Projet de Fin d'Études

2025 – 2026 [R|E]

REPORT RESEARCH PROJECT

Generative AI for staining transfer

Master degree in Computer Science, UBINET

Student:
SPATHIS Giuseppe

Supervisor
DESCOMBES Xavier

Université Nice Sophia Antipolis

POLYTECH NICE-SOPHIA

Abstract

Digital pathology consists of scanning histology slides into Whole Slide Images (WSIs) and analyzing them with machine learning. In routine practice, the standard Hematoxylin and Eosin (H&E) stain is available for most patients, while Immunohistochemistry (IHC) markers are usually acquired only when needed, because they require extra tissue sections, reagents, and laboratory time. Moreover, H&E and IHC are typically applied to adjacent (serial) sections, so perfectly aligned image pairs are rarely available.

This project studies virtual staining: generating an IHC-like image from an H&E image with a generative model, aiming to obtain synthetic images that look realistic while preserving tissue morphology. The work is carried out with CHU Nice and the Morpheme team (INRIA/I3S/IBV) on kidney cancer data, using H&E tiles from Nice as inputs and CK7 IHC tiles from the Lyon cohort as targets.We adopt PixCell, a diffusion-based (iterative denoising) foundation model for pathology, and adapt it to the target domain with a lightweight LoRA adapter trained on unpaired target IHC images while keeping the PixCell backbone frozen. Because the setting is unpaired, evaluation relies on distribution-level quantitative metrics and qualitative inspection. Beyond visual plausibility, the longer-term motivation is to provide additional synthetic CK7-like inputs that could support downstream computational pathology models (e.g., tumour classification) when real IHC is limited.

Contents

1. General Project Description

1.1. Framework/Context

The project is situated within computational pathology, where Whole Slide Images (WSIs) enable large-scale analysis of histology slices with machine learning. This research is part of a collaboration between CHU Nice and the Morpheme team, and focuses on kidney cancer histopathology to support diagnosis and prognosis after surgery.

In routine workflows, Hematoxylin and Eosin (H&E) provides the primary morphological view of tissue, while Immunohistochemistry (IHC) markers provide complementary molecular information and are typically acquired only when clinically needed. This motivates virtual staining: a computational approach that predicts an IHC-like appearance from an available H&E image, with the aim of generating realistic synthetic targets while preserving underlying tissue morphology.

What virtual staining means in practice. Virtual staining does not replace the wet-lab process for a given physical slide; instead, it is a computational method that predicts how a tissue section would look if it were stained using a different protocol. In other words, it is a stain-to-stain (or modality-to-modality) translation problem: the input can be H&E, but it can also be another routine stain, and the target can be any histochemical stain. In routine pathology, different stains are typically applied to different serial sections cut from the same tissue block, because staining protocols chemically process the section and cannot be freely repeated on the exact same tissue slice. As a result, the practical scenario is often that one section is stained with H&E (to assess morphology), while adjacent sections are stained with additional protocols depending on the diagnostic question. These can include IHC markers (e.g., CK7, Ki-67, HER2) to detect specific proteins, but also special stains such as PAS (highlighting basement membranes and glycogen), Masson's trichrome (highlighting collagen and fibrosis), or reticulin/silver stains (highlighting reticular fibers), among others. Virtual staining aims to infer, from an available source stain, the kind of signal and appearance that would be observed under another stain, potentially reducing the need to request additional laboratory stains in situations where extra sections are limited or where additional staining is costly and time-consuming.

1.2. Motivations

The primary motivation for this work is the prohibitive cost and logistical complexity associated with physical multiplexed staining. Obtaining IHC data is expensive, time-consuming, and requires the consumption of valuable patient tissue samples. In many clinical and research scenarios, only H&E slides are readily available, which limits the depth of possible molecular analysis. Virtual staining offers a computational alternative by generating IHC markers directly from standard H&E slides. This capability allows researchers and clinicians to "re-stain" existing slides without wet-lab procedures, thereby saving time and resources. Furthermore, this technology can support data augmentation and enable training of more robust downstream models.

1.3. Challenges

The main challenge of virtual staining is that the problem is rarely paired in real clinical practice. In principle, the cleanest supervision would require the same tissue section observed twice: first stained with H&E and then stained with the target IHC marker, so that the two images are perfectly aligned pixel-by-pixel. In practice, this is not feasible because staining protocols are wet-lab procedures that chemically process the tissue and typically consume or alter the section; therefore, pathologists usually produce serial sections (adjacent thin cuts) from the same tissue block and apply different stains on different sections. As a consequence, H&E and IHC images may correspond to nearby but not identical tissue, and perfect alignment is not guaranteed. More generally, hospitals do not systematically produce both H&E and IHC for every case, because additional IHC stains require extra reagents, technician time, and extra tissue sections; IHC is usually requested only when needed to refine the diagnosis. This makes the training setting predominantly unpaired and strongly motivates methods that can learn from separate H&E and IHC collections.

A second major difficulty is domain shift. Even when the organ is the same (kidney) and the task is the same (H&E→IHC), slides can look different across hospitals due to differences in fixation, staining protocols, scanners, and color calibration. A model pre-trained on a large multi-center dataset may therefore perform well on some domains but degrade on a new hospital cohort unless it is adapted. This practical issue is central for deploying foundation models on proprietary data.

Finally, histopathology images have an extremely high-resolution. Whole Slide Images are gigapixel-scale, so training and inference must be performed on smaller tiles while preserving fine-grained structures (cell morphology, tissue architecture) and the correct spatial distribution of the marker signal.

1.4. Goals

The goal of this project is to enable virtual staining from H&E to IHC CK7 in order to generate realistic synthetic IHC-like images that remain faithful to the underlying tissue morphology. Beyond visual plausibility, the intended use is to support downstream computational pathology pipelines—for instance, providing synthetic CK7 inputs to automated models (e.g., tumor classification)—especially in settings where real IHC is limited, delayed, or costly.

1.5. Medical Background: Histopathology and Staining

Histopathology is the medical discipline that studies diseases by examining tissue under a microscope. In oncology, it is a cornerstone for diagnosing tumours and for estimating prognosis, i.e., the expected clinical course and outcome for the patient, and for designing a proper treatment. In practice, tissue is collected during surgery or biopsy, processed in the laboratory, embedded in paraffin, cut into very thin sections (a few micrometres), and mounted on glass slides for microscopic examination. Because most biological structures are nearly transparent, slides are usually coloured with stains to make cells and tissue architecture visible and interpretable.

The most common reference stain is Hematoxylin and Eosin (H&E). Hematoxylin colours

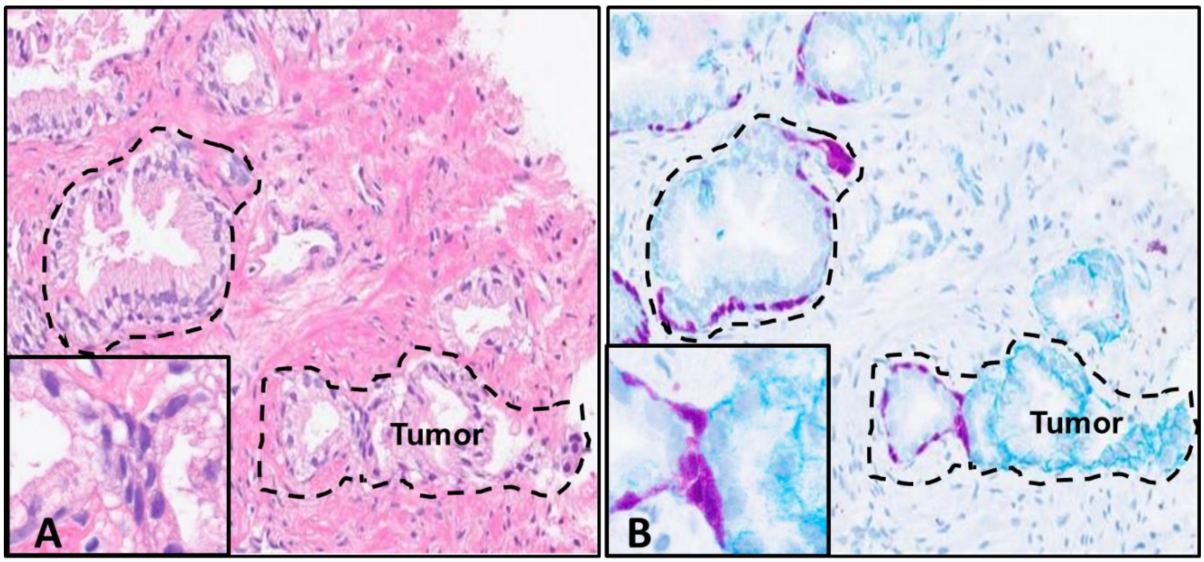


Figure 1: Example of renal tumour tissue stained with Hematoxylin and Eosin (H&E). Dashed contours indicate tumour regions.

nuclei in blue/purple, while eosin colours cytoplasm and extracellular structures in pink, as illustrated in Fig. 1. H&E provides a broad structural overview of the tissue and is typically the first step for diagnosis; however, morphology alone can be ambiguous when different renal tumour subtypes share similar architecture. This is particularly relevant in renal cell carcinoma, which includes several subtypes (e.g., clear cell, papillary, chromophobe, oncocytoma) that can be difficult to separate using morphology only.

To obtain more specific biological information, pathologists often use Immunohistochemistry (IHC). IHC is a laboratory technique used to visualize the presence of a specific protein directly inside a tissue section. The protein of interest is called the antigen, and the experiment starts by applying a primary antibody designed to bind specifically to that antigen. Because the primary antibody itself is not easily visible under the microscope, a detection system is added: most commonly, a secondary antibody (which recognizes the primary antibody) is coupled to an enzyme. When a chromogenic substrate such as DAB (3,3'-diaminobenzidine) is subsequently applied, the enzyme converts it into an insoluble brown precipitate that deposits exactly where the target protein is located. As a result, cells expressing the antigen become brown in the expected cellular compartment, while cells that do not express the antigen do not develop the brown signal. To preserve tissue readability and provide morphological context, slides are almost always counterstained with hematoxylin, which colors nuclei blue. Therefore, "marker-negative" cells (no detectable antigen expression) appear mostly blue because only the counterstain is visible, whereas "marker-positive" cells show a clear brown DAB signal on top of the blue nuclear background, as illustrated in Fig. 2.

