Imperial College London

Individual Project Report

Imperial College London

Department of Computing

Generating Histopathology Images With Segmentation Masks, and Gigapixel Images With Ultra-Resolution Cascaded Diffusion Models

Author:
James Ball

Supervisor:
Dr Bernhard Kainz

Second Marker:
Dr Benjamin Hou

Abstract

With the recent advancements in AI image generation from the likes of Stable Diffusion, DALL-E 2, and Imagen, there are a lot of examples of using diffusion models to generate unseen high-quality images. One clear application of this is to enrich medical datasets with synthetic images where any performance gains can have a big impact.

This project uses a novel method to generate synthetic pairs of images and segmentation masks to enrich such datasets. We train an unconditional diffusion model on histopathology images, along with a baseline segmentation model, to generate synthetic segmentation masks. Then, we fine-tune this diffusion model, conditioning it on segmentation masks, so that we can generate variations of the same image. Finally, we can generate a synthetic dataset that can be used to enrich the training of a segmentation model.

Furthermore, we introduce a novel architecture for generating ultra-resolution images that are larger than a gigapixel in size using diffusion models. Ultra-Resolution Cascaded Diffusion Models (URCDMs) consist of three Cascaded Diffusion Models (CDMs) operating at increasing magnifications. The first CDM generates a full-scale but low-resolution image, and later CDMs are conditioned on this image and 'zoom in' on the centre of it.

Overall, the novel method for synthesising segmentation and image pairs can lead to state-of-the-art performance on downstream segmentation tasks. In addition, we evaluate their realism by developing an evaluation platform to carry out a user study with expert pathologists. These results show that formally trained experts cannot distinguish between real and synthetic images.

Furthermore, URCDMs successfully generate ultra-resolution images with long-distance spatial coherency that look more realistic at multiple scales than outpainting. Whilst there is room for improvement in generating higher-quality fine details, this is a big step towards extremely high-resolution synthetic images that are indistinguishable from their real counterparts.

Acknowledgements

I would like to thank the following people that have helped me complete and evaluate this project:

First of all, a big thank you to **Sarah Cechnicka** for working with me to evaluate the performance of the diffusion models on downstream segmentation tasks, and for your help in evaluating the realism of synthetic images.

I'm extremely grateful for the evaluation carried out by expert medical renal pathologists working at North West London Pathology and Guy's and St Thomas' NHS Trust: **Dr Catherine Horsfield**, **Dr Naomi Simmonds**, and **Dr Andrew Smith**, and with a special thank you to **Dr Candice Roufosse** from the Department of Immunology & Inflammation, Imperial College, who helped organise this small team and has been helpful for expert feedback throughout.

Thank you very much, **Dr Bernhard Kainz** for supervising this project, pushing me to extend the project beyond what I originally envisioned, and giving useful guidance and advice.

Thank you, **Dr Benjamin Hou** for providing advice as a second marker.

I'd also like to thank **Friedrich-Alexander-Universität Erlangen-Nürnberg** for providing access to their high-performance computing cluster that made this project possible.

Finally, thank you to my friends and family that have been great support throughout.

Contents

Chapter 1

Introduction

1.1 Objectives

This project has two main objectives. The first objective is to use diffusion models to generate realistic-looking histopathology imagery, along with segmentation masks. By using these images to enrich data-limited datasets, we aim to improve performance on downstream machine learning tasks as an advanced method of data augmentation. This objective is explored in Section 3.

Unconditionally generated synthetic histopathology image patches are valuable on their own as they can be used as an alternative to real medical image datasets that are hard to obtain. However, synthetic images that are conditioned on segmentation masks are much more useful. Labelling histopathology images is a very costly and time-consuming job that requires formally-trained pathologists, so synthesising this data is valuable. Furthermore, they allow variations of an image to be generated, by segmenting an image and then re-generating the image using the same segmentation mask. Combined with conditional image generation using clinical parameters, such as tissue type (see Figure 5.8e-h), this can be quite powerful.

