
GeneFlow: Translation of Single-cell Gene Expression to Histopathological Images via Rectified Flow

Mengbo Wang^{*1,4}

Shourya Verma^{*1}

Aditya Malusare^{2,4}

Luopin Wang^{1,4}

Yiyang Lu²

Vaneet Aggarwal^{1,2,4}

Mario Sola^{3,4}

Ananth Grama^{†1,4}

Nadia Atallah Lanman^{†3,4}

Purdue University

Abstract

Spatial transcriptomics technologies can be used to align transcriptomes with histopathological morphology, presenting exciting new opportunities for biomolecular discovery. Using spatial transcriptomic gene expression and corresponding histology data, we construct a novel framework, GeneFlow, to map single- and multi-cell gene expression onto paired cellular images. By combining an attention-based RNA encoder with a conditional UNet guided by rectified flow, we generate high-resolution images with different staining methods (e.g., H&E, DAPI) to highlight various cellular/ tissue structures. Rectified flow with high-order ODE solvers creates a continuous, bijective mapping between expression and image manifolds, addressing the many-to-one relationship inherent in this problem. Our method enables the generation of realistic cellular morphology features and spatially resolved intercellular interactions under genetic or chemical perturbations. This enables minimally invasive disease diagnosis by revealing dysregulated patterns in imaging phenotypes. Our rectified flow based method outperforms diffusion methods and baselines in all experiments. <https://github.com/wangmengbo/GeneFlow>.

1 Introduction

Spatial transcriptomics has revolutionized our understanding of gene expression within tissue architecture, providing unprecedented insights into biological processes and disease mechanisms [1, 2]. Combined with co-registered high-resolution histology images, spatial transcriptomes provide exciting opportunities for understanding the relationship between cellular transcriptomes and the corresponding image phenotypes. Existing computational approaches focus primarily on inferring gene expression from histological images [3, 4]. We address the largely unexplored inverse problem: Generating realistic histopathology images from transcriptomic data.

Spatial transcriptomics technologies such as Slide-seq, Stereo-seq, Visium and Xenium [5, 6, 7, 8] simultaneously capture morphological features through histological imaging (stained with H&E or DAPI) and transcriptomic profiles through spatially resolved gene expression measurements. This multi-modal approach reveals tissue architecture, cellular heterogeneity, and molecular mechanisms with greater depth than either modality alone, particularly in complex tissues where spatial organization impacts function. Previous machine learning applications in this field have focused on the

^{*}Equal Contribution (wang4887@purdue.edu, verma198@purdue.edu), [†] Corresponding Authors

¹ Computer Science, ² Industrial Engineering, ³ Comparative Pathobiology, ⁴ Institute For Cancer Research.

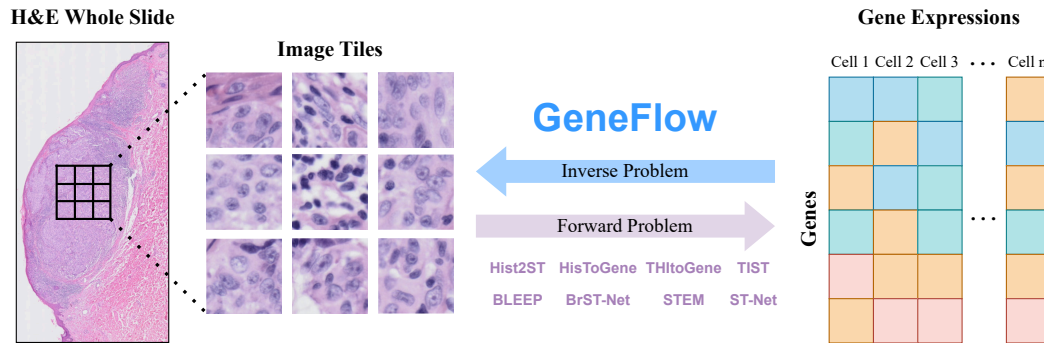


Figure 1: Mapping from histopathology images to gene expression (forward problem) and vice versa (inverse problem) through GeneFlow.

forward problem of predicting gene expression from histology images or integrating both modalities to enhance spatial clustering and gene expression analysis [9, 10, 11, 12, 13, 14, 15, 16]. However, the inverse problem, illustrated in Figure 1, remains largely unexplored. We introduce GeneFlow, the first method that generates histopathology images from single- and multi-cell gene expression transcriptomic profiles.

The ability to generate histopathology images from transcriptomic data has profound implications for cancer research and precision medicine. Cancer exhibits complex molecular alterations that manifest in diverse histological patterns, making it an important use case for our generative approach [17, 18, 19, 20]. Although histopathology remains the gold standard for cancer diagnosis, molecular profiling has become increasingly important for understanding cancer biology and guiding treatment decisions [21, 22, 23]. Studies using spatial transcriptomics data across diverse cancer types highlight the potential impact of integrating both data types. GeneFlow bridges these critical modalities with several potential applications, such as visualizing the histological manifestations of specific gene expression patterns, hypothesis generation, and biomarker discovery.

2 Related Work

Inferring Transcriptomes from Histology Images. The forward problem of predicting gene expression from histology images has been extensively studied. Wang et al. [9] provided a comprehensive benchmark for spatial transcriptome prediction, revealing variability across tissue types and platforms. Transformer-based models such as HisToGene [10] and THltoGene [11] capture long-range spatial dependencies in H&E images to infer gene expression patterns. TIST [13] introduced a self-supervised framework leveraging unlabeled histology data, while BLEEP [14] employed contrastive learning to align histological and transcriptomic features in a shared latent space, enabling bidirectional cross-modal queries. HIST2ST [15] used graph neural networks to model cellular interactions, and STEM [16] integrated multi-scale tissue representations to capture hierarchical biological organization. These methods collectively demonstrate the feasibility of inferring molecular profiles from morphology but address the forward problem, in contrast to our framework, which tackles the inverse mapping from transcriptomes to histology.

Mapping Transcriptomes to Histology Images. While predicting gene expression from histology has been widely explored, the inverse task of generating histopathology images from transcriptomic data remains largely unaddressed. To our knowledge, no prior work directly synthesizes realistic single- or multi-cell H&E or DAPI-stained images from spatial transcriptomics. Existing approaches only partially tackle this problem: RNA-GAN [24] generates histology-like tiles from bulk RNA-seq but lacks single-cell resolution and spatial structure modeling, while HistoXGAN [25] reconstructs cancer histology using multimodal embeddings that depend on pre-extracted histological features rather than gene expression. Consequently, the inverse mapping from transcriptomes to histology, particularly within spatial transcriptomics, remains an open challenge. Our framework, **GeneFlow**, is the first to directly address this task by employing rectified flow to learn the high-dimensional correspondence between spatial gene expression, cellular morphology, and tissue organization, establishing a foundation for bidirectional multi-modal integration in spatial biology.

Rectified Flow. Rectified flow is a generative modeling framework that constructs continuous bijective mappings between probability distributions via ordinary differential equations (ODEs). Proposed by Liu et al. [26], it extends normalizing flows [27] and continuous normalizing flows [28] by learning straight-line transport paths for efficient and stable generation. Unlike diffusion models [29] that rely on stochastic Markovian denoising, rectified flow deterministically transports probability mass between distributions. Connections to optimal transport [30] further explain its improved sample quality over diffusion models. In this work, we apply rectified flow to the transcriptomic domain, conditioning the flow on gene expression features through an attention mechanism [31] that modulates trajectory dynamics. High-order ODE solvers are employed for precise integration, capturing the nonlinear correspondence between transcriptomic profiles and histological structures.

3 Methods

We formulate the problem of generating histopathological images from gene expression data as follows. Given a single-cell resolution spatial transcriptomics dataset $\mathcal{D} = (X_i, I_i)_{i=1}^N$, where $X_i \in \mathbb{R}^{C_i \times G}$ represents the gene expression matrix for the i -th image tile with C_i cells and G genes, and $I_i \in \mathbb{R}^{H \times W \times K}$ denotes the corresponding histopathological image, our goal is to learn a mapping function $f_\theta : \mathbb{R}^{C \times G} \rightarrow \mathbb{R}^{H \times W \times K}$ that generates realistic histopathological images from gene expression profiles. Our rectified flow approach constructs a continuous bijective mapping between a simple prior distribution (Gaussian noise) and the target distribution of histopathological images conditioned on gene expression embedding and extra control embeddings such as the number of cells. The model learns a time-dependent vector field $v_\theta(x(t), t, X)$ that guides the transformation from random noise $x(0) \sim \mathcal{N}(0, I)$ to a realistic histopathological image $x(1) \approx I$ conditioned on gene expression matrix X . The input to our model consists of single-cell or multi-cell gene expression matrices, where each row corresponds to a gene and each column represents a cell. The output is a high-resolution (256×256 pixels) histopathological image with multiple channels, including H&E staining and optional auxiliary channels such as DAPI for visualizing nuclei.

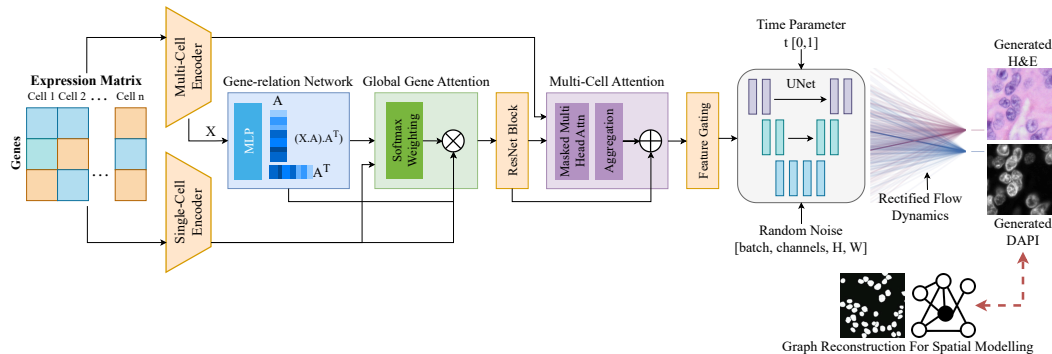


Figure 2: Architecture of the GeneFlow model for mapping transcriptomes to histology images.

3.1 GeneFlow Architecture

Our GeneFlow architecture, shown in Figure 2, establishes a framework to translate RNA expression data into histology images using rectified flow dynamics. We develop two distinct encoding pathways to handle different biological contexts: a single-cell encoder and a multi-cell encoder. The single-cell encoder generates image tiles with gene expression of one selected cell while the multi-cell encoder generates image tiles with gene expressions of many cells.

Gene Relation Network: Our encoder processes a batch of image patches, where each patch contains multiple cells, represented by a tensor $X \in \mathbb{R}^{B \times C_{\max} \times G}$, with B as batch size, $C_{\max}$ as maximum number of cells per patch, and G as the gene dimension. First, we flatten this tensor for processing $X_{\text{flat}} = \text{reshape}(X) \in \mathbb{R}^{BC_{\max} \times G}$, then the base network produces cell-specific embeddings that are used to predict parameters for a low-rank factorization of gene-gene relationships $E = f_{\text{base}}(X_{\text{flat}}) \in \mathbb{R}^{BC_{\max} \times 256}$. This factorization allows us to model complex interactions between genes while keeping the parameter count manageable $[U, V] = f_{\text{factors}}(E)$. Here, $U \in \mathbb{R}^{BC_{\max} \times G \times K}$

and $V \in \mathbb{R}^{BC_{\max} \times K \times G}$ are cell-specific factor matrices with rank K significantly smaller than the gene dimension G . The interaction expression values incorporate learned gene relationships, where α is a scaling factor (tuned during experiments to 0.1) that controls the influence of the learned relationships: $X_{\text{inter}}[i, j] = X[i, j] + \alpha \cdot ((X[i, j] \cdot U[i \cdot C_{\max} + j]) \cdot V[i \cdot C_{\max} + j])$.

Global Gene Attention: We apply global gene attention weights to focus on biologically relevant genes, and the weighted expressions are processed through a deep neural network with residual connections to produce cell embeddings with dimension D .

$$X_{\text{weight}}[i, j] = X_{\text{inter}}[i, j] \odot \text{softmax}(a) \quad (1)$$

$$H = \text{reshape}(f_{\text{cell_enc}}(X_{\text{weight, flat}})) \in \mathbb{R}^{B \times C_{\max} \times D} \quad (2)$$

Multi-Head Cell Attention: A critical challenge in tissue modeling is aggregating information from variable numbers of cells per patch. We implement a mask to handle this variable-length input $M[i, j] = 1$ if $j < \text{num_cells}[i]$, otherwise 0. Our multi-head attention mechanism learns to focus on relevant cells, allowing the model to identify cell populations that contribute most significantly to the tissue's visual characteristics:

$$A = f_{\text{cell_attn}}(H) \in \mathbb{R}^{B \times C_{\max} \times H_{\text{agg}}} \quad (3)$$

$$A_{\text{masked}} = A \cdot M - (1 - M) \cdot \infty \quad (4)$$

$$A_{\text{weights}} = \text{softmax}(A_{\text{masked}}^T) \in \mathbb{R}^{B \times H_{\text{agg}} \times C_{\max}} \quad (5)$$

Each attention head h produces a different weighting over cells, enabling the model to capture various aspects of cellular composition. The attention weights are applied to head-specific projections of cell embeddings:

$$P^{(h)} = f_{\text{proj}}^{(h)}(H) \in \mathbb{R}^{B \times C_{\max} \times D} \quad (6)$$

$$Z_{\text{agg}} = \frac{1}{H_{\text{agg}}} \sum_{h=1}^{H_{\text{agg}}} A_{\text{weights}}[:, h, :] \cdot P^{(h)} \in \mathbb{R}^{B \times D} \quad (7)$$

This design allows various heads to specialize in different aspects of cell behavior, such as identifying rare cell types or focusing on cells with distinctive expression patterns. The outputs from all heads are aggregated and processed through a final encoding layer with feature gating $Z_{\text{gate}} = f_{\text{final}}(Z_{\text{agg}})$, with $Z_{\text{final}} = Z_{\text{gate}} \odot \sigma(W_g \cdot Z_{\text{final}})$. This feature gating mechanism allows the model to selectively emphasize important features in the final representation. This helps control information flow and improves gradient propagation during training.