In this project, the IHC slides from Lyon use CK7 (Cytokeratin 7), a cytoplasmic marker (i.e., it is expected to appear mainly in the cytoplasm rather than in the nucleus). CK7 was selected in the reference study because cytoplasmic staining tends to cover larger areas and is therefore easier to quantify than very focal patterns; in CK7 IHC, positive regions appear brown due to DAB, while negative regions remain primarily blue because of the counterstain.

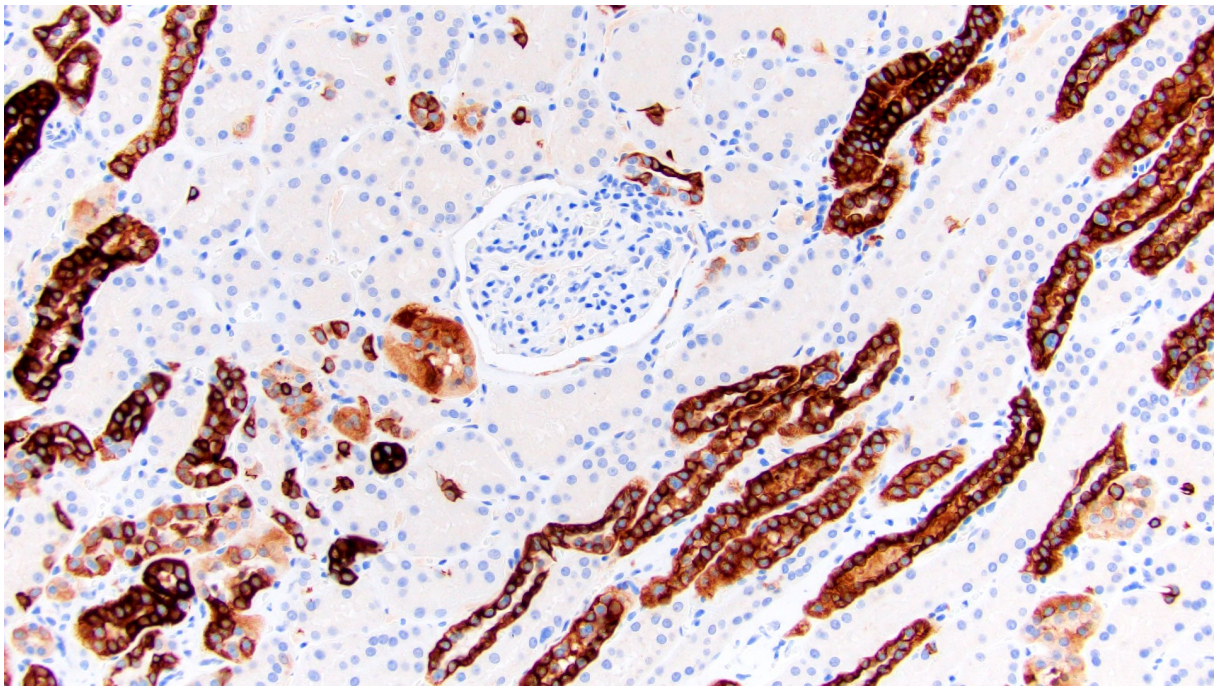


Figure 2: Example of CK7 immunohistochemistry (IHC) on renal tissue. Positive CK7 expression is visualized by a brown DAB signal (typically cytoplasmic/membranous), while nuclei remain blue due to hematoxylin counterstaining.

Finally, modern pathology increasingly relies on digitised slides. Instead of looking through a microscope, slides are scanned into Whole Slide Images (WSIs), which are extremely large multi-resolution images that can be explored at different magnifications, similarly to a microscope. Because WSIs are gigapixel-scale, computational pipelines rarely process them as a single image; instead, they are subdivided into smaller regions called tiles or patches. For downstream machine learning, it is also common to annotate tissue regions (e.g., tumour vs non-tumour) using dedicated software; these annotations can be exported as polygon files (e.g., GeoJSON) and converted into masks, which encode which pixels correspond to tumour, non-tumour, or background.

1.6. Declaration of AI tool usage.

In accordance with the UniCA AI Usage Charter, I declare that AI tools were used in a supportive role during this project and that all content included in this report was critically reviewed by me, with claims and sources verified and cited where applicable. Regarding the writing of the report, I used ChatGPT 5.2 in the initial phase to help outline a coherent structure (sections and subsections) and, later, to improve English clarity by correcting grammatical errors and refining style toward a more academic register. For the implementation work, I used GitHub Copilot as assistance while writing code, and in the most critical moments where I was blocked I used ChatGPT 5.2 Deep Thinking to help reason through debugging steps and implementation choices. AI assistance was also used to support the analysis and interpretation of experimental results as a way to check whether the experimentation was progressing in a consistent direction; however, the final interpretation, validation, and presentation of results are my responsibility.

2. State of the Art

During the preliminary phase of this project, three diffusion-based approaches were studied for virtual staining: VIMs [1], His-MMDM [3], and PixCell [4]. The goal is to translate a standard histology stain (H&E) into an immunohistochemistry (IHC) marker, while preserving tissue structure and enabling high-resolution generation. Since each method makes different trade-offs (data requirements, conditioning strategy, computational cost, and ease of adaptation), comparing them helps justify the final choice adopted in this work.

Why diffusion models for stain translation. Historically, most virtual staining systems in computational pathology were built with GANs or, less frequently, VAEs. GAN-based approaches can translate stains quickly at inference time, but they are often engineered for a specific source→target pair and can be challenging to train robustly without task-specific losses and careful stabilization tricks. VAE-based methods are typically easier to optimize, yet their reconstructions can be overly smooth and may wash out fine-grained histological details that matter at the cellular level. More recently, the field has started to adopt diffusion models for stain translation, motivated by their strong performance in conditional image generation and their state-of-the-art visual fidelity and texture realism in many image translation settings, albeit at higher computational cost. In histopathology, this trade-off is often acceptable because diffusion models better preserve morphology while synthesizing plausible marker-specific signal under realistic constraints such as imperfect pairing.

VIMs. VIMs (Virtual Immunohistochemistry Multiplex Staining) is designed to generate different IHC markers from the same H&E image using text prompts instead of training one separate model per marker. In practice, text is encoded with a CLIP-based prompt encoder (CLIP is a pretrained vision–language model used here to embed text prompts) and used to steer the generation toward the requested marker; this is attractive because it reduces the need for complex multiplex datasets where multiple stains must be perfectly aligned on the same tissue section. Another strong point is speed: classical diffusion models generate images through many denoising steps, whereas VIMs is engineered as a single-step diffusion model and uses adversarial objectives to retain high visual fidelity while keeping inference time close to GAN-like generation. The method was validated on a small but carefully prepared dataset where slides were first stained with H&E and then re-stained and re-scanned with specific antibodies, enabling accurate alignment and evaluation.

However, for this project the main limitation is the conditioning mechanism: the project requires transferring visual structural information from H&E to IHC with minimal ambiguity, while text prompts can be less constrained than image-based conditioning when the objective is diagnostic faithfulness. In addition, the main validation domain of the original model is not renal pathology, so applying it to kidney tumour slides would likely require non-trivial re-training and careful domain adaptation. Finally, the unavailability of the source code further reduces its practicability as a primary candidate for hands-on experiments.

His-MMDM. His-MMDM is a conditional diffusion framework designed for multi-domain translation, meaning it can translate among many categories rather than learning a single fixed pair (e.g., not only H&E→one marker, but potentially H&E→multiple IHC markers, or even tumour-type transformations). In addition to categorical domains (such as stain type or tissue preparation protocol), it can also accept omics information (genomics or transcriptomics) as a condition. In simple terms, this means the model can be instructed not only to "change stain" but also to "simulate how tissue appearance could change under a given molecular profile", which is scientifically interesting even if it is beyond the scope of this project. Architecturally, it follows the standard diffusion paradigm: a forward process gradually adds noise to an image, and a backward process uses a U-Net (a convolutional encoder–decoder architecture widely used for image-to-image tasks) to iteratively denoise while being conditioned on the desired target domain.

A major strength of His-MMDM is that it is coupled with a clear methodology to verify biological correctness, not just visual realism. In IHC, the diagnostic signal is often the brown DAB deposit; the authors quantify it via color deconvolution and verify that the positive signal appears in the expected cellular compartment (nucleus vs cytoplasm) depending on the marker. This is valuable because it links generation quality to medically meaningful constraints, rather than relying exclusively on generic perceptual metrics.

The key drawback is computational cost: because the method relies on iterative denoising, inference can be slow (minutes per batch in the reported setting), which is a practical bottleneck for rapid experimentation and high-throughput virtual staining. Moreover, the most detailed reported results concerning virtual staining experiments and datasets are centered on brain tumours, so applying it directly to kidney cancer would require a substantial domain shift and engineering effort within the project timeline.

PixCell. PixCell is a diffusion-based (iterative denoising) foundation model trained at large scale on histopathology patches (PanCan-30M), explicitly including kidney tissue, which makes it a strong candidate for our renal setting. It is implemented as a latent diffusion model, i.e., diffusion is performed in a compressed latent space rather than directly in RGB pixels, and uses a Diffusion Transformer (DiT) denoiser (a transformer network trained to predict and remove noise). PixCell supports generation up to 1024×1024, a resolution that is relevant in pathology where clinically meaningful patterns span multiple spatial scales. Unlike text-conditioned diffusion models, PixCell is conditioned on image embeddings extracted by a pathology encoder (UNI-2h, a self-supervised model that maps patches to feature vectors), so the conditioning signal is directly derived from the input image and is better suited to structure-preserving translation. In addition, PixCell can be efficiently adapted through LoRA fine-tuning (training small low-rank adapters while keeping the backbone frozen) on unpaired IHC data, which is important under real hospital domain shift. For these reasons—renal coverage in pretraining, high-resolution generation, image-based conditioning, and practical target-domain adaptation—PixCell was selected as the backbone for this project.

