BigCell: Generating Gigapixel Whole-Slide Images

Srikar Yellapragada[1] **Alexandros Graikos**[1] **Zilinghan Li**[2] **Kostas Triaridis**[1]
Tarak Nath Nandi[2,3] **Karen Bai**[1] **Beatrice Knudsen**[4] **Tahsin Kurc**[1] **Prateek Prasanna**[1]
Rajarsi R. Gupta[1] **Ravi K. Madduri**[2,3] **Joel Saltz**[1,5] **Dimitris Samaras**[1]

[1]Stony Brook University [2]Argonne National Laboratory [3]University of Chicago
[4]The University of Utah [5]The University of North Carolina at Chapel Hill

Abstract

Whole-slide images (WSIs) are gigapixel-scale digital scans of histopathology tissue and serve as the primary basis for cancer diagnosis. Existing histopathology generative models operate at the patch scale, producing fixed-size tiles of at most 1024×1024 pixels which encompass only a small fraction of the WSI. In contrast, text reports for clinical diagnoses and slide-level diagnostic labels – the supervision that exists at scale – operate at the whole-slide level. We present BigCell, the first approach to generate coherent whole-slide histopathology images at native gigapixel resolution. BigCell trains a flow-matching transformer at 2048×2048 pixels on over 120,000 publicly available WSIs spanning multiple organ types. Conditioning on embeddings from pathology foundation models and using an inference-time sliding-window pipeline, BigCell generates coherent images of up to $32{,}768\times32{,}768$ pixels, outperforming prior tile generators on both tile-level and slide-level fidelity metrics. To make gigapixel synthesis practical, we introduce an adaptive scheduling strategy that prioritizes important slide regions, speeding up image generation by over $3\times$with minimal quality degradation. Furthermore, we propose the first text-to-image synthesis at 8k resolution by training a separate model that produces dense conditioning grids directly from pathology text reports. We demonstrate that augmenting with synthetic samples improves slide-level few-shot classification performance by up to 18% over baselines trained on real data.

1 Introduction

Diffusion and flow-matching have enabled high-fidelity natural image synthesis at megapixel resolution [52, 46, 10]. Approaches that push beyond the megapixel-scale rely on the self-similarity of natural scenes, synthesizing fine-grained material detail from coarser inputs [33]. In this paper, we address a fundamentally different challenge: the synthesis of histopathology whole-slide images (WSIs). WSIs are high-resolution digital scans of tissue sections that often exceed $50{,}000\times50{,}000$ pixels, and are the primary source of information for pathologists to diagnose diseases, including cancer. Crucially, they exhibit diverse structures at each magnification: from cellular morphology at the finer scales to glandular and tumor architecture at the coarser scales. Moreover, distant regions of a WSI are tightly coupled as staining properties are globally consistent, and cell morphology must be compatible with the surrounding tissue type (e.g., tumor cells should not appear in a region of normal tissue). In natural scenes, by contrast, such global constraints are fewer and usually weaker. Pathologists rely on spatial structures that span thousands of pixels to perform accurate diagnoses, making the full resolution of a WSI key to its diagnostic value.

Corresponding author: `myellapragad@cs.stonybrook.edu`. Code and models provided at the project page.

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

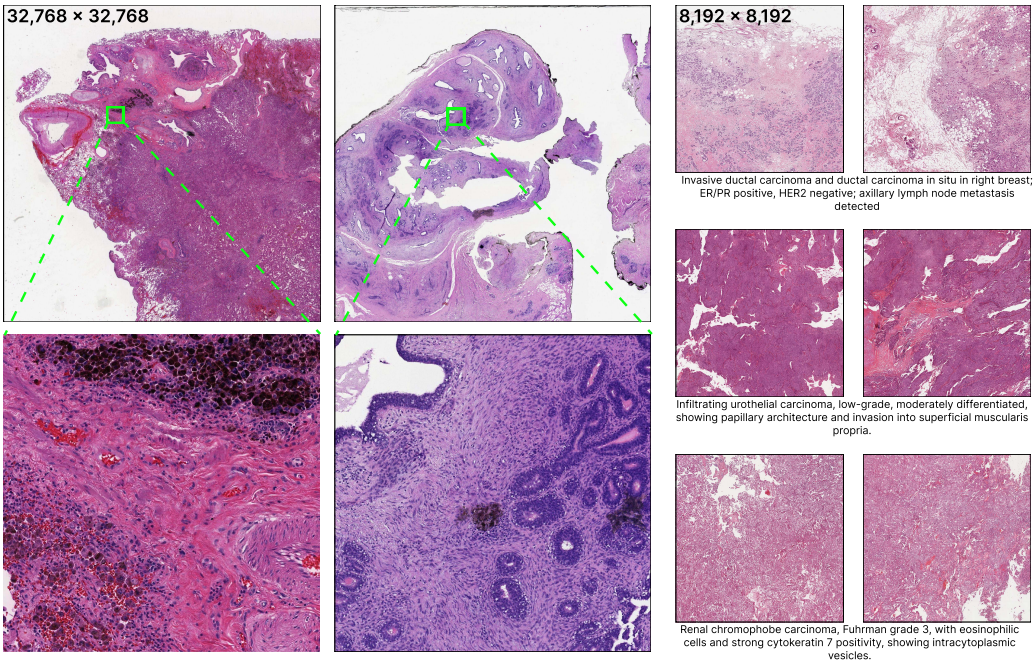

Figure 1: We present **synthetic** images generated using BigCell. Left: 32768×32768 images along with 2048×2048 crops, highlighting the realism achieved both at the macro-scale and cellular detail level. Right: Two 8192×8192 images per report generated from text. Best viewed zoomed in.

Existing generative methods in histopathology operate at the patch-level, synthesizing fixed-size tiles of WSIs (typically 256 to 1024 pixels) conditioned on semantic maps [2, 3], textual descriptions [7, 19, 45], or learned representations [14, 17, 49, 50]. One of the best-performing models, PixCell [49], generates tiles of up to 1024×1024 pixels natively, still orders of magnitude below whole-slide resolution. The scale gap is only one part of the problem: patch-level framing is often misaligned with pathology reporting. Pathologists work with WSIs, not tiles, and the clinical reports they produce describe findings at the level of the entire slide or case, with the latter comprising multiple slides. Pixel-level annotation at any meaningful scale is prohibitively expensive (Graikos et al. [17] report 40 hours of pathologist time to annotate tiles from a single WSI with captions), while slide-level reports and labels are routinely produced in clinical practice.

Modern computational pathology systems are increasingly being trained and evaluated at levels approaching whole-slides [5, 13, 24, 40, 54]. A collection of independently generated tiles cannot faithfully represent slide-level morphology, report-level diagnostic content, or long-range tissue organization. A generative model that aims to capture the distribution of pathology data and scale with the supervision that is widely available must therefore operate at the whole-slide level.

We present BigCell, the *first* method to generate coherent whole-slide histopathology images at native gigapixel resolution in a single stage. Since training directly at this scale is computationally infeasible, we propose a scheme that progressively increases the base training resolution from 256 to 2048, controlling generation with embeddings from foundation models. Upon training a large-enough tile generator, we adopt a sliding-window inference-time algorithm [17] that synthesizes images beyond the training resolution, using the dense foundation model conditioning embeddings as a global blueprint, dictating exactly what tissue morphology belongs where. Training on over 120,000 publicly available WSIs spanning multiple organ types, our approach scales to produce images of up to 32,768×32,768 pixels, the typical size of a small to medium WSI (Figure 1-left).

Sliding-window generation at gigapixel scale, however, is expensive: a single 32k pixel image requires denoising hundreds of overlapping windows at every timestep. We observe that not all windows require the same effort: complex regions rely heavily on the conditioning signal, while uniform regions (such as background or homogeneous stroma) only require coarse conditioning information to be synthesized accurately. We measure this per-window importance as the difference between conditional and unconditional model predictions (akin to classifier-free guidance [21]) and

use it to reduce the number of generation steps in the identified low-importance regions. This strategy speeds up generation by over 3 times with minimal loss in image quality, is training-free, and requires only a single interpretable hyperparameter controlling the fraction of important windows.

To rigorously evaluate generation at this scale, we decouple local and global metrics: we measure local fidelity using standard patch-level similarity and Fréchet distance scores, and global coherence using slide-level foundation model similarity. When using previous histopathology tile generators for 8192×8192 image generation, we find that although they achieve high patch-level quality, their limited training resolution degrades the structural coherence across broader regions. Consequently, BigCell achieves superior macro-scale fidelity, reflected in higher foundation model similarity and improved bottom-5th-percentile patch-level similarity.

The above pipeline requires a real WSI to extract the conditioning embedding grid from. The most commonly available (and often the only) information about WSIs is the pathology report that accompanies it. Pathology reports describe tissue type, morphology, diagnosis, grade, and ancillary findings and are abundant in clinical archives. We train a flow-matching model, the conditioning generation model (CGM), that operates in the foundation model's representation space, generating dense conditioning grids suitable for BigCell's tile generator, directly from report text. The full grid is generated jointly, capturing the long-range dependencies present at this scale. The resulting pipeline produces 8192×8192 pixel histopathology images from free-form text descriptions, generating tissue whose alignment with the input report, as measured by both board-certified pathologists and multiple pathology foundation models, approaches that of real paired data (Figure 1-right). To our knowledge, BigCell is the first attempt to achieve text-controlled generation at this scale.

The application we evaluate our generations on is synthetic data augmentation for computational pathology. We use BigCell to generate diverse and class-consistent variations of real training samples in various few-shot classification settings. On the TCGA-UT-8K [13] and PANDA [9] datasets, augmenting the real training set with synthetic samples from BigCell yields up to 18% relative improvement in performance over baselines trained on real data alone. Beyond data augmentation, we expect the report-to-image technology to impact clinical practice in many ways. For example, it can empower clinicians to explore counterfactual scenarios by generating side-by-side comparisons of subtle diagnostic alternatives, which will be useful for difficult differential diagnoses; it can enhance diagnostic quality control by visualizing report findings to cross-check against the patient's slides. It can also improve histopathology education by generating teaching cases of rare morphologic variants.

In summary, our contributions are:

(1) We present BigCell, the first method to generate whole-slide images at native gigapixel resolution, outperforming prior generative methods in global coherence and image fidelity.

(2) We develop an inference-time strategy that uses classifier-free guidance magnitude to accelerate sliding-window generation, achieving over 3 times speedup with minimal quality degradation.

(3) We show how learning the distribution of conditioning embeddings from clinical pathology reports enables the first text-to-pathology image generation at 8192×8192 resolution.

(4) We demonstrate that synthetic samples from BigCell improve few-shot classification performance on two downstream datasets and convey the same diagnostic content as real tissue when verified by board-certified pathologists.

The entire BigCell pipeline is trained exclusively on publicly available datasets with no proprietary data. Our trained models and code are publicly released to facilitate further research.

