	520	**25.06**	**16.3**	**64.66**	**63.0**
		DiT-RF	1,024	50.77	48.9	149.51	172.3
	4K	`QuaDiT`	1,501	**22.64**	**13.7**	**38.51**	**30.0**
		DiT-RF	4,096	46.84	47.9	73.73	74.2
		DiT-DDPM	4,096	84.79	103.5	88.28	88.4
		JiT	4,096	106.83	123.7	190.83	180.2
	8K	`QuaDiT`	6,004	21.18	10.9	18.81	13.5
Hydrogel	1K	`QuaDiT`	199	**21.86**	–	54.53	–
		DiT-RF	256	26.01	–	**35.20**	–
		DiT-DDPM	256	35.67	–	35.60	–

All generators receive the class label. FID is compared within each resolution, since the real-image distributions differ across resolutions. Appendices D and E report validation-layout FID, a memorization audit, and all metrics with real-image controls.

4.2 Main results on S8C

Table 1 reports the main generation results. At 4K and 100k updates, `QuaDiT` achieves global FID 22.64 and Patch-FID 38.51, compared with 46.84 and 73.73 for DiT-RF. Its 1,501-token representation uses 63.4% fewer spatial tokens than the 4,096-token dense baseline. DiT-DDPM and JiT have higher FID under their evaluated recipes; KID ranks all models identically. On native patches, `QuaDiT` has the highest recall (16.8% at 4K versus 7.0% for DiT-RF), covering more of the real texture distribution, while DiT-RF has higher precision (55.6% versus 38.1%); their F-scores are 23.3 and 12.5. Global precision and recall, with real-image controls, are reported in Appendix E. This gain does not stem from a better codec: UniAE has a higher reconstruction FID than the dense baselines' SD-VAE (Appendix F).

At 2K, `QuaDiT` also obtains lower global and patch FID than DiT-RF at the common 100k endpoint. At 8K, the 6,004-token model achieves global FID 21.18 and Patch-FID 18.81 at 100k updates, demonstrating generation at 8192×8192 pixels. No completed native-8K dense baseline is available. Appendix C reports extended 120k runs. Training layouts do not lead the model to reproduce their source images: among all 53,432 training images, the source is never the nearest neighbor of any of 192 8K samples (Appendix D).

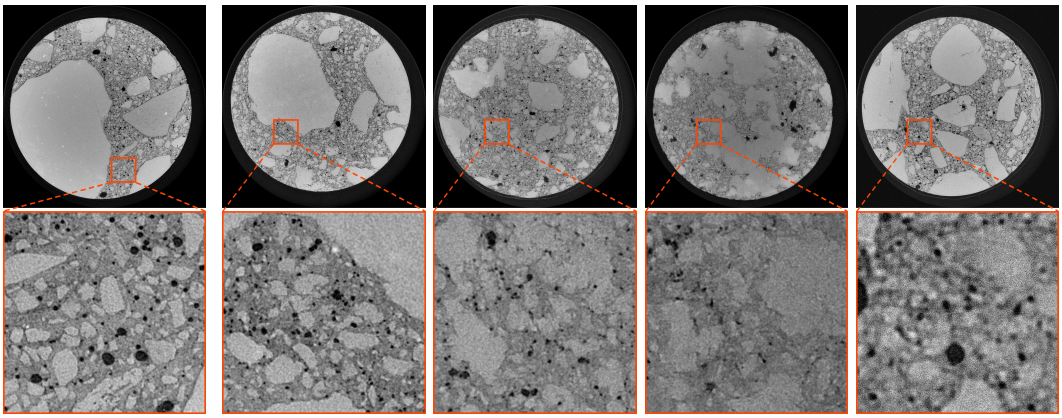
In Figure 4, all models produce the circular specimen and large aggregates, but at native resolution only `QuaDiT` renders sharp aggregate boundaries and isolated round pores in the mortar, matching the real texture.

Figure 5 shows FID across checkpoints. `QuaDiT` reaches global FID 25.08 at 50k updates, already below the 100k scores of every baseline, and its Patch-FID stays lowest at every evaluated checkpoint.

4.3 Compute and memory

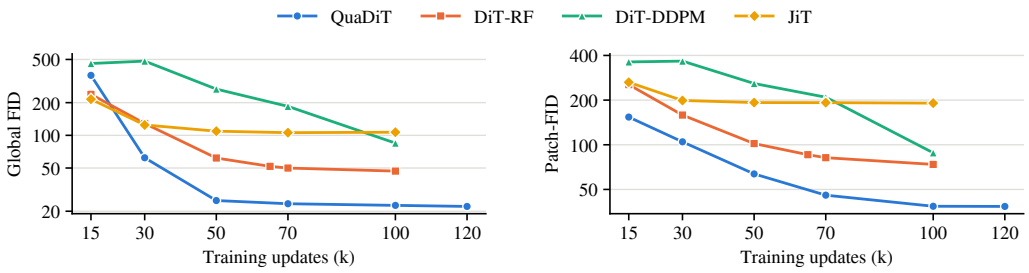
With 1,501 tokens, `QuaDiT` trains at $2.25\times$ the throughput of DiT-RF with 54.6% lower peak memory (Table 2), although it has more parameters, since transformer cost is dominated by token count. Autoencoder training and latent pre-encoding are excluded for all latent models.

At 8K, a dense DiT-XL/8 operates on 16,384 tokens; without activation checkpointing or gradient accumulation, DiT-RF and DiT-DDPM exceed the 184 GiB of a GB200 even at local batch 8, whereas `QuaDiT` trains with 6,004 tokens.



(a) Real (reference) (b) QuaDiT (ours) (c) DiT-RF (d) DiT-DDPM (e) JiT

Figure 4: Generated S8C images at 4K ($c = 1$). (a) A real slice as texture reference. (b)–(e) Samples drawn from noise by the 100k checkpoints of Table 1; they are not reconstructions of (a), so aggregate arrangements differ by design. Top: full 4096^2 images. Bottom: the boxed native 512×512 region, at the same coordinates in every sample.



(a) Global FID (b) Patch-FID

Figure 5: S8C 4K FID over training (EMA weights, batch 256). The FID axes are logarithmic. Each curve uses one sampler throughout: Euler50 for QuaDiT and DiT-RF, DDPM250 for DiT-DDPM, and Heun50 for JiT. QuaDiT is additionally evaluated at 120k after its learning-rate decay.

4.4 Mask decoding and paired synthesis

We evaluate semantic decoding using paired simulator images and labels: five-class masks on S8D and binary masks on Hydrogel. On S8D, five-class macro Dice on the test split is 0.968 for the class-0 appearance and 0.996 for class 1, with validation scores of 0.967 and 0.992. On Hydrogel, validation macro Dice is 0.863. Scores on the held-out splits closely match the training scores in every case. Appendix G contains the protocol and full tables.