UNet Architecture: Both encoding pathways condition a shared UNet backbone [32] that implements the rectified flow dynamics. The UNet consists of a series of downsampling blocks, a middle block, and upsampling blocks with skip connections. The RNA embedding z (either z_{single} or z_{multi}) is combined with a time embedding $\gamma(t)$: $\gamma(t) = \text{Embed}(t) \in \mathbb{R}^{4d}$, where d is the base model channel dimension (128 in our implementation). Each residual block in the UNet incorporates these embeddings: $h_{\text{out}} = h_{\text{in}} + \text{Conv}(\text{SiLU}(\text{GroupNorm}(h_{\text{in}}) + \text{Linear}(\gamma(t) + \text{Linear}(z))))$. The UNet predicts the velocity field $v_{\theta}(x, t)$ that guides the generative process from random noise to fully-formed histological images. During training, the model learns to match the ground truth velocities derived from the rectified flow path: $\mathcal{L}(\theta) = \mathbb{E}_{x_1, t, \text{noise}} [|v_{\theta}(x(t), t) - v^*(x(t), t)|^2] + \lambda |W_1|_1$, where $\lambda = 0.001$ is a regularization parameter and W_1 represents the weights of the first layer in the encoder, encouraging sparsity in gene utilization.

The advantages of this method lie in low-rank factorization of gene relationships and the multi-head attention mechanism for cell aggregation. These encoding techniques capture meaningful interactions between gene expressions without requiring extra parameters, making the model more efficient and less prone to overfitting. It also provides explainability by revealing which cells contribute most to the tissue's visual characteristics. The encoder also handles variable numbers of cells per image patch, making it robust for real-world applications where cell density varies significantly across samples.

3.2 Generative Modeling With Rectified Flow

Our generative modeling with rectified flow defines a deterministic mapping between noise and data distributions via a continuous-time ODE: $\frac{dx(t)}{dt} = v_{\theta}(x(t), t)$, $t \in [0, 1]$, where v_{θ} is a learnable

vector field parameterized by neural network parameters θ . Unlike diffusion models with stochastic trajectories, rectified flow employs straight-line dynamics for efficient and stable generation. For each data point x_1 , we construct a sinusoidal interpolation path with small stochastic perturbation $\epsilon_t = (1 - t)\sigma z$, where $\sigma = 0.05$ and $z \sim \mathcal{N}(0, I)$:

$$x(t) = \sin(t\pi/2) x_1 + (1 - \sin(t\pi/2)) \text{noise} + \epsilon_t, \quad (8)$$

$$v^*(x(t), t) = \frac{dx(t)}{dt} = (x_1 - \text{noise}) \frac{\pi}{2} \cos(t\pi/2) - \frac{d\epsilon_t}{dt}, \quad (9)$$

$$\mathcal{L}(\theta) = \mathbb{E}_{x_1, t, \text{noise}} [\|v_\theta(x(t), t) - v^*(x(t), t)\|^2]. \quad (10)$$

Here, $t \sim \mathcal{U}[0, 1]$. A noise schedule $\sigma(t) = \sigma_{\min} + (\sigma_{\max} - \sigma_{\min})(1 - t)$, $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80.0$, controls noise magnitude along the trajectory. During inference, we solve the ODE using a fifth-order Runge–Kutta integrator [33] with adaptive step size, ensuring accurate, stable transformation of initial noise into high-resolution H&E or multi-channel (e.g., DAPI) histological images.

We trained our models for 100 epochs using the AdamW optimizer [34] with a batch size of 96. The learning rate followed a cosine annealing schedule [35] with a minimum learning rate set to 1% of the initial value, helping the model converge more smoothly during later stages of training. All experiments were conducted on a single NVIDIA H100 GPU, with training times ranging around 12 hours per experiment (on full sample) requiring up to 78 GB of VRAM.

4 Results

4.1 Datasets

To curate our training data with high resolution H&E stained image and real single-cell level resolved spatial transcriptomics data, we used three large publicly available spatial transcriptomics datasets prepared with 10x Genomics’ Xenium platform [36], all derived from Formalin-Fixed Paraffin-Embedded (FFPE) human melanoma samples. Among these samples, two were prepared using standard gene panels or with add-on custom gene targets, including around 300 genes. Another sample was from the Xenium Prime panel with 5000 targeting genes. For identification, we name these datasets Xenium_{C1}, Xenium_{C2}, and Xenium_{P1}. We collected 40X H&E stained images and aligned images with auxiliary staining such as DAPI and 18S (stain nucleus and cell boundaries respectively), which were used for cell segmentation by 10x Xenium Analyzer. Based on identified cell boundaries, we locate the cell at the center and extracted 256×256 pixels square-size image for consistency, with or without cell boundary mask. Only 126 genes are shared across the aforementioned samples, while the majority of genes remain substantially different due to various designs of Xenium panel. This results in gene expression data that are effectively collected from heterogeneous distributions, making the dataset a good fit to assess the generalizability of our models. We also extended experiments to 59 human Xenium samples from 12 organs in the HEST-1k dataset [37], totaling 1.6M paired patches.

Transcriptomics data were processed following standard protocols. Pre-identified Cells with unusually low or high gene counts were removed during quality check. Gene expression profiles were normalized and log-transformed to stabilize variance and reduce the influence of outliers. We further removed bottom 5% cells with the lowest total gene count to exclude low-quality or degraded cells. For single-cell modeling, we aligned and paired each individual cell’s gene expression data with its corresponding image tile, including auxiliary channels when available. To

simulate tissue-level heterogeneity and cell-cell interactions, we also created patch-level data using a sliding window approach with 256×256 pixel windows and 100-pixel overlap. Within each patch, we aggregated transcriptomic profiles from cells completely enclosed in the window to ensure accurate context matching. Cell types shown in Table 1 are identified by canonical cell type markers widely used by previous melanoma studies and further verified by differentially expressed gene and pathway analyses. To test model performance in the presence of class imbalance and potential catastrophic forgetting, we created subsets of the data, focusing solely on melanoma or non-melanoma cells,

Table 1: Dataset Cellularity

	Xenium _{C1}	Xenium _{C2}	Xenium _{P1}
Total patches	9394	39334	13832
Total cells	106980	70178	137927
B/Plasma cells	-	-	3435
Endothelial cells	4123	4182	5110
Epithelial cells	11105	4203	2425
Fibroblasts	12091	9694	11003
Macrophages	2739	12088	15728
Melanoma cells	70539	33309	47423
T cells	-	15272	12871

which served as the basis for targeted ablation experiments. Signatures and differentially expressed genes used for cell type annotation can be found in Appendix A.

4.2 Quantitative Image Analysis

We benchmarked the GeneFlow architecture against a baseline diffusion-based generative model, which uses the same structure for the single- and multi-cell gene encoder, but differs in its image generation dynamics. Both models were trained separately on each of the spatial transcriptomics datasets. For each dataset, we performed 3-fold cross-validation to ensure robustness and generalizability of our results across tissue variations. To assess the quality of the generated histological images, we used three widely accepted evaluation metrics, each capturing different aspects of visual fidelity and biological plausibility. Structural Similarity Index Measure (SSIM) evaluates how perceptually similar the generated images are to the ground truth by comparing structural elements such as texture, contrast, and brightness. Higher SSIM scores indicate greater visual and structural resemblance. Fréchet Inception Distance (FID) [38] measures how close the overall distribution of generated images is to that of real histology images. It does this by comparing statistical summaries; specifically, the means and covariances of image features extracted by a pretrained neural network. Lower FID scores indicate greater realism and similarity to the real image distribution. Feature Distance in Inception Space quantifies the average difference between individual generated and real image pairs by comparing their features in the latent space of the same pretrained network. Unlike FID, which offers a global view, this metric focuses on localized, sample-by-sample feature differences.

We evaluated our rectified flow method against a diffusion baseline across three datasets, using both single-cell and multi-cell models trained and tested on all cell types (Table 2). Rectified flow consistently outperforms the baseline across all metrics, delivering substantially better image quality with FID scores 3-6 times lower. Single-cell models generally perform better than multi-cell ones, with the Xenium_{C1} single-cell model achieving the best FID (20.73), suggesting stronger capture of intra-cellular gene expression morphology relationships. Notably, our model obtained comparable level of performance over all three evaluation metrics on multi-cell mode, which indicates our model's capability to learn tile level inter-cellular features and local tissue structures.

Table 2: Rectified Flow and Diffusion models trained and tested on all cell types

Sample	Model	Rectified Flow			Diffusion		
		↓ FID	↑ SSIM±	↓ FeatDist±	↓ FID	↑ SSIM±	↓ FeatDist±
Xenium _{P1}	multi	34.31±5.65	0.23±0.11	13.41±2.19	213.60±17.20	0.18±0.19	17.90±2.56
	single	27.43±5.91	0.17±0.03	14.53±2.15	132.09±57.05	0.20±0.07	17.11±2.22
Xenium _{C1}	multi	47.95±7.38	0.28±0.10	14.50±2.80	189.08±13.40	0.30±0.18	19.31±2.47
	single	20.73±8.45	0.24±0.04	14.90±2.55	171.06±81.98	0.22±0.07	18.53±2.81
Xenium _{C2}	multi	45.50±4.10	0.40±0.09	15.61±2.26	208.86±40.93	0.24±0.15	20.46±2.75
	single	42.61±4.50	0.35±0.06	15.65±2.28	119.22±40.21	0.36±0.12	18.24±2.47

Table 3: Rectified Flow model trained and tested on melanoma and non-melanoma cells

Sample	Model	Melanoma Cells			Non-Melanoma Cells		
		↓ FID±	↑ SSIM±	↓ FeatDist±	↓ FID	↑ SSIM±	↓ FeatDist±
Xenium _{P1}	multi	181.72±3.12	0.36±0.21	15.68±2.99	264.06±19.63	0.24±0.08	18.42±3.18
	single	14.18±1.86	0.17±0.02	13.95±2.01	47.01±14.81	0.18±0.04	14.82±2.23
Xenium _{C1}	multi	104.50±2.99	0.37±0.18	13.71±2.48	348.21±1.82	0.35±0.10	20.43±3.49
	single	22.86±1.98	0.22±0.03	14.12±2.48	96.28±51.64	0.21±0.03	16.25±2.43
Xenium _{C2}	multi	24.43±0.17	0.52±0.05	13.94±2.08	272.79±18.30	0.39±0.06	20.12±2.21
	single	46.30±14.44	0.37±0.05	15.74±2.31	65.24±6.77	0.40±0.05	15.97±2.20

We further compared generation performance between models trained on datasets containing all cell types versus those trained exclusively on either melanoma or non-melanoma cells (Table 3). Models trained on non-melanoma cells generally showed degraded performance, likely due to the heterogeneity of immune cell types, imbalance in cell abundance, and their non-uniform spatial distribution. Additionally, because non-melanoma cells often co-occur with melanoma cells in tumor regions, training data labeled as non-melanoma may still contain partial melanoma features, which can confuse the model's gene-to-image mapping. In contrast, models trained on more curated and homogeneous melanoma-only datasets performed comparably or better than models trained on all cell

types. This highlights the model’s ability to learn precise, cell type-specific morphological features from gene expression data when provided with sufficiently clean and targeted training samples.

To assess generalization, we train on one dataset and test on another (Table 4). Despite limited gene panel overlap (126 shared genes), rectified flow maintains strong cross-dataset performance. Models trained on Xenium_{C1} and tested on P1 yield the best results (FID: 67.00 single-cell, 79.86 multi-cell), with stable SSIM and feature distance scores, demonstrating robust transferability and ability to learn dataset-agnostic gene morphology mappings. Benchmarking details and auxiliary experiments can be found in Appendix B.1.

Train Sample	Test Sample	Model	↓ FID±	↑ SSIM±	↓ FeatDist±
Xenium _{P1}	Xenium _{C1}	multi	89.16±10.22	0.30±0.17	16.11±2.35
	Xenium _{C1}	single	90.88±2.01	0.24±0.05	16.13±2.04
	Xenium _{C2}	multi	127.79±13.28	0.38±0.11	17.27±2.33
	Xenium _{C2}	single	144.49±3.32	0.34±0.09	17.40±2.07
Xenium _{C1}	Xenium _{P1}	multi	79.86±11.45	0.30±0.18	15.80±2.33
	Xenium _{P1}	single	67.00±11.61	0.20±0.07	15.91±2.06
	Xenium _{C2}	multi	112.25±11.02	0.37±0.12	17.18±2.15
	Xenium _{C2}	single	98.78±12.09	0.36±0.09	17.20±2.09
Xenium _{C2}	Xenium _{P1}	multi	148.15±0.72	0.29±0.19	17.59±2.01
	Xenium _{P1}	single	164.35±7.36	0.19±0.07	18.02±1.93
	Xenium _{C1}	multi	146.78±29.10	0.34±0.17	18.36±2.36
	Xenium _{C1}	single	145.44±31.13	0.24±0.06	18.21±2.31

Table 4: Rectified Flow model cross-dataset performance evaluation

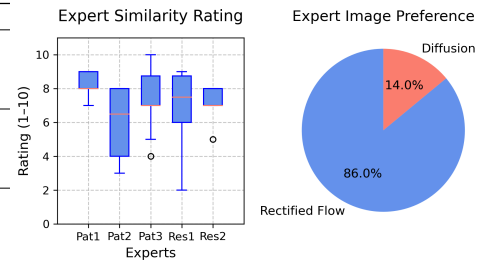


Figure 3: Evaluation by 3 ACVP board certified pathologists and 2 residents.