2 Related Work

Natural image synthesis has become synonymous with latent diffusion [38] and transformer-based denoisers [36, 34], with flow matching [29, 1] emerging as the prevailing approach. By scaling the latent diffusion recipe, prior work has pushed native generation to megapixel resolution (e.g., Stable Diffusion 3 [16] and FLUX [26]). Pushing beyond the training resolution is typically handled by inference-time approaches that synthesize an image in overlapping windows [4, 27] or by fusing

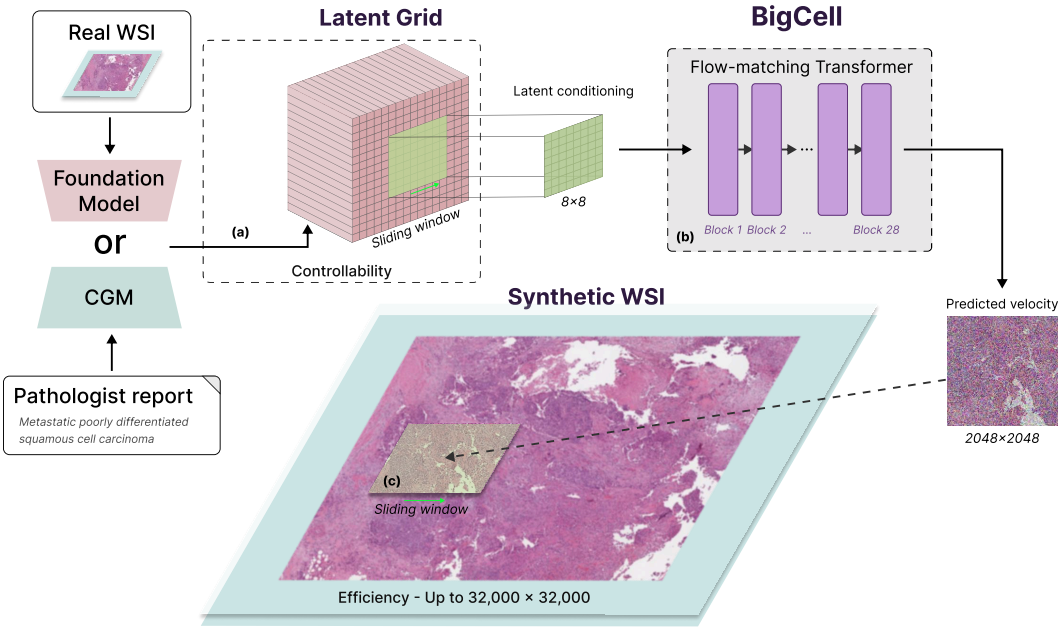

Figure 2: Overview of the BigCell pipeline. (a) We obtain a dense grid of conditioning embeddings either from a reference WSI or sampled using a model conditioned on text (CGM). This serves as the controllable blueprint that defines the large-scale structures of the whole-slide image. (b) The tile generator uses the local conditioning to predict how to transform the slide locally. (c) Using the tile generator on overlapping windows, we aggregate the per-window velocities into a global velocity field and iteratively denoise the noisy slide into a gigapixel image, allowing us to efficiently synthesize WSIs without explicitly training on their scale.

predictions from multiple resolution branches to reach higher resolutions [37]. These approaches rely on the assumption that local appearance dominates fidelity, so distinct regions can be rendered independently without impacting global semantics. Whole slide images break this assumption: cellular morphology must match with the surrounding tissue type across thousands of pixels, and staining must remain globally consistent.

Computational pathology typically follows a two-stage approach for downstream tasks: a patch-level foundation model [11, 53, 8, 32, 6] provides dense tissue embeddings, and a multiple instance learning (MIL) model [22, 31] aggregates them into a slide-level prediction. More recently, the field is moving toward slide-level foundation models [13, 40, 47, 24] and context-aware patch encoders [5, 54] whose representations depend on their neighbors. These works have demonstrated that modeling dependencies beyond a single patch yields stronger representations for downstream tasks.

Generative models in histopathology have thus far operated at the patch scale, conditioned on semantic segmentation maps [3, 2], text captions [48, 7, 19, 45], and embeddings from discriminative pathology foundation models [17, 50, 49, 14]. At the whole-slide level, Harb et al. [20] produce outputs of up to 65,536×65,536 pixels with a multi-stage cascade: an unconditional model samples a 512×512 thumbnail, which successive super-resolution stages progressively upsample. However, their WSI is sampled from a learned distribution only once, at the thumbnail level, and every subsequent stage refines a local, blurred crop of the stage below. DiffInfinite [3] synthesizes large mask-image pairs by utilizing a per-pixel tissue-class segmentation mask with parallel patch diffusion. Distant patches are coupled only through the mask, which does not encode global properties such as staining, and the method has not been demonstrated at gigapixel scale.

The representation-conditioned line of work is closest to our setting: LRDM [17] introduced sliding-window inference over an embedding-conditioned latent diffusion model, ZoomLDM [50] added magnification-aware training and cross-scale consistency [18], and PixCell [49] progressively trained from 256 up to 1024 native pixels on over 69,000 slides, reaching 4096 through inference-time stitching. BigCell denoises the full latent canvas at 20× in a single stage, conditioning every window

on a dense grid of 1536-dimensional foundation model embeddings that is either extracted from a reference slide or sampled jointly for the entire region by the CGM (Section 3.4).

3 Method

3.1 Tile generator

The core of BigCell is a flow-matching transformer (Figure 2(b)) that generates 2048×2048 pixel tiles conditioned on a dense grid of pathology foundation model embeddings. Following the latent diffusion framework, we operate in the latent space of a pretrained VAE [25], which encodes a 2048×2048 tile into a 256×256×32 latent z. The generative model is a Scalable Interpolant Transformer (SiT) [34] parameterized as a velocity predictor $v_\theta(z_t, t, y)$, where z_t is the noisy latent at timestep $t \in [0, 1]$ and y is the conditioning grid. We refer to this tile generator as BigCell $_{\text{tile}}$.

To condition the model, we follow the LRDM formula [17], which uses features extracted from discriminative foundation models (Figure 2(a)). For each 2048×2048 tile, we extract UNI-2 [11] features from the 64 non-overlapping 256×256 patches, yielding an 8×8×1536 conditioning grid y that is injected into the transformer via cross-attention layers.

3.2 Sliding-window generation

Training directly at whole-slide resolution is infeasible. To generate images larger than the native 2048 resolution, we extend the LRDM framework [17] to flow matching (Appendix C). Given a target image of $H \times W$ pixels, we first assemble a global conditioning grid $y \in \mathbb{R}^{(H/256)\times(W/256)\times 1536}$ of UNI-2 features, extracted from a reference WSI, or generated from text as in Section 3.4. At each denoising step, we run BigCell $_{\text{tile}}$ independently on overlapping windows and aggregate the per-window velocities into a global velocity field, which we then use to take a deterministic Euler step (Figure 2(c)). After the final step, the VAE decodes the latent canvas to pixels.

The sliding-window procedure is training-free and can be applied to any tile generator. The fidelity of the final image, however, depends directly on the underlying tile generator's ability to faithfully render the conditioning blueprint. Using a higher native resolution, which captures a sufficiently large spatial context per velocity window, yields better fidelity at the gigapixel scale; in Appendix E.1, we ablate how gigapixel image quality improves monotonically with native resolution from 256 up to 2048, with diminishing returns after 1024. We also demonstrate that BigCell $_{\text{tile}}$ outperforms tile generators from prior work [49, 14, 45] across tile-level and slide-level metrics (Section 5.1). We refer to the sliding-window pipeline as BigCell $_{\text{large}}$.

3.3 Efficient generation

BigCell $_{\text{large}}$ applies the tile generator on every window, which at gigapixel resolution, amounts to hundreds of forward passes per generation step, making it prohibitively expensive. A straightforward way to reduce this cost is to denoise only a subset of windows at each step, but this raises the question of which windows to prioritize.

We observe that windows have varying dependence on the conditioning signal. Some regions contain complex morphological structures (tumor boundaries, heterogeneous stroma) with the conditioning enforcing a particular cellular arrangement. Other regions (background, homogeneous stroma) are largely shaped by the unconditional prior, making the conditional and unconditional velocities nearly identical. We posit that such "simpler" windows also require fewer steps to be synthesized faithfully. Therefore, we propose to reuse the same velocity from the previous step as an approximation to the current velocity for less important windows, reducing the total number of forward passes.

Importance via classifier-free guidance. To distinguish between "harder" and "easier" windows, we measure how much each window relies on the conditioning during inference. During generation we already use classifier-free guidance (CFG) [21], which requires both the conditional $v_\theta(z_{ij}, t, y_{ij})$ and unconditional velocities $v_\theta(z_{ij}, t, \emptyset)$. The magnitude of their difference is therefore available for free, and we can use it to define a per-window importance score, $I_{ij}(t) = ||v_\theta(z_{ij}, t, y_{ij}) - v_\theta(z_{ij}, t, \emptyset)||_1$. A high $I_{ij}(t)$ indicates that the conditioning signal strongly influences the velocity prediction. We provide a visualization of this importance signal in Appendix Figure 10.

The idea of allocating compute across tokens was proposed by Wu et al. [44] in the context of a single image. Their method discards information from low-importance tokens within a forward pass and cannot be used to share information across tiles. Our formulation achieves that by caching per-window velocities across denoising steps and reusing them when a window is deemed low-importance.

Adaptive scheduling. Let $r \in (0, 1]$ denote the fraction of windows to denoise per step. At the first denoising step, we compute $I_{ij}(t)$ for all windows. At each subsequent step, we compute velocities only for the top-r most important windows from the previous step, and update their importance scores accordingly. Skipped windows reuse the cached velocity from the most recent step in which they were denoised. This strategy is training-free, requires only a single interpretable hyperparameter r, and reduces compute by approximately $1/r$. We ablate this tradeoff in Section 5.2, and show that we can achieve over 3 times speedup with minimal quality degradation.

3.4 Conditioning generation model (CGM)

BigCell $_{\text{large}}$ and BigCell $_{\text{tile}}$ require a dense embedding grid y as conditioning. In the default setting, y is extracted from a reference WSI, which ties generation to existing slides. To enable novel generation from free-form text, we train a separate generative model that produces y directly from a pathology report. We call this model the conditioning generation model (CGM).

Architecture and training. The CGM is trained on a subset of WSIs paired with pathology reports describing tissue type, morphological findings, and diagnosis. Each report is encoded using the TITAN text encoder [13], yielding a sequence of text embeddings c. CGM is also a flow-matching transformer that learns to sample from the conditional distribution $p(y|c)$, operating directly in the UNI representation space. Given a target UNI grid $y \in \mathbb{R}^{32\times32\times1536}$ that corresponds to a 8192×8192 pixel region, we apply the same linear interpolant formulation as BigCell $_{\text{tile}}$ to train a model that samples y from the given text tokens c.

Text-to-image generation. At inference, we sample a UNI grid from the CGM conditioned on TITAN text embeddings, pass it to BigCell $_{\text{large}}$ and produce an 8192×8192 pixel image. In Section 5.3 we showcase how this two-stage process synthesizes gigapixel images from text that exhibit the desired properties specified in the given prompt. To the best of our knowledge, our method is the first to achieve histopathology text-to-image generation at this scale.

3.5 Synthetic data augmentation

The CGM also enables a targeted form of data augmentation for downstream classification tasks. Given a real image from the training set, we can use the CGM to synthesize class-consistent variations by perturbing and denoising the embedding grid under a class-specific text prompt.

For a classification task with label set $\mathcal{L}$, we first construct m synthetic pathology reports per class by prompting an LLM with the class name, yielding text embeddings $\{c_l\}_{l\in\mathcal{L}}$. Given a real training example (x, l) with UNI grid y_0, we add noise to the embeddings up to an intermediate timestep t', and then denoise it back to $t = 0$ using the CGM conditioned on c_l. The resulting UNI grid $\hat{y}_0$ is passed through BigCell $_{\text{large}}$ to produce a synthetic image $\hat{x}$, and the pair $(\hat{x}, l)$ is added to the training set. This preserves the global structure of the original image while injecting class-consistent variations driven by the text prompt. For all augmentation experiments, we use $t' = 0.5$.

4 Experimental Setup

4.1 Data

Training data. Our training data consists of 120,703 WSIs from six publicly available sources: TCGA [43], CPTAC [15], GTEx [30], HistAI [35], PTRC-HGSOC [12], and NLST [41]. We extract 2048×2048 pixel tiles at 20×magnification, and split the dataset at the slide level to obtain 15,996,332 tiles for training and 852,435 for evaluation.

CGM training data. The CGM is trained on the subset of slides with paired pathology reports, drawn from TCGA diagnostic and HistAI, totaling approximately 30k patient-level reports. A single report may correspond to multiple slides from the same patient. We extract all 8192×8192 pixel regions across associated slides and construct target UNI grids, yielding 500k paired (grid, text embedding)

samples for CGM training. We choose 8k as our operating point because at larger resolutions (e.g., 32k), the number of regions drops sharply, leaving too few samples to train a CGM. We hold out 150 patient-level reports and the corresponding slides as a test set for all text-conditioned evaluations.