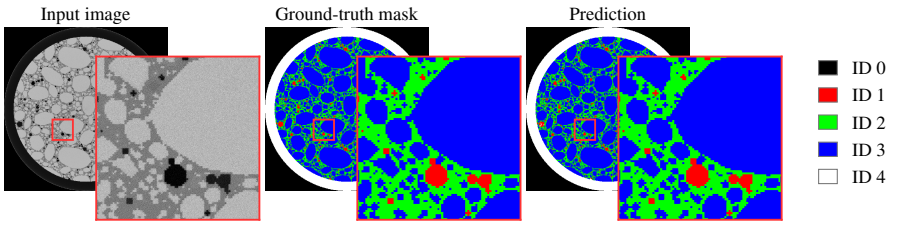
S8C has no paired reference masks, so Figure 6 shows qualitative predictions for a real S8C image, from its encoded latents, and a QuaDiT-generated image, decoded directly from the generated latents, next to a held-out S8D test example.

4.5 Hydrogel results

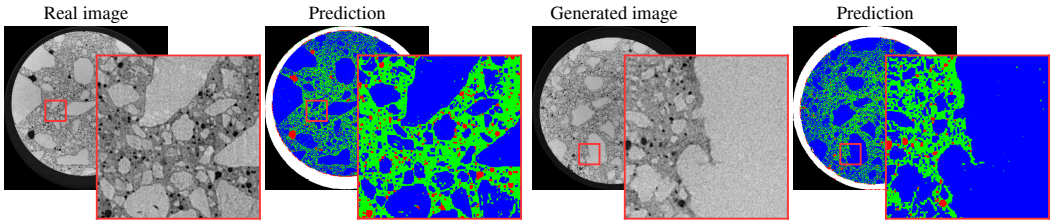
On Hydrogel at 1024^2, with models trained separately from S8C, QuaDiT uses 199 tokens versus 256 for the dense baselines and reaches global FID 21.86 at the common 75k endpoint, compared with 26.01 for DiT-RF and 35.67 for DiT-DDPM (Table 1). It has the lowest global FID at every shared checkpoint (Table 16), while its 75k Patch-FID of 54.53 is higher than those of DiT-RF (35.20) and DiT-DDPM (35.60).

Table 2: Generative-model training cost at 4K on 16 GB200 GPUs, global batch 256, local batch 16, BF16, no gradient accumulation. Params refer to the generative model. Peak memory is PyTorch peak allocated tensor memory per GPU.

Model	Parameters (M)	Tokens	ms/step ↓	Train images/s ↑	Peak GiB ↓
`QuaDiT` (ours)	751.8	1,501	**214.2**	**1,195.3**	**45.60**
DiT-RF	680.8	4,096	481.0	532.2	100.45
DiT-DDPM	678.3	4,096	458.3	558.6	100.87
JiT	679.8	4,096	494.0	518.2	103.29



(a) S8D test sample



(b) Real S8C image

(c) `QuaDiT`-generated image

Figure 6: Qualitative mask-head results. (a) A sample drawn from the S8D test set with the $c = 1$ appearance condition: input image, simulator ground-truth mask, and prediction. (b) A real S8C image from `No_014` and its predicted mask. (c) A `QuaDiT`-generated image and its mask prediction decoded from the generated latents. All masks share the same color map, with pixel-class IDs 0–4. Red boxes mark native 512×512 regions enlarged in the insets, with identical crop coordinates within each group. The real and generated crops are selected to show relatively clean aggregate interiors and boundaries, rather than average segmentation quality.

5 Conclusion

We presented `QuaDiT`, a framework for high-resolution scientific image–mask synthesis built on spatially adaptive latent tokens. A budgeted quadtree and a shared autoencoder allocate representation capacity across variable-size regions, while a geometry-aware transformer generates their latent features. At 4K on concrete CT, `QuaDiT` uses 1,501 tokens and achieves lower global and patch FID than the evaluated baselines, with $2.25\times$ the generative-model training throughput of DiT-RF and 54.6% lower peak allocated memory. It also supports native 8192×8192 image generation. A simulator-supervised semantic head decodes aligned mask predictions from the same features, with quantitative mask evaluation on labeled synthetic data. On synthetic Hydrogel, lower global FID coexists with worse patch FID at the final checkpoint, leaving local texture fidelity as an open challenge. The transfer of semantic supervision from simulations to real and generated CT also requires further study: mask evaluation on these images remains qualitative. Expert annotations and downstream evaluations are needed to establish the utility of the generated image–mask pairs for scientific analysis.

AI USE STATEMENT

In this work, we used generative AI tools for code generation, for running experiments and organizing experimental data (e.g., collecting metrics from experiment logs into tables), for literature search, for drafting parts of the paper, and as an editorial aid for writing and grammar. We have not used generative AI tools for research ideation: the research questions, the experimental design, the analyses and all claims are the authors' own. We have reviewed all AI-assisted work; generated code was reviewed by the authors and its outputs verified against the results reported here, references found with AI assistance were checked against the original publications, and all AI-assisted text was checked and revised by the authors. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Code will be made publicly available at `https://anonymous.4open.science/r/quadit-FCF3`. Implementation details and hyperparameters are provided in Appendix.

REFERENCES

Anwai Archit, Luca Freckmann, Sushmita Nair, Nabeel Khalid, Paul Hilt, Vikas Rajashekar, Marei Freitag, Carolin Teuber, Melanie Spitzner, Constanza Tapia Contreras, Genevieve Buckley, Sebastian von Haaren, Sagnik Gupta, Marian Grade, Matthias Wirth, Günter Schneider, Andreas Dengel, Sheraz Ahmed, and Constantin Pape. Segment anything for microscopy. *Nature Methods*, 22(3):579–591, 2025. doi: 10.1038/s41592-024-02580-4.

Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*, 2022.

Benjamin Billot, Douglas N. Greve, Oula Puonti, Axel Thielscher, Koen Van Leemput, Bruce Fischl, Adrian V. Dalca, Juan Eugenio Iglesias, and ANDI. SynthSeg: Segmentation of brain MRI scans of any contrast and resolution without retraining. *Medical Image Analysis*, 86:102789, 2023. doi: 10.1016/j.media.2023.102789.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

Min Jin Chong and David Forsyth. Effectively unbiased FID and inception score and where to find them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6070–6079, 2020.

Rohan Choudhury, JungEun Kim, Jinhyung Park, Eunho Yang, László A. Jeni, and Kris M. Kitani. Accelerating vision transformers with adaptive patch sizes. In *International Conference on Learning Representations*, 2026. URL `https://arxiv.org/abs/2510.18091`.

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*, 2021.

Fernando García-Torres, Rocío del Amor, Sandra Morales, Álvaro Barroso, Peter Heiduschka, Björn Kemper, and Valery Naranjo. DualDiT: A conditional dual-output diffusion transformer for joint OCT image and segmentation mask generation. *arXiv preprint arXiv:2607.29337*, 2026. URL `https://arxiv.org/abs/2607.29337`.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, 2023.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2511.13720.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Jiawei Mao, Yuhan Wang, Yucheng Tang, Daguang Xu, Kang Wang, Yang Yang, Zongwei Zhou, and Yuyin Zhou. MedSegFactory: Text-guided generation of medical image-mask pairs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21525–21535, 2025.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *International Conference on 3D Vision (3DV)*, pp. 565–571, 2016.

Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in PyTorch, 2020. URL https://doi.org/10.5281/zenodo.4957738. Version 0.3.0.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2337–2346, 2019.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11410–11420, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. URL https://arxiv.org/abs/2002.05202.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1874–1883, 2016.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2107–2116, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Weijia Wu, Yuzhong Zhao, Hao Chen, Yuchao Gu, Rui Zhao, Yefei He, Hong Zhou, Mike Zheng Shou, and Chunhua Shen. DatasetDM: Synthesizing data with perception annotations using diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Donglin Yang, Paul Vicol, Xiaojuan Qi, Renjie Liao, and Xiaofan Zhang. QDM: Quadtree-based region-adaptive sparse diffusion models for efficient image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 5044–5053, 2026.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.06940.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. PixelDiT: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2511.20645.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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*, 2024. doi: 10.1109/SC41406.2024.00082.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3836–3847, 2023.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. DatasetGAN: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10145–10155, 2021.

Zhengqiang Zhang, Rongyuan Wu, Lingchen Sun, and Lei Zhang. GPSToken: Gaussian parameterized spatially-adaptive tokenization for image representation and generation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2509.01109.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.11690.

Amirkoushyar Ziabari and Mohammed Alnaggar. PyCMG-based simulation of volumetric concrete microstructure, 2025. URL https://doi.org/10.13139/ORNLNCCS/2548178. Dataset, Oak Ridge National Laboratory.

SUPPLEMENTARY MATERIALS

A FIXED-LENGTH QUADTREE CONSTRUCTION

Algorithm 1 builds the layout of Section 3.1. Starting from the whole image, it repeatedly splits the leaf with the highest priority into four children. Each split adds three leaves, so $(N-1)/3$ splits yield exactly N leaves. Leaves larger than $s_{\max}$ are split first, and leaves of side $s_{\min}$ are never split, so every leaf side lies in $[s_{\min}, s_{\max}]$. The final Morton sort gives the Z-order used as the token order.

Algorithm 1 Edge-guided quadtree with fixed sequence length

Input: Image $X \in \mathbb{R}^{C_{\text{in}} \times H \times H}$; Gaussian kernel size k; Canny thresholds $\tau_{\text{low}}, \tau_{\text{high}}$; leaf size limits $s_{\min} \le s_{\max}$ with $H/s_{\min}$ and $H/s_{\max}$ powers of two; target length N with $(H/s_{\max})^2 \le N \le (H/s_{\min})^2$ and $(N-1) \bmod 3 = 0$; split-priority function $q(b; M)$.
Output: Morton-ordered layout $\mathcal{B} = (b_1, \ldots, b_N)$.

$X_g \leftarrow \mathit{GaussianBlur}(\mathit{Luminance}(X); k)$
$M \leftarrow \mathit{CannyEdge}(X_g; \tau_{\text{low}}, \tau_{\text{high}})$
$\mathcal{T} \leftarrow \{(0, 0, H, H)\}$
$\mathcal{Q} \leftarrow$ max-priority queue of splittable leaves in $\mathcal{T}$, keyed lexicographically by $\big(\mathbb{1}[\mathit{SideLength}(b) > s_{\max}],\ q(b; M)\big)$
for $j = 1$ to $(N-1)/3$ **do**
  $b \leftarrow \mathit{PopMax}(\mathcal{Q})$
  $\mathcal{T} \leftarrow (\mathcal{T} \setminus \{b\}) \cup \mathit{SplitIntoFour}(b)$
  for all $b' \in \mathit{SplitIntoFour}(b)$ **do**
    if $\mathit{SideLength}(b') > s_{\min}$ **then**
      Insert b' into $\mathcal{Q}$
    end if
  end for
end for
$\mathcal{B} \leftarrow \mathit{MortonSort}(\mathcal{T})$
return $\mathcal{B}$, with $|\mathcal{B}| = N$

B DATASET-SPECIFIC CONFIGURATION

The method in Section 3 is stated independently of any dataset. This appendix lists the settings chosen per dataset: layout construction (Table 3), autoencoder and output dimensions (Table 4), and the simulator appearance transformation T_c used to train the mask branch.

Table 3: Layout construction settings for Algorithm 1. S8C uses Canny edges (Canny, 1986) with a $k \times k$ Gaussian blur and condition-dependent thresholds. Hydrogel substitutes simulator centerlines for Canny edges; dashes denote inapplicable parameters.

Dataset	H	N	$s_{\min}$	$s_{\max}$	k	$(\tau_{\text{low}}, \tau_{\text{high}})$	
						$c=0$	$c=1$
S8C	2048	520	64	512	7	(50, 100)	(100, 200)
S8C	4096	1501	64	1024	7	(50, 100)	(100, 200)
S8C	8192	6004	64	2048	7	(50, 100)	(100, 200)
Hydrogel	1024	199	64	128	–	–	

S8C layout construction. Each slice is first processed with a circular specimen ROI detector. The quadtree is then built with two size caps: leaves inside the ROI are subdivided until their side is at

most 128 pixels, whereas leaves outside the ROI may remain as large as $s_{\max}$. The remaining split budget up to N is allocated by the edge mass $q(b; M)$ inside the ROI. Canny thresholds depend on the image-level condition: $(50, 100)$ for $c = 0$ (`No_001`) and $(100, 200)$ for $c = 1$ (`No_014` and `No_015`), both after a 7×7 Gaussian blur.

Hydrogel layout construction. The layout starts from an 8×8 grid of 128×128 parents. We count simulator centerline pixels in each parent and subdivide the 45 parents with the highest counts into four 64×64 leaves each. Ties are broken by row-major parent order. This yields $45 \times 4 = 180$ fine leaves and $64 - 45 = 19$ coarse leaves, ordered by Morton (Z) order. Fine leaves cover 70.3125% of the image area. The regular tree above the 128-pixel scale is implicit. No ROI restriction or Canny preprocessing is applied. UniAE crops leaves at their native resolution on demand.

Table 4: UniAE and output dimensions. e: entry resolution; W: trunk width at entry; d: token dimension, with its spatial latent shape; K: pixel classes of the mask head; A: image-level classes. Mask heads are trained only at 4K; dashes denote codecs without a mask head.

Dataset	H	e	W	d	K	A
S8C	2048	32	64	1024 ($16 \times 8 \times 8$)	–	2
	4096	32	64	1024 ($16 \times 8 \times 8$)	5	2
	8192	32	64	1024 ($16 \times 8 \times 8$)	–	2
Hydrogel	1024	32	64	1024 ($16 \times 8 \times 8$)	2	1

S8C conditions and S8D appearance augmentation. The image-level condition is $c = 0$ for S8C specimen `No_001` and $c = 1$ for `No_014` and `No_015`. These conditions are shared by the CT generator and mask head and are separate from the five semantic pixel classes. The label-preserving transformation T_c augments the grayscale appearance of the S8D simulations according to the intensity characteristics of the corresponding S8C condition. Geometry and semantic labels are preserved. Mask evaluation uses each condition separately on the S8D training, validation, and test splits; the split sizes are listed in Table 14.