3. PixCell in Detail

PixCell is a generative foundation model for digital pathology built on the Latent Diffusion Model (LDM) paradigm, meaning that generation is performed in the latent space of a Variational Autoencoder (VAE), i.e., a model that compresses an image into a compact representation and reconstructs it back to RGB. Instead of generating images directly in pixel space, PixCell operates in the compressed latent space produced by the VAE, following the same high-level strategy as Stable Diffusion. Concretely, PixCell uses the Stable Diffusion 3 VAE to encode a histopathology patch into latent features and then applies a diffusion-based generative model in that latent space. This design is motivated by the extreme size and detail of pathology images: working in latent space reduces computational cost while preserving clinically relevant morphology, provided that the VAE reconstruction quality is sufficiently high for histology.

A key architectural choice is that the denoising network is a Diffusion Transformer (DiT), adapted from PixArt-σ, rather than a U-Net. In practice, the transformer backbone is used to model global structure and long-range dependencies that are common in histology (e.g., repeated glandular patterns, stromal organization), while the latent representation keeps the computational budget manageable.

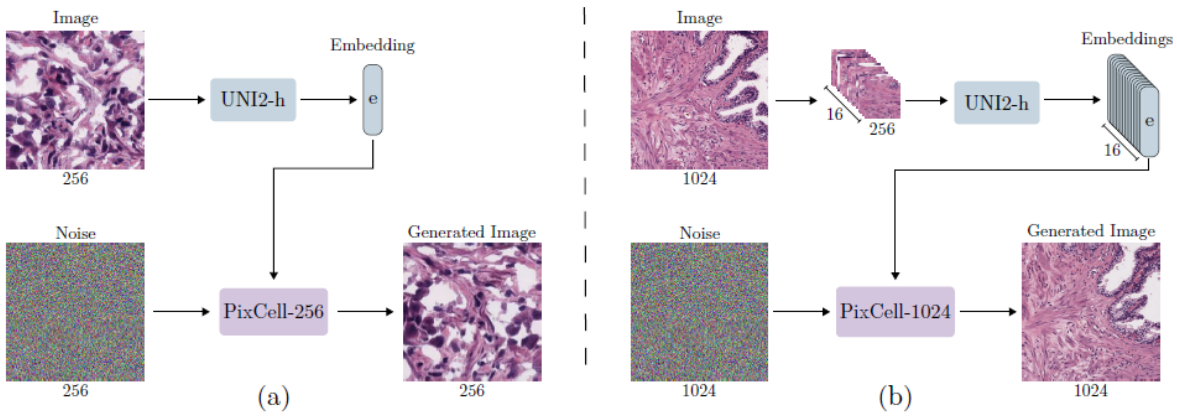


Figure 3: Global overview of PixCell multi-scale conditioning and generation. Panel (a) sketches the 256×256 stage (PixCell-256) conditioned on a UNI-2h embedding (optionally with mask-based control), while panel (b) sketches the 1024×1024 stage (PixCell-1024) used for virtual staining with spatial UNI-2h conditioning and stain adaptation.

Conditioning mechanism (UNI-2h embeddings). Unlike classical text-to-image diffusion models that rely on textual prompts, PixCell is conditioned on UNI-2h embeddings, i.e., feature vectors extracted from a self-supervised pathology image encoder and used as a compact description of the input image. In practice, the input patch is first encoded into UNI-2h embeddings that provide the conditioning signal, then the Diffusion Transformer (DiT) denoises in latent space while being guided by these embeddings, and finally the denoised latents are decoded back to RGB by the VAE decoder. Fig. 3 illustrates this conditioning concept at both 256 and 1024 resolutions, it provides the global picture; Fig. 4 and Fig. 5 zoom into the low- and high-resolution stages, respectively.

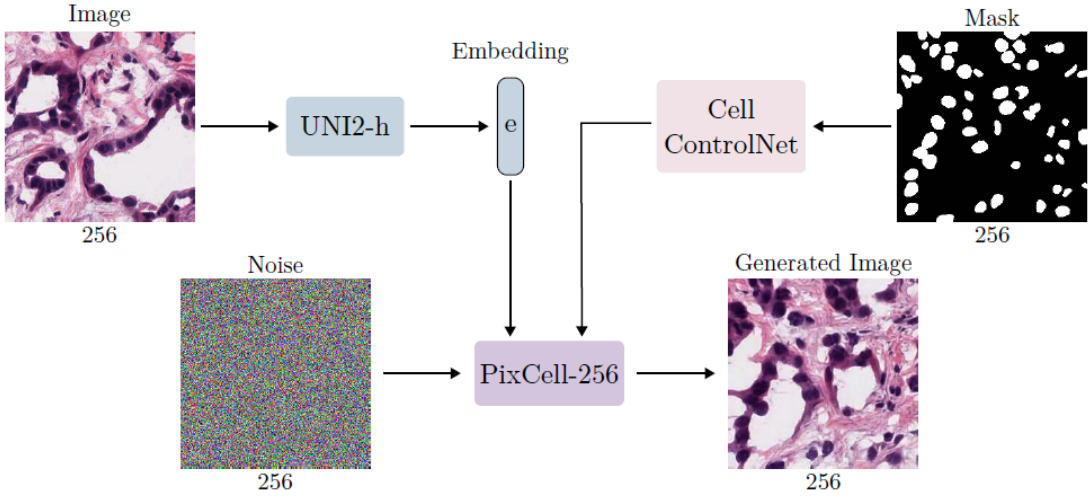


Figure 4: Detailed schematic of PixCell-256 generation at 256×256, conditioned on a UNI-2h embedding. A cell mask can optionally be injected through a Cell ControlNet branch to constrain cellular layout.

PixCell-256 (low-resolution stage). Fig. 4 summarizes the 256×256 variant used as a low-resolution generation stage. A 256×256 input patch is encoded by the UNI-2h encoder into a single conditioning embedding e, which guides the diffusion denoising process starting from noise. Optionally, a binary cell mask can be provided and injected through a dedicated Cell ControlNet branch, so that PixCell-256 is guided not only by the global UNI-2h descriptor but also by explicit cellular layout constraints. The model outputs a generated patch at the same 256×256 resolution.

Progressive training from 256 to 1024. Training a diffusion model directly at 1024×1024 on tens of millions of patches is extremely expensive, so PixCell uses a progressive training strategy that increases resolution in stages. The authors first train a model at low resolution (PixCell-256), then fine-tune at intermediate resolution (512), and finally fine-tune at full resolution (PixCell-1024). Importantly, the conditioning signal also becomes more spatially detailed across stages: at higher resolution the model uses a grid of UNI-2h embeddings (e.g., a 2×2 block at 512, and the full embedding grid at 1024) so that the conditioning carries localized information rather than a single global vector. To reduce training overhead, VAE latents and UNI-2h embeddings are precomputed for the training set, so that the expensive encoders are not repeatedly executed during diffusion training.

PixCell-1024 (high-resolution stage for virtual staining). Fig. 5 illustrates the full-resolution 1024×1024 model used for virtual staining. At this scale, conditioning is spatial: the H&E patch is encoded into a grid of UNI-2h embeddings (here 16 tokens) that carry localized information across the field of view. To perform H&E→IHC translation, these H&E embeddings are mapped into IHC-style embeddings, which are then used to condition PixCell-1024 during denoising from noise. An additional IHC-specific LoRA module is applied within PixCell-1024 to adapt stain-dependent texture and color statistics, improving visual alignment with the target

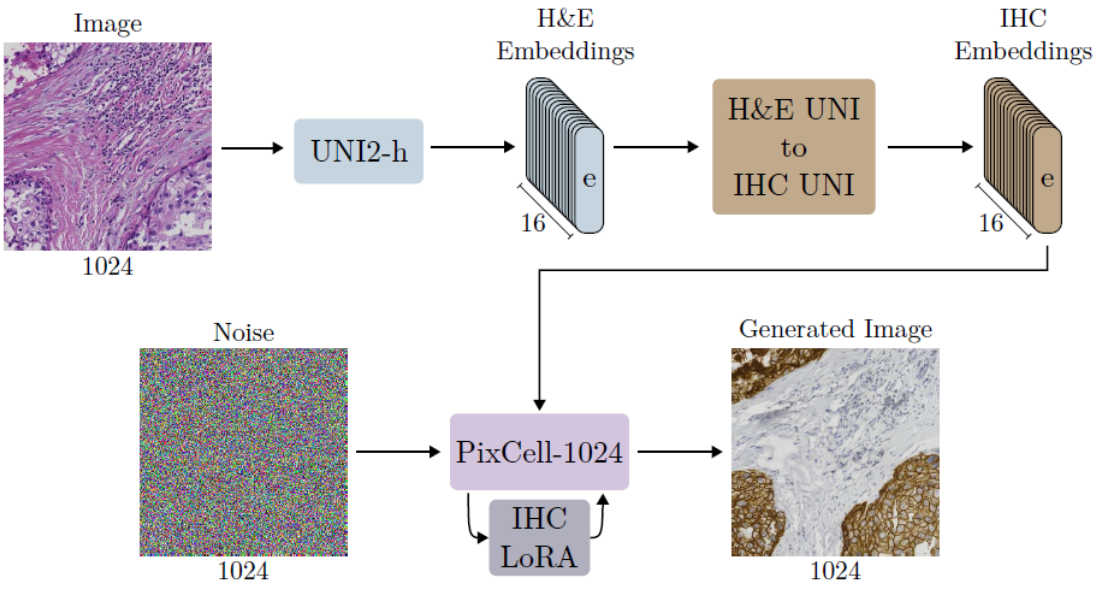


Figure 5: PixCell-1024 virtual staining pipeline at 1024×1024: H&E UNI-2h embeddings are translated into IHC-style embeddings and used to condition PixCell-1024, with an additional LoRA adaptation to match IHC appearance statistics.

IHC domain while preserving morphology driven by the embeddings.

Virtual staining via embedding translation (rectified flow) and LoRA. Although PixCell is trained on H&E patches, it can be driven by embeddings extracted from other visual domains. Feeding an embedding extracted from a real IHC image allows PixCell to generate an image consistent with that stain, indicating that conditioning with UNI-2h embeddings can generalize beyond the exact training domain. Building on this, PixCell performs H&E→IHC translation by operating in embedding space rather than pixel space: a rectified flow model (a lightweight model trained to map H&E embeddings into "IHC-style" embeddings) is trained to map an H&E embedding into an "IHC-style" embedding, which is then used as the conditioning signal for PixCell-1024. In practice, directly swapping embeddings can yield semantically plausible results but with limited diagnostic-quality texture; therefore, a lightweight Low-Rank Adaptation (LoRA) module is fine-tuned on unpaired IHC images to better match stain-specific appearance statistics (textures and color distribution) without full model fine-tuning. This combination (embedding translation + LoRA) is the core mechanism that makes PixCell practically usable for virtual staining with weakly paired or unpaired data.