Downstream classification data. We perform k-shot classification on two datasets: PANDA [9], a prostate cancer dataset with 6 ISUP grade classes, and TCGA-UT-8K [13], which contains 8192×8192 pixel regions across the 32 target cancer types.

4.2 Metrics

We evaluate generated images on tile-level fidelity, slide-level coherence, report-level agreement, and text alignment:

Tile-level fidelity. We extract non-overlapping 256×256 tiles from each generated slide and their corresponding reference slides. For each tile pair, we compute cosine similarity in the feature space of Virchow-2 [53] and H-Optimus-1 [8]. We report two statistics per slide: the average tile similarity ($V2_{mean}$, $H1_{mean}$), and the bottom 5th percentile tile similarity ($V2_{5th}$, $H1_{5th}$). The latter captures the prevalence of severe local artifacts that average similarity may miss. Additionally, we compute the Fréchet distances in both encoder spaces ($V2_{FD}$ and $H1_{FD}$) on the tiles.

Slide-level fidelity. To evaluate coherence beyond individual tiles, we compute cosine similarity between the generated and reference images in the feature space of slide-level foundation models: TITAN [13] and MOOZY [24].

Report-level agreement. To complement embedding-based metrics with a measure that does not rely on pathology foundation models, we use MedGemma-4B [39], a general-purpose medical VLM. We generate structured reports and measure the fraction of synthetic/real pairs whose reports agree on the organ, on the diagnosis, and on all three fields. (details in Appendix F).

Text alignment. For CGM-generated outputs, we evaluate alignment with the conditioning report by computing CLIP-style similarity between image and text embeddings under TITAN and PRISM [40]. We also report report-to-image retrieval recall@1 and recall@5 over the 150 report test-set.

4.3 Baselines and evaluation procedure

Tile generators. We compare BigCell $_{\text{tile}}$ against three prior histopathology tile generators: PixCell [49], CytoSyn [14], and MuPaD [45]. For each method, we use the publicly released model weights and apply the same sliding-window procedure described in Section 3.2 to generate 8192×8192 pixel images. The window size for each baseline matches its native resolution: 1024 for PixCell, 512 for MuPaD, and 224 for CytoSyn, with stride set to half the window size. We additionally compare against the large-image histopathology methods LRDM [17] and ZoomLDM [50], using their publicly released weights and their own large-image sampling procedures. We use 100 8192×8192 pixel ROIs sampled from the held-out test slides for evaluation and measure statistical significance via a two-sided paired Wilcoxon signed-rank test vs. the best baseline (not applied to Fréchet distance).

Efficient generation. We evaluate the adaptive window scheduling of Section 3.3 at 32k resolution under matched compute budgets measured in number of function evaluations (NFEs). Evaluation is performed on 256 randomly sampled slides from the held-out test set, with one 32k pixel region generated per slide. The full baseline runs all windows at every step for 20 denoising steps. We compare against four operating points of our adaptive schedule with $r \in \{0.05, 0.09, 0.25, 0.41\}$. We compare against a **fewer-steps baseline**: the full sliding-window pipeline (all windows denoised at every step) but with the number of denoising (Euler) steps reduced to match the same NFE budget as our adaptive schedule.

Downstream classification. For each dataset, we sample k real training examples per class ($k \in \{1, 2, 4\}$), and generate synthetic variations using the CGM-based augmentation procedure described in Section 3.5. We compare the baseline trained on only the k real images per class, and an augmented version trained on k real + k synthetic per class. For all variants, we use TITAN's vision encoder as the frozen feature extractor for both the real and synthetic samples. We then train a logistic regression classifier on top, and report the mean over ten random seeds of classifier training. Significance is assessed via a one-sided Wilcoxon signed-rank test across seeds on weighted kappa for PANDA and balanced accuracy for TCGA-UT-8K.

Table 1: Comparison at 8192×8192 resolution. V2: Virchow-2; H1: H-Optimus-1; FD: Fréchet distance (no test). *: $p < 0.001$, two-sided paired Wilcoxon vs. best baseline ($N = 100$). Resolution- and data-matched comparisons are in Appendix E.2.

	Slide-level Similarity		Patch-level Similarity				Fréchet Distance	
Generator	TITAN↑	MOOZY↑	$V2_{mean}$↑	$V2_{5th}$↑	$H1_{mean}$↑	$H1_{5th}$↑	$V2_{FD}$↓	$H1_{FD}$↓
PixCell	.941	.900	.778	.632	.667	.509	1023	696
CytoSyn	.929	.864	.753	.502	.640	.457	895	604
MuPaD	.646	.755	.462	.258	.342	.196	981	641
LRDM	.531	.693	.513	.334	.395	.247	**731**	**502**
ZoomLDM	.708	.801	.646	.464	.477	.325	815	553
BigCell	**.956***	**.973***	**.869***	**.737***	**.735***	**.586***	907	621

Pathologist evaluation. Three board-certified pathologists each completed 100 trials. In each trial, a query 8k image was shown alongside four candidate pathology reports; the pathologist selected the report believed to describe the query (multi-select with an *unsure* option), and classified the query as real or synthetic. Of the 100 trials, 50 were real images and 50 were BigCell generations. The four candidate reports consisted of the true report and three randomly sampled reports from the test set.

5 Results

5.1 Comparison against prior methods

In Table 1, we compare BigCell $_{\text{tile}}$ against three prior tile generators – PixCell-1024 [49], CytoSyn-224 [14], and MuPaD-512 [45] – plugging each into the shared sliding-window pipeline to produce 8k images, as well as against the large-image histopathology methods LRDM [17] and ZoomLDM [50]. BigCell $_{\text{tile}}$ achieves the highest similarity to reference images under both slide-level encoders. Slide-level encoders aggregate information across the entire 8k image, and these gains indicate that BigCell $_{\text{tile}}$ produces outputs whose global tissue arrangement more faithfully matches the conditioning grid. Note that prior tile generators were trained at different native resolutions; our ablation in Appendix E.1 shows that higher native resolution leads to better fidelity at 8k, so these results reflect the combined effect of architecture and training resolution.

BigCell $_{\text{tile}}$ also leads on patch-level cosine similarity under both Virchow-2 and H-optimus-1, in both mean and bottom-5th-percentile statistics. The gap is particularly large on the tail metrics: $V2_{5th}$ is 0.737 for BigCell $_{\text{tile}}$ vs 0.632 for PixCell, 0.502 for CytoSyn, indicating substantially fewer regions of degraded local fidelity. LRDM and ZoomLDM attain the best Fréchet distances while obtaining the lowest agreement with the reference images; Fréchet distance is computed over unpaired 256×256 crops and does not measure agreement with the conditioning.

Table 2 evaluates the same generations with report-level agreement using MedGemma-4B, which does not rely on pathology foundation model embeddings. BigCell again leads on all three fields, agreeing with the reference on the organ in 69% of cases and on the diagnosis in 46%, compared to 63% and 42% for PixCell. As all methods are evaluated on identical ROIs, these rates are meaningful relative to one another rather than in absolute terms. Qualitative comparisons across the tile generators are shown in Appendix Figure 9.

5.2 Efficient gigapixel generation

We evaluate the adaptive window scheduling strategy of Section 3.3 at 32k resolution, and compare it against the fewer-steps baseline at matched compute budgets. Figure 3 reports TITAN similarity and Fréchet distance with Virchow-2 and H-Optimus-1 as the encoders. Across all evaluated compute budgets, our adaptive schedule strictly improves over the fewer-steps baseline, with the gap being more pronounced at low compute. At NFE $\approx$ 4000, importance scheduling achieves a TITAN similarity of 0.76 while the fewer-steps baseline gets 0.63. At NFE $\approx$ 11000, our method yields a TITAN similarity of 0.951, matching the 0.957 similarity obtained at full compute (NFE = 38440).

Table 2: Report-level agreement at 8192×8192 resolution ($N = 100$, same ROIs as Table 1): fraction of generated/reference pairs whose MedGemma-4B reports agree on the tissue of origin, the diagnosis, and all three fields (organ, malignancy, diagnosis).

Generator	Same organ↑	Same diagnosis↑	All three fields↑
PixCell	63%	42%	38%
CytoSyn	54%	26%	25%
MuPaD	41%	17%	17%
ZoomLDM	61%	40%	36%
LRDM	59%	31%	30%
BigCell	**69%**	**46%**	**41%**

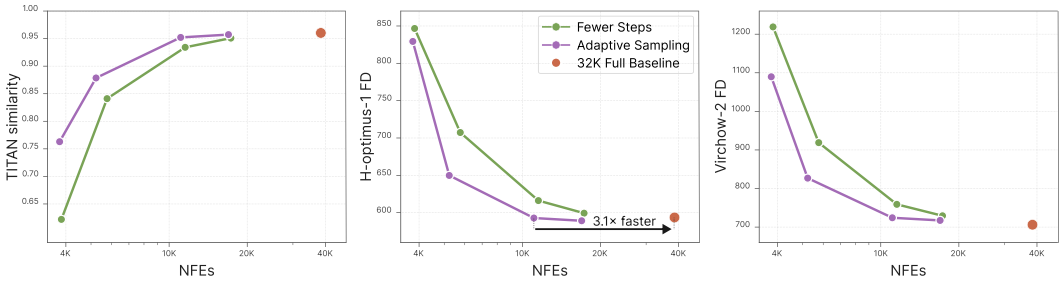

Figure 3: Adaptive window scheduling compared with simple, fewer-step sampling and the full 38440 NFE baseline at 32k resolution (N=256), evaluated at $r \in \{0.05, 0.09, 0.25, 0.41\}$. At $r = 0.25$, we obtain a 3.1×speedup (15.6 h → 4.97 h on a single Intel Data Center GPU Max 1550).

This leads to a 3.1× speedup with negligible quality degradation, which, in wall-clock terms, reduces 32k generation from 15.6 hours to 4.97 hours on a single Intel Data Center GPU Max 1550.

5.3 Text-conditioned generation

Table 3 shows the text-image alignment for 8k images generated by the CGM + BigCell $_{\text{large}}$ for 150 held-out reports. We compare against the upper limit of real images paired with their corresponding reports. Generated images achieve text-image similarity that approaches the real paired data under both TITAN and PRISM encoders (TITAN: 0.487 vs 0.551; PRISM: 0.454 vs 0.499). As a sanity check, we also compute the similarity of randomly paired images and reports, which is significantly lower than what our model generated (TITAN: 0.118; PRISM: 0.203). This indicates that synthetic images faithfully reflect the content of the conditioning report. Retrieval recall further strengthens this conclusion: recall@5 under TITAN matches the real paired baseline (0.655), and recall@1 reaches 0.169 vs 0.211 for real data.

Table 3: TITAN and PRISM CLIP-style similarity and report-to-image retrieval recall on 150 held-out reports. CGM-generated images approach the alignment of real data under both encoders.

	TITAN			PRISM		
Method	Sim	Recall@1	Recall@5	Sim	Recall@1	Recall@5
Real data (paired)	0.551	0.211	0.655	0.499	0.148	0.430
BigCell	0.487	0.169	0.655	0.454	0.106	0.387

Pathologist evaluation. In Figure 4, we present the results of the pathologist evaluation. Pathologists match synthetic queries to their source reports as accurately as real queries (0.57), indicating that synthetic images convey the same diagnostic content as real ones. The matching accuracy is upper-bounded by the task design: each report describes a patient but covers multiple distinct 8k regions, so

Figure 4: Pathologist evaluation. Report-matching accuracy is comparable for real and synthetic queries.