C Additional experimental details

Generative-model recipes. Table 5 lists the training and sampling recipes of the three latent generative models compared on S8C. `QuaDiT` and DiT-RF share the same modern recipe (rectified flow, SwiGLU, Q/K RMS normalization, and the same AdamW settings); DiT-DDPM follows the original DiT recipe. The DiT-RF versus DiT-DDPM comparison therefore changes a package of training and architecture choices rather than a single factor. For `QuaDiT`, the learning rate is constant through 100k updates; the extended runs decay it linearly to zero at 120k. JiT-XL/64 operates on native grayscale pixels with 4,096 spatial tokens, 32 in-context class tokens, RMSNorm (Zhang & Sennrich, 2019), rotary position encoding (Su et al., 2024), SwiGLU, and an effective learning rate of 2×10^{-4}. Its evaluation uses EMA decay 0.9999, 49 two-evaluation Heun intervals (Karras et al., 2022) and a final Euler interval. The 2K DiT-DDPM training became unstable; its 100k checkpoint reaches global FID 461.59 and Patch-FID 404.58. No RAE generative stage was run.

Data and preprocessing. The source SPring-8 XCT collection comprises 46 specimens of highway concrete and asphalt composites scanned using the SPring-8 BL28B2 high-energy XCT system at approximately 200 keV; S8C uses only the three specimens named in Section 4.1. We model individual two-dimensional slices. The 53,432 training and 2,856 validation images come from 337 and 18 case folders, respectively; the same specimen identifiers occur in both splits. S8C images are supplied at 8192×8192 pixels. The 4K and 2K images are obtained by downsampling these native 8K slices. The S8C downstream split excludes 832 images from five case folders and retains 56,288 images in total. UniAE and RAE use the retained training list. SD-VAE's loader excludes validation folders but does not apply the five-folder exclusion: the 2K log records 54,264 training images, and the 4K recipe has the same loader behavior. The 4K external validation-case file is no longer available for an exact comparison. We therefore do not assume identical training pools for

Table 5: Training and sampling recipes of the latent generative models on S8C. All three use the DiT-XL backbone (28 blocks, width 1,152, 16 heads, adaLN-Zero), global batch 256 on 16 GPUs without gradient accumulation, BF16, zero weight decay, EMA decay 0.9999, and guidance scale 1. Parameter counts are for the 4K models.

	`QuaDiT`	DiT-RF	DiT-DDPM
Latent codec	UniAE, 1,024 per leaf	SD-VAE f8c16	SD-VAE f8c16
Tokens at 4K	1,501	4,096	4,096
Position encoding	Leaf center + geometry	Fixed 2D grid	Fixed 2D grid
Objective	Rectified-flow velocity	Rectified-flow velocity	ϵ-prediction, learned var.
Time sampling	$\sigma(\mathcal{N}(0,1))$	$\sigma(\mathcal{N}(0,1))$	Uniform over 1,000 steps
Feed-forward	SwiGLU	SwiGLU	GELU MLP
Q/K normalization	RMS	RMS	–
Timestep LayerNorm	Yes	Yes	–
AdamW betas	(0.9, 0.95)	(0.9, 0.95)	(0.9, 0.999)
Learning rate	2×10^{-4}	2×10^{-4}	1×10^{-4}
Gradient clipping	1.0	1.0	–
Class dropout	0.1	0	0
Sampler	Euler, 50 steps	Euler, 50 steps	DDPM, 250 steps
Parameters (M)	751.8	680.8	678.3

every autoencoder. Dense DiT training and normalization use the retained 53,432-image training list.

`QuaDiT` standardizes the 1,024 latent coordinates using statistics pooled over leaves from 200 validation images, separately at each resolution. Dense DiT instead uses channel statistics from the complete training split. `QuaDiT`'s validation-derived normalization is part of the reported protocol. The 4K quadtree layouts are inherited from earlier 8K trees by halving bounding-box coordinates; the 6,004-token 8K experiment uses separately prepared geometry.

Autoencoders. The S8C 2K, 4K, and 8K experiments use separate UniAE codecs with 520, 1,501, and 6,004 leaves per image. The 2K and 4K codecs have native heads through 512 pixels; the 8K codec has heads through 2,048 pixels. In the 4K layout, leaves larger than 512 use the 512-pixel head and are resized to their bounding boxes at decoding. The selected UniAE checkpoints each represent ten training epochs. The 2K/4K/8K global image batches are 192/12/48, with peak learning rates 8×10^{-4}/2×10^{-4}/4×10^{-4}. All use AdamW, two warmup epochs, cosine decay, and no EMA. SD-VAE uses separate 2K and 4K EMA checkpoints at 200k steps, trained on native-scale 256×256 crops. The RAE checkpoint uses a frozen DINOv2-B encoder and a trainable merger and ViT-XL (Dosovitskiy et al., 2021) decoder trained for four epochs. The evaluated RAE first resizes the 4K input to 3584^2, then encodes 8×8 tiles of 448^2 pixels independently.

The dense 4K training cache uses tiled encoding with shared GroupNorm statistics, whereas its rFID evaluation performs fresh full-image encoding and decoding. Thus, that rFID does not measure the exact encoding path used for the cached training latents.

S8C FID implementation. FID uses `torch-fidelity 0.3.0` (Obukhov et al., 2020), TensorFlow-compatible Inception-v3 features of dimension 2,048, float32 feature extraction with TF32 disabled, and float64 feature moments. The full image or native crop is resized to 299×299 for feature extraction; resizing and compression choices affect FID values (Parmar et al., 2022). No additional contrast normalization is applied. Generation seeds are indexed from 0 and crop seeds from 1,000,000 in validation-list order. ROI or crop failures are errors, without a full-frame fallback. There are 880 class-0 and 1,976 class-1 validation images. FID estimates have finite-sample bias (Chong & Forsyth, 2020), and crops from a single image are correlated. The recorded results do not include independent training-seed repetitions or confidence intervals.

Extended training. Extending the `QuaDiT` runs from 100k to 120k updates, with the learning rate decayed linearly to zero, changes global/patch FID from 25.06/64.66 to 23.06/65.27 at 2K, from 22.64/38.51 to 22.14/38.41 at 4K, and from 21.18/18.81 to 21.44/21.65 at 8K. The main com-

parison retains the common 100k endpoint. Table 6 lists every evaluated checkpoint with training-set layouts.

Table 6: S8C QuaDiT FID across checkpoints with training-set layouts (global / patch). Lower is better.