Evaluation protocol for virtual staining (H&E→IHC). The virtual staining pipeline (H&E→IHC) is evaluated on the MIST-HER2 dataset, which provides roughly aligned (i.e., not pixel-perfect) pairs of H&E patches and HER2 IHC patches for the same tissue regions, where HER2 (Human Epidermal Growth Factor Receptor 2) denotes the immunohistochemical marker targeted by the stain. Quantitative evaluation is performed on the MIST-HER2 test set by comparing generated IHC patches to the corresponding real IHC ground truth using a combination of structural and perceptual metrics: SSIM (structure preservation) and FID/KID/Crop FID

Table 1: Quantitative evaluation on HER2 stain translation (H&E→IHC) on the MIST-HER2 benchmark.

Method	SSIM↑	FID↓	KID↓	Crop FID↓
Baseline [2]	**0.1945**	54.28	16.0	33.22
PixCell-1024 (no LoRA)	0.1880	67.68	19.1	41.54
PixCell-1024	0.1892	**52.32**	**13.4**	**20.87**

(distributional and perceptual distances). The main comparison is reported in Table 1, where the complete PixCell-1024 stain-translation pipeline (rectified-flow embedding translation plus LoRA adaptation) improves perceptual quality over the CycleGAN baseline, achieving lower FID, KID, and Crop FID while keeping SSIM at a comparable level. The same table shows that omitting LoRA substantially degrades perceptual scores, supporting the role of LoRA as a lightweight adaptation step that recovers stain-specific texture and color statistics without full model fine-tuning. Qualitative results are shown in Fig. 6 through side-by-side comparisons including the H&E input, PixCell outputs with and without LoRA, and the IHC ground truth; visually, the LoRA-adapted outputs better match the appearance of HER2 staining while preserving the underlying tissue semantics inferred from H&E.

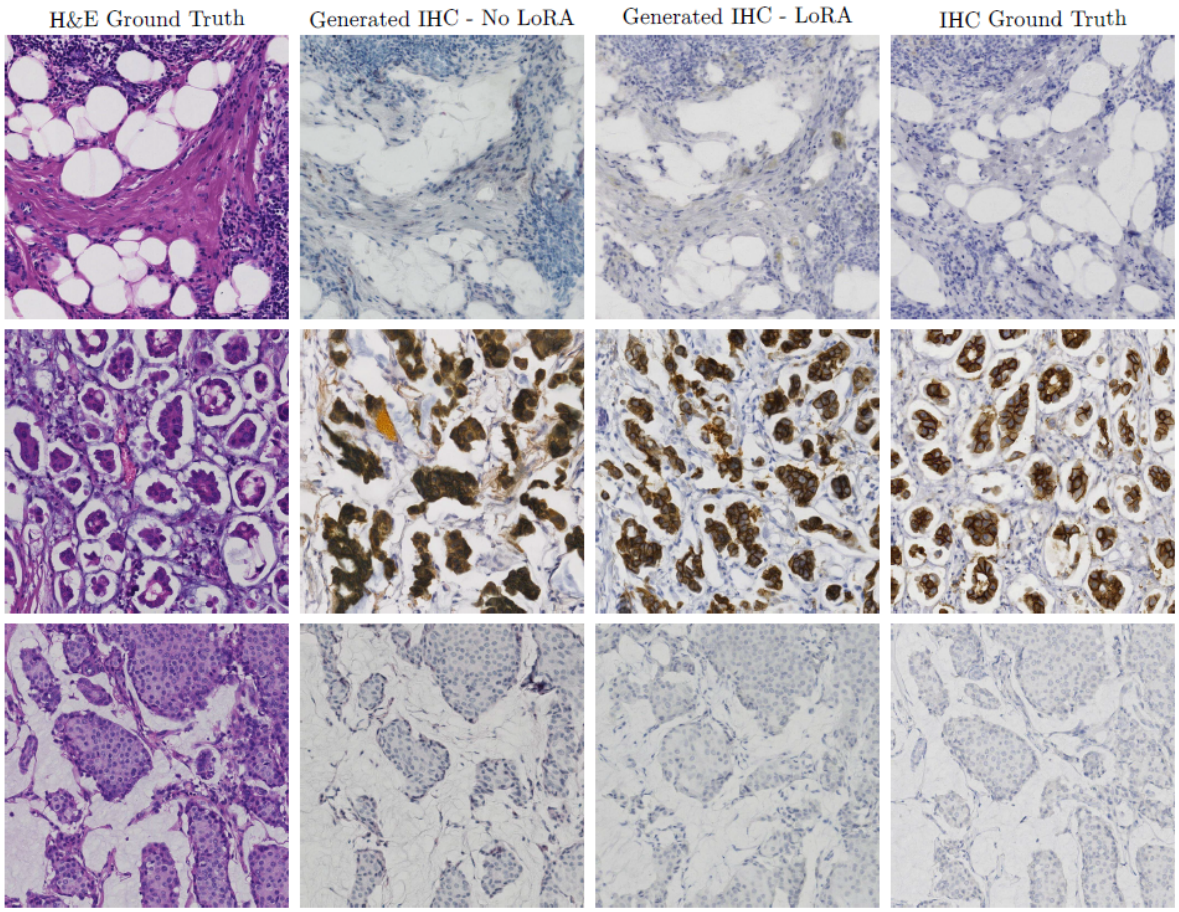


Figure 6: Qualitative examples of H&E→IHC translation with PixCell-1024, comparing generation without and with LoRA adaptation against the IHC ground truth. Since MIST-HER2 pairs are only roughly aligned (adjacent serial sections), local structural differences are expected; the figure mainly highlights the impact of LoRA on IHC-style/marker rendering rather than pixel-level geometric fidelity.

4. Dataset and Preprocessing

The experiments rely on two distinct data sources, corresponding to the input domain (H&E) and the target domain (IHC), both consisting of kidney cancer tissue collected at CHU Nice and in the Lyon cohort. The raw material consists of digitized Whole Slide Images (WSIs), primarily stored as `.svs` files (with related WSI formats also supported by the preprocessing pipeline). The `.svs` format, commonly produced by Aperio/Leica scanners, stores a multi-resolution pyramidal representation of a histology slide, often at gigapixel scale. This property is essential for diagnostic inspection but makes direct training on full slides impractical; consequently, the workflow adopted in this project is patch-based, where WSIs are converted into collections of fixed-size tiles. The resulting tiles are stored as `.npz` files, i.e., compressed NumPy archives containing an array of samples. Each sample stores an extracted RGB tile and, depending on the dataset format, may include additional fields (e.g., labels). Concretely, the preprocessing script reads the WSI pyramid through OpenSlide and extracts tiles using the `DeepZoomGenerator`, producing a collection of fixed-size patches that can be efficiently shuffled and loaded during training. Finally, the extracted samples are serialized using `np.savez_compressed(...)` into

an `object`-typed NumPy array.

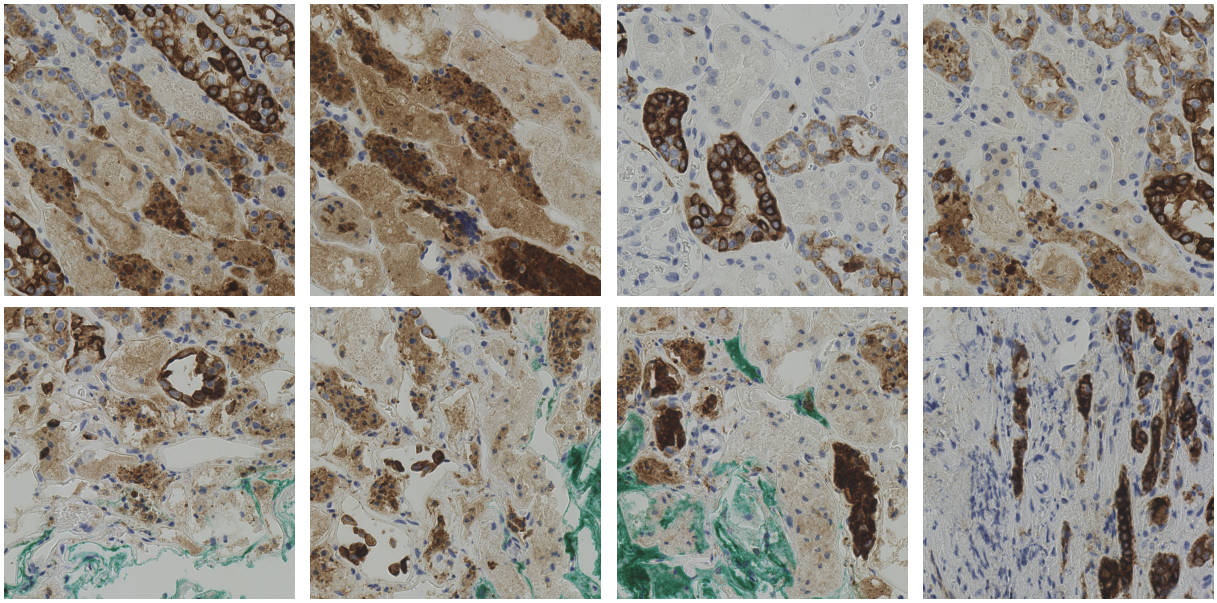
H&E tiles originate from the hospital of Nice, and for this subset the preprocessing step had already been completed prior to this project: the original WSIs were already available as `.npz` archives. The Nice H&E collection comprises 80 `.npz` files for a total of 55,490 tiles, corresponding to an average of ∼694 tiles per file. The naming convention follows the pattern `stain_patientID_replicate.npz`, where the optional suffix `_replicate` indicates multiple acquisitions or multiple analyses for the same patient identifier; for instance, a file such as `he_1811474_3.npz` denotes H&E tiles associated with patient `1811474` and a third replicate, which can correspond to distinct tissue fragments, additional sections, or repeated processing.