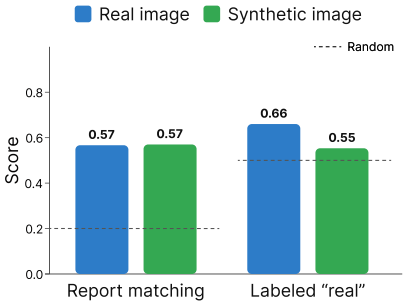

Table 4: Augmenting the real training set with synthetic samples from BigCell improves classifier performance over baselines trained on real data alone. * denotes $p < 0.05$.

		TCGA-UT-8k		PANDA	
Shots	Split	Bal-acc	AUROC	κ	Bal-acc
K=1	Real	.696	.970	.405	.304
	+Syn	**.751***	**.979**	**.442***	**.315**
K=2	Real	.761	.982	.480	.336
	+Syn	**.785***	**.984**	**.531***	**.358**
K=4	Real	.801	.987	.493	.375
	+Syn	**.804**	.987	**.583***	**.402**

a distractor with a similar diagnosis can fit the query as well as the true report. On the real-vs-synthetic task, pathologists label synthetic queries as real 55% of the time and real queries as real 66% of the time. Overall detection accuracy is 55.3% (95% CI 50–61%), and a permutation test that shuffles the real/synthetic labels across trials shows that this gap is not significant. Inter-rater agreement is at chance (Cohen's $\kappa = 0.020$), indicating that the three pathologists in our study did not reliably distinguish BigCell outputs from real tissue.

Synthetic data augmentation. Table 4 shows few-shot classification performance on TCGA-UT-8K and PANDA datasets with and without CGM-based augmentation. Note that PANDA is held out from CGM training, providing an out-of-distribution evaluation. Augmentation improves performance across both datasets and all shot counts, with relative gains of up to 18% in PANDA and 7% in TCGA-UT-8K. Augmenting directly at the scale at which the classifier operates is key to improving prediction accuracy. With a tile-level model that independently augments tiles in a slide, we are not guaranteed to maintain the large-scale structures that may be critical in making the correct prediction.

6 Limitations

We note three key limitations. First, text-conditioned generation is restricted to 8192×8192 pixel resolution (see Sec. 4.1). Scaling text-to-image synthesis to gigapixel resolution would require substantially larger paired datasets than currently publicly available. Second, gigapixel generation remains expensive. Our adaptive scheduling reduces 32k generation from 15.6 hours to 4.97 hours on a single GPU, but the absolute runtime is still impractical for high-throughput applications. Third, synthetic histopathology data carries a risk of misuse; generated images are intended for research only and should not substitute for real tissue in clinical or diagnostic settings.

7 Conclusion

We presented BigCell, the first approach to generate coherent whole-slide images at native gigapixel resolution. BigCell produces images of up to 32,768×32,768 pixels, enabled by our adaptive scheduling strategy that speeds up generation by over 3×. Generated images outperform prior methods in both tile-level and slide-level fidelity metrics; we further guide it with text using the conditioning generation model that produces dense grids of pathology foundation model embeddings directly from reports. BigCell demonstrates the first text-to-image generation at 8192×8192 resolution, with text-generated samples improving few-shot classification performance on two downstream datasets and conveying the same diagnostic content as real tissue when verified by board-certified pathologists.

Our work opens promising directions for future research: generating a WSI from its report would allow imputation of slides for historical archives that predate digitization and for regions without scanning infrastructure. Beyond text, our method can support other modalities, such as spatial transcriptomics or multiplex imaging, enabling multi-modal generation. Providing the means to multi-modal generative models in digital pathology at the WSI-scale is paramount to building digital twin systems with the aim of accurately simulating medical scenarios and intervention outcomes.

Acknowledgments and Disclosure of Funding

This research was partially supported by National Science Foundation (NSF) grants IIS-2123920, IIS-2212046, National Institutes of Health (NIH) grant 1R21CA258493-01A1, as well as Stony Brook Profund 2022 seed funding and private support from Bob Beals and Betsy Barton. This research used resources of the Argonne Leadership Computing Facility, a U.S. Department of Energy (DOE) Office of Science user facility at Argonne National Laboratory, and is based on research supported by the U.S. DOE Office of Science-Advanced Scientific Computing Research Program, under Contract No. DE-AC02-06CH11357.

References

[1] M. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

[2] S. Alfasly, W. Uegami, M. E. HOQ, G. Alabtah, and H. Tizhoosh. Semantic and visual crop-guided diffusion models for heterogeneous tissue synthesis in histopathology. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[3] M. Aversa, G. Nobis, M. Hägele, K. Standvoss, M. Chirica, R. Murray-Smith, A. M. Alaa, L. Ruff, D. Ivanova, W. Samek, et al. Diffinfinite: Large mask-image synthesis via parallel random patch diffusion in histopathology. *Advances in Neural Information Processing Systems*, 36:78126–78141, 2023.

[4] O. Bar-Tal, L. Yariv, Y. Lipman, and T. Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. In *International Conference on Machine Learning*, pages 1737–1752. PMLR, 2023.

[5] V. Belagali, S. Kapse, P. Marza, S. Das, Z. Li, S. Boutaj, P. Pati, S. Yellapragada, T. N. Nandi, R. K. Madduri, et al. Ticon: A slide-level tile contextualizer for histopathology representation learning. *arXiv preprint arXiv:2512.21331*, 2025.

[6] V. Belagali, S. Yellapragada, A. Graikos, S. Kapse, Z. Li, T. N. Nandi, R. K. Madduri, P. Prasanna, J. Saltz, and D. Samaras. Gen-sis: Generative self-augmentation improves self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2886–2896, 2026.

[7] M. Bhosale, A. Wasi, Y. Zhai, Y. Tian, S. Border, N. Xi, P. Sarder, J. Yuan, D. Doermann, and X. Gong. Pathdiff: Histopathology image synthesis with unpaired text and mask conditions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22415–22424, 2025.

[8] Bioptimus. H-optimus-1. `https://huggingface.co/bioptimus/H-optimus-1`, 2025.

[9] W. Bulten, K. Kartasalo, P.-H. C. Chen, P. Ström, H. Pinckaers, K. Nagpal, Y. Cai, D. F. Steiner, H. Van Boven, R. Vink, et al. Artificial intelligence for diagnosis and gleason grading of prostate cancer: the panda challenge. *Nature medicine*, 28(1):154–163, 2022.

[10] J. Chen, C. Ge, E. Xie, Y. Wu, L. Yao, X. Ren, Z. Wang, P. Luo, H. Lu, and Z. Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pages 74–91. Springer, 2024.

[11] R. J. Chen, T. Ding, M. Y. Lu, D. F. Williamson, G. Jaume, A. H. Song, B. Chen, A. Zhang, D. Shao, M. Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 30(3):850–862, 2024.

[12] S. Chowdhury, J. J. Kennedy, R. G. Ivey, O. D. Murillo, N. Hosseini, X. Song, F. Petralia, A. Calinawan, S. R. Savage, A. B. Berry, et al. Proteogenomic analysis of chemo-refractory high-grade serous ovarian cancer. *Cell*, 186(16):3476–3498, 2023.

[13] T. Ding, S. J. Wagner, A. H. Song, R. J. Chen, M. Y. Lu, A. Zhang, A. J. Vaidya, G. Jaume, M. Shaban, A. Kim, et al. A multimodal whole-slide foundation model for pathology. *Nature medicine*, pages 1–13, 2025.

[14] T. Duboudin, X. Fontaine, E. Andrier, L. Guillou, A. Filiot, T. Baiocco-Rodrigues, A. Olivier, A. Romagnoni, J. Klein, and J.-B. Schiratti. Cytosyn: a foundation diffusion model for histopathology–tech report. *arXiv preprint arXiv:2603.18089*, 2026.

[15] N. J. Edwards, M. Oberti, R. R. Thangudu, S. Cai, P. B. McGarvey, S. Jacob, S. Madhavan, and K. A. Ketchum. The cptac data portal: a resource for cancer proteomics research. *Journal of proteome research*, 14(6):2707–2713, 2015.

[16] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[17] A. Graikos, S. Yellapragada, M.-Q. Le, S. Kapse, P. Prasanna, J. Saltz, and D. Samaras. Learned representation-guided diffusion models for large-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8532–8542, 2024.

[18] A. Graikos, N. Jojic, and D. Samaras. Fast constrained sampling in pre-trained diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[19] M. Han, Y. Liu, Y. Liu, Z. Chen, J. Tang, X. Wu, D. Yang, and L. Zhang. Beyond pixel simulation: Pathology image generation via diagnostic semantic tokens and prototype control. *arXiv preprint arXiv:2512.21058*, 2025.

[20] R. Harb, T. Pock, and H. Müller. Diffusion-based generation of histopathological whole slide images at a gigapixel scale. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5131–5140, 2024.

[21] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[22] M. Ilse, J. Tomczak, and M. Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018.

[23] J. Kefeli and N. Tatonetti. Tcga-reports: A machine-readable pathology report resource for benchmarking text-based ai models. *Patterns*, 5(3), 2024.

[24] Y. Kotp, V. Q.-H. Trinh, C. Pal, and M. S. Hosseini. Moozy: A patient-first foundation model for computational pathology. *arXiv preprint arXiv:2603.27048*, 2026.

[25] B. F. Labs. FLUX.2: Frontier Visual Intelligence. `https://bfl.ai/blog/flux-2`, 2025.

[26] B. F. Labs, S. Batifol, A. Blattmann, F. Boesel, S. Consul, C. Diagne, T. Dockhorn, J. English, Z. English, P. Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[27] Y. Lee, K. Kim, H. Kim, and M. Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. *Advances in Neural Information Processing Systems*, 36:50648–50660, 2023.

[28] X. Leng, J. Singh, Y. Hou, Z. Xing, S. Xie, and L. Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18262–18272. IEEE, 2025.

[29] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[30] J. Lonsdale, J. Thomas, M. Salvatore, R. Phillips, E. Lo, S. Shad, R. Hasz, G. Walters, F. Garcia, N. Young, et al. The genotype-tissue expression (gtex) project. *Nature genetics*, 45(6):580–585, 2013.

[31] M. Y. Lu, D. F. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6):555–570, 2021.

[32] M. Y. Lu, B. Chen, D. F. Williamson, R. J. Chen, I. Liang, T. Ding, G. Jaume, I. Odintsov, L. P. Le, G. Gerber, et al. A visual-language foundation model for computational pathology. *Nature Medicine*, 30(3):863–874, 2024.

[33] J. Ma, V. Jayaram, B. Curless, I. Kemelmacher-Shlizerman, and S. Seitz. Ultrazoom: Generating gigapixel images from regular photos. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–10, 2025.

[34] N. Ma, M. Goldstein, M. S. Albergo, N. M. Boffi, E. Vanden-Eijnden, and S. Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.

[35] D. Nechaev, A. Pchelnikov, and E. Ivanova. Histai: An open-source, large-scale whole slide image dataset for computational pathology, 2025.

[36] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[37] H. Qiu, S. Zhang, Y. Wei, R. Chu, H. Yuan, X. Wang, Y. Zhang, and Z. Liu. Freescale: Unleashing the resolution of diffusion models via tuning-free scale fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16893–16903, 2025.

[38] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[39] A. Sellergren, S. Kazemzadeh, T. Jaroensri, A. Kiraly, M. Traverse, T. Kohlberger, S. Xu, F. Jamil, C. Hughes, C. Lau, et al. Medgemma technical report. *arXiv preprint arXiv:2507.05201*, 2025.

[40] G. Shaikovski, A. Casson, K. Severson, E. Zimmermann, Y. K. Wang, J. D. Kunz, J. A. Retamero, G. Oakley, D. Klimstra, C. Kanan, et al. Prism: A multi-modal generative foundation model for slide-level histopathology. *arXiv preprint arXiv:2405.10254*, 2024.

[41] N. L. S. T. R. Team. Reduced lung-cancer mortality with low-dose computed tomographic screening. *New England Journal of Medicine*, 365(5):395–409, 2011.