Single-cell models (Tables 2-3) outperform multi-cell models due to both methodological and biological factors. Unlike sliding-window multi-cell generation, single-cell patches are centered on valid cells with transcript overlap above a threshold and use spatial weighting that decreases toward patch margins. This emphasizes target-cell morphology while reducing peripheral effects. Patch sizes accommodate full cells and variability, yielding more homogeneous patches. Single-cell conditioning provides focused gene context, whereas multi-cell patches mix heterogeneous cell types, complicating gene-to-morphology mapping. In melanoma, the diverse tumor-immune-stromal microenvironment further degrades multi-cell performance in boundary resolution and texture consistency.

We evaluate performance using histopathology specific metrics derived from the UNI2-h foundational model [39] in Table 5. These include UNI2-h FID for pathology-specific image quality, UNI2-h embedding similarity for comparing feature distributions, and nuclear morphometric similarity quantifying circularity, eccentricity, and solidity of segmented nuclei. We further assess spatial energy similarity from gray-level co-occurrence matrices, as well as spatial complexity and feature magnitude from UNI2-h embeddings to capture tissue structure. These metrics provide finer insights than standard vision metrics, diagnosing limitations in boundary definition and texture fidelity. In table 6 we benchmark against the diffusion model along with an implementation of conditional UNet baseline with our gene encoder trained using MSE and perceptual losses. Ablations show the transformer-based RNA encoder consistently outperforms simpler encoders across all metrics. While dropping components causes modest performance declines, each contributes uniquely to interpretability, supporting identification of important genes and their relationships. The encoder’s complexity is justified, balancing strong performance with interpretable gene-morphology links.

Table 5: Domain-specific evaluation metrics for C1 model

Metric	C1 Multi Diff.	C1 Multi Rect.	C1 Single Diff.	C1 Single Rect.
<i>Image Quality Metrics ↓</i>				
FID Overall UNI2-h	404.06	87.96	405.98	39.27
Inception Feat. Dist.	23.05±1.86	16.72±1.64	20.18±1.61	14.50±2.52
<i>Biological Feature Similarity ↑</i>				
UNI2-h Embedding Sim.	0.967±0.007	0.979±0.004	0.969±0.007	0.983±0.003
Nuclear Circularity Sim.	0.835±0.043	0.844±0.037	0.839±0.049	0.874±0.028
Nuclear Eccentricity Sim.	0.869±0.046	0.954±0.015	0.880±0.039	0.964±0.012
Nuclear Solidity Sim.	0.714±0.048	0.888±0.036	0.721±0.077	0.867±0.028
<i>Spatial Feature Metrics ↑</i>				
Spatial Energy Sim.	–	0.283±0.083	–	0.678±0.176
Spatial Complexity Sim.	0.167±0.138	0.506±0.080	0.130±0.111	0.571±0.085
Spatial Feat. Magnitude Sim.	0.167±0.143	0.514±0.077	0.135±0.119	0.571±0.084

To explicitly preserve the spatial organization of cells in generated histopathology images, we introduced a spatial graph loss that enforces consistency in local tissue architecture. We propose two complementary approaches, (1) a segmentation-based method that models nuclear morphology and spatial relationships and (2) a fast alternative gradient-based method that captures texture patterns through local image derivatives and neighborhood similarity. Both approaches construct kNN graphs in spatial coordinates and penalize discrepancies in local appearances between generated and ground

Table 6: C1 Single-Cell model baseline comparison and ablation study. Including larger HEST-1k dataset for Multi-Cell model

Metric	UNet (MSE)	Diffusion	Rectified	-Gene Att.	-Gene Att./Rel.	-Gene-/Multi Att.	HEST-1k
<i>Image Quality Metrics ↓</i>							
FID Overall UNI2-h	525.51	405.98	39.27	26.46	27.25	29.50	62.54
Inception Feat. Dist.	21.76±1.29	20.18±1.61	14.50±2.52	14.89±2.52	15.02±2.70	15.39±2.68	15.26±2.70
<i>Biological Feature Similarity ↑</i>							
UNI2-h Embedding Sim.	0.964±0.004	0.969±0.007	0.983±0.003	0.991±0.003	0.990±0.003	0.990±0.003	0.974±0.004
Nuclear Circularity Sim.	0.659±0.038	0.839±0.049	0.874±0.028	0.949±0.022	0.941±0.021	0.947±0.020	0.924±0.027
Nuclear Eccentricity Sim.	0.655±0.024	0.880±0.039	0.964±0.012	0.959±0.013	0.954±0.015	0.957±0.017	0.955±0.014
Nuclear Solidity Sim.	0.479±0.037	0.721±0.077	0.867±0.028	0.950±0.020	0.943±0.019	0.949±0.020	0.921±0.028
<i>Spatial Feature Metrics ↑</i>							
Spatial Energy Sim.	0.018±0.034	—	0.678±0.176	0.723±0.130	0.735±0.055	0.719±0.048	0.744±0.049
Spatial Complexity Sim.	0.099±0.060	0.130±0.111	0.571±0.085	0.749±0.069	0.704±0.072	0.721±0.067	0.641±0.061
Spatial Feat. Magnitude Sim.	0.112±0.061	0.135±0.119	0.571±0.084	0.759±0.068	0.718±0.068	0.735±0.069	0.635±0.059

truth images. The spatial loss is gradually introduced during training which regularize that generated images maintain biologically plausible cell arrangements and tissue microarchitecture, improving both visual fidelity and downstream biological interpretability. Details can be found in appendix E.

4.3 Gene Importance Analysis

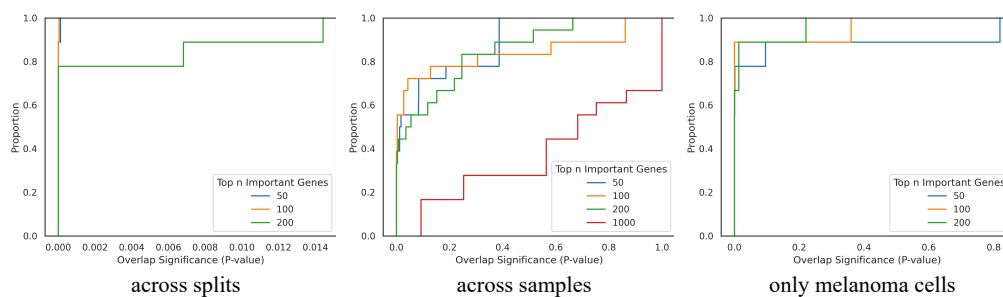


Figure 4: Overlap significance of influential genes

To interpret gene expression-phenotype relationships learned by our generative model, we performed a gradient-based sensitivity analysis to quantify each gene's influence on cellular morphology. Specifically, we computed the partial derivatives of the predicted velocity's squared L2-norm with respect to individual gene expression levels across multiple stochastic evaluations and generative timesteps. This allowed us to rank genes based on their impact on morphological outcomes, providing potential insight into how high-dimensional gene expression patterns shape complex visual phenotypes. These importance scores can guide hypotheses about gene function and regulatory mechanisms, and help prioritize candidates for downstream experimental validation.

We evaluated the consistency of the gene sets with the highest importance scores by counting the overlap of the top 50, 100, and 200 important genes across multiple model variants, data splits, and datasets with differing cellular composition and gene panels. The statistical significance of the overlapping genes was assessed using a hypergeometric test. As shown in Figure 4, we consistently observed statistically significant overlaps in influential genes across different splits, with 60 to 80% of comparisons showing substantial agreement, even though only 126 genes were shared across the three gene panels. This highlights our model's ability to generalize across sample-specific gene panels. When comparing models trained on all cell types versus melanoma-only cells (right panel), over 80% of comparisons showed significant gene overlap. This is consistent with melanoma cells being the dominant population in all datasets. In contrast, there was no significant overlap between influential genes from melanoma-only and non-melanoma-only models, reflecting distinct underlying gene-phenotype relationships in these cell populations.

To further validate biological relevance, we performed a gene set enrichment analysis [40] on the 34 shared genes out of top 50 influential genes between two samples with standard panels and custom add-on (Xenium_{C1} and Xenium_{C2}). The results (see Appendix C) revealed that EMT and extracellular

matrix organization pathways are significantly enriched. These pathways play critical roles in melanoma progression, with EMT-like phenotype switching contributing to metastatic potential and ECM remodeling facilitating invasion. Our model independently prioritized genes within these pathways without any explicit biological knowledge encoded in its architecture; demonstrating the model's ability to successfully capture the fundamental relationship between gene expression and cellular morphology visible in histopathological images.

4.4 Qualitative Image Analysis

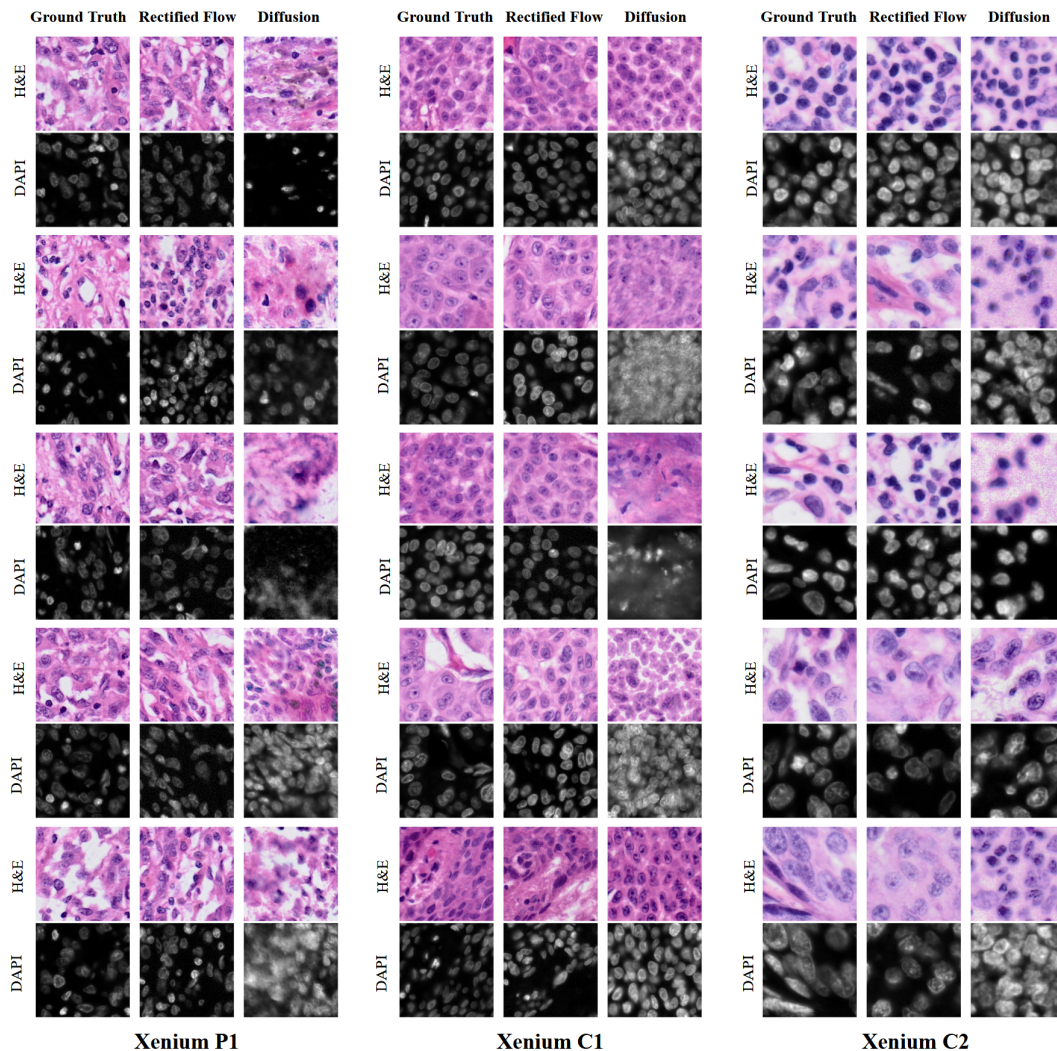


Figure 5: Comparison of ground-truth and generated images for Rectified Flow and Diffusion models