Updates	2K	4K	8K
15k	196.40 / 197.04	356.69 / 153.73	278.90 / 167.34
30k	92.76 / 136.55	62.14 / 104.68	101.43 / 74.64
50k	29.43 / 85.13	25.08 / 63.61	30.06 / 29.14
70k	25.73 / 69.46	23.44 / 45.74	24.10 / 29.07
100k	25.06 / 64.66	22.64 / 38.51	21.18 / 18.81
120k	23.06 / 65.27	22.14 / 38.41	21.44 / 21.65

8K samples. Figure 7 shows uncurated 8K samples from the 120k checkpoint.

D LAYOUT SOURCE AND MEMORIZATION AUDIT

Training-set layouts. The S8C results in the main text draw QuaDiT's layouts from the training split. For each resolution, 2,856 layouts are sampled uniformly without replacement within each specimen, matching the validation counts: 880 from No_001 (class 0), and 1,064 and 912 from No_014 and No_015 (class 1); the selection seed is 20260924. Generation reads only the bounding boxes and the class label, and uses EMA weights at 100k updates, 50 Euler steps, guidance scale 1, and generation seed 0. The FID reference is the 2,856-image validation set, whose case folders are disjoint from the training split. Patch-FID crops of a generated image lie within the specimen ROI of its layout's source image, and validation crops within their own ROIs; the two crop sets are compared as distributions, without pairing.

Validation-set layouts. An alternative protocol generates one image from the layout of each validation image. Table 7 compares the two. Validation layouts share the coarse specimen geometry of the reference images and give lower global FID, whereas training layouts give lower Patch-FID at every resolution. Both protocols rank QuaDiT ahead of DiT-RF at 2K and 4K (Table 1). The 8K comparison uses the 120k checkpoint, the only 8K checkpoint evaluated with validation layouts. Table 8 lists the class-specific scores with training layouts.

Table 7: S8C QuaDiT FID by layout source. 2K and 4K use 100k updates; 8K uses 120k. Lower is better.

	Global FID ↓		Patch-FID ↓	
Resolution	Training layouts	Validation layouts	Training layouts	Validation layouts
2K	25.06	20.14	64.66	77.51
4K	22.64	17.98	38.51	40.98
8K	21.44	18.04	21.65	22.50

Table 8: Class-specific S8C QuaDiT FID with training-set layouts. Class 0 has 880 generated and validation images, class 1 has 1,976. 2K and 4K use 100k updates; 8K uses 120k.

	Global FID ↓			Patch-FID ↓		
Resolution	All	Class 0	Class 1	All	Class 0	Class 1
2K	25.06	26.71	29.29	64.66	120.72	56.80
4K	22.64	30.91	23.44	38.51	55.77	38.69
8K	21.44	24.48	24.80	21.65	47.42	18.13

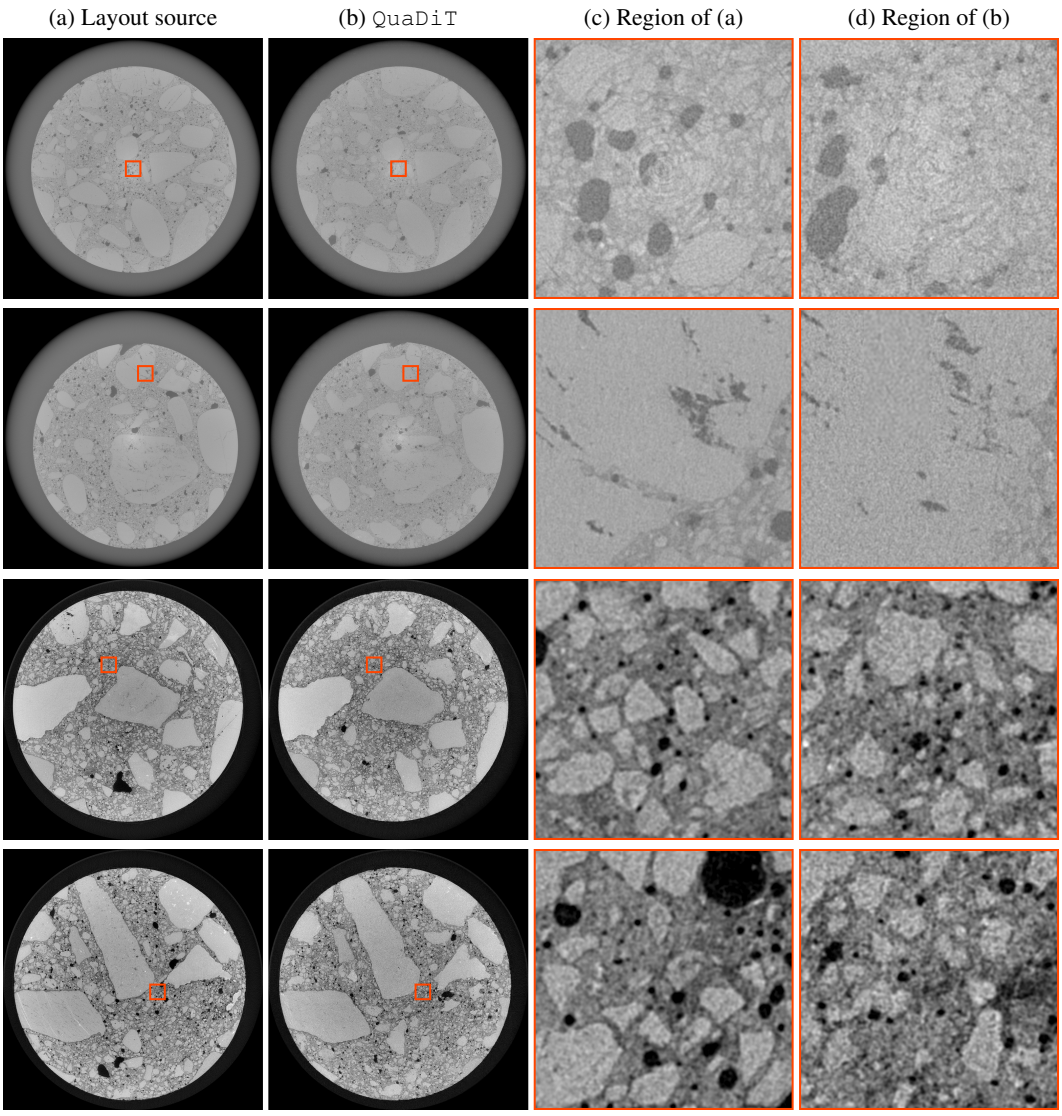


Figure 7: Uncurated S8C samples at 8192×8192 (EMA, 120k updates, 50 Euler steps, guidance scale 1). Rows use two class-0 layouts (`No_001`) and two class-1 layouts (`No_014`, `No_015`), drawn at random from the training split without regard to sample quality. (a) The real image whose 6,004-leaf layout conditions (b). (c), (d) The boxed native 512×512 region of (a) and (b); it is the 512-pixel window containing the most 64-pixel leaves of the layout, so it is chosen from the layout alone.