[42] Q. Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

[43] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, and J. M. Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

[44] H. Wu, J. Xu, H. Le, and D. Samaras. Importance-based token merging for efficient image and video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4983–4995, 2025.

[45] J. Xiang, M. Li, S. Hou, Y. Chen, X. Luo, Y. Ji, X. Zhou, E. Adeli, A. Chaudhari, C. P. Langlotz, et al. A generative foundation model for multimodal histopathology. *arXiv preprint arXiv:2604.03635*, 2026.

[46] E. Xie, J. Chen, J. Chen, H. Cai, H. Tang, Y. Lin, Z. Zhang, M. Li, L. Zhu, Y. Lu, and S. Han. SANA: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *The Thirteenth International Conference on Learning Representations*, 2025.

[47] H. Xu, N. Usuyama, J. Bagga, S. Zhang, R. Rao, T. Naumann, C. Wong, Z. Gero, J. González, Y. Gu, et al. A whole-slide foundation model for digital pathology from real-world data. *Nature*, 630(8015):181–188, 2024.

[48] S. Yellapragada, A. Graikos, P. Prasanna, T. Kurc, J. Saltz, and D. Samaras. Pathldm: Text conditioned latent diffusion model for histopathology. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5182–5191, 2024.

[49] S. Yellapragada, A. Graikos, Z. Li, K. Triaridis, V. Belagali, T. N. Nandi, K. Bai, B. S. Knudsen, T. Kurc, R. R. Gupta, P. Prasanna, R. K. Madduri, J. Saltz, and D. Samaras. Pixcell: A generative foundation model for digital histopathology images, 2025.

[50] S. Yellapragada, A. Graikos, K. Triaridis, P. Prasanna, R. Gupta, J. Saltz, and D. Samaras. Zoomldm: Latent diffusion model for multi-scale image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23453–23463, 2025.

[51] A. Zhang, G. Jaume, A. Vaidya, T. Ding, and F. Mahmood. Accelerating data processing and benchmarking of ai models for pathology. *arXiv preprint arXiv:2502.06750*, 2025.

[52] J. Zhang, Q. Huang, J. Liu, X. Guo, and D. Huang. Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23464–23473, 2025.

[53] E. Zimmermann, E. Vorontsov, J. Viret, A. Casson, M. Zelechowski, G. Shaikovski, N. Tenenholtz, J. Hall, D. Klimstra, R. Yousfi, et al. Virchow2: Scaling self-supervised mixed magnification models in pathology. *arXiv preprint arXiv:2408.00738*, 2024.

[54] E. Zimmermann, J. Viret, M. Zelechowski, J. B. Hall, N. Tenenholtz, A. Casson, G. Shaikovski, E. Vorontsov, S. Liu, and K. A. Severson. Mixed magnification aggregation for generalizable region-level representations in computational pathology. *arXiv preprint arXiv:2602.22176*, 2026.

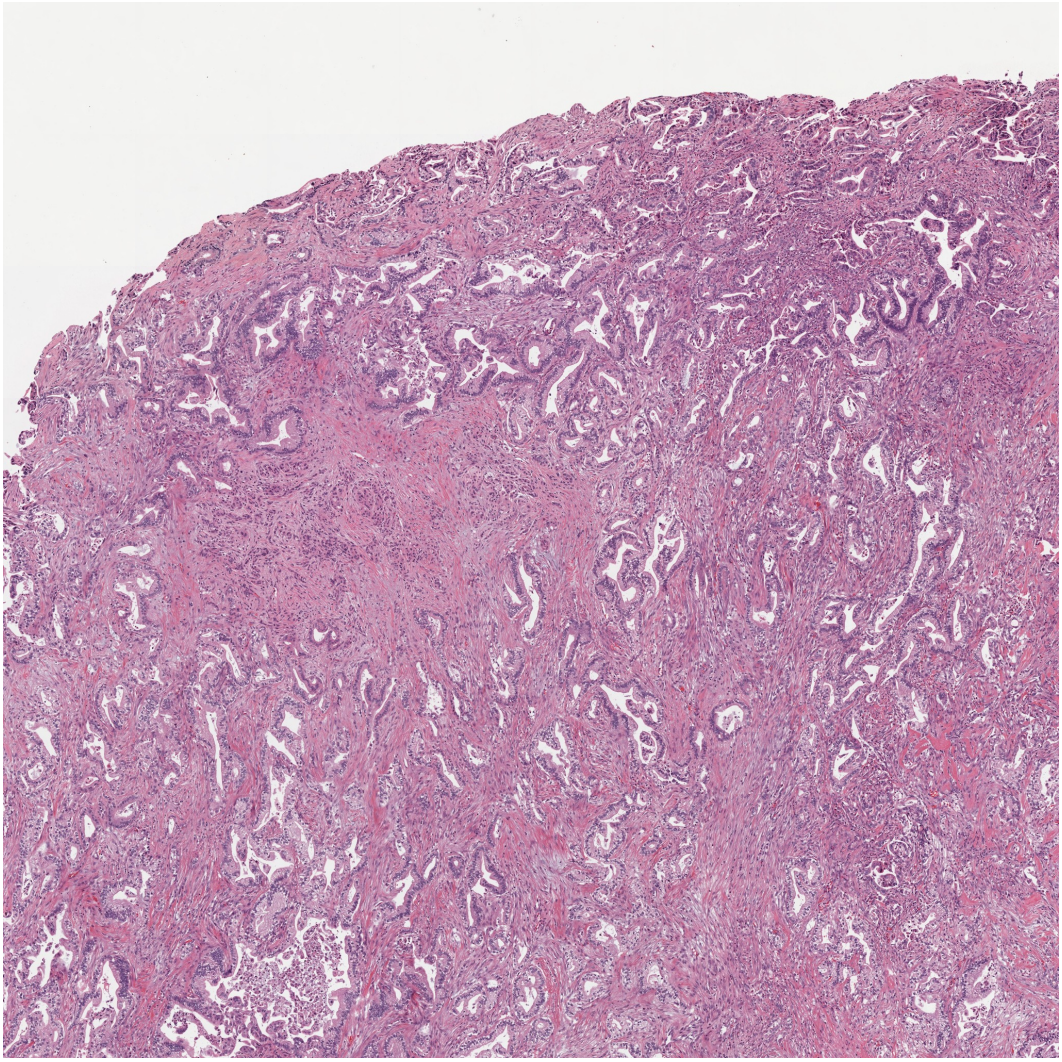

Figure 5: Synthetic 8192×8192 image generated by BigCell from a pathology text report, showing extensive pancreatic ductal adenocarcinoma.

A Motivation for whole-slide generation

Generating a coherent WSI requires modeling biological dependencies that span the entire slide. Existing generative methods in histopathology operate at the patch level, and any approach that composes such outputs into a WSI must explicitly coordinate them across the canvas. Without such coordination, independent patch generation produces incoherent or biologically implausible outputs even when each patch is locally realistic.

Diagnosis depends on the global spatial structure. Pathologists interpret local findings in relation to the surrounding tissue. Distinguishing invasive carcinoma from in situ disease, for example, requires continuity across the tumor-stroma interface and assessment of growth patterns over large regions. Patches remove this context and cannot support reliable diagnostic reasoning.

Pathology reports aggregate features at the slide level. Key elements of a clinical report (tumor grade, margin status, extent of invasion, spatial heterogeneity) are defined by patterns spanning large portions of the slide. These features cannot be inferred from independently generated patches because they depend on consistent relationships across distant regions.

Clinical trust requires global coherence. For a generated image to be useful in a clinical workflow, staining, morphology, and tissue organization must remain consistent across the entire slide. Even

subtle inconsistencies introduced by patch-based synthesis can undermine confidence and limit adoption.

B Clinical Applications

Generating high-resolution synthetic 8192×8192 images of regions of interest (ROIs) based on complex text descriptions would be highly valuable for a wide variety of clinical and translational research applications. This technology can be utilized to generate controllable examples of "ground truth" for algorithmic development, iterative refinement, and benchmarking quality control for computational pathology. Figure 5 is a synthetic example that shows extensive pancreatic ductal adenocarcinoma (PDAC) surrounding a small focal area of chronic pancreatitis with residual pancreatic acinar tissue and parenchymal fibrosis. BigCell can be used to readily generate multiple examples of pancreaticobiliary cancers invading the duodenum, peripancreatic adipose tissue, and regional lymph nodes that vary in terms of extent of invasion (cancer staging), grade/differentiation (growth pattern and glandular formation), and inclusion of rare variant cell morphology.

Considering the diversity of cancer as a disease and its uniqueness in each patient, this technology will be leveraged to visually examine common and rare morphologic findings side-by-side with synthetic hypothetical examples from massive real datasets to serve as a robust educational tool for residents, fellows, and other trainees. Having access to an inexhaustible library of histologic variants of cancer and other rare histological entities will be invaluable for the development of their professional microscopic examination and diagnostic skills, as well as support their preparation for board certification exams. Beyond the conventional PDAC shown in Figure 5, BigCell can be utilized to generate synthetic examples of adenosquamous (glandular and squamous components), signet ring cell (poorly cohesive cancer cells with cytoplasmic mucin), and undifferentiated (lacking gland formation +/- osteoclast-like giant multinucleated cells) variants of PDAC, which would be extremely helpful for improving diagnostic recognition and slide navigation to salient ROIs for these kinds of rare entities associated with more aggressive disease than conventional PDAC. Similarly, BigCell can generate diverse examples to study cancer evolution and potentially predict scenarios associated with varying disease progression and clinical outcomes.

C Extending sliding window generation to flow-matching

Here we discuss the extension of the sliding window algorithm of [4, 17] to flow-matching models, where the trained model is predicting velocities instead of noise.

At each generation step, we have a 'noisy canvas' $x_t \in \mathbb{R}^{H\times W\times D}$, which is larger than what the tile-level model is trained with. Our goal is to compute the velocity for the entire canvas v_t, by using only the local velocities $v_t^i = v_\theta(x_t^i, t, c^i)$ for all windows x_t^i.

We denote the window extraction operation as $P_i x_t = x_t^i$, and we assume that the windowed global velocity should be similar to the local velocities, i.e., $P_i v_t = v_t^i$.

Following MultiDiffusion [4] and LRDM [17], we pose the task of finding the global velocity as minimizing the least squares problem

$$v_t^* = \arg\min_{v_t} \sum_i \|W^i \odot (P^i v_t - v_t^i)\|_2^2, \tag{1}$$

where W^i is a windowing function that handles overlapping pixels (for example, a uniform or Gaussian kernel). Looking at an individual location p, which indexes both the global and local velocities, we can rewrite the problem as

$$v_t^*(p) = \arg\min_{v_t} \sum_i w^i(p)^2 (v_t(p) - v_t^i(p))^2, \tag{2}$$

which is straightforward to solve by setting

$$\frac{\partial v_t^*(p)}{\partial v_t(p)} = 2\sum_i w^i(p)^2 (v_t(p) - v^i(p)) = 0. \tag{3}$$

The solution

$$v_t^*(p) = \frac{\sum_i w^i(p)^2 v^i(p)}{\sum_i w^i(p)^2} \tag{4}$$

tells us that for the uniform weighting $w^i(p) = 1$ that we use in our experiments, the global velocity can be approximated by averaging all (overlapping) local velocities at every location p in the global canvas. This solution can be seen as the flow-matching extension of the result of [4], which was applied to a diffusion model.

Having computed the global velocity, we can continue sampling by just replacing its value in our (Euler) sampler as

$$x_{t+\Delta t} = x_t + \Delta t \, v_t^*. \tag{5}$$

D Implementation details

D.1 Tile generator

BigCell $_{\text{tile}}$ is an SiT-XL/2 [34] flow-matching transformer with 830M parameters. We condition the model with UNI-2 [11] features via cross-attention in every block and operate in the Flux-2 VAE [25] latent space, which compresses 2048×2048×3 images into 256×256×32.