Figure 5 shows the ground-truth and generated images through the rectified flow and diffusion models from three different datasets. Rectified flow model clearly outperforms the diffusion model by producing more accurate H&E and DAPI images of cellular and tissue morphology given gene expression data. The generated cellular, nuclear and nucleoli morphologies are visually consistent with ground truth. Further generated image results from all experiments can be found in Appendix B.3. We conducted a human evaluation study with three ACVP board certified pathologists and two residents, using similarity and preference tasks. In the similarity task, pathologists were shown 20 pairs of rectified flow-generated and ground truth images from the test set, and asked to blindly rate their similarity on a scale of 1-10. In the preference task, they were shown 20 pairs of generated images one from the rectified flow model and one from the diffusion model and asked to blindly choose which they preferred for cell and tissue classification clarity. As shown in Figure 3, all

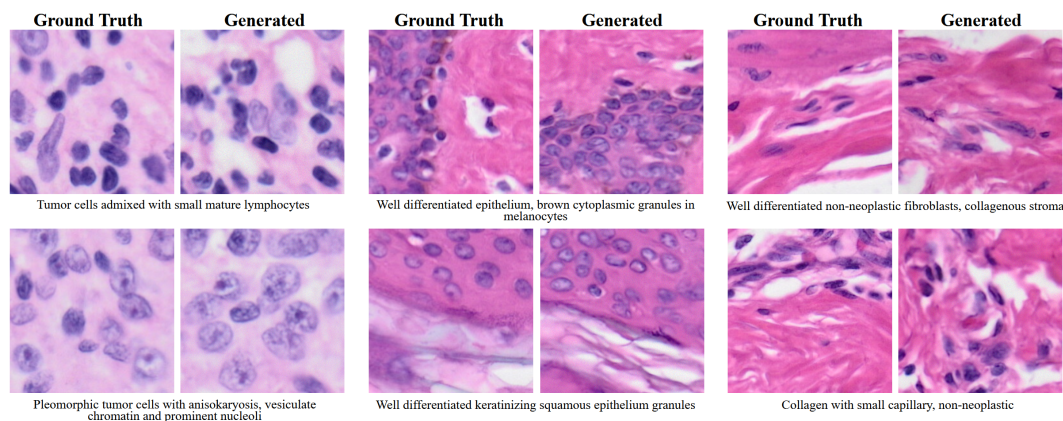


Figure 6: Ground-truth and generated H&E image with diagnosis by ACVP certified pathologists.

pathologists rated similarity above a median of 6, and preferred the image generated using the rectified flow model over diffusion in 86% of the cases.

To demonstrate a diagnostic application of GeneFlow, Figure 6 presents representative examples of generated histology patches alongside their corresponding ground-truth images from both tumor (melanoma) and healthy skin tissues. ACVP board certified pathologist were asked to provide diagnoses based solely on the generated cellular and microenvironmental morphology. Neoplastic regions characterized by clonal cellular proliferation include both benign and malignant tumor cells, while non-neoplastic regions encompass normal epithelium, inflammatory infiltrates, fibroblastic stroma, and reactive tissue changes. The generated images faithfully reproduce key diagnostic features such as pleomorphic nuclei, keratinizing squamous epithelium, and collagenous stroma, enabling pathologist to reach consistent interpretations with high confidence relative to ground truth. These results highlight GeneFlow's potential to synthesize diagnostically coherent tissue morphologies from transcriptomic data, supporting minimally invasive, transcriptome-guided pathology.

5 Discussion

GeneFlow bridges the gap between transcriptomics and cellular morphology by introducing a novel framework that translates gene expression profiles into realistic histopathological images. Leveraging rectified flow dynamics, our method consistently outperforms diffusion-based alternatives, as demonstrated by both quantitative benchmarks and pathologist assessments. By effectively modeling the complex relationship between gene expression and histological features, GeneFlow enables new opportunities for minimally invasive diagnostics and virtual tissue reconstruction. The observed performance gap between single-cell and multi-cell models also offers valuable insights into model scalability. While the single-cell variant currently achieves higher accuracy, this highlights a promising direction for improving the multi-cell architecture to better capture intercellular dynamics and tissue-level structure.

While GeneFlow offers a strong foundation, several limitations highlight important directions for future work. Our current implementation operates on 256×256 pixel tiles, serving as a proof-of-concept that can be extended to whole-slide image synthesis using hierarchical generation or sliding-window strategies with boundary-aware stitching. Additionally, our reliance on Xenium data featuring true single-cell resolution limits compatibility with a broader range of spatial transcriptomics datasets that offer only near single-cell resolution. Scaling to datasets with more comprehensive gene panels may also pose challenges, as the model must handle higher-dimensional input. This could be addressed by integrating foundational models trained on single-cell or tissue-level gene expression.

Finally, our current cell alignment method treats cell inclusion as binary. Incorporating weighted contributions based on partial overlap would enable more nuanced modeling of cell context. These limitations point to natural and promising extensions of our framework. Future work will also explore next-generation generative models and large-scale training on whole-slide images to further advance the mapping between gene expression and visual cellular phenotypes at high resolution.

Acknowledgement

This work was supported in part by the Walther Cancer Foundation and the Purdue Institute for Cancer Research (Grant P30CA023168).

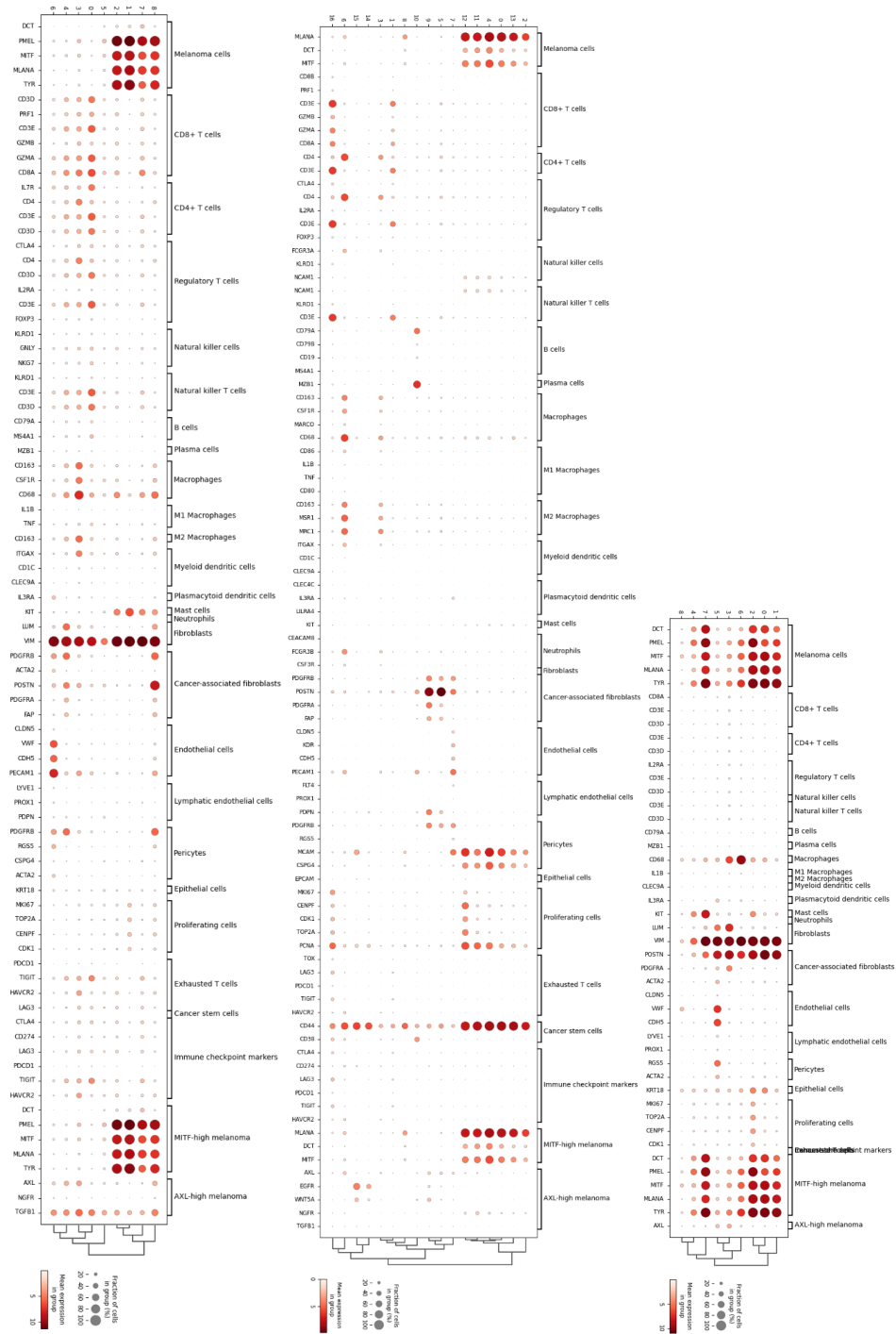
References

- [1] Y. Jin, Y. Zuo, G. Li, W. Liu, Y. Pan, T. Fan, X. Fu, X. Yao, and Y. Peng, “Advances in spatial transcriptomics and its applications in cancer research,” *Molecular Cancer*, vol. 23, no. 1, p. 129, 2024.
- [2] J. Du, Y.-C. Yang, Z.-J. An, M.-H. Zhang, X.-H. Fu, Z.-F. Huang, Y. Yuan, and J. Hou, “Advances in spatial transcriptomics and related data analysis strategies,” *Journal of translational medicine*, vol. 21, no. 1, p. 330, 2023.
- [3] D. Cifci, S. Foersch, and J. N. Kather, “Artificial intelligence to identify genetic alterations in conventional histopathology,” *The Journal of pathology*, vol. 257, no. 4, pp. 430–444, 2022.
- [4] Z. Zeng, Y. Li, Y. Li, and Y. Luo, “Statistical and machine learning methods for spatially resolved transcriptomics data analysis,” *Genome biology*, vol. 23, no. 1, p. 83, 2022.
- [5] T.-Y. Chen, L. You, J. A. U. Hardillo, and M.-P. Chien, “Spatial transcriptomic technologies,” *Cells*, vol. 12, no. 16, p. 2042, 2023.
- [6] Y. Wang, B. Liu, G. Zhao, Y. Lee, A. Buzdin, X. Mu, J. Zhao, H. Chen, and X. Li, “Spatial transcriptomics: Technologies, applications and experimental considerations,” *Genomics*, vol. 115, no. 5, p. 110671, 2023.
- [7] A. Rao, D. Barkley, G. S. França, and I. Yanai, “Exploring tissue architecture using spatial transcriptomics,” *Nature*, vol. 596, no. 7871, pp. 211–220, 2021.
- [8] L. Tian, F. Chen, and E. Z. Macosko, “The expanding vistas of spatial transcriptomics,” *Nature Biotechnology*, vol. 41, no. 6, pp. 773–782, 2023.
- [9] C. Wang, A. S. Chan, X. Fu, S. Ghazanfar, J. Kim, E. Patrick, and J. Y. Yang, “Benchmarking the translational potential of spatial gene expression prediction from histology,” *Nature Communications*, vol. 16, no. 1, p. 1544, 2025.
- [10] M. Pang, K. Su, and M. Li, “Leveraging information in spatial transcriptomics to predict super-resolution gene expression from histology images in tumors,” *BioRxiv*, pp. 2021–11, 2021.
- [11] Y. Jia, J. Liu, L. Chen, T. Zhao, and Y. Wang, “Thitogene: a deep learning method for predicting spatial transcriptomics from histological images,” *Briefings in Bioinformatics*, vol. 25, no. 1, p. bbad464, 2024.
- [12] M. M. Rahaman, E. K. Millar, and E. Meijering, “Breast cancer histopathology image-based gene expression prediction using spatial transcriptomics data and deep learning,” *Scientific Reports*, vol. 13, no. 1, p. 13604, 2023.
- [13] Y. Shan, Q. Zhang, W. Guo, Y. Wu, Y. Miao, H. Xin, Q. Lian, and J. Gu, “Tist: transcriptome and histopathological image integrative analysis for spatial transcriptomics,” *Genomics, Proteomics & Bioinformatics*, vol. 20, no. 5, pp. 974–988, 2022.
- [14] R. Xie, K. Pang, S. Chung, C. Perciani, S. MacParland, B. Wang, and G. Bader, “Spatially resolved gene expression prediction from histology images via bi-modal contrastive learning,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 70626–70637, 2023.
- [15] Y. Zeng, Z. Wei, W. Yu, R. Yin, Y. Yuan, B. Li, Z. Tang, Y. Lu, and Y. Yang, “Spatial transcriptomics prediction from histology jointly through transformer and graph neural networks,” *Briefings in Bioinformatics*, vol. 23, no. 5, p. bbac297, 2022.
- [16] S. Zhu, Y. Zhu, M. Tao, and P. Qiu, “Diffusion generative modeling for spatially resolved gene expression inference from histology images,” *arXiv preprint arXiv:2501.15598*, 2025.
- [17] R. Zhou, G. Yang, Y. Zhang, and Y. Wang, “Spatial transcriptomics in development and disease,” *Molecular Biomedicine*, vol. 4, no. 1, p. 32, 2023.
- [18] A. A. Ahmed and M. Abedalthagafi, “Cancer diagnostics: the journey from histomorphology to molecular profiling,” *Oncotarget*, vol. 7, no. 36, p. 58696, 2016.