IHC tiles originate from the Lyon cohort and required full preprocessing within this work, starting from WSIs and producing `.npz` archives with the same overall strategy (WSI tiling and compressed storage). All Lyon IHC slides are stained with CK7 (Cytokeratin 7), a cytoplasmic marker frequently used in renal tumour characterization and typically visualized with a brown DAB signal when positive; the presence of CK7 provides molecular information that complements H&E morphology and motivates the virtual staining task. The Lyon IHC collection comprises 21 `.npz` files for a total of 37,268 tiles, corresponding to an average of ∼1,775 tiles per file. Filenames follow the same convention as the H&E subset; for example, `ihc_24009747_1.npz` denotes CK7 IHC tiles for patient `24009747`, with suffix `_1` indicating that multiple IHC acquisitions/analyses exist for that same patient.

A key practical issue was the overall quality of the Lyon IHC data. Compared to the Nice H&E tiles, the Lyon slides contained a large proportion of uninformative regions, including extensive white/background areas and low-content fields, which can negatively impact both training and evaluation. To mitigate this problem, the SVS→NPZ conversion applies an aggressive selection strategy during extraction. In particular, the filtering logic discards tiles that are blurry (quantified by a low Laplacian variance computed on the grayscale image), tiles that have weak staining or low chromatic content (quantified by a low fraction of pixels with sufficiently high saturation in HSV space), and tiles dominated by background (quantified by the fraction of near-white pixels). The exact implementation of this informativeness filter is provided in Appendix A.1. After preprocessing, both domains share a uniform representation—fixed-size RGB tiles stored inside `.npz` archives—which enables efficient random access, shuffling, and batched loading on the cluster, and matches the practical requirements of PixCell, especially for high-resolution settings such as 1024×1024 tiles.

Despite the aggressive informativeness filtering, a small number of suboptimal tiles remain in the Lyon CK7 set. In particular, occasional patches can still appear slightly out of focus (reduced sharpness) or exhibit atypical color contamination, most notably a greenish cast that is not expected for standard CK7 IHC (DAB brown with hematoxylin counterstain) and is likely attributable to acquisition or staining artefacts. Representative examples are shown in Fig. 7: the top row illustrates tiles with clear morphology and coherent CK7 signal, whereas the bottom row reports residual low-quality cases. Importantly, these outliers represent only a minor fraction of the overall dataset and do not dominate the post-filtered distribution.

Figure 7: Representative Lyon CK7 IHC tiles after preprocessing.



4.1. Compute Infrastructure (DR-1 Cluster)

All experiments were executed on DR-1, a shared GPU server at Polytech Nice Sophia used for ML research and accessed through Slurm (a job scheduler that allocates GPU resources to submitted jobs) via the LiCO interface (a web UI for job submission and monitoring). Because resources are shared among multiple users, experiment turnaround time was often affected by queueing and GPU availability. In practice, jobs are scheduled as full GPU "slots", which limits fine-grained resource matching: even lightweight runs occupy an entire GPU allocation. Under our inference settings, generating the full synthetic CK7 set for a single checkpoint required approximately **2 days** of wall-clock time on one NVIDIA A40 GPU; to evaluate multiple checkpoints within the project timeline, generations were submitted as independent Slurm jobs and executed in parallel whenever multiple GPUs were available. Hardware details are reported in Appendix A.3.

4.2. Fine-Tuning

To reduce the domain gap between the public PixCell checkpoint and the Lyon cohort for CK7 virtual staining, we adapted PixCell-1024 by training a lightweight LoRA module while keeping the backbone frozen. We did not use the PixCell-256 Cell-ControlNet (mask conditioning) because our objective was $H\&E \rightarrow IHC$ appearance translation with PixCell-1024. LoRA adapters were inserted on the transformer cross-attention projections (the layers where conditioning embeddings are injected into the transformer). The Lyon CK7 dataset was split into 70% training and 30% testing, and optimization was performed only on the training split, with the test split held out for evaluation. To keep training within the available compute/IO budget, we trained the LoRA adapter on a capped subset of the CK7 training tiles rather than the full collection; this may under-represent rare patterns and increase variance, but allowed completing the run within the project timeline. Training ran for 20 epochs with batch size 4 using AdamW and a constant

learning rate of 10^{-4}. Checkpoints were saved at the end of each epoch. Implementation details and the core training loop are reported in Appendix A.2.

4.3. Evaluation

The goal of the evaluation is to assess how closely each checkpoint reproduces the appearance of real CK7 IHC from the Lyon cohort when translating from H&E, in a setting where no pixel-aligned ground truth is available. For this reason, the evaluation combines two complementary viewpoints: a quantitative comparison of real-vs-generated distributions through perceptual distances, and a qualitative inspection of representative samples to identify typical behavior and recurrent failure modes. Before running the full quantitative evaluation, we performed a preliminary quantitative comparison between the two available public configurations (`mist_pr` vs `mist_er`) on a small subset of Lyon tiles using the same distributional metrics adopted in this section; since `mist_pr` consistently achieved lower distances to the real CK7 distribution, we selected `mist_pr` as the flow target for the fine-tuned `lora20` checkpoint in all correlated experiments.

4.3.1. Quantitative Metrics

The Lyon CK7 test set does not provide paired H&E/IHC patches, i.e., there is no pixel- or region-aligned ground truth between input H&E and target CK7. For this reason, SSIM and other pairwise structural similarity metrics cannot be computed in a meaningful way. We therefore adopt distributional/perceptual metrics in the FID/KID family, while stressing that absolute values are not directly comparable to those reported in the PixCell paper because the evaluation is performed in a different domain (kidney CK7), with a different acquisition site, dataset, and sampling/preprocessing pipeline.

For each checkpoint, we compare the distribution of generated CK7 tiles against real CK7 tiles from the Lyon cohort using FID, KID, and CropFID, i.e., distribution-level distances computed on deep feature embeddings rather than pixel-aligned pairs. Concretely, FID (vanilla definition) measures the Fréchet distance between two Gaussians fitted to Inception-v3 feature embeddings extracted from full tiles; CropFID applies the same FID computation on random 256×256 crops (4 crops per tile) to emphasize local texture statistics; and KID (Kernel Inception Distance) measures a kernel two-sample discrepancy (MMD^2) in the same Inception feature space and, in our implementation, it is computed on the same crop stream used for CropFID. Formal definitions and formulas are provided in Appendix A.4–A.6.

In our implementation, embeddings are extracted with an Inception-v3 network through the standard TorchMetrics FID/KID pipeline; therefore, the reported values should be interpreted as relative distances under a fixed protocol rather than as directly comparable across different feature extractors or implementations. All metrics are computed on the same number of images for real and fake domains and with identical deterministic file selection and data-loading logic across methods. To estimate variability, we repeat the crop-based evaluation for 5 different seeds: the seed controls the sampling of the random 256×256 crops (4 crops per 1024×1024 tile), thus affecting CropFID and, because KID is computed on the same crop stream, also the KID estimates. Table 2 reports the mean across the 5 seeds for each checkpoint (lower is better for

Table 2: Quantitative evaluation on Lyon CK7 (means over 5 seeds). Lower is better.

Checkpoint	FID ↓	KID ↓	Crop FID ↓
lora20	**154**	**0.111**	**116**
mist_pr	182	0.117	126
mist_er	190	0.158	161
noFlow	221	0.206	193

all metrics). Overall, the LoRA-adapted CK7 checkpoint (`lora20`) provides the closest match to the real CK7 distribution across all reported metrics, while the `noFlow` ablation performs worst, supporting the role of the flow-based embedding translation and/or stain-specific adaptation for matching CK7 appearance statistics in this setting; among public checkpoints, `mist_pr` is consistently closer to the Lyon CK7 distribution than `mist_er` under our protocol.

4.3.2. Qualitative Evaluation Protocol

Quantitative metrics provide distribution-level signals but do not guarantee histological plausibility, so we complement them with two qualitative figures aimed at highlighting both typical behavior and failure modes across checkpoints. Figure 8 presents 6 randomly sampled examples (one patch per file, same tile index across checkpoints) after tissue-content filtering (at least 60% tissue on the full tile and at least 60% tissue on the central 512×512 crop). In these random cases, `lora20` more consistently produces an IHC-like appearance characterized by a pale background with localized brown DAB-like signal and visible hematoxylin counterstain, while `noFlow` often preserves an H&E-like color palette and contrast (purple/pink dominance) that looks closer to the input modality than to an IHC rendering. In the same rows, the public checkpoints (`mist_pr` and `mist_er`) frequently generate a much denser and more uniform "dot-like" staining pattern and/or stronger global tint, which can make the output look less CK7-like for this dataset and less consistent with the heterogeneous staining intensity typically observed across tissue structures; in addition, some examples show background casts and intensity saturation that reduce the visual separation between stain signal and counterstain. At the same time, while the models often succeed at transferring the global stain appearance (background tone, counterstain, and DAB-like chromogen), fine-grained morphological fidelity is not always preserved: local cellular boundaries and micro-architectural details can be blurred or mildly distorted in the generated IHC. This limitation is expected in our unpaired setting and in an embedding-conditioned diffusion pipeline, where no pixel-level supervision enforces strict geometric correspondence. Figure 9 then focuses on `lora20` by selecting the top-3 best and top-3 worst cases using a proxy-based ranking computed on generated tiles after the same tissue filtering, where the score penalizes global chromatic flatness, repetitive texture artifacts, and blur/over-sharpening tendencies (via HSV saturation variability, shift-based autocorrelation, and Laplacian variance). The best cases illustrate situations where `lora20` yields sharp cellular morphology with controlled background and coherent DAB-like signal distribution, and the side-by-side columns show that the same regions tend to be rendered by `noFlow` in a more H&E-like fashion and by `mist_pr`/`mist_er` with heavier, more uniformly distributed dotting and stronger brown dominance. The worst cases expose typical failure modes: locally unstable staining intensity, occasional structured ar-

tifacts, and regions where the stain appearance becomes less selective and more globally tinted; comparing columns on the same indices suggests that some failures are input-driven (challenging morphologies or low-contrast regions) but that `lora20` still produces outputs that are, overall, visually closer to the Lyon CK7 style than the other checkpoints. These qualitative observations support the quantitative ranking, indicating `lora20` as the most suitable checkpoint in our setting; however, a definitive qualitative assessment of CK7 plausibility and diagnostic validity should be performed with the support of a trained pathologist, since visual similarity alone is not sufficient to establish clinical correctness.