Table 5 traces the tensor shapes through the pipeline. The generated object is the latent z; the final resolution is restored by a single decode through the Flux-2 VAE at a fixed 8× spatial factor. The conditioning grid y enters only through cross-attention and is not part of the decode path. For larger canvases, the same relationship holds by scaling both tensors: each sliding window denoises a $256 \times 256 \times 32$ latent region conditioned on the corresponding $8 \times 8 \times 1536$ slice of the global grid, the per-window velocities are aggregated into a global velocity field (Section 3.2, Appendix C), and the canvas is decoded once after the final denoising step.

Table 5: Tensor shapes in BigCell for a single 2048×2048 tile and a 32,768×32,768 canvas. At 32k, each sliding window operates on a $256 \times 256 \times 32$ latent region with the corresponding $8 \times 8 \times 1536$ slice of the conditioning grid.

Step	Tensor	2048×2048 tile	32,768×32,768 canvas
Input / output image	image x	$2048 \times 2048 \times 3$	$32768 \times 32768 \times 3$
VAE encode / decode	latent z	$256 \times 256 \times 32$	$4096 \times 4096 \times 32$
UNI-2 on 256×256 patches	conditioning grid y	$8 \times 8 \times 1536$	$128 \times 128 \times 1536$
Denoising (per window)	velocity $v_\theta(z_t, t, y)$	$256 \times 256 \times 32$	$256 \times 256 \times 32$

We train under the linear interpolant formulation of SiT with velocity prediction. Following PixCell [49], we progressively increase the training resolution from 256 to 2048 pixels. At each stage, we initialize the model from its previous stage weights. We also chose a shifted schedule for higher resolutions, as proposed by Stable Diffusion 3 [16]. For training at lower resolutions (256, 512, 1024), we extract image crops and smaller UNI sub-grids from the same 2048×2048 dataset. We summarize our design choices and configuration in Tables 6 and 7.

Table 6: Progressive training schedule for BigCell $_{\text{tile}}$. We train for one epoch at each resolution, and initialize the model from the previous stage weights.

Resolution	Batch size	Conditioning	lr	t schedule
256	4800	1×1536	2e-4	Uniform
512	2400	4×1536	1e-4	Uniform
1024	720	16×1536	1e-4	Shifted, $\alpha = 3$
2048	360	64×1536	1e-4	Shifted, $\alpha = 3$

D.2 CGM

The CGM is a DiT-DH-Base [28] flow-matching transformer operating directly in UNI-2 CLS token space, with 362M parameters. The model uses RoPE positional embeddings, RMSNorm, and SwiGLU activations. We condition the model with 128×768 unpooled TITAN text embeddings via cross-attention. We train under the linear interpolant formulation with velocity prediction and logit-normal timestep sampling, with a time-distribution shift of $\sqrt{(32 \times 32 \times 1536)/4096}$ to account for latent dimensionality. The CGM training configuration is summarized in Table 7.

Table 7: Training configuration for BigCell $_{\text{tile}}$ and CGM

Component	BigCell $_{\text{tile}}$	CGM
Optimizer	AdamW	AdamW
β_1, β_2	0.9, 0.999	0.9, 0.95
Weight decay	0	0
EMA decay	0.9999	0.9995
Gradient clipping	1.0	1.0
Warmup	5,000 steps (linear)	40 epochs (linear)
CFG dropout	0.1	0.1

D.3 Compute details

We use an Intel GPU cluster for training. Each node contains 6 Intel Data Center GPU Max 1550 and 2 Intel Xeon CPU Max 9470C processors. We use PyTorch with the Intel Extension for PyTorch (IPEX) library and perform distributed training using 12 ranks per node via MPI. In total, we consume 8,000 node-hours for this paper, including exploration and failed runs.

D.4 Data pipeline

We use the TRIDENT library [51] for tissue segmentation and patch extraction at 20$\times$magnification. UNI-2 grids, VAE features, and TITAN features were pre-computed and stored on disk. We did not apply stain normalization or data augmentation during BigCell $_{\text{tile}}$ training.

D.5 Sampling details

For the tile generator, we use the Euler ODE sampler with 20 sampling steps and a classifier-free guidance scale of 2.0. We apply the same configuration for all baseline tile generators (PixCell, CytoSyn, MuPaD) for fair comparison. For the CGM, we sample UNI-2 grids from text reports using the Euler ODE sampler with 50 sampling steps and a classifier-free guidance scale of 2.0.

Table 8: Training cost across native resolutions, measured on the Intel GPU cluster. Cost scales quadratically, making training at 4096 impractical. The 4096 row is a single-iteration benchmark; we did not train a model at this resolution.

Resolution	Batch size	Sec/Iter	Compute/Sample
1024	720	1.35	1.0$\times$
2048	360	5.59	8.3$\times$
4096	360	65	96$\times$

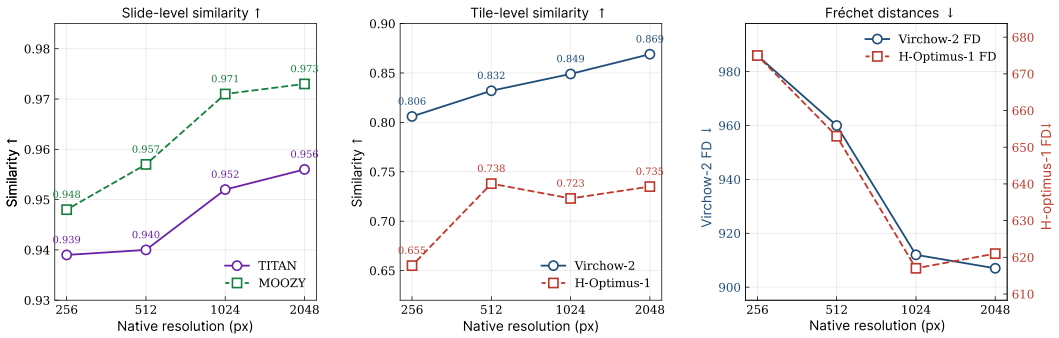

Figure 6: Effect of tile generator's native resolution on 8192×8192 generation quality. Both slide-level and tile-level metrics improve with resolution, with diminishing returns beyond 1024.

E Ablations

E.1 Effect of tile generator resolution

We train BigCell $_{\text{tile}}$ at four native resolutions as part of our progressive training pipeline (256, 512, 1024, 2048). Here we evaluate each model checkpoint at 8192×8192 generation using the sliding-window pipeline. As shown in Figure 6, both slide-level and tile-level metrics improve monotonically with native resolution. Most of the improvement occurs below 1024 px, with the 1024 → 2048 step yielding smaller gains.

Per-sample training cost scales by 8×from 1024 to 2048 (Table 8), reflecting the quadratic attention cost in latent sequence length. A single-iteration benchmark at 4096 pixels measures a further 12×increase, putting full 4096 training at ∼ 96 times our 1024×1024 budget, which is impractical at the dataset scale used in this paper. Combined with the diminishing fidelity gains beyond 1024, this leads us to adopt 2048 as the final operating point.

E.2 Resolution-matched comparisons

Table 1 compares tile generators trained at different native resolutions. Since fidelity increases with native resolution (Appendix E.1), that comparison conflates model design with training resolution. We disentangle the two with two additional comparisons at 8192×8192 on the same 100 held-out ROIs (Table 9).

Matched native resolution. Our progressive training strategy produces a 1024-native BigCell checkpoint (Appendix E.1), which we compare against PixCell-1024 [49]. At identical native resolution and identical inference, BigCell leads on every slide- and patch-level metric.

Matched native resolution, data, and compute. We train a 2048-native model that keeps our backbone (SiT-XL, Flux-2 VAE latent space), training data, and sampler fixed, and replaces our dense embedding conditioning with the summarizer-based conditioning of ZoomLDM [50], under the same training budget as BigCell. Following ZoomLDM, we train on fixed-size tiles at five magnifications (20× to 1.25×) and use its joint sampling algorithm to synthesize 8192×8192 images at 20×, guided by a 2048×2048 image at 5×. The summarizer-conditioned model (BigCell-summarizer) underperforms not only BigCell but also PixCell-1024 on every metric. We attribute this to ZoomLDM's pooling of the conditioning grid to a fixed 64×1536 budget: at 2048 native resolution, a 20× tile (a 2048×2048 region), a 5× tile (an 8192×8192 region), and a 1.25× tile (a 32,768×32,768 region) are all conditioned with the same 64×1536 grid. We believe this is the bottleneck when scaling ZoomLDM in resolution, and it warrants further dedicated investigation.

E.3 Importance-based window selection

We isolate the contribution of the importance signal by comparing our top-r CFG-based selection against a random-r baseline that selects the same fraction of windows uniformly at random. Both schedules use the same number of denoising steps and velocity caching; the only difference is which

Table 9: Resolution-matched comparisons at 8192×8192 resolution ($N = 100$, same ROIs as Table 1). Top: BigCell and PixCell at the same 1024 native resolution. Bottom: BigCell against a model with identical backbone, data, sampler, and training budget that uses ZoomLDM's summarizer-based conditioning.

	Slide-level Similarity		Patch-level Similarity		Fréchet Distance	
Generator	TITAN↑	MOOZY↑	$V2_{mean}$↑	$H1_{mean}$↑	$V2_{FD}$↓	$H1_{FD}$↓
Matched native resolution						
PixCell-1024	.941	.900	.778	.667	1023	696
BigCell-1024	**.952**	**.971**	**.849**	**.723**	**912**	**617**
Matched native resolution, data, and compute						
BigCell-summarizer-2048	.926	.868	.627	.505	1229	813
BigCell-2048	**.956**	**.973**	**.869**	**.735**	**907**	**621**

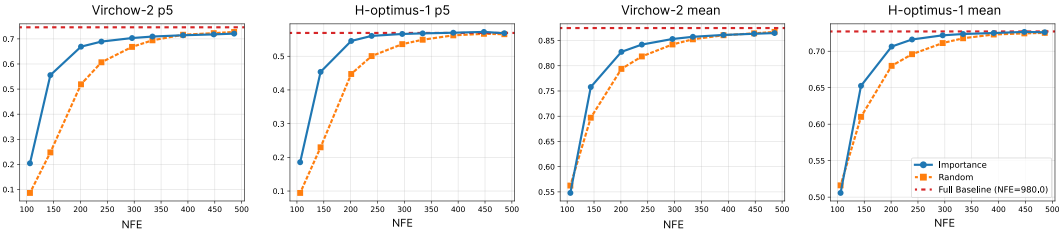

Figure 7: Effect of window selection strategy on 8192×8192 generation quality. Our CFG importance-based strategy outperforms random selection across all metrics.

windows get fresh vs. cached velocities. We sweep r to vary NFE and evaluate on 8192×8192 generation.

As shown in Figure 7, importance-based selection outperforms random selection across the full NFE range, with larger gaps on tail metrics at low compute. The gap closes as NFE grows and both schedules approach the full-compute ceiling. This confirms that the CFG magnitude signal identifies windows that genuinely benefit from fresh velocity computation, while random selection is more likely to waste compute on unimportant windows.

F Report-level agreement with MedGemma

To assess generated images without relying on pathology foundation model embeddings, we process each generated image and its reference with MedGemma-4B [39], a general-purpose medical vision-language model, and compare the decoded text. Patching follows the high-dimensional pathology notebook released with the model. Each image is presented in a single turn with a fixed template that the model must fill:

```
TISSUE OF ORIGIN: <one organ>
MALIGNANT: <yes or no>
DIAGNOSIS: <one most likely specific diagnosis>
```

We report the fraction of generated/reference pairs whose reports agree on the tissue of origin, on the diagnosis, and on all three fields. The 8192×8192 comparison across methods is shown in Table 2.

Gigapixel generation. We repeat this evaluation on the 256 matched real/synthetic 32,768×32,768 pairs used in Section 5.2, generated at full compute ($r = 1$). As a chance baseline, we compute the rate at which a real report matches the synthetic report of a different slide, averaged over all such pairings. Table 10 shows that agreement between BigCell generations and their reference slides far exceeds chance on all three criteria, indicating that our 32k generations carry the diagnostic content of the tissue they are conditioned on.