- [19] M. A. Cilento, C. J. Sweeney, and L. M. Butler, "Spatial transcriptomics in cancer research and potential clinical impact: a narrative review," *Journal of Cancer Research and Clinical Oncology*, vol. 150, no. 6, p. 296, 2024.
- [20] Q. Yu, M. Jiang, and L. Wu, "Spatial transcriptomics technology in cancer research," *Frontiers in Oncology*, vol. 12, p. 1019111, 2022.
- [21] L.-J. Tseng, A. Matsuyama, and V. MacDonald-Dickinson, "Histology: The gold standard for diagnosis?," *The Canadian Veterinary Journal*, vol. 64, no. 4, p. 389, 2023.
- [22] E. R. Malone, M. Oliva, P. J. Sabatini, T. L. Stockley, and L. L. Siu, "Molecular profiling for precision cancer therapies," *Genome medicine*, vol. 12, pp. 1–19, 2020.
- [23] H.-E. Park, S. H. Jo, R. H. Lee, C. P. Macks, T. Ku, J. Park, C. W. Lee, J. K. Hur, and C. H. Sohn, "Spatial transcriptomics: technical aspects of recent developments and their applications in neuroscience and cancer research," *Advanced Science*, vol. 10, no. 16, p. 2206939, 2023.
- [24] F. Carrillo-Perez, M. Pizurica, M. G. Ozawa, H. Vogel, R. B. West, C. S. Kong, L. J. Herrera, J. Shen, and O. Gevaert, "Synthetic whole-slide image tile generation with gene expression profile-infused deep generative models," *Cell Reports Methods*, vol. 3, no. 8, 2023.
- [25] F. M. Howard, H. M. Hieromnimon, S. Ramesh, J. Dolezal, S. Kochanny, Q. Zhang, B. Feiger, J. Peterson, C. Fan, C. M. Perou, *et al.*, "Generative adversarial networks accurately reconstruct pan-cancer histology from pathologic, genomic, and radiographic latent features," *Science Advances*, vol. 10, no. 46, p. eadq0856, 2024.
- [26] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," *arXiv preprint arXiv:2209.03003*, 2022.
- [27] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3964–3979, 2020.
- [28] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in neural information processing systems*, vol. 31, 2018.
- [29] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [30] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *arXiv preprint arXiv:2210.02747*, 2022.
- [31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [32] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pp. 234–241, Springer, 2015.
- [33] J. R. Dormand and P. J. Prince, "Runge-kutta triples," *Computers & Mathematics with Applications*, vol. 12, no. 9, pp. 1007–1017, 1986.
- [34] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
- [35] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.
- [36] A. Janesick, R. Shelansky, A. D. Gottscho, F. Wagner, S. R. Williams, M. Rouault, G. Beliakoff, C. A. Morrison, M. F. Oliveira, J. T. Sicherman, *et al.*, "High resolution mapping of the tumor microenvironment using integrated single-cell, spatial and in situ analysis," *Nature communications*, vol. 14, no. 1, p. 8353, 2023.
- [37] G. Jaume, P. Doucet, A. Song, M. Y. Lu, C. Almagro Pérez, S. Wagner, A. Vaidya, R. Chen, D. Williamson, A. Kim, *et al.*, "Hest-1k: A dataset for spatial transcriptomics and histology image analysis," *Advances in Neural Information Processing Systems*, vol. 37, pp. 53798–53833, 2024.

- [38] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” *Advances in neural information processing systems*, vol. 30, 2017.
- [39] 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*, vol. 30, no. 3, pp. 850–862, 2024.
- [40] Z. Xie, A. Bailey, M. V. Kuleshov, D. J. Clarke, J. E. Evangelista, S. L. Jenkins, A. Lachmann, M. L. Wojciechowicz, E. Kropiwnicki, K. M. Jagodnik, *et al.*, “Gene set knowledge discovery with enrichr,” *Current protocols*, vol. 1, no. 3, p. e90, 2021.
- [41] L. Jerby-Arnon, P. Shah, M. S. Cuoco, C. Rodman, M.-J. Su, J. C. Melms, R. Leeson, A. Kandodia, S. Mei, J.-R. Lin, *et al.*, “A cancer cell program promotes t cell exclusion and resistance to checkpoint blockade,” *Cell*, vol. 175, no. 4, pp. 984–997, 2018.
- [42] I. Tirosh, B. Izar, S. M. Prakadan, M. H. Wadsworth, D. Treacy, J. J. Trombetta, A. Rotem, C. Rodman, C. Lian, G. Murphy, *et al.*, “Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq,” *Science*, vol. 352, no. 6282, pp. 189–196, 2016.

A Datasets Details



expression data for hundreds of targeted genes, enabling us to map transcript locations and quantify expression levels within specific tissue regions and individual cells. We extracted cell centroid from pre-processed melanoma Xenium data, locating cell centroid at the center of image with size of 256 pixels. We do not mask cell by its boundary due to 1) cells may overlap with each other which may cause inaccurate fluorescence-based cell boundary identification 2) actual cell size can be affected by the sample preparation that cells more densely collapsed together or loosely isolated may lead to abnormal cell size and 3) without masking our model can learn more than intrinsic cell features but the environments where that cell resides via the environment information (neighboring cells, etc.) provided in the unmasked image, which is key to single-cell level accurate image reconstruction for precise pathological diagnosis.

For each sample, we further processed the transcriptomics data and clustering the cells based on their transcriptomes profiles. We annotated cell types referring to marker genes from [41, 42], Marker genes expressions are visualized in the dot plots.

B Experiments

B.1 Quantitative Results

Sample	Model	FID	SSIM±	Feature Dist±
Xenium _{P1}	multi	105.54±6.65	0.28±0.13	13.88±2.38
Xenium _{P1}	single	32.43±16.75	0.17±0.03	14.22±2.08
Xenium _{C1}	multi	63.41±3.83	0.33±0.13	12.61±2.69
Xenium _{C1}	single	25.47±16.10	0.23±0.03	14.56±2.54
Xenium _{C2}	multi	41.45±6.80	0.51±0.05	14.84±2.13
Xenium _{C2}	single	21.85±5.76	0.49±0.05	15.10±2.22

Table 7: Rectified Flow model trained on all cell types but tested only on melanoma cells

Sample	Model	FID	SSIM±	Feature Dist±
Xenium _{P1}	multi	50.09±11.37	0.32±0.16	13.23±2.41
Xenium _{P1}	single	37.08±4.61	0.17±0.03	14.83±2.19
Xenium _{C1}	multi	94.79±4.82	0.41±0.14	16.03±3.04
Xenium _{C1}	single	34.59±15.01	0.25±0.05	15.74±2.38
Xenium _{C2}	multi	54.18±2.97	0.37±0.09	15.85±2.34
Xenium _{C2}	single	51.38±12.59	0.27±0.04	15.75±2.30

Table 8: Rectified Flow model trained on all cell types but tested only on non-melanoma cells

We conducted a series of evaluation experiments using Rectified Flow models trained on all cell types but tested under three specific conditions: (1) only on melanoma cells, (2) only on non-melanoma cells, and (3) separately on each non-melanoma cell type. Tables 7–9 report the corresponding evaluation metrics across different samples and configurations. Models tested exclusively on melanoma cells (Table 7) exhibited strong performance, particularly in the single-cell conditioned setting. Notably, while multi-cell conditioning resulted in marginally lower FID scores in some cases, it consistently led to higher SSIM values, indicating better structural similarity. This suggests that multi-cell generation better preserves spatial structure when the sample contains a mixture of cell types, as is typical even in melanoma-dominated tissues. Evaluation on non-melanoma cells (Table 8) showed that models trained on all cell types maintained good generalization, though the variability in FID and SSIM scores increased across samples. Again, the multi-cell conditioned models tended to produce more structurally consistent images (higher SSIM), whereas single-cell conditioned models sometimes yielded lower FID scores, indicating perceptually closer images. The most granular analysis testing on individual non-melanoma cell types (Table 9) revealed a slight drop in performance across all metrics. This decline can likely be attributed to the reduced diversity in conditioning data: conditioning on a single cell type eliminates contextual transcriptomic variation that may assist the model in disambiguating cell-specific morphological features. In particular, cell types such as epithelial and endothelial cells demonstrated higher FID and lower SSIM, suggesting that their complex spatial

Gene/Cell Type (P1)	FID	SSIM±	Feature Dist±
B/Plasma	59.02±8.59	0.17±0.03	14.69±2.01
Endo	49.75±2.52	0.17±0.03	14.72±2.05
Epith	191.30±9.21	0.21±0.04	17.60±1.97
Fibro	53.22±9.68	0.18±0.03	15.02±2.15
Mono/Mac	32.28±3.64	0.17±0.03	14.47±2.04
T cells	37.63±3.13	0.16±0.03	14.57±2.04

Gene/Cell Type (C1)	FID	SSIM±	Feature Dist±
Endo	69.82±17.87	0.26±0.05	16.40±2.29
Epith/Kera	51.14±28.75	0.27±0.05	15.68±2.48
Fibro/Stroma	39.57±13.39	0.24±0.04	15.76±2.25
Macro/Myeloid	40.76±3.41	0.23±0.04	14.83±2.27

Gene/Cell Type (C2)	FID	SSIM±	Feature Dist±
Endo	80.71±10.73	0.29±0.04	16.36±2.37
Epith/Kera	71.04±10.03	0.42±0.06	16.47±2.42
Fibro	61.50±11.25	0.29±0.05	15.98±2.30
Macro/Mono	59.19±12.77	0.29±0.04	15.85±2.18
T cells	56.93±16.43	0.24±0.03	15.11±2.19

Table 9: Rectified Flow model tested on non-melanoma cell types in (top) P_1 , (middle) C_1 , and (bottom) C_2 .

morphologies or transcriptional profiles are harder to reconstruct accurately under single-cell-type conditioning. Overall, our results indicate that while the model performs robustly across all settings, multi-cell-type conditioning offers a tangible advantage, especially for maintaining spatial fidelity in heterogeneous tissues. The detailed evaluation metrics for each configuration are reported in Tables 7, 8, and 9.

B.2 Ablation Study

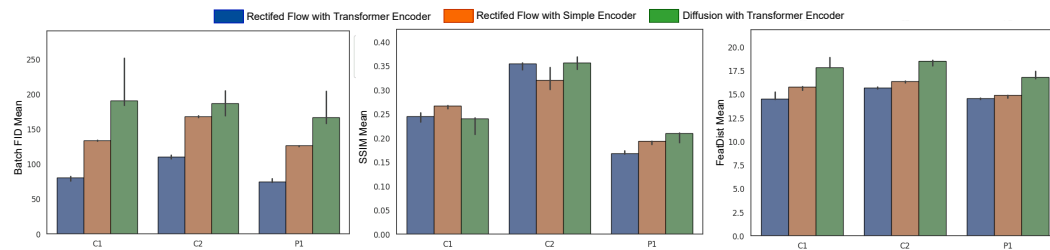


Figure 8: Ablation study metric comparison between Rectified Flow with a transformer encoder vs Rectified Flow with a simple encoder vs Diffusion with a transformer encoder.

To assess the effectiveness of our transformer-based RNA encoder, we conducted an ablation study comparing three configurations: (1) Rectified Flow with a transformer RNA encoder, (2) Rectified Flow with a simple RNA encoder, and (3) a standard diffusion model with a transformer encoder. The simple RNA encoder replaces attention mechanisms, residual blocks, and feature gating with a basic stack of fully connected linear layers to encode gene expression data. Figure 8 presents a comparative evaluation across three key metrics FID, SSIM, and Feature Distance on three representative samples (C1, C2, and P1). Models using the transformer-based RNA encoder consistently outperformed those with the simple encoder, particularly in terms of FID and Feature Distance. This demonstrates that incorporating attention and residual connections facilitates better conditioning on gene expression, leading to more realistic and feature-faithful image generation. Interestingly, when comparing Rectified Flow and Diffusion models both using transformer encoders, Rectified Flow generally achieved lower FID and Feature Distance scores, especially on C1 and P1, indicating superior visual quality and fidelity. However, Diffusion models showed competitive or slightly better SSIM on C2 and P1, suggesting that while they may achieve better pixel-wise similarity, Rectified Flow captures global image realism more effectively. These results validate the importance of architectural choices in the encoder design, highlighting the robustness and effectiveness of the transformer-based encoder in the context of spatial transcriptomic image synthesis.

B.3 Qualitative Results

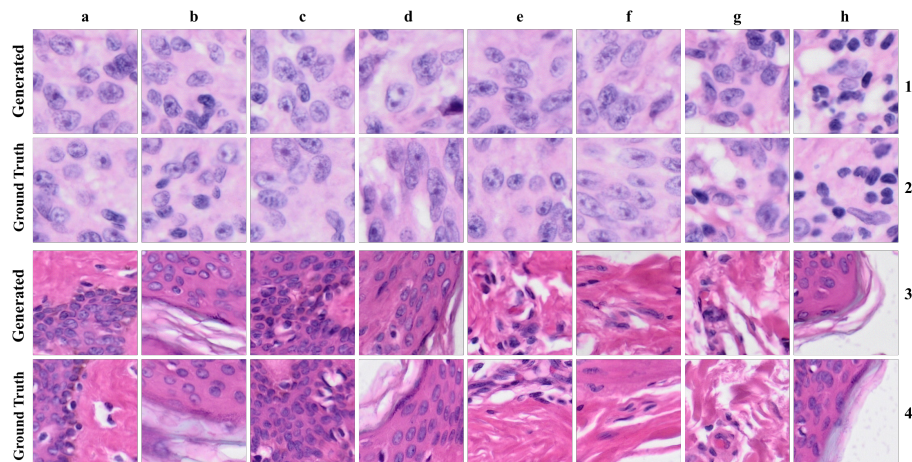


Figure 9: Extra examples of neoplastic tumor cells (Top) and non-neoplastic cells (Bottom), diagnosed by ACVP board certified pathologist.