Memorization audit. A layout computed from a training image carries that image's coarse geometry, so we test whether generating on it reproduces the source image. We generate 192 8K images (EMA, 120k updates, 50 Euler steps, guidance scale 1) from training layouts, 64 per specimen, each from a different case folder, with noise seeds disjoint from those of the FID runs. Each sample is compared with its layout's source image and, by exhaustive search, with all 53,432 training images, using the 2,048-dimensional Inception features of the FID pipeline. SSIM (Wang et al., 2004) is computed after Lanczos resizing to 1024^2 with Gaussian weighting ($\sigma = 1.5$), on the full image and on the central 80% of its width and height, which reduces the contribution of the shared black exterior. As a calibration, each source image is also compared with its nearest other training image.

The source is never the nearest training image under Inception L2 distance (0/192; minimum rank 2, median rank 692 of 53,432), and no source distance approaches zero. No generated image reaches a center SSIM of 0.80 to its source (maximum 0.606), whereas neighboring real slices reach a median of 0.888 (Table 9). The generated images are also far from their nearest training images, so the search reveals no retrieval of a different training image either. Similarity is highest for `No_001`, whose layouts convey large aggregate regions (median center SSIM 0.552), but remains well below that group's real-to-real baseline of 0.960; the most similar samples (Figure 8) share coarse aggregate arrangement but differ in pores, inclusions, and fine texture. The audit thus finds no evidence that `QuaDiT` reproduces the source image of a training layout; it does not address every form of training-data memorization.

Table 9: Training-layout memorization audit at 8K (192 samples). Center SSIM uses the central 80% of each image.

	Inception L2	Center SSIM	
Comparison	median	median	max
Generated vs. layout source	7.00	0.170	0.606
Generated vs. nearest training image	5.55	0.161	0.599
Source vs. nearest other training image	3.13	0.888	0.965

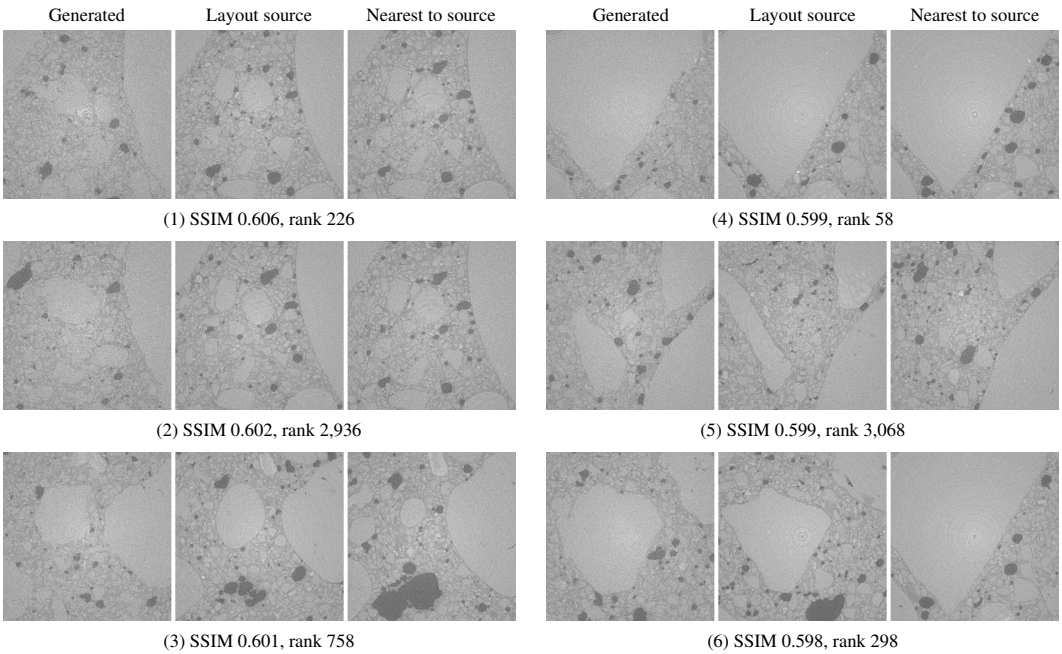


Figure 8: The six 8K samples of the memorization audit with the highest center SSIM to their layout sources, all from `No_001`, shown as central details. Each case shows the generated image, its layout source, and the training image nearest to the source under Inception L2. Labels give the generated-to-source center SSIM and the rank of the source among the 53,432 training images by Inception L2 distance to the generated image.

E ADDITIONAL GENERATION METRICS

Protocol. KID (Bińkowski et al., 2018) uses the 2,048-dimensional Inception features of the FID pipeline and the cubic polynomial kernel $k(x, y) = (x^{\mathsf{T}}y/2048 + 1)^3$ with the unbiased estimator, averaged over 100 random subsets of 1,000 samples per side (seed 20260924); we report the mean $\pm$ the standard deviation over subsets, scaled by 10^3. Precision and recall (Kynkäänniemi et al., 2019) use 4,096-dimensional VGG16 features, $k = 3$, and all 2,856 global or 51,408 patch features per side, with exact nearest-neighbor search. Precision is the fraction of generated features inside at least one real k-NN ball, and recall the fraction of real features inside at least one generated k-NN ball. All scores use the samples of Table 1.

Table 10: KID ($\times 10^3$, mean $\pm$ subset SD), precision (P, %), and recall (R, %) on S8C at 100k updates. Lower KID and higher P/R are better.

		Global			Patch		
Res.	Model	KID	P	R	KID	P	R
2K	`QuaDiT`	16.3 ± 0.5	0.0	26.1	63.0 ± 1.7	7.9	0.4
	DiT-RF	48.9 ± 0.9	3.0	0.4	172.3 ± 3.2	27.4	0.2
4K	`QuaDiT`	13.7 ± 0.6	0.0	19.4	30.0 ± 1.3	38.1	16.8
	DiT-RF	47.9 ± 1.0	0.1	2.0	74.2 ± 2.4	55.6	7.0
	DiT-DDPM	103.5 ± 1.5	0.1	0.1	88.4 ± 2.7	26.8	1.9
	JiT	123.7 ± 1.7	0.0	0.0	180.2 ± 2.7	0.0	0.0
8K	`QuaDiT`	10.9 ± 0.4	0.1	21.3	13.5 ± 1.1	60.0	34.1

Results. KID reproduces the FID ranking at every resolution and scale (Table 10). `QuaDiT` has the highest recall among the compared models at 2K and 4K, globally and on patches, and at 4K its patch recall of 16.8% is more than twice that of DiT-RF. DiT-RF has the higher patch precision.