Table 10: Report-level agreement at 32,768×32,768 resolution ($N = 256$, full compute). Chance is the rate at which a real report matches the synthetic report of a different slide, averaged over all such pairings. 95% Wilson intervals in brackets.

Field agreement	BigCell vs. reference	Chance
Same tissue of origin	85.9% [81.1, 89.6]	9.1% [7.7, 10.7]
Same diagnosis	72.2% [66.4, 77.3]	9.2% [7.4, 11.1]
All three fields identical	71.0% [65.1, 76.2]	6.0% [4.6, 7.7]

G LLM usage

G.1 Report summarization for CGM training

Pathology reports in TCGA and HistAI are unstructured free-form text. To produce concise summaries suitable for conditioning the CGM, we prompt the Qwen-3-4B-Instruct [42] model to rewrite each report into a concise summary, capturing key diagnostic information. We generate five paraphrases per report to provide diversity during CGM training. The HistAI dataset [35] includes reports along with the images. For TCGA, we use cleaned reports from Kefeli and Tatonetti [23]. We use the following prompt template:

```
    You are a pathology lab assistant. You are given an unstructured pathology
report describing a tissue sample. Follow these instructions carefully and
to the letter:
- Extract a detailed summary of the diagnosis and the examined tissue from
  the report in a sentence under 50 words.
- Mention tissue type if it is mentioned in the report.
- If the report includes results of immunohistochemical studies or molecular
  tests, include a short summary of the most important positive results.
  Keep the overall sentence length below 50 words.
- Do not mention any negative test results or absence of something.
- Minimize prose, be succinct, use as few words as possible.
- Following the instructions above, rewrite the report 5 times, for a total
  of 5 sentences, each with a summary written in a slightly different way.
  Make sure all summaries are consistent with each other.
- Output the resulting 5 sentences as a list in JSON format. Don't output
  anything else.
```

G.2 Synthetic report generation for data augmentation

The CGM-based augmentation procedure of Section 3.5 requires a class-conditional text prompt for each label. For both TCGA-UT-8K (32 cancer types) and PANDA (6 ISUP grades), we use Claude Opus 4.7 to generate four short plausible pathology reports per class, following this prompt template:

```
    I want to perform a class-conditional augmentation task.
The task is to predict <label name> (<label range or class list>). For each
label, give me four plausible pathology reports. The report should be no
more than 25 words. Return the output in json.
```

H Broader impacts

BigCell enables high-quality generation of gigapixel histopathology images, which can be a powerful tool for research and education. As it is trained on publicly available data, the model reflects the biases present in those datasets. Generated images are intended for research use only and should not be used as a substitute for real tissue in clinical or diagnostic settings.

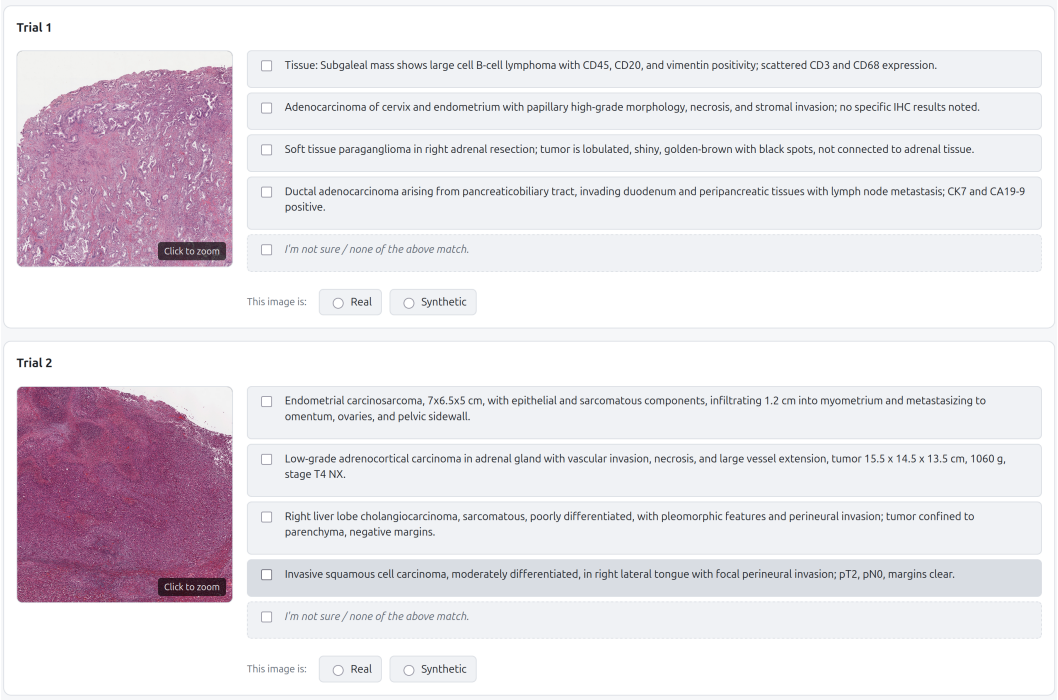

Figure 8: Pathologist evaluation interface. Each trial shows an 8192×8192 image (real or synthetic) and four candidate reports (one of which is true). Pathologists indicate matching report(s) and classify the image as real or synthetic.

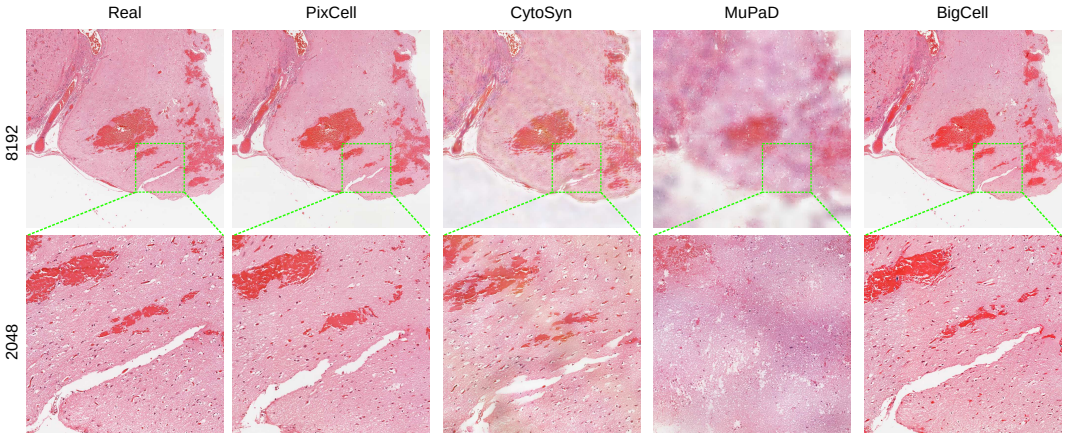

Figure 9: Qualitative comparison of 8192×8192 generation between BigCell $_{\text{tile}}$ and baselines. Top row: full 8k image. Bottom row: 2k crop from the boxed region.

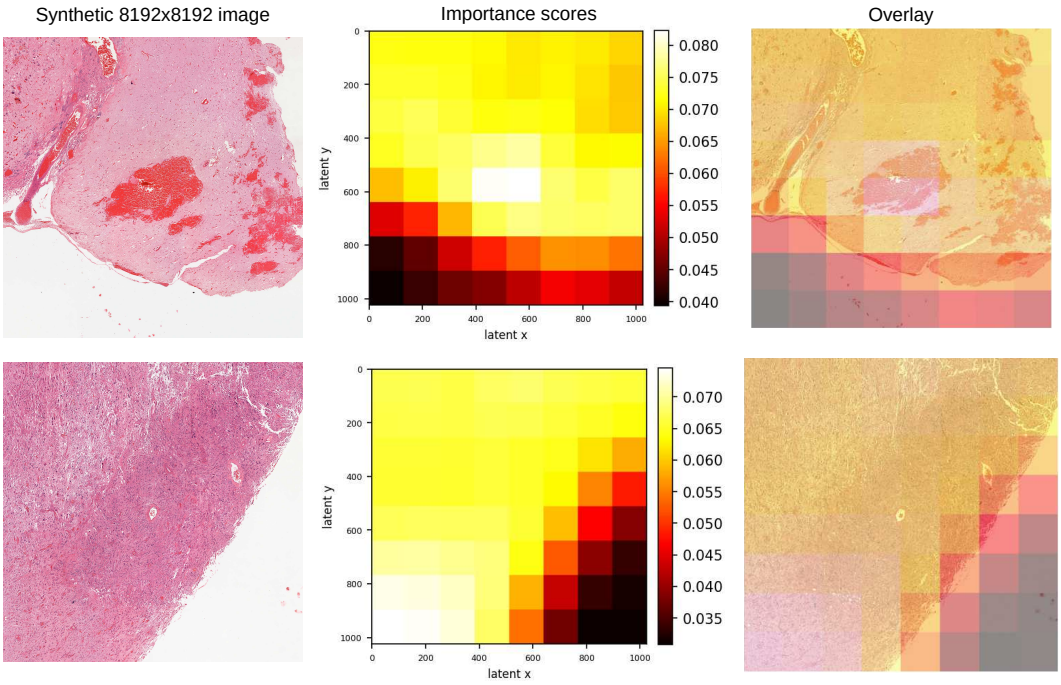

Figure 10: Visualization of CFG-magnitude importance scores on two synthetic 8192×8192 images. Tissue regions receive high importance, while background regions receive low importance.

I Additional qualitative results

Figure 9 shows 8192×8192 samples from BigCell $_{\text{tile}}$ and baseline tile generators, along with the reference real image. BigCell $_{\text{tile}}$ preserves both the global tissue structure and fine cellular details faithfully. PixCell preserves local details but generates structures that are inconsistent with the reference. CytoSyn captures the overall structure but exhibits blurriness, consistent with its lower similarity metrics. MuPaD struggles at this resolution – the global layout collapses, and tissue morphology is poorly preserved, in line with its substantially worse metrics.

In Figure 10, we visualize the importance assigned to different windows in an image during the efficient sliding-window generation. As expected, the model adaptively selects the regions with more details as the ones to perform more denoising steps on, while leaving background areas to be generated with fewer model forward passes. This speeds up generation significantly without impacting the overall quality of the gigapixel image produced.

In Figures 11, 12, 13, we show synthetic 32,768×32,768 pixel images generated by BigCell. Full-resolution pyramidal TIFFs of additional synthetic 32,768×32,768 samples are available at `https://figshare.com/s/f52821e239de442b7967` and can be viewed interactively using the QuPath[1] software.