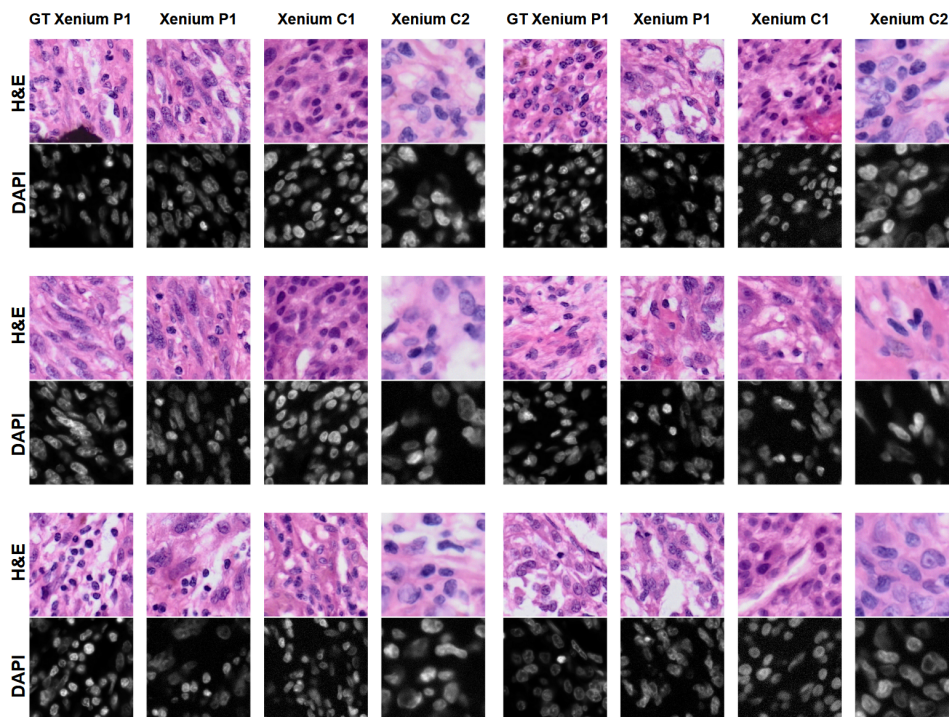


Figure 10: Comparison of ground-truth and generated images for Rectified Flow model trained on Xenium P1 and tested on Xenium P1, Xenium C1, Xenium C2. There are 2 columns with 3 rows.

Figures 10, 11, 12 show comparison of ground truth and generated images for the cross dataset evaluation task. This task corresponds to the results in Table 4. Here the Rectified Flow model is trained on the 3 datasets separately and tested on the held out datasets. These models were trained on the 126 overlapping shared genes.

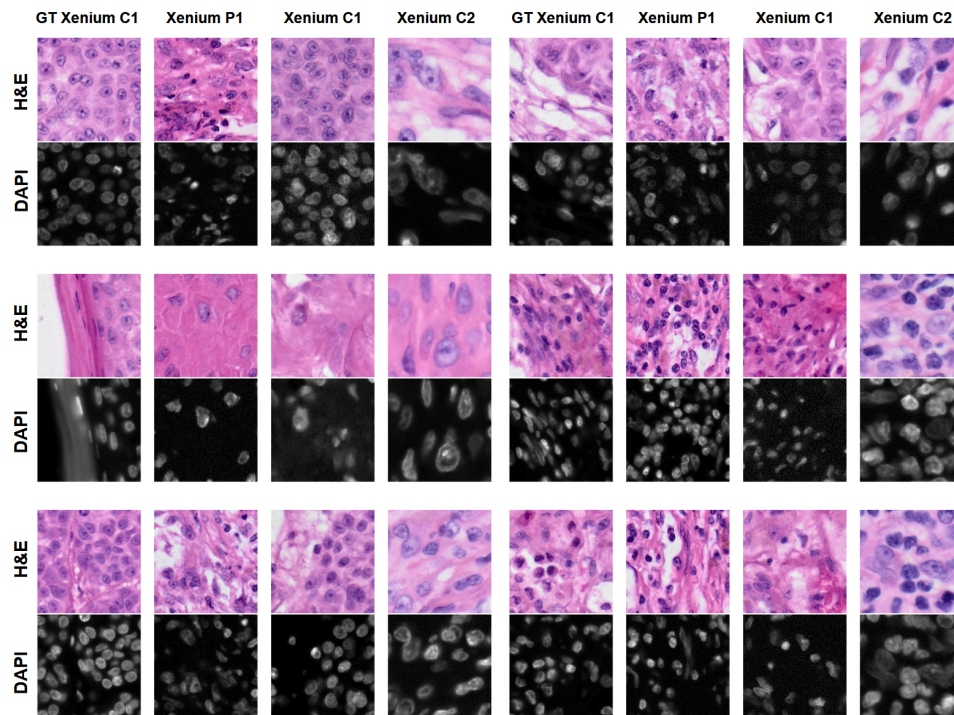


Figure 11: Comparison of ground-truth and generated images for Rectified Flow model trained on Xenium C1 and tested on Xenium P1, Xenium C1, Xenium C2. There are 2 columns with 3 rows.

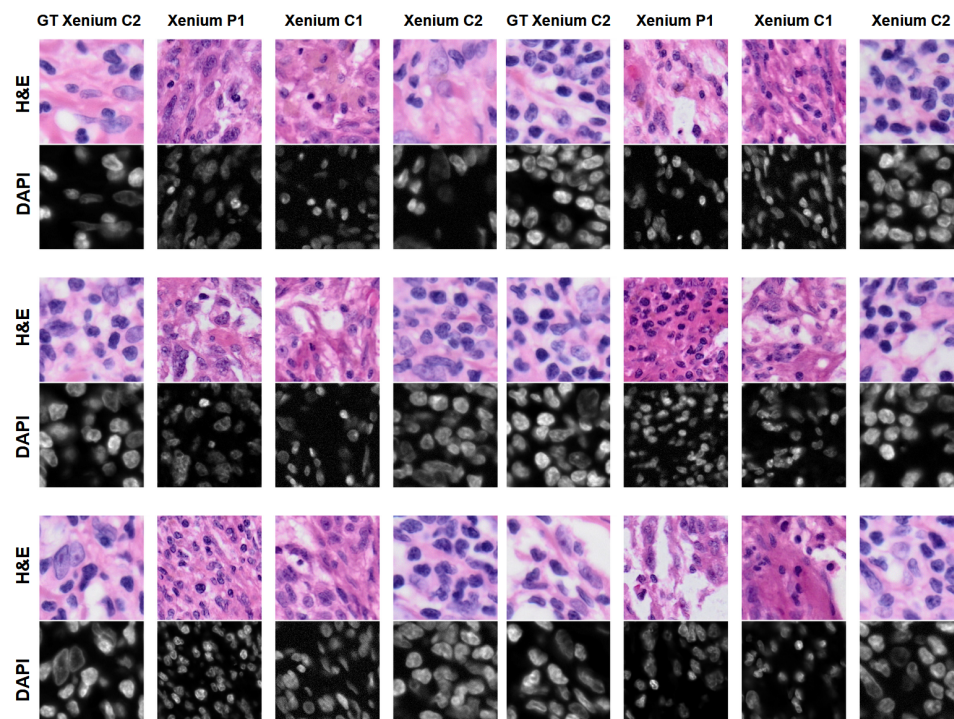


Figure 12: Comparison of ground-truth and generated images for Rectified Flow model trained on Xenium C2 and tested on Xenium P1, Xenium C1, Xenium C2. There are 2 columns with 3 rows.

C Gene Influence Analysis

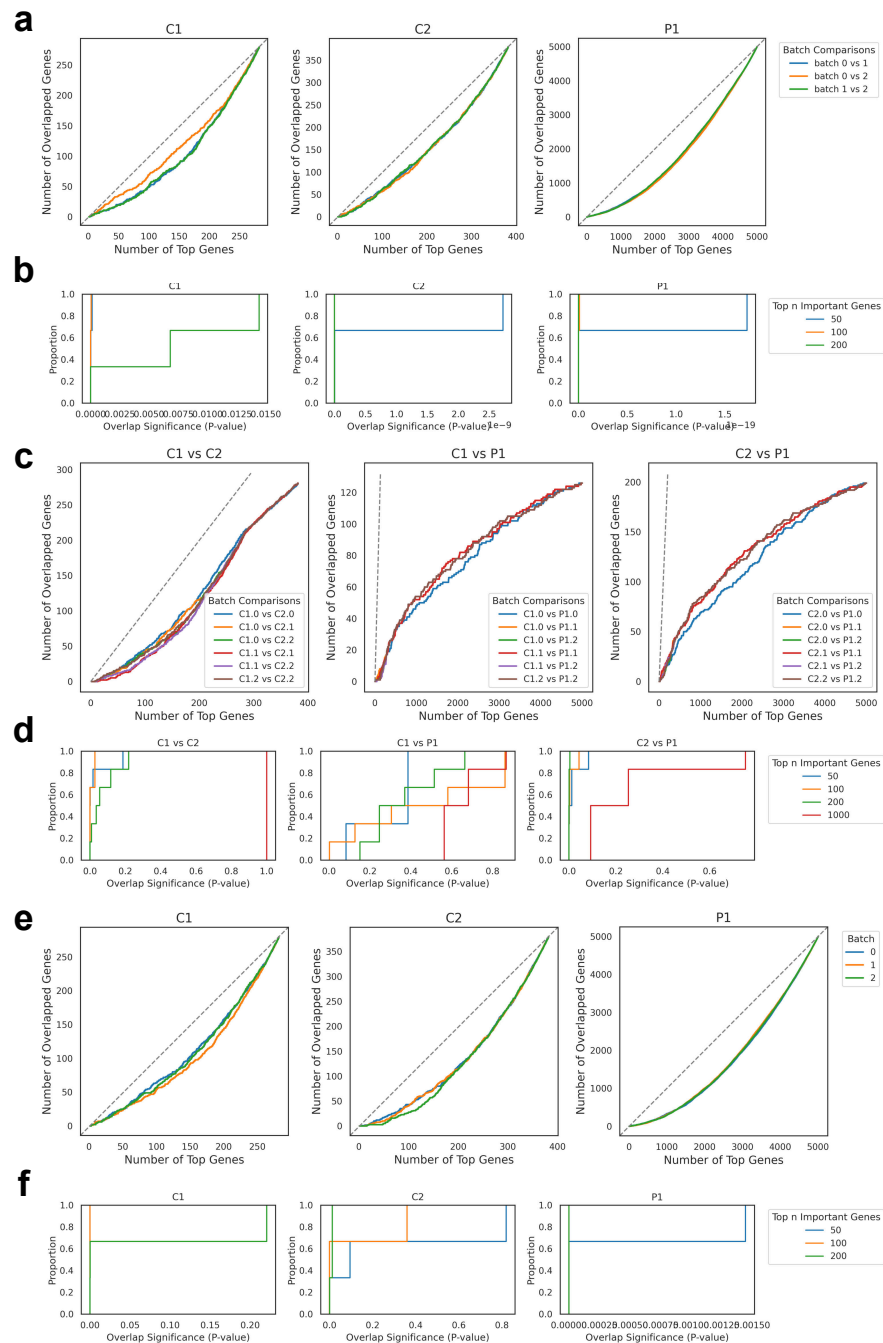


Figure 13: Line charts and cumulative density plots for influential gene analysis. **a, c, e.** Line charts showing the number of overlapping genes as a function of the total number of top influential genes examined. The dashed diagonal line indicates perfect overlap, where the number of overlaps equals the number of top genes considered. **b, d, f.** Cumulative density plots of p-values for the overlap of top influential genes under varying thresholds (top 50, 100, 200, and 1000 genes when Xenium P_1 is included). The x-axis shows p-values; the y-axis shows cumulative density. Different colors indicate different thresholds. **a, b.** Comparison across cross-validation folds. **c, d.** Comparison across biological samples. **e, f.** Comparison between models trained on all cell types vs. models trained only on melanoma cells.

We evaluated gene importance with respect to gene attention map as well as gradient flow analysis from all experiments, with respect to consistency across folds, consistency across models trained on different samples, and consistency across models trained on all cell types versus models trained on only melanoma cells. Our experiments demonstrated overall very high consistency with significant overlap across top important/influential genes from across folds evaluation scenario13(a,b), with p-value much less than 0.05 from one-tailed (greater) fisher-exact, regardless evaluating overlap with top 50, 100 or 200 influential genes. It indicated our model can stably capture marker genes contributing to cellular morphology during generation.

When comparing models trained on one sample while evaluated on unseen samples13(c,d), we observed high overlap between models trained on one of the two samples with standard gene panels and custom add-on and tested on the other, which is as expected since they share a major portion of target genes. All test cases with top 100 influential genes are statistically significant (p-value less than 0.05). Over 80% cases with top 50 have significant overlap.

Since the majority of cells from our curated dataset are melanoma cells, we expect our model to focus more on genes that are more important to picture melanoma cell morphology. Our most influential gene analysis revealed that there's significant overlap of top important genes between model trained on all cell types and model trained on only melanoma cells, demonstrating that our model's successfully focused on melanoma cells out of all cell types. This also matched the better metrics over evaluation against melanoma cells compared to against each other cell type from quantitative results analyses.