4.3.3. Summary

Overall, the quantitative metrics consistently rank `lora20` as the closest match to the real Lyon CK7 distribution, with `mist_pr` second, `mist_er` third, and the `noFlow` ablation clearly worst, suggesting that stain-specific adaptation and the flow-based embedding translation are important to reproduce CK7 appearance statistics in this setting. The qualitative figures support the same conclusion: `lora20` more often yields an IHC-like rendering with a pale background, localized brown DAB-like signal, and visible counterstain, whereas `noFlow` tends to remain H&E-like and the public checkpoints frequently produce heavier, more uniform dot-like staining and stronger global tints. At the same time, qualitative inspection also reveals input-driven difficult regions and occasional artifacts (e.g., locally unstable intensity or structured repetition), indicating that metric improvements do not guarantee histological plausibility at the slide level. A definitive assessment of CK7 plausibility and diagnostic suitability should therefore be conducted with the support of a trained pathologist, since visual similarity and distributional metrics alone are insufficient to establish clinical correctness.

5. Difficulties Encountered

A first major limitation during this work was the restricted availability of computational resources on the DR-1 cluster. The system provides only eight concurrent job slots, and in practice these slots were frequently saturated. As a consequence, the turnaround time for running experiments was often dominated not by computation itself, but by queueing delays, which in some cases extended to multiple days. This constraint was particularly problematic given the tight timeline of the project. In addition, the DR-1 configuration effectively forces users to allocate large GPUs even for lightweight tasks: even when an experiment requires modest GPU memory or compute, the job still occupies one of the limited high-end GPU slots, accelerating slot exhaustion and reducing overall flexibility. Beyond these structural constraints, the cluster also experienced technical issues that temporarily made it unusable. The combination of limited parallel capacity, inefficient resource matching, and unexpected downtime significantly reduced the number of iterations that could be performed and slowed down the overall experimental cycle.

A second difficulty concerned qualitative evaluation. The initial plan was to complement numerical metrics with a clinician-driven assessment of the generated virtual stains, in order to judge marker plausibility in a medically meaningful way (e.g., localization patterns, tissue structures, and typical failure modes). However, the collaborating physician was unavailable throughout January and February, which made it impossible to carry out the planned expert review within the project timeframe. As a result, the evaluation strategy had to be adapted, relying more heavily on internal qualitative inspection and quantitative measures, with the understanding that these proxies cannot fully replace domain-expert validation.

Finally, data quality posed an additional challenge. The IHC data obtained from the Lyon hospital were of relatively low quality, with artifacts and variability that can negatively impact both training and evaluation. Since virtual staining models are highly sensitive to the statistical properties of the target domain, noisy or inconsistent IHC references can reduce achievable fidelity and may partially explain limitations observed in the generated outputs. In this context, dataset quality is not merely a practical inconvenience but a core factor that conditions the upper bound of model performance and the interpretability of quantitative comparisons.

6. Future Work

Several directions naturally follow from this work. First, it would be valuable to run experiments with the His-MMDM framework and perform a direct comparison with PixCell under the same data and evaluation protocol. Such a study would clarify the trade-off between computational cost, image fidelity, and biological plausibility across different diffusion-based approaches.

Second, a longer-term objective is to design and develop a new virtual staining model tailored to the constraints and needs observed in this project, which would constitute the basis of the upcoming internship work. A key limitation emerging from our qualitative analysis is that, while the model often achieves convincing stain appearance transfer, fine-grained structural details from the H&E input are not always preserved in the generated IHC; as a result, the current outputs should not be considered sufficient for medically grounded validation without expert review and stronger morphology-preservation guarantees. Several complementary technical directions could be explored to mitigate this issue, including increasing the capacity of the latent representation (at the cost of training or re-training parts of the model), decoupling structure and color by operating in alternative color spaces such as Lab (e.g., processing luminance and chrominance components separately), and working in stain-separated space via color deconvolution to process hematoxylin/eosin components independently and better control marker-like signal.

Third, expanding the dataset is essential. Obtaining additional data from other hospitals would improve diversity in acquisition protocols, scanners, and staining procedures, enabling more reliable evaluation of generalization and reducing the risk that results are overly dependent on a single site. In parallel, higher-quality IHC targets would strengthen both training supervision and the credibility of downstream validation.

Additionally, a practical next step would be to broaden the exploration of PixCell by testing the full set of publicly available checkpoints, rather than restricting experiments to `mist_pr` and `mist_er`. Since PixCell has been trained on heterogeneous data sources and domains, different checkpoints may implicitly encode different staining protocols, scanner characteristics, or tissue distributions. Systematically evaluating multiple checkpoints could therefore help identify a source domain that is closer to the Lyon hospital data, potentially improving performance without additional training, simply by selecting a better-matched pretrained initialization. In the current work, experiments were limited to the `mist_pr` and `mist_er` checkpoints due to time constraints, so a more exhaustive checkpoint screening remains an important piece of future work. As a complementary and scalable evaluation signal, one could compute and cache UNI embeddings for both generated IHC tiles and real Lyon IHC tiles, and quantify their similarity in embedding space (e.g., cosine distance, optionally alongside Euclidean distance). While this would not replace histology- or biology-driven validation, it would provide an efficient proxy to compare checkpoints and fine-tuning variants at scale and to support retrieval-style inspection.

Finally, the current work focuses on translation from H&E to a specific IHC setting. Extending virtual staining to multiple stain types and multiple target domains—not only H&E→IHC but also across different stains and markers—would be an important step toward a more general and clinically useful tool. This would also support broader applications such as multi-marker synthesis, comparative analyses across markers, and richer downstream pipelines for diagnosis and prognosis.

7. Conclusion

This project investigated virtual staining for kidney cancer histopathology, with the goal of generating a CK7 immunohistochemistry (IHC)-like appearance from an available H&E input. The work uses internal hospital data collected with CHU Nice and the Morpheme team: H&E tiles from Nice define the source domain, while CK7 IHC tiles from the Lyon cohort define the target domain. We selected PixCell as the generative backbone and adapted it to the Lyon CK7 style with lightweight LoRA fine-tuning. Since no pixel-aligned H&E/CK7 pairs are available in this setting (serial sections and heterogeneous acquisitions), the evaluation necessarily focuses on cohort-level distributional metrics combined with qualitative visual inspection, rather than per-image similarity measures.

Across the cohort, both the quantitative metrics (Table 2) and the qualitative inspection (Figs. 8–9) consistently rank the LoRA-adapted checkpoint (`lora20`) as the best-performing configuration, followed by `mist_pr`, `mist_er`, and the `noFlow` ablation. This supports the conclusion that, under domain shift, lightweight target-domain adaptation remains beneficial and that embedding translation coupled with modest adaptation can partially bridge the gap between a broadly pretrained foundation model and a hospital-specific staining protocol. Qualitatively, `lora20` most often matches the expected CK7 IHC style for the Lyon cohort, whereas `noFlow` remains closer to the source (H&E-like) appearance and the public checkpoints more frequently introduce systematic style biases.

At the same time, several constraints limit the strength of what can be claimed at this stage. The Lyon CK7 tiles include residual quality issues (e.g., blur, background-dominant regions, and occasional atypical color contamination), which can cap achievable fidelity and may influence distributional metrics even after filtering. Moreover, without a pathologist-driven review within the project timeframe, the evaluation cannot establish biological correctness of CK7 localization beyond distributional proximity and visual plausibility. Finally, limited compute availability restricted the breadth of ablations and hyperparameter exploration. Overall, this project demonstrates the feasibility of adapting a diffusion-based foundation model to a new clinical cohort for CK7 virtual staining in an unpaired setting and delivers a reproducible pipeline for preprocessing, LoRA fine-tuning, and cohort-level evaluation. In the longer term, sufficiently realistic synthetic CK7 images generated from widely available H&E could serve as additional input or training signal for downstream computational pathology workflows (for instance tumour classification), provided that future work incorporates expert validation and broader multi-center testing to ensure medical reliability and generalization.

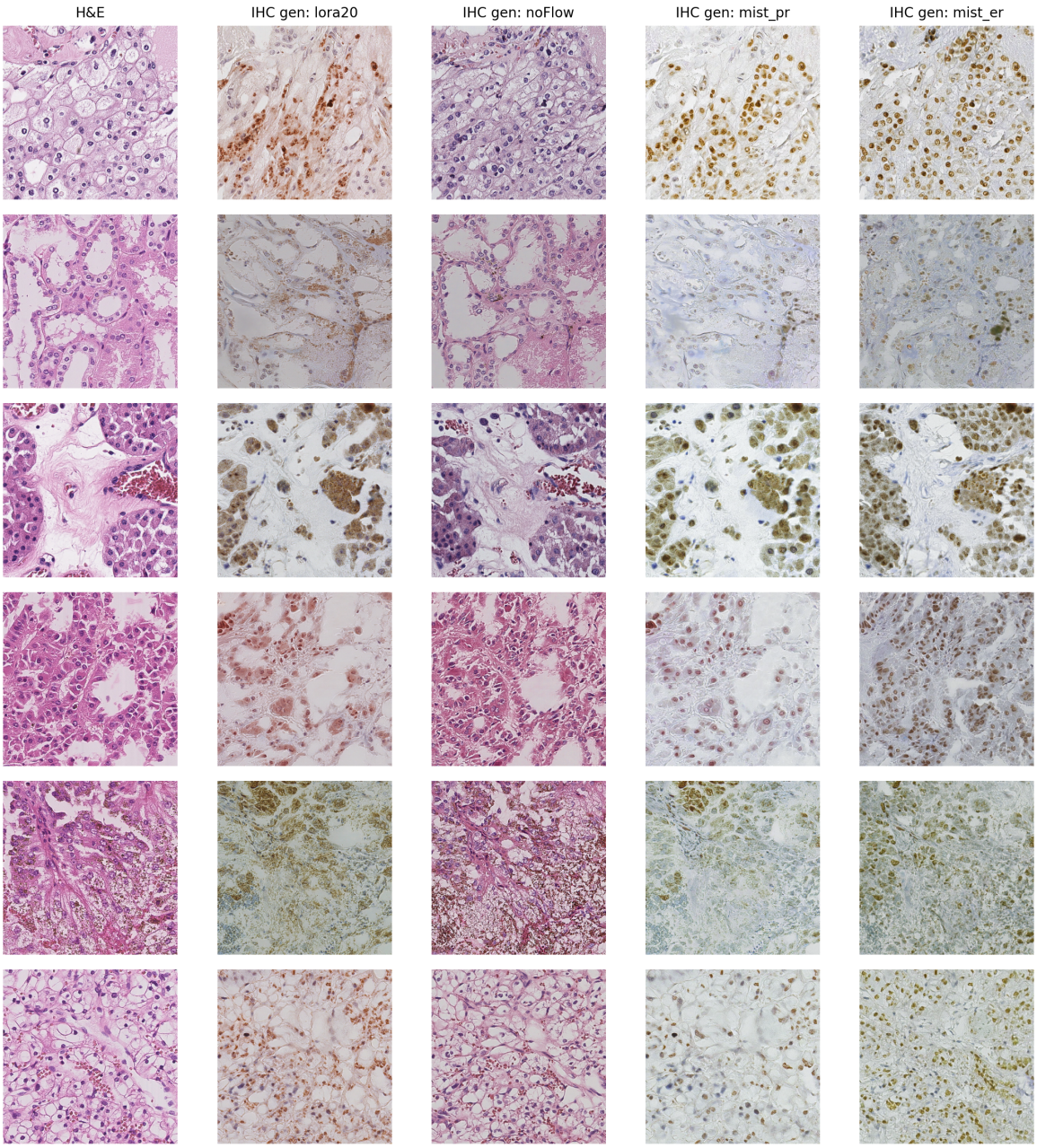


Figure 8: Random samples (6 cases). Same tile index across checkpoints.