[1] `https://qupath.github.io`

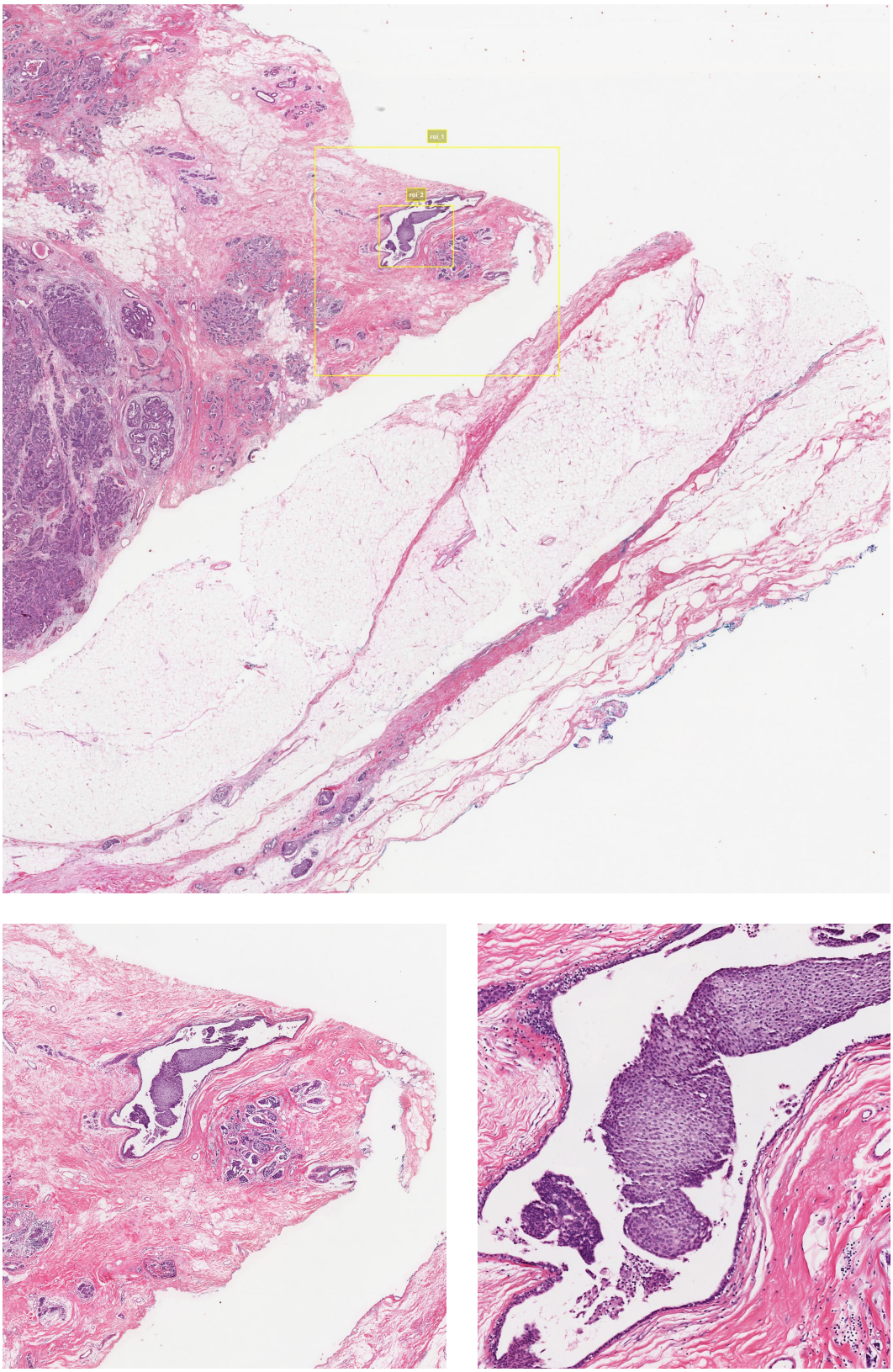

Figure 11: 32,768×32,768 pixel breast tissue image generated by BigCell (top), with progressive zoom-in crops from the marked regions (bottom).

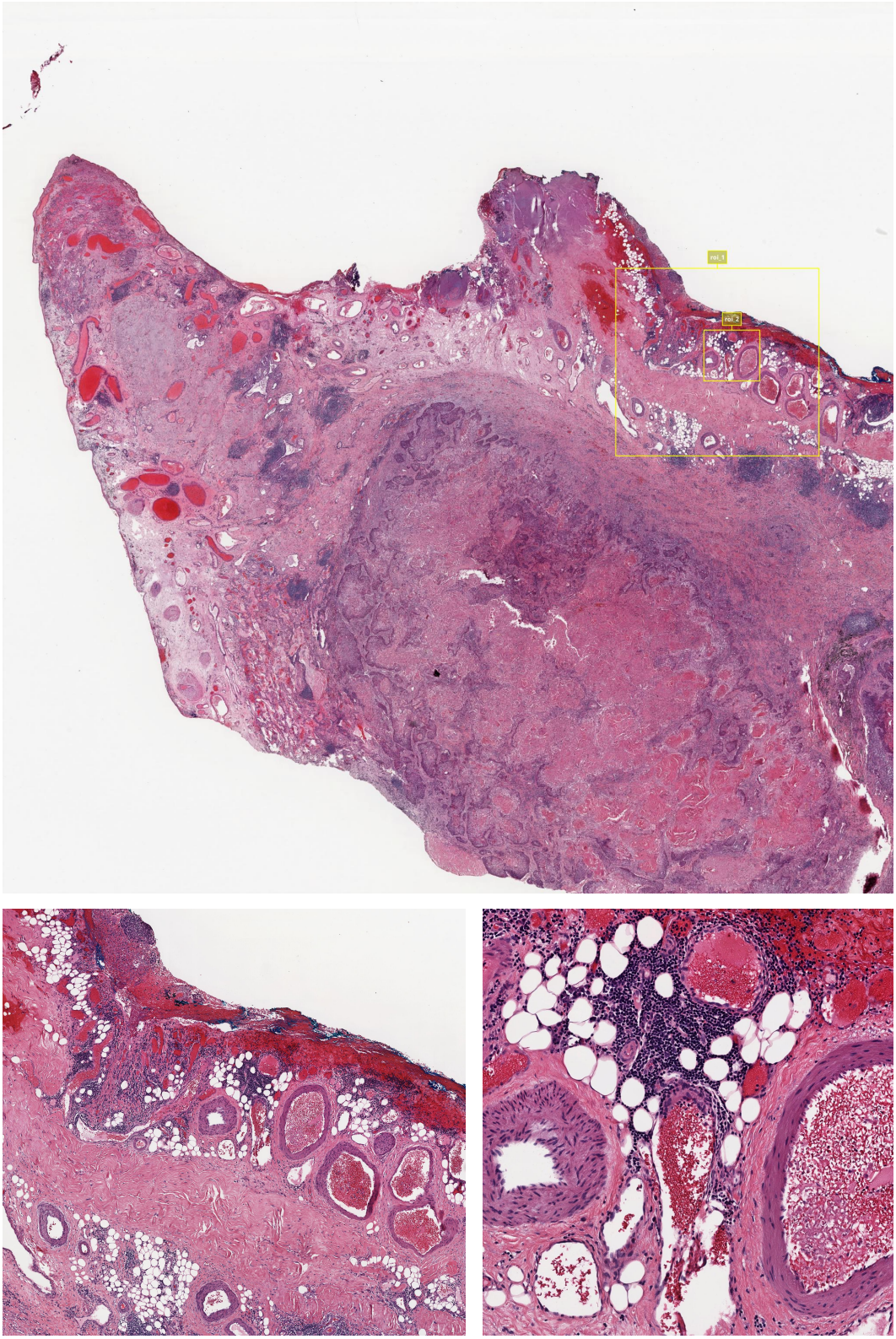

Figure 12: 32,768×32,768 pixel lung tissue image generated by BigCell (top), with progressive zoom-in crops from the marked regions (bottom).

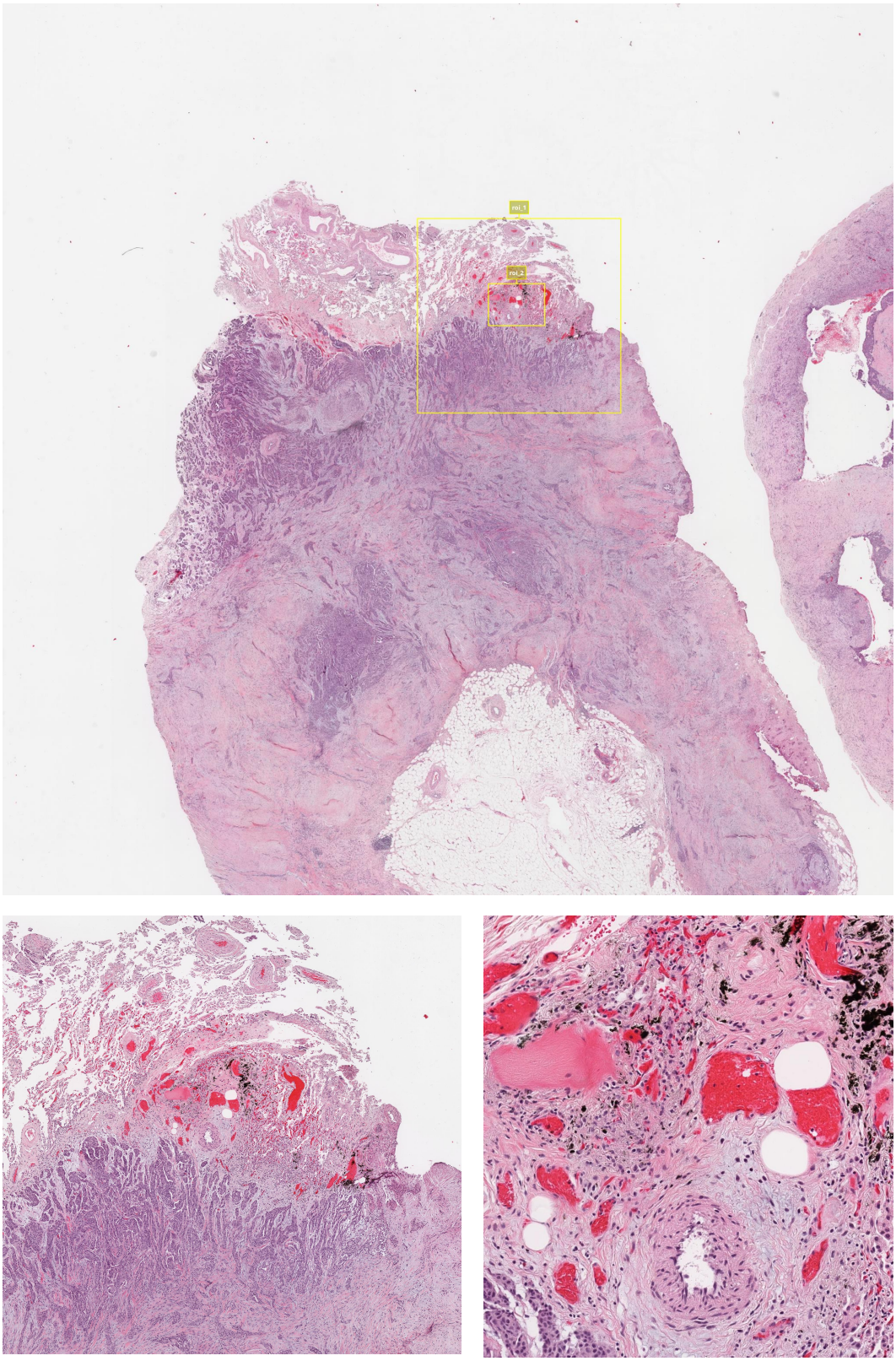

Figure 13: 32,768×32,768 pixel mesothelial tissue image generated by BigCell (top), with progressive zoom-in crops from the marked regions (bottom).

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 results shown in the paper reflect the contributions described in the abstract and introduction.

   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: We discuss the limitations of our work in Section 6.

   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: We do not include theoretical results in this paper.

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: We provide all the information needed to reproduce the main experimental results in Appendix Section D.

   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: The entire BigCell pipeline is trained exclusively on publicly available datasets with no proprietary data. We open source our code and pretrained models at this link.

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: Please refer to Section 4 and Appendix D for all the details of our experimental setting.

   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: We perform statistical significance tests for the data augmentation experiments and comparison against prior tile generators.

   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: Please refer to Appendix Section D.

   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: We follow the NeurIPS Code of Ethics.

   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: We discuss this in Appendix Section H.

    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: [Yes]

    Justification: Users of our released models and datasets will be required to adhere to usage guidelines and restrictions to access the model.

    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: We cite all the original assets used in our paper.

    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: We open-source our code and pre-trained weights at this link.

    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: [Yes]

    Justification: We ask expert pathologists to evaluate the synthetic images. We provide a screenshot of the evaluation platform in Figure 8.

    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: Evaluating pathologists were members of the research team.

    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: [Yes]

Justification: We use LLMs as part of our data pipeline, described in Appendix Section G.

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.