Global precision and real-image controls. Global precision is close to zero for every model. To calibrate it, we score real images against the same validation reference (Table 11): the source images of the 2,856 training layouts reach only 1.6% and 2.0% precision at 2K and 4K, whereas two alternating halves of the validation set reach nearly 100% against each other. More than 99.9% of the three nearest neighbors of a validation image come from its own case folder, so at $k = 3$ the real balls describe variation among related slices of one case, and an image from any other case, real or generated, rarely falls inside them. Precision rises with k for both generated and real training images (Table 12), and $k = 3$ in Inception feature space also gives zero `QuaDiT` precision. Global precision therefore measures overlap with the neighborhoods of the sampled validation cases, and we compare models on the patch metrics, KID, and FID.

Table 11: Global precision controls (VGG16, $k = 3$, %). Probes are scored against the validation reference; the last column is the share of validation 3-NN relations within the same case folder.

Res.	`QuaDiT`	Training real	Validation halves	Same-case 3-NN
2K	0.00	1.65	99.86	99.93
4K	0.00	2.03	99.93	99.97

F AUTOENCODER RECONSTRUCTION

UniAE is a deterministic autoencoder trained with pixel-weighted L1 reconstruction, without KL, perceptual, or adversarial losses; each leaf maps to $16 \times 8 \times 8$, or 1,024 values, irrespective of its native size. The dense baseline's SD-style KL autoencoder is trained with reconstruction, perceptual, KL, and adversarial objectives, and an 8×8 latent patch also contains 1,024 values. The RAE comparison uses a tiled DINOv2-B (Oquab et al., 2024) encoder with a trainable token merger and ViT-XL decoder; only its reconstruction stage was run. SD-VAE has the lowest measured

Table 12: Global precision (%) as a function of k at 100k updates, for `QuaDiT` samples and real training images scored against the validation reference.

	2K		4K	
k	`QuaDiT`	Training real	`QuaDiT`	Training real
3	0.00	1.65	0.00	2.03
5	0.04	3.75	0.04	3.82
10	0.21	7.04	0.42	7.77
20	0.95	15.90	2.52	19.64
50	9.73	47.20	18.52	53.12

reconstruction FID and the highest PSNR (Table 13). UniAE's PSNR rises with resolution, from 31.33 dB at 2K to 38.82 dB at 8K. Thus, `QuaDiT`'s lower generation FID is not explained by a lower autoencoder rFID. Reconstruction quality and the difficulty of modeling a latent representation are distinct aspects of the complete system; rFID is not a guaranteed lower bound on generation FID. Codec configurations are given in Appendices B and C.

Table 13: S8C autoencoder reconstruction. rFID is measured at 4K, and tokens denote transformer inputs at 4K; RAE uses tiled encoding. PSNR uses the same 2,856 validation images at each resolution, computed on the full frame after rounding reconstructions to uint8, with peak value 255, and averaged over images. Dashes denote codecs not trained at that resolution.

			4K rFID ↓		PSNR (dB) ↑		
Autoencoder	Tokens	Values/token	Global	Patch	2K	4K	8K
UniAE	1,501	1,024	12.62	20.37	31.33	35.41	38.82
SD-VAE f8c16	4,096	1,024	**0.33**	**0.47**	**35.79**	**44.77**	–
RAE (tiled)	4,096	768	132.21	28.04	–	21.70	–

G Mask evaluation protocol and results

S8D provides simulated concrete images and semantic labels produced with the concrete microstructure simulation pipeline (Ziabari & Alnaggar, 2025); we use its 4K Noise_1 image–label pairs. S8D supplies both mask supervision and labeled evaluation data, whereas the S8C image generator is trained on real CT images. The CT mask head learns five pixel classes from S8D image–label pairs while the image autoencoder remains frozen. Grayscale augmentation matches the two S8C image-level conditions, and the same condition c is supplied to the mask head. For quantitative evaluation, simulator images are encoded and their predicted masks are compared with the paired simulator labels. S8D Dice is computed separately for $c = 0$ and $c = 1$ on each of the training, validation, and test splits (Table 14), using the epoch-19 mask-head checkpoint. We report macro Dice over all five pixel classes on the full image. Dice exceeds 0.966 on every split for $c = 0$ and 0.991 for $c = 1$, and validation and test scores closely match training scores under both conditions.

Hydrogel uses a binary mask head and simulator-provided binary masks. Its Dice evaluation compares predictions from encoded dataset images with their paired labels on the training and validation splits (Table 15), using the epoch-18 checkpoint. We report macro Dice over the background and foreground classes. Validation Dice is 0.863, matching the training score of 0.862. These evaluations assess semantic decoding on paired synthetic data.

Real and generated CT images. S8C does not provide paired reference masks. Figure 6 presents mask-head predictions on a real S8C image and a `QuaDiT`-generated image, alongside a paired S8D test example. The S8D example is the middle entry of the domain-1 test manifest, selected without quality filtering; its prediction uses no post-processing. Real-image predictions use encoded image latents; generated-image predictions are decoded directly from the generated latents, without re-encoding the image. The S8C panels assess visual agreement between images and masks; quantitative Dice is reported on the paired synthetic data. The figure insets show native 512×512

Table 14: S8D mask evaluation by S8C appearance condition and data split. Class 0 matches `No_001`; class 1 matches `No_014`/`No_015`. Both conditions use the same five pixel classes and simulator reference labels. Dice is the macro average over the five pixel classes; higher is better.

Appearance condition	Split	Images	Dice
Class 0	Training	2,503	0.969
	Validation	266	0.967
	Test	290	0.968
Class 1	Training	2,503	0.996
	Validation	266	0.992
	Test	290	0.996

Table 15: Hydrogel binary mask evaluation against paired simulator labels. Image counts refer to the dataset splits. Dice is the macro average over background and foreground; higher is better.

Split	Images	Dice
Training	8,550	0.862
Validation	450	0.863

crops without mask smoothing or relabeling. Their top-left (x, y) coordinates in the 4096^2 images are $(1216, 2304)$ for S8D, $(1024, 1856)$ for real S8C, and $(1168, 2944)$ for `QuaDiT` synthesis. Real and generated crops were visually selected for relatively clean aggregate interiors with visible boundaries; they are not random samples of local segmentation quality.

H HYDROGEL DATASET AND ADDITIONAL RESULTS

Dataset groups and split. `hydrogel_synth` v2 contains 9,000 synthetic 1024^2 grayscale images mimicking mineral-stained transmission electron microscopy (TEM) images of hydrogel networks, with paired binary masks and centerline maps. Centerlines determine the quadtree layout but are not an additional generator input, and the generator and binary mask head use no image-level class conditioning. The three parameter groups, `1101`, `1201`, and `1301`, each contain 3,000 images and are tuned to statistics of corresponding real-image groups. From `1101` to `1301`, the simulator increases network density and reduces dangling chains, with parameters randomized within each group. A stratified 95/5 split assigns 2,850 training and 150 validation images per group (split seed 20260917). Each image has 180 leaves of side 64 and 19 of side 128, without Canny edges or an ROI restriction; Appendix B gives the subdivision rule.