C.1 Enrichment gene-sets/pathways from top influential genes

We evaluated top influential genes from comparisons across datasets, where we found highest consistency lying in sample C_1 and C_2 . We selected shared genes from top 50 most influential genes from both datasets, sorted by average importance score, and ran gene-set enrichment analysis with Enrichr[40]. We found epithelial mesenchymal transition pathway is mostly enriched, along with KRAS signaling pathway, angiogenesis, apoptosis, myogenesis and coagulation which were identified to be closely relevant to melanoma development and progression by previous studies. This further demonstrates our model's capability on identifying disease relevant genes during generation cell/tissue images guided by transcriptomics. Details of significantly enriched pathways are attached as follows.

Term	P-value	Adj. P-value	Odds Ratio	Combined Score	Genes
Epithelial Mesenchymal Transition	2.09E-05	5.02E-04	17.48	188.37	POSTN;LUM;MMP2;IGFBP2;MGP
KRAS Signaling Dn	3.56E-04	4.27E-03	13.45	106.80	TFAP2B;KRT15;IGFBP2;KRT5
Angiogenesis	1.70E-03	1.36E-02	36.64	233.56	POSTN;LUM
Apoptosis	2.55E-03	1.53E-02	12.13	72.44	CCND1;LUM;MMP2
Myogenesis	4.69E-03	2.25E-02	9.71	52.07	ACHE;APOD;CRYAB
Coagulation	2.30E-02	9.18E-02	9.11	34.39	C1QA;MMP2

Table 10: Gene set enrichment analysis of the most influential genes identified by our model. The table shows the top 10 enriched pathways ranked by adjusted p-value.

D Model Architecture

D.1 RNA Encoder

The RNA Encoder is a deep neural encoder designed to transform gene expression profiles from multiple single cells into a compact, biologically enriched embedding. This encoder supports enhanced cell representation through three mechanisms: (i) gene-gene relational modeling via low-rank factorization, (ii) global gene importance attention, and (iii) cell-wise aggregation using multi-head attention. It is optimized to handle variable numbers of cells per input sample, with optional masking support.

Given an input tensor of shape $[B, C_{\max}, G]$ where B is the batch size, $C_{\max}$ is the maximum number of cells per patch, and G is the number of genes the encoder outputs a single $[B, D_{\text{out}}]$ embedding per patch. The architecture comprises the following main components: **Gene-Gene Relation Module:** Models contextual dependencies between genes per cell using low-rank matrix factorization. **Gene Attention Module:** Applies learned global attention weights to emphasize biologically informative genes. **Cell Encoder Stack:** Processes gene expression per cell through residual blocks to obtain

latent cell embeddings. **Multi-Head Attention Aggregation:** Aggregates cell embeddings into a single patch embedding using learned cell-wise attention across multiple heads. **Final Projection and Gating:** Refines the aggregated feature through normalization, projection, and optional gating.

Gene-Gene Relation Module: This module enhances input gene vectors via low-rank matrix factorization:

$$\tilde{x}_i = x_i + 0.1 \cdot [(x_i U_i) V_i], \quad \text{where } U_i \in \mathbb{R}^{G \times K}, V_i \in \mathbb{R}^{K \times G}$$

The U_i and V_i matrices are predicted per cell using a lightweight feed-forward subnetwork. The gene relation parameters $U_i \in \mathbb{R}^{G \times K}$ and $V_i \in \mathbb{R}^{K \times G}$ for each cell are generated via a two-layer MLP:

$$\begin{aligned} h_i &= \text{Dropout}(\text{SiLU}(\text{LayerNorm}(W_1 x_i + b_1))) \in \mathbb{R}^{256} \\ \theta_i &= W_2 h_i + b_2 \in \mathbb{R}^{2 \cdot G \cdot K} \\ \theta_i &= \text{concat}(\text{vec}(U_i), \text{vec}(V_i)) \end{aligned}$$

Here $x_i \in \mathbb{R}^G$ is the input gene expression vector for cell i . $W_1 \in \mathbb{R}^{256 \times G}$ and $W_2 \in \mathbb{R}^{2 \cdot G \cdot K \times 256}$ are learned projection weights. θ_i is reshaped into matrices $U_i \in \mathbb{R}^{G \times K}$ and $V_i \in \mathbb{R}^{K \times G}$.

Gene Attention Module: A softmax-normalized attention vector $\alpha \in \mathbb{R}^G$ is learned across all genes:

$$x'_i = x_i \odot \text{softmax}(\alpha)$$

Cell Encoder Stack: Each weighted gene expression vector $x_i \in \mathbb{R}^G$ (or $x_i \in \mathbb{R}^{2G}$ if concatenated with auxiliary features) is passed through a stack of two residual blocks to produce a latent cell embedding $e_i \in \mathbb{R}^{256}$.

Residual Block 1:

$$h_i^{(1)} = \text{Dropout}(\text{SiLU}(\text{Linear}(\text{LayerNorm}(x_i)))) \in \mathbb{R}^{512}$$

Residual Block 2:

$$e_i = \text{Dropout}(\text{SiLU}(\text{Linear}(\text{LayerNorm}(h_i^{(1)})))) \in \mathbb{R}^{256}$$

These operations are applied independently to each cell i in the sample. The resulting set $\{e_i\}_{i=1}^C$ serves as the input to the multi-head attention aggregator.

Cell Aggregation via Multi-Head Attention: The encoded cell embeddings $\{e_i\}_{i=1}^N$, where $e_i \in \mathbb{R}^d$, are aggregated using a multi-head attention mechanism. Each head learns to focus on different aspects of the cell population. Let H be the number of attention heads. For each head $h = 1, \dots, H$: Attention Logits compute unnormalized attention weights via a head-specific scoring network:

$$a_i^{(h)} = w^{(h)\top} \cdot \tanh(W^{(h)} e_i + b^{(h)}) \quad \text{for } i = 1, \dots, N$$

Attention Weights: Normalize logits using a softmax across all N cells. If masking is used to ignore padded or invalid cells, the masked version is applied:

$$\begin{aligned} \alpha_i^{(h)} &= \frac{\exp(a_i^{(h)})}{\sum_{j=1}^N \exp(a_j^{(h)})} \quad (\text{no masking}) \\ \alpha_i^{(h)} &= \frac{\exp(a_i^{(h)}) \cdot m_i}{\sum_{j=1}^N \exp(a_j^{(h)}) \cdot m_j}, \quad m_i \in \{0, 1\} \quad (\text{with masking}) \end{aligned}$$

Projection: Linearly project each embedding into a head-specific space:

$$\tilde{e}_i^{(h)} = V^{(h)} e_i$$

Aggregation: Compute the weighted sum of projected embeddings:

$$z^{(h)} = \sum_{i=1}^N \alpha_i^{(h)} \tilde{e}_i^{(h)}$$

Head Fusion: The final aggregated representation is obtained by averaging over all heads:

$$z = \frac{1}{H} \sum_{h=1}^H z^{(h)}$$

This vector $z \in \mathbb{R}^{d'}$ serves as the RNA-level embedding summarizing the full set of cell embeddings through attention-driven aggregation.

Final Encoding and Feature Gating: The aggregated embedding is passed through a final linear projection and optional sigmoid gating:

$$z = \text{LayerNorm}(\text{Linear}(\text{LayerNorm}(a))), \quad \hat{z} = z \odot \sigma(\text{Linear}(z))$$

Output: The encoder outputs a batch of embeddings $[B, D_{\text{out}}]$, where D_{out} is a configurable dimension (e.g., 512). These embeddings can be used for downstream tasks such as conditioning image synthesis via UNet.

For clarity, we summarize the main tensor shapes used:

- $x \in \mathbb{R}^{C \times H \times W}$: Feature map at each U-Net level
- $t \in \mathbb{R}^{d_t}$: Diffusion timestep embedding
- $r \in \mathbb{R}^{d_r}$: RNA embedding vector (output of RNA encoder)
- $x_i \in \mathbb{R}^G$: Raw gene expression vector for cell i
- $e_i \in \mathbb{R}^{d'}$: Encoded cell embedding for cell i
- $z \in \mathbb{R}^{d'}$: Aggregated RNA embedding

D.2 Conditioned U-Net Architecture with RNA and Timestep Embeddings

The conditioned U-Net integrates residual blocks with both timestep and RNA conditioning throughout its encoder, bottleneck, and decoder. This design allows the model to perform spatiotemporal image generation or transformation while incorporating gene expression context.

Input and Conditioning Embeddings: Let: $x_0 \in \mathbb{R}^{C \times H \times W}$ be the input image. $t \in \mathbb{R}^{d_t}$ be the diffusion timestep embedding. $r \in \mathbb{R}^{d_r}$ be the RNA embedding vector. These are passed to all residual blocks throughout the network.

Encoder Path: The encoder consists of L levels. Each level performs One or more **ResBlocks** with conditioning:

$$x^{(l)} = \text{ResBlock}(x^{(l-1)}, t, r)$$

A **Downsampling** operation (e.g., strided convolution or pooling):

$$x_{\text{down}}^{(l)} = \text{Down}(x^{(l)})$$

Intermediate outputs are stored as **skip connections**:

$$\text{skip}^{(l)} = x^{(l)}$$

Bottleneck: At the lowest resolution level, additional **ResBlocks** with conditioning are applied:

$$x_{\text{bottleneck}} = \text{ResBlock}(\text{ResBlock}(x_{\text{down}}^{(L)}, t, r), t, r)$$

Decoder Path: The decoder also has L levels and mirrors the encoder: Upsample the bottleneck or previous output:

$$x_{\text{up}}^{(l)} = \text{Up}(x^{(l+1)})$$

Concatenate with the corresponding skip connection:

$$x_{\text{cat}}^{(l)} = \text{Concat}(x_{\text{up}}^{(l)}, \text{skip}^{(l)})$$

Apply one or more **ResBlocks** with conditioning:

$$x^{(l)} = \text{ResBlock}(x_{\text{cat}}^{(l)}, t, r)$$

Final Output Layer: After the final decoder level a final convolution layer maps the result to the desired number of output channels:

$$\hat{x}_0 = \text{Conv}_{\text{out}}(x^{(0)})$$

The U-Net leverages **ResBlocks with timestep and RNA conditioning** at every level to guide feature transformations based on both dynamic diffusion context (t) and transcriptomic content (r). Skip connections to preserve high-resolution spatial information across the network. Symmetric encoder-decoder structure to downsample and then upsample features, enabling efficient learning of hierarchical and context-aware representations.

Residual Block with Timestep and RNA Conditioning: The input feature map $x \in \mathbb{R}^{C \times H \times W}$. A timestep embedding vector $t \in \mathbb{R}^{d_t}$. An RNA feature embedding vector $r \in \mathbb{R}^{d_r}$. Both t and r are projected and added to the intermediate representations of x during processing.

Input Transformations: The input x is passed through two convolutional blocks (Conv-BN-GELU):

$$\begin{aligned} h_1 &= \text{Conv}(\text{BN}(\text{GELU}_1(x))) \\ h_2 &= \text{Conv}(\text{BN}(\text{GELU}_2(h_1))) \end{aligned}$$

Conditioning via Additive Projections: The timestep embedding t and RNA embedding r are each passed through separate MLPs (typically implemented as linear layers followed by non-linearities) and reshaped to be broadcastable across spatial dimensions:

$$\begin{aligned} t_{\text{proj}} &= \text{MLP}_t(t) \in \mathbb{R}^C \\ r_{\text{proj}} &= \text{MLP}_r(r) \in \mathbb{R}^C \end{aligned}$$

These are added to h_2 :

$$h_{\text{cond}} = h_2 + t_{\text{proj}}[:, \text{None}, \text{None}] + r_{\text{proj}}[:, \text{None}, \text{None}]$$

Final Convolution and Residual Connection: The conditioned output h_{cond} is passed through a final convolution:

$$h_{\text{out}} = \text{Conv}_{\text{final}}(\text{GELU}(h_{\text{cond}}))$$

A skip connection is applied. If the input and output channels differ, a 1×1 convolution (projection) is applied to x :

$$\begin{aligned} x_{\text{res}} &= \begin{cases} x, & \text{if } C_{\text{in}} = C_{\text{out}} \\ \text{Conv}_{1 \times 1}(x), & \text{otherwise} \end{cases} \\ \text{Output} &= x_{\text{res}} + h_{\text{out}} \end{aligned}$$

The block allows feature maps to be conditioned on both temporal context (via timestep embedding) and transcriptomic context (via RNA embedding), enabling the model to modulate its computations dynamically based on both spatial and external biological signals. For clarity, the overall residual block transformation can be written as:

$$\text{ResBlock}(x, t, r) = \text{Conv}_{\text{final}}(\text{GELU}(\text{Conv}_2(\text{GELU}(\text{Conv}_1(x)) + \text{MLP}_t(t) + \text{MLP}_r(r))) + x_{\text{res}}$$

E Spatial Regularization

E.1 Spatial Graph Loss Framework

Let $\mathbf{X}_{\text{real}} \in \mathbb{R}^{B \times C \times H \times W}$ and $\mathbf{X}_{\text{gen}} \in \mathbb{R}^{B \times C \times H \times W}$ denote batches of real and generated images, where B is the batch size, C is the number of channels, and $H \times W$ is the spatial resolution. For each image pair in the batch, we construct a spatial graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where vertices $\mathcal{V}$ correspond to spatial locations and edges $\mathcal{E}$ connect spatially proximate regions.