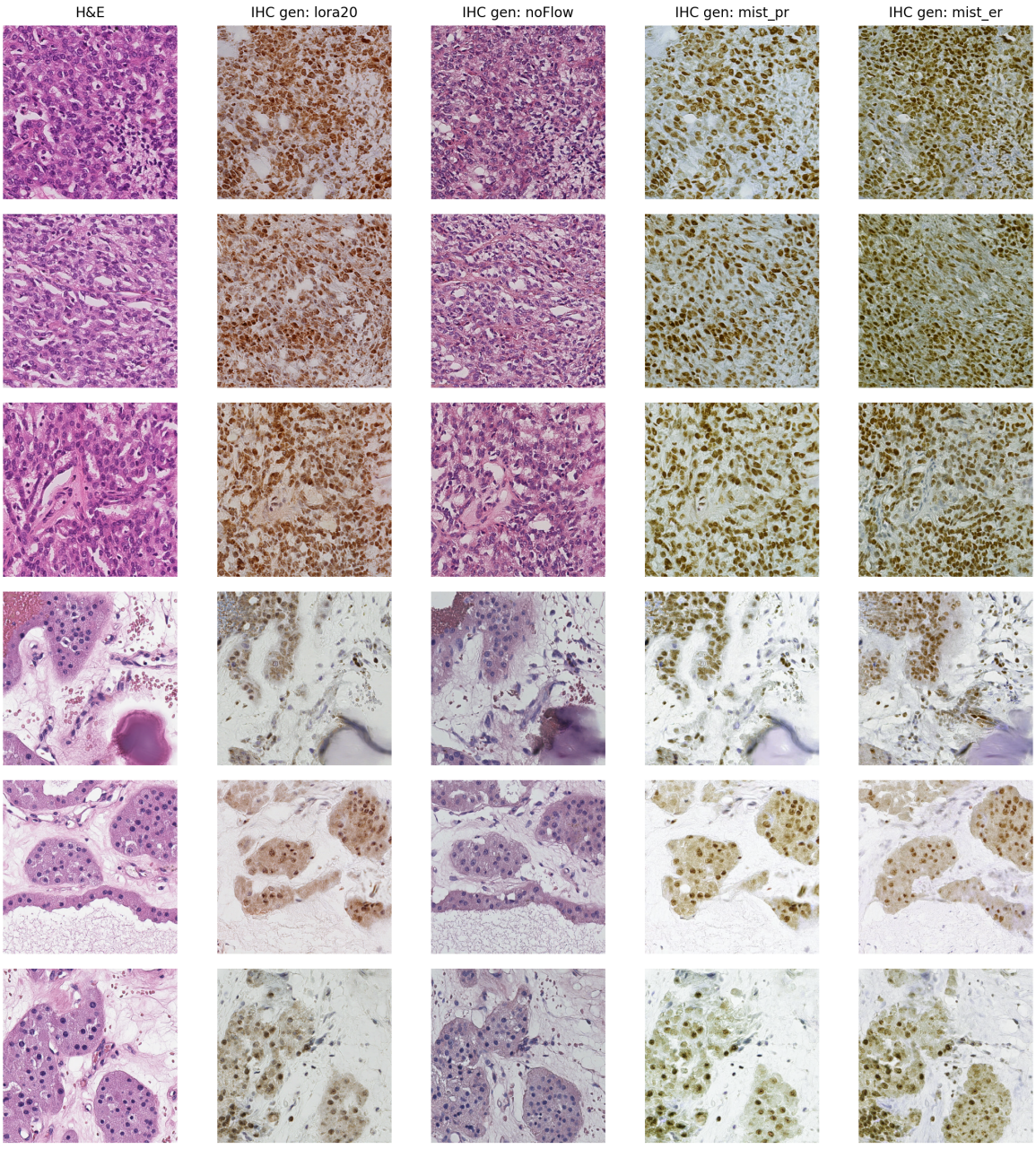


Figure 9: Best (top 3) / Worst (top 3) cases ranked on `lora20` using proxy-based scoring; other checkpoints shown on the same indices.

8. Bibliography

[1] Shikha Dubey, Yosep Chong, Beatrice Knudsen, and Shireen Y Elhabian. Vims: virtual immunohistochemistry multiplex staining via text-to-stain diffusion trained on uniplex stains. In *International Workshop on Machine Learning in Medical Imaging*, pages 143–155. Springer, 2024.

[2] Fangda Li, Zhiqiang Hu, Wen Chen, and Avinash Kak. Adaptive supervised patchnce loss for learning h&e-to-ihc stain translation with inconsistent groundtruth image pairs. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 632–641. Springer, 2023.

[3] Zhongxiao Li, Tianqi Su, Bin Zhang, Wenkai Han, Sibin Zhang, Guiyin Sun, Yuwei Cong, Xin Chen, Jiping Qi, Yujie Wang, et al. His-mmdm: Multi-domain and multi-omics translation of histopathological images with diffusion models. *Advanced Science*, page e18066, 2024.

[4] Srikar Yellapragada, Alexandros Graikos, Zilinghan Li, Kostas Triaridis, Varun Belagali, Saarthak Kapse, Tarak Nath Nandi, Ravi K Madduri, Prateek Prasanna, Tahsin Kurc, et al. Pixcell: A generative foundation model for digital histopathology images. *arXiv preprint arXiv:2506.05127*, 2025.

A. Appendix

A.1. Tile filtering logic used during SVS→NPZ conversion

The following excerpt reports the core "informativeness" filter used during tile extraction. A tile is discarded if it is blurry (low Laplacian variance), poorly saturated (mostly gray/unstained), or dominated by background/white pixels.

```python
import numpy as np
import cv2

def is_informative(im, size):
    """
    Returns True if a tile contains enough signal to be kept.
    Filters out:
      (1) blurry tiles (low Laplacian variance),
      (2) low-saturation tiles (mostly gray),
      (3) background-dominated tiles (too many near-white pixels).
    """

    img_array = np.array(im)

    # (1) Blur filtering (focus measure via Laplacian variance)
    gray = cv2.cvtColor(img_array, cv2.COLOR_RGB2GRAY)
    laplacian_var = cv2.Laplacian(gray, cv2.CV_64F).var()
    if laplacian_var < 100:
        return False

    # (2) Low-saturation filtering (proxy for weak staining / low
        content)
    hsv = cv2.cvtColor(img_array, cv2.COLOR_RGB2HSV)
    saturation = hsv[:, :, 1]
    if np.mean(saturation > 30) < 0.10:
        return False

    # (3) Background filtering (fraction of near-white pixels)
    white_fraction = np.sum(np.mean(img_array, axis=2) > 235) / (size *
        size)
    if white_fraction > 0.60:
        return False

    return True
```

Listing 1: Core tile informativeness filter used in `svs_to_npz.py`.

A.2. LoRA fine-tuning details and code

PixCell-1024 was instantiated from the official weights and augmented with LoRA adapters on the transformer cross-attention projections; LoRA parameters were trained while the backbone

remained frozen. The Stable Diffusion 3.5 Large VAE was used only to encode CK7 tiles into latent space and kept frozen, and UNI-2h was used to compute conditioning embeddings and kept frozen. To keep training within the available compute and I/O budget, we did not use the full Lyon CK7 collection (37,268 tiles) but trained on a capped subset of the training split by limiting to 300 tiles per NPZ file (4,061 tiles total), which may under-represent rare patterns but allowed completing experiments within the project timeline. Training followed the standard diffusion noise-prediction objective: at each iteration, a random timestep in $[0, 1000)$ was sampled, Gaussian noise was added to VAE latents according to the scheduler cumulative alphas, and the LoRA-augmented denoiser predicted the noise conditioned on UNI embeddings. The loss was the mean squared error between predicted and true noise. Classifier-free conditioning dropout was applied with probability 0.1 by replacing conditioning embeddings with the unconditional embedding. Optimization used AdamW with constant learning rate 10^{-4}, batch size 4, and gradient accumulation steps equal to 1. Mixed precision `bf16` (bfloat16, a reduced-precision floating-point format) was enabled via `Accelerate` (a Hugging Face library that manages device placement and mixed precision), and TF32 (NVIDIA TensorFloat-32 mode) was enabled to speed up matrix operations on the GPU.

```
    sd3_vae.requires_grad_(False)
    uni_model.requires_grad_(False)
    lora_parameters = [p for p in lora_transformer.parameters() if p.
        requires_grad]
    optimizer = torch.optim.AdamW(lora_parameters, lr=1e-4)

    for epoch in range(num_epochs):
    lora_transformer.train()
    for step, batch in enumerate(train_dataloader):
    with accelerator.accumulate(lora_transformer):
    he, ihc = batch
    ihc = ihc.to(device, non_blocking=True)

    with accelerator.autocast():
    uni_patches = einops.rearrange(
    ihc, "b c (d1 h) (d2 w) -> (b d1 d2) c h w", d1=4, d2=4
    )
    uni_input = uni_transform(uni_patches).to(device, non_blocking=True)
    with torch.inference_mode():
    uni_emb = uni_model(uni_input)
    uni_emb = uni_emb.unsqueeze(0).reshape(ihc.size(0), 16, -1)

    ihc_in = (2.0 * (ihc - 0.5)).to(dtype=vae.dtype)
    latents = vae.encode(ihc_in).latent_dist.sample()
    latents = (latents - vae_shift) * vae_scale

    t = torch.randint(0, 1000, (ihc.size(0),), device=device, dtype=
        torch.int64)
    atbar = alphas_cumprod[t].view(ihc.size(0), 1, 1, 1)
    epsilon = torch.randn_like(latents)
```

```
    noisy = torch.sqrt(atbar) * latents + torch.sqrt(1.0 - atbar) *
        epsilon

    if uncond_prob > 0:
    uncond = lora_transformer.caption_projection.uncond_embedding.clone
        ().tile(ihc.size(0), 1, 1)
    mask = (torch.rand((ihc.size(0), 1, 1), device=device) < uncond_prob
        ).float()
    uni_emb = (1.0 - mask) * uni_emb + mask * uncond

    eps_pred = lora_transformer(
    noisy, encoder_hidden_states=uni_emb, timestep=t, return_dict=False
    )[0]
    loss = ((eps_pred[:, :16, :, :] - epsilon) ** 2).mean()

    accelerator.backward(loss)
    optimizer.step()
    optimizer.zero_grad(set_to_none=True)
```