Synthetic network and image formation. The v2 simulator is calibrated to a reference collection of 45 real hydrogel TEM image–label pairs, organized into groups `1101`, `1201`, and `1301`. These identifiers denote simulator parameter groups. A smooth Gaussian random field creates spatially varying density. Nodes combine a non-uniform Poisson process with Thomas-cluster points; nearby nodes are probabilistically connected by worm-like chains. Dangling chains, random breaks, side spurs, and star-shaped clusters add structural variation. Key geometry and imaging parameters receive multiplicative log-normal jitter within each group.

Chains are rendered as discrete Gaussian beads on a weak continuous mass baseline. Gaussian point-spread blur, soft saturation, and exponential attenuation convert the mass map into grayscale intensity, with background fluctuations and density-dependent shading. Added Gaussian noise is colored using a radial power spectrum measured from the real images. Images are quantized to 8-bit grayscale and JPEG-compressed at quality 70–85 before PNG storage. The simulator uses per-image random seeds determined by the base seed 2, group, and image index, and records randomized parameters for each image. The disjoint, exhaustive training and validation lists split independently generated images within each group; they do not hold out an entire parameter group.

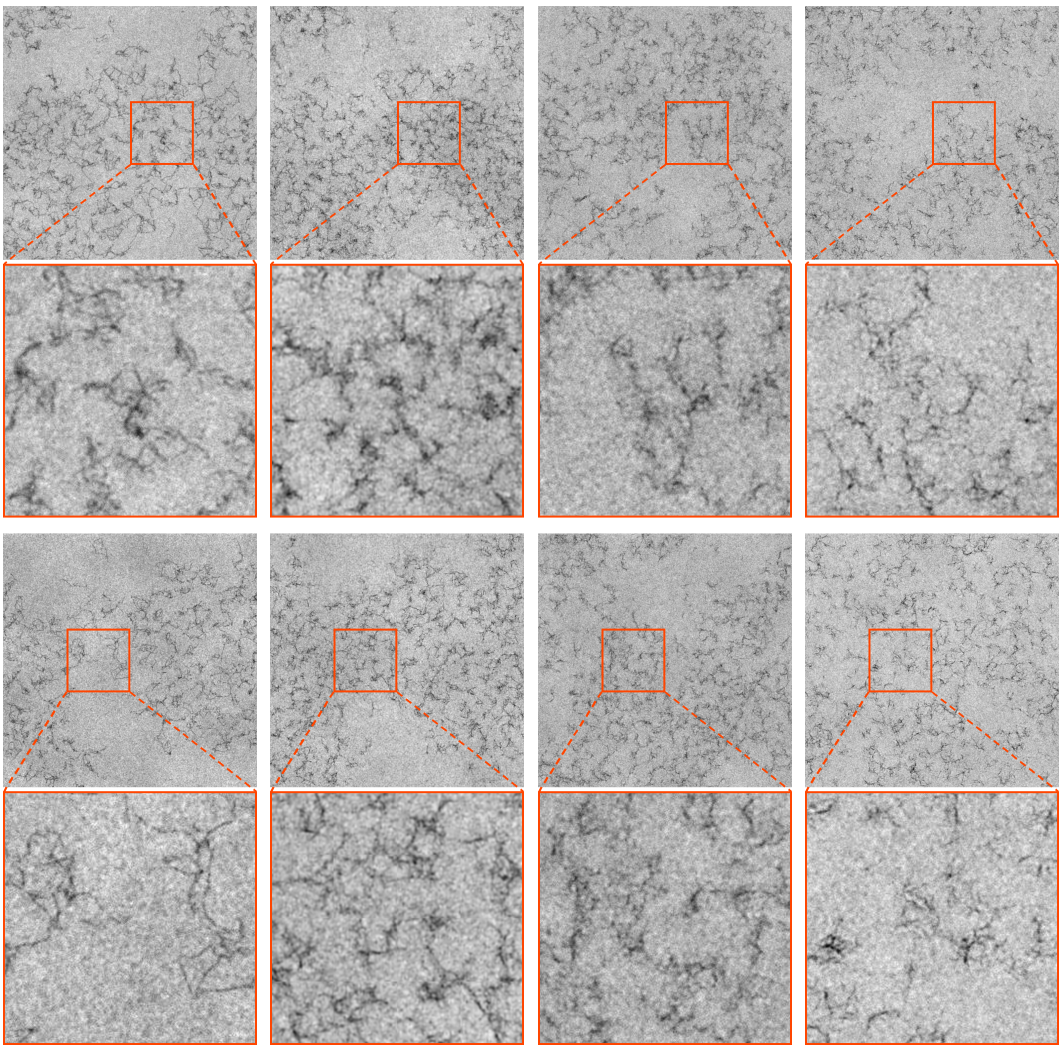
Labels and scope. The simulator exports both exact one-pixel rasterized centerlines and annotation-style binary masks. Masks combine radius-1 dilation of the centerlines, radius-2 dila-

tion in a spatially smooth region covering 4–12% of the image, and small discs at approximately half of the crosslink nodes and at star-cluster cores. These masks provide the binary supervision; centerlines supply layout geometry as described in Appendix B. Labels are generated procedurally for each synthetic image, while the simulator's appearance and annotation style are informed by real reference data. The dataset documentation reports more fragmented synthetic masks, less pronounced large dark regions and star clusters, and thinner, less smooth annotations than in the real data. Evaluation on this synthetic split therefore does not establish performance on real TEM images.

Table 16: Hydrogel generation across shared training checkpoints. Lower FID is better; bold marks the lowest score at each checkpoint for each metric. `QuaDiT` scores retain the three decimal places of the supplied results; baseline scores were reported to two.

	Global FID			Patch256 FID		
Updates	`QuaDiT`	DiT-DDPM	DiT-RF	`QuaDiT`	DiT-DDPM	DiT-RF
15k	**155.122**	271.54	199.28	**86.402**	335.46	110.66
30k	**57.867**	157.81	111.77	**61.644**	155.91	68.75
50k	**28.995**	157.11	45.39	53.416	79.75	**40.88**
75k	**21.857**	35.67	26.01	54.526	35.60	**35.20**

Generation across checkpoints. Table 16 compares `QuaDiT` with both dense DiT baselines at 15k, 30k, 50k, and 75k updates. `QuaDiT` has the lowest global FID at all four checkpoints. Its Patch256 FID improves from 86.402 at 15k to 53.416 at 50k, then increases slightly to 54.526 at 75k. Both baseline patch scores improve monotonically across the reported checkpoints. The main comparison uses the common 75k endpoint; Figure 9 shows samples from the 75k checkpoints.



(a) Data (layout source) (b) `QuaDiT` (ours) (c) DiT-RF (d) DiT-DDPM

Figure 9: Generated Hydrogel images at 75k updates, two examples. (a) A `hydrogel_synth` validation image whose 199-leaf layout conditions the `QuaDiT` sample in (b). (b)–(d) Samples drawn from noise; they are not reconstructions of (a). For each example, the first row shows the full 1024^2 image and the second row the boxed native 256×256 region, at the same coordinates in every column.