The overall training objective combines the base rectified flow loss with the spatial graph loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{RF}}(\mathbf{v}_{\theta}, \mathbf{v}_{\text{target}}) + \lambda_s \cdot w(t) \cdot \mathcal{L}_{\text{spatial}}(\mathbf{X}_{\text{gen}}, \mathbf{X}_{\text{real}}) \quad (11)$$

where $\mathcal{L}_{RF}$ is the rectified flow velocity matching loss, λ_s is the spatial loss weight, $w(t)$ is a warmup schedule, and $\mathcal{L}_{spatial}$ enforces spatial consistency.

The warmup schedule is defined as:

$$w(t) = \begin{cases} 0 & t < t_{start} \\ \frac{t-t_{start}}{t_{warmup}} & t_{start} \leq t < t_{start} + t_{warmup} \\ 1 & t \geq t_{start} + t_{warmup} \end{cases} \quad (12)$$

where t is the current epoch, t_{start} is the epoch to begin spatial loss, and t_{warmup} is the number of warmup epochs.

E.2 Segmentation-Based Spatial Loss

The segmentation-based approach explicitly models cell nuclei through instance segmentation, enabling fine-grained morphological analysis.

E.2.1 Nuclear Segmentation

We apply a pretrained Cellpose model to segment individual nuclei. For an image $\mathbf{X}$, Cellpose produces an instance segmentation mask $\mathbf{M} \in \mathbb{Z}^{H \times W}$ where $\mathbf{M}(i, j) = n$ indicates pixel (i, j) belongs to nucleus n .

The set of detected nuclei is:

$$\mathcal{N} = \{n_1, n_2, \dots, n_K\} \quad (13)$$

For each nucleus $n \in \mathcal{N}$, we extract its centroid:

$$\mathbf{c}_n = (\bar{i}_n, \bar{j}_n) = \left(\frac{1}{|\mathcal{R}_n|} \sum_{(i,j) \in \mathcal{R}_n} i, \frac{1}{|\mathcal{R}_n|} \sum_{(i,j) \in \mathcal{R}_n} j \right) \quad (14)$$

where $\mathcal{R}_n = \{(i, j) : \mathbf{M}(i, j) = n\}$ is the region of nucleus n .

E.2.2 Morphological Feature Extraction

For each nucleus, we compute morphological features:

1. **Area:** $A_n = |\mathcal{R}_n|$
2. **Perimeter:** $P_n = \sum_{(i,j) \in \mathcal{R}_n} \mathbb{1}[\exists (i', j') \in \mathcal{N}_8(i, j) : \mathbf{M}(i', j') \neq n]$
3. **Circularity:** $C_n = \frac{4\pi A_n}{P_n^2}$
4. **Eccentricity:** Computed from the eigenvalues $\lambda_1 \geq \lambda_2$ of the covariance matrix of pixel positions:

$$E_n = \sqrt{1 - \frac{\lambda_2}{\lambda_1}} \quad (15)$$

5. **Solidity:** $S_n = \frac{A_n}{A_{convex_hull}(n)}$

The morphological feature vector for nucleus n is:

$$\mathbf{f}_{morph}(n) = [A_n, P_n, C_n, E_n, S_n] \quad (16)$$

E.2.3 Nuclear Spatial Graph

We construct a k-nearest neighbor graph in the space of nuclear centroids:

$$\mathcal{G}_{nuclei} = (\mathcal{N}, \mathcal{E}_{nuclei}) \quad (17)$$

where edges connect spatially proximate nuclei:

$$\mathcal{E}_{nuclei} = \{(n_i, n_j) : j \in \text{top-}k \text{ nearest neighbors of } i\} \quad (18)$$

based on centroid distances $d(\mathbf{c}_{n_i}, \mathbf{c}_{n_j}) = \|\mathbf{c}_{n_i} - \mathbf{c}_{n_j}\|_2$.

E.2.4 Morphological Consistency Loss

For matched nuclei between real and generated images (matched by spatial proximity of centroids), we penalize morphological differences:

$$\mathcal{L}_{morph} = \frac{1}{|\mathcal{N}_{matched}|} \sum_{n \in \mathcal{N}_{matched}} \|\mathbf{f}_{morph}^{real}(n) - \mathbf{f}_{morph}^{gen}(n)\|_2 \quad (19)$$

where $\mathcal{N}_{matched}$ is the set of matched nuclei pairs.

E.2.5 Spatial Arrangement Loss

We enforce consistency in the spatial arrangement of neighboring nuclei. For each nucleus n_i and its neighbors $\mathcal{N}_k(n_i)$:

$$\mathcal{L}_{arrangement} = \frac{1}{|\mathcal{N}|} \sum_{n_i \in \mathcal{N}} \frac{1}{k} \sum_{n_j \in \mathcal{N}_k(n_i)} |d_{real}(\mathbf{c}_{n_i}, \mathbf{c}_{n_j}) - d_{gen}(\mathbf{c}_{n_i}, \mathbf{c}_{n_j})| \quad (20)$$

This term encourages similar inter-nuclear distances in generated images.

E.2.6 Nuclear Density Consistency

We compare local nuclear density using kernel density estimation. The nuclear density at location (i, j) is:

$$\rho(i, j) = \sum_{n \in \mathcal{N}} \mathcal{K}_h(\|\mathbf{c}_n - (i, j)\|_2) \quad (21)$$

where $\mathcal{K}_h$ is a Gaussian kernel with bandwidth h :

$$\mathcal{K}_h(d) = \frac{1}{\sqrt{2\pi}h^2} \exp\left(-\frac{d^2}{2h^2}\right) \quad (22)$$

The density consistency loss is:

$$\mathcal{L}_{density} = \frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W |\rho_{real}(i, j) - \rho_{gen}(i, j)| \quad (23)$$

E.2.7 Combined Segmentation-Based Loss

$$\mathcal{L}_{spatial}^{segment} = \beta_{morph} \cdot \mathcal{L}_{morph} + \beta_{arr} \cdot \mathcal{L}_{arrangement} + \beta_{dens} \cdot \mathcal{L}_{density} \quad (24)$$

where $\beta_{morph}, \beta_{arr}, \beta_{dens}$ are weighting hyperparameters.

E.3 Gradient-Based Spatial Loss

The gradient-based approach captures local texture patterns through image derivatives and neighborhood similarity. For each spatial location (i, j) , we extract a local patch and compute its features.

E.3.1 Gradient Feature Extraction

We compute spatial gradients using Sobel operators:

$$\mathbf{G}_x = \mathbf{X} * \mathbf{K}_x, \quad \mathbf{G}_y = \mathbf{X} * \mathbf{K}_y \quad (25)$$

where $*$ denotes convolution and $\mathbf{K}_x, \mathbf{K}_y$ are Sobel kernels:

$$\mathbf{K}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \mathbf{K}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (26)$$

The gradient magnitude and orientation at each pixel are:

$$M(i, j) = \sqrt{G_x(i, j)^2 + G_y(i, j)^2}, \quad \theta(i, j) = \arctan\left(\frac{G_y(i, j)}{G_x(i, j)}\right) \quad (27)$$

E.3.2 Texture Feature Extraction

We extract local texture features using patch statistics. For a patch centered at (i, j) with radius r :

$$\mathcal{P}_{i,j} = \{\mathbf{X}(i', j') : |i' - i| \leq r, |j' - j| \leq r\} \quad (28)$$

The texture feature vector $\mathbf{f}_{texture}(i, j)$ includes:

$$\mathbf{f}_{texture}(i, j) = [\mu(\mathcal{P}_{i,j}), \sigma(\mathcal{P}_{i,j}), s(\mathcal{P}_{i,j}), k(\mathcal{P}_{i,j})] \quad (29)$$

where μ, σ, s, k are the mean, standard deviation, skewness, and kurtosis of the patch.

E.3.3 Spatial Graph Construction

We construct a k -nearest neighbor graph in spatial coordinates. For a downsampled grid of locations $\{(i_1, j_1), \dots, (i_N, j_N)\}$, we find the k nearest neighbors for each location based on Euclidean distance:

$$\mathcal{N}_k(i_m, j_m) = \{(i_n, j_n) : d((i_m, j_m), (i_n, j_n)) \in \text{top-}k \text{ smallest}\} \quad (30)$$

where $d((i_m, j_m), (i_n, j_n)) = \sqrt{(i_m - i_n)^2 + (j_m - j_n)^2}$.

E.3.4 Gradient-Based Spatial Loss

The gradient component of the spatial loss compares gradient patterns between spatially neighboring locations:

$$\mathcal{L}_{gradient} = \frac{1}{N} \sum_{m=1}^N \frac{1}{k} \sum_{(i_n, j_n) \in \mathcal{N}_k(i_m, j_m)} \|\mathbf{G}_{real}(i_m, j_m) - \mathbf{G}_{gen}(i_m, j_m) - (\mathbf{G}_{real}(i_n, j_n) - \mathbf{G}_{gen}(i_n, j_n))\|_2 \quad (31)$$

where $\mathbf{G} = [G_x, G_y]$ is the gradient vector. This formulation enforces that gradient differences between neighbors should be similar in real and generated images.

E.3.5 Texture-Based Spatial Loss

The texture component compares local texture statistics:

$$\mathcal{L}_{texture} = \frac{1}{N} \sum_{m=1}^N \frac{1}{k} \sum_{(i_n, j_n) \in \mathcal{N}_k(i_m, j_m)} \|\mathbf{f}_{texture}^{real}(i_m, j_m) - \mathbf{f}_{texture}^{gen}(i_m, j_m)\|_2 \quad (32)$$

E.3.6 Combined Simple Spatial Loss

$$\mathcal{L}_{spatial}^{simple} = \alpha_{grad} \cdot \mathcal{L}_{gradient} + \alpha_{tex} \cdot \mathcal{L}_{texture} \quad (33)$$

where α_{grad} and α_{tex} are weighting factors (typically $\alpha_{grad} = 1.0$, $\alpha_{tex} = 0.5$).

E.4 Implementation Details

Both methods use:

- **k-nearest neighbors:** $k = 5$
- **Spatial loss weight:** $\lambda_s = 0.1$
- **Warmup epochs:** 5 epochs with linear ramp-up
- **Activation threshold:** Spatial loss begins at 70% of total epochs or when validation loss drops below a predetermined threshold

The gradient-based method is computed at every training step with negligible overhead ($\sim 5\%$ increase in training time), while the segmentation-based method uses cached segmentation masks updated every few epochs to balance accuracy and computational cost.

F Data Availability and License

We used the following 10x Xenium demo data:

- **Dataset 1 (Xenium C1):**
Human Skin Preview Data (Xenium Human Skin Gene Expression Panel), In Situ Gene Expression dataset analyzed using Xenium Onboard Analysis 1.6.0, 10x Genomics, (2023-09-19).
- **Dataset 2 (Xenium C2):**
Human Skin Preview Data (Xenium Human Skin Gene Expression Panel with Custom Add-On), In Situ Gene Expression dataset analyzed using Xenium Onboard Analysis 1.7.0, 10x Genomics, (2023-12-08).
- **Dataset 3 (Xenium P1):**
Preview Data: FFPE Human Skin Primary Dermal Melanoma with 5K Human Pan Tissue and Pathways Panel, In Situ Gene Expression dataset analyzed using Xenium Onboard Analysis 3.0.0, 10x Genomics, (2024-08-01).

These datasets are licensed under the Creative Commons Attribution 4.0 International (CC BY 4.0) license, as indicated in the dataset documentation.

HEST-1k: The dataset is public available on HuggingFace. The dataset is distributed under the Attribution-NonCommercial-ShareAlike 4.0 International license (CC BY-NC-SA 4.0 Deed).

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: Yes all the claims made in the abstract and introductions are accurately reflected to the best of our knowledge.

Guidelines:

- The answer NA 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 NA 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: Yes we discuss the limitations of the work in the Discussion section.

Guidelines:

- The answer NA 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: [\[NA\]](#)

Justification: Our method has no theoretical results and we make no assumptions and provide no mathematical proofs. All the results and methods are purely application based.

Guidelines:

- The answer NA 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: Yes we have disclosed all the information needed to reproduce the main experimental results. We will open source the code and pre-processed datasets after the double blind review.

Guidelines:

- The answer NA 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: Yes we provide the links to the original datasets used. We will open source the pre-processed data for reproducibility after double blind review.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: Yes we provide all details regarding the data splits, hyperparameters, optimizer type and their reason. We will open source the code for reproducibility after double blind review.

Guidelines:

- The answer NA 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: Yes we provide the mean and standard deviation of all the defined metrics for all experiments.

Guidelines:

- The answer NA 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: Yes we provide all the details about batch size, GPU specifications, VRAM requirements, time of execution, and epoch numbers for all experiments.

Guidelines:

- The answer NA 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: No code of ethics have been violated in our methods, data usage, and experiments.

Guidelines:

- The answer NA 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: Yes the paper discuss the positive societal impacts of our methods specifically in the bioscience and cancer research domain.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: None of our methods pose any risk to society and have no potential of misuse.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: All previous methods have been extensively and explicitly cited and given proper credit.

Guidelines:

- The answer NA 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: All the original code we wrote has been well documented and will be made open source after double blind review.

Guidelines:

- The answer NA 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: [\[NA\]](#)

Justification: Our work does not involve crowdsourcing and research with human subjects.

Guidelines:

- The answer NA 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: [\[NA\]](#)

Justification: Our work does not involve crowdsourcing and research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLMs were only used for editing and grammatical improvements of the written text. LLMs were not used for any original methodology implementations.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.