Listing 2: Core LoRA fine-tuning loop used in `train_lora.py`.

A.2.1. Quantitative metrics computation

```
    def compute_full_fid(
    real_imgs: List[np.ndarray],
    fake_imgs: List[np.ndarray],
    device: str,
    ) -> float:
    fid = FrechetInceptionDistance(feature=2048, normalize=True).to(
        device)

    def update(imgs: List[np.ndarray], real: bool):
    batch = []
    for im in imgs:
    t = torch.from_numpy(im).permute(2, 0, 1).unsqueeze(0)  # uint8 CHW
    batch.append(t)
    if len(batch) >= 32:
    x = torch.cat(batch, dim=0).to(device)
    fid.update(x, real=real)
    batch = []
    if batch:
    x = torch.cat(batch, dim=0).to(device)
    fid.update(x, real=real)

    update(real_imgs, real=True)
    update(fake_imgs, real=False)
    return float(fid.compute().detach().cpu().item())

```

```
    def compute_crop_fid_kid(
    real_imgs: List[np.ndarray],
    fake_imgs: List[np.ndarray],
    crop_size: int,
    crops_per_image: int,
    seed: int,
    device: str,
    kid_subsets: int,
    kid_subset_size: int,
    ) -> Tuple[float, float, float]:
    rng = np.random.default_rng(seed)

    fid = FrechetInceptionDistance(feature=2048, normalize=True).to(
        device)
    kid = make_kid_metric(
    device=device, subset_size=kid_subset_size, subsets=kid_subsets,
        normalize=True
    )

    def update_metrics(imgs: List[np.ndarray], real: bool):
    batch = []
    for im in imgs:
    for cr in random_crops(im, crop_size, crops_per_image, rng):
    t = torch.from_numpy(cr).permute(2, 0, 1).unsqueeze(0)  # uint8 CHW
    batch.append(t)
    if len(batch) >= 32:
    x = torch.cat(batch, dim=0).to(device)
    fid.update(x, real=real)
    kid.update(x, real=real)
    batch = []
    if batch:
    x = torch.cat(batch, dim=0).to(device)
    fid.update(x, real=real)
    kid.update(x, real=real)

    update_metrics(real_imgs, real=True)
    update_metrics(fake_imgs, real=False)

    cropfid = float(fid.compute().detach().cpu().item())
    kid_out = kid.compute()

    if isinstance(kid_out, (tuple, list)) and len(kid_out) == 2:
    kid_mean, kid_std = kid_out
    elif isinstance(kid_out, dict) and "kid_mean" in kid_out and "
        kid_std" in kid_out:
    kid_mean, kid_std = kid_out["kid_mean"], kid_out["kid_std"]
    else:
    kid_mean, kid_std = kid_out, torch.tensor(0.0)
```

```
70
71      kid_mean = float(kid_mean.detach().cpu().item())
72      kid_std = float(kid_std.detach().cpu().item())
73      return cropfid, kid_mean, kid_std
```

Listing 3: Quantitative metrics computation used in `compute_fid_kid.py`.

A.2.2. Qualitative evaluation figure generation

```
1  def tissue_fraction_rgb_uint8(rgb):
2      hsv = Image.fromarray(rgb, mode="RGB").convert("HSV")
3      hsv = np.asarray(hsv).astype(np.float32)
4      s = hsv[..., 1] / 255.0
5      v = hsv[..., 2] / 255.0
6      tissue = (v < 0.92) & (s > 0.05)
7      return float(tissue.mean())
8
9  def passes_tissue_filters(rgb):
10     tf_full = tissue_fraction_rgb_uint8(rgb)
11     if tf_full < tissue_min: return False
12     cc = center_crop(rgb, center_crop_size)
13     tf_c = tissue_fraction_rgb_uint8(cc)
14     return tf_c >= center_tissue_min
15
16 # proxies on generated tile
17 sat_std = hsv_saturation_std(gen_img)           # low => global tint /
       low chroma
18 ac = autocorr_shift_score(gen_img, shift=16)   # high => repetition
       artifacts
19 lap = laplacian_var(gen_img)                   # extreme => blur/
       oversharp
20
21 # z-normalize and build a single badness score
22 badness = (-sat_z) + (ac_z) + (0.5 * abs(lap_z))
23 best = argsort(badness)[:topk]
24 worst = argsort(badness)[-topk:][::-1]
```

Listing 4: Key logic for qualitative figures: tissue filtering and best/worst selection of `generate_figures.py` file.

A.3. DR-1 cluster details

DR-1 is a shared compute server operated in a "cluster-like" fashion at Polytech Nice Sophia. Jobs are submitted through Slurm, with the LiCO web interface used for simplified submission and monitoring. Access is restricted to research members of the SophiaTech Campus laboratories (i3S, LEAT, Polytech Lab), so availability depends on concurrent usage.

From a hardware standpoint, DR-1 provides 512 GiB of RAM, dual Intel Xeon Gold 6326 CPUs (32 physical cores / 64 logical cores total), and eight NVIDIA A40 GPUs with 48 GiB of

memory each, along with local NVMe storage in RAID-0. The presence of eight GPUs defines the maximum GPU parallelism, and workloads are typically scheduled as whole-GPU "slots", which is an operational constraint when many users compete for limited accelerators. In our setting, this translated into queueing delays that sometimes dominated the wall-clock iteration cycle, in addition to the intrinsic runtime of long inference jobs.

A.4. FID

Let $f_{\text{Inc}}(\cdot)$ be a fixed Inception-v3 feature encoder and let $z = f_{\text{Inc}}(x) \in \mathbb{R}^d$ denote the embedding of an image x (we use the standard 2048-dimensional pool3 features). Given real and generated embedding sets $\mathcal{Z}_r = \{z_i^r\}_{i=1}^N$ and $\mathcal{Z}_g = \{z_j^g\}_{j=1}^M$, we compute empirical means and covariances:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} z_i, \qquad \Sigma = \frac{1}{n-1}\sum_{i=1}^{n}(z_i - \mu)(z_i - \mu)^\top.$$

FID is the Fréchet distance between the two fitted Gaussians:

$$\text{FID}(\mathcal{Z}_r, \mathcal{Z}_g) = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\Big(\Sigma_r + \Sigma_g - 2\,(\Sigma_r\Sigma_g)^{1/2}\Big).$$

Lower values indicate closer feature distributions.

A.5. KID

KID measures the discrepancy between two distributions via the squared Maximum Mean Discrepancy (MMD^2) in feature space. Given a positive definite kernel $k(\cdot,\cdot)$ and two sample sets $\mathcal{Z}_r = \{z_i^r\}_{i=1}^N$ and $\mathcal{Z}_g = \{z_j^g\}_{j=1}^M$ (Inception features), the unbiased estimator is:

$$\widehat{\text{MMD}^2}_u(\mathcal{Z}_r, \mathcal{Z}_g) = \frac{1}{N(N-1)}\sum_{i\neq j} k(z_i^r, z_j^r) + \frac{1}{M(M-1)}\sum_{i\neq j} k(z_i^g, z_j^g) - \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k(z_i^r, z_j^g).$$

In practice, KID is estimated by averaging $\widehat{\text{MMD}^2}_u$ over multiple random subsets; in our implementation (TorchMetrics) this yields a mean and an empirical standard deviation.

A.6. Crop FID

Crop FID applies the same vanilla FID computation, but on random crops to emphasize local texture. Let $c(\cdot;\omega)$ be a random crop operator extracting a 256×256 crop from a tile, with randomness ω. We build crop embedding sets $\mathcal{Z}_r^{\text{crop}} = \{f_{\text{Inc}}(c(x_i;\omega_{i,t}))\}$ and $\mathcal{Z}_g^{\text{crop}} = \{f_{\text{Inc}}(c(\tilde{x}_j;\tilde{\omega}_{j,t}))\}$, and compute FID using the standard formula in Appendix A.4. Crop sampling is seed-controlled.

A.7. Useful Links

The resources related to this project are available at the following links:

- **Synthetic Image Dataset:** The dataset generated in this project can be downloaded from the following shared folder: `https://liveunibo-my.sharepoint.com/:f:/g/personal/`

giuseppe_spathis_studio_unibo_it/IgBB795oHqRsR6NDZd5Qz8udAaKdjCy0DGtqybqrlZMgo7g?e=jp4ONH.
Note: Access to this link is restricted to institutional email accounts from the University of Bologna. If you are not affiliated with the University of Bologna and would like to request access to the dataset, please contact: giuseppe.spathis@studio.unibo.it.

- **Code and Scripts:** The code and scripts used in this project are available in this public GitHub repository: https://github.com/GiuseppeSpathis/virtualStaining.
- **LoRA Checkpoint:** The checkpoint of LoRA tuned can be downloaded from Google Drive: https://drive.google.com/file/d/1_JvJ33qZJLya_uUExNigbneXCuYvPu9Z/view?usp=sharing.