The second objective is to extend this capability to ultra-resolution images: those that are greater than a gigapixel (1,000,000,000 pixels) in size. This poses several challenges, such as memory constraints, sampling times, and lack of training data. This objective is explored in Section 4.

A Whole Slide Image (WSI) is an extremely high-resolution digital image of a glass slide with tissue, captured by a slide scanner. These ultra-resolution images are usually split into patches to increase the amount of training data and computational complexity. However, pathologists typically analyse Whole Slide Images in their entirety; zooming in and out at multiple scales. Synthesising ultra-resolution imagery could allow for more complex downstream algorithms that operate on the entire image at different scales due to the additional training data it provides. Furthermore, high-resolution images are already used at multiple scales in other domains such as satellite imagery [1]. A novel method of synthesising ultra-resolution images is evaluated on the Whole Slide Image dataset discussed in Section 5.1.1.

1.2 Novel Contributions

On top of unconditionally generating images with existing architectures in the new context of renal histopathology, this project will explore some novel applications of diffusion models:

Whilst generating histopathology patches conditioned on segmentation masks has already been demonstrated [2], we further condition generation on clinical parameters such as the tissue type. We also develop a novel method of generating synthetic pairs of segmentation masks and images to enrich datasets that leads to state-of-the-art Dice scores on data-limited datasets.

Contextually accurate large-scale image generation of large regions of kidney tissue by utilising outpainting [3, 4] is a challenging area to explore. Methods such as Anyres-GAN [5] achieve this using GANs and allow for any resolution image generation at inference time, but only evaluate sub-gigapixel image generation. The Ultra Resolution Cascaded Diffusion Model architecture is a novel method of achieving high-quality and extremely high-resolution images applied to diffusion models that are known to outperform GANs [6].

1.3 Ethical Issues

Human samples used in this research project were obtained from the Imperial College Healthcare Tissue & Biobank (ICHTB). ICHTB is supported by the National Institute for Health Research (NIHR) Biomedical Research Centre based at Imperial College Healthcare NHS Trust and Imperial College London. ICHTB is approved by Wales REC3 to release human material for research (22/WA/2836) and the samples for this project are from research application number R18040.

	Yes	No
Section 1: Humans		
Does your project involve human participants?	✓	
Section 2: Protection of Personal Data		
Does your project involve personal data collection and/or processing?	✓	
Does it involve the collection and/or processing of sensitive personal data (e.g. health, sexual lifestyle, ethnicity, political opinion, religious or philosophical conviction)?	✓	
Does it involve processing of genetic information?		✓
Does it involve tracking or observation of participants? It should be noted that this issue is not limited to surveillance or localization data. It also applies to Wan data such as IP address, MACs, cookies etc.		✓
Does your project involve further processing of previously collected personal data (secondary use)? For example Does your project involve merging existing data sets?	✓	
Section 3: Animals		
Does your project involve animals?		✓
Section 4: Developing Countries		
Does your project involve developing countries?		✓
If your project involves low and/or lower-middle income countries, are any benefit-sharing actions planned?		✓
Could the situation in the country put the individuals taking part in the project at risk?		✓
Section 5: Environmental Protection and Safety		
Does your project involve the use of elements that may cause harm to the environment, animals or plants?		✓
Does your project involve the use of elements that may cause harm to humans, including project staff?		✓
Section 6: Dual Use		
Does your project have the potential for military applications?		✓
Does your project have an exclusive civilian application focus?		✓
Will your project use or produce goods or information that will require export licenses in accordance with legislation on dual use items?		✓

Does your project affect current standards in military ethics – e.g., global ban on weapons of mass destruction, issues of proportionality, discrimination of combatants and accountability in drone and autonomous robotics developments, incendiary or laser weapons?		✓
SECTION 7: MISUSE		
Does your project have the potential for malevolent/criminal/terrorist abuse?		✓
Does your project involve information on/or the use of biological-, chemical-, nuclear/radiological-security sensitive materials and explosives, and means of their delivery?		✓
Does your project involve the development of technologies or the creation of information that could have severe negative impacts on human rights standards (e.g. privacy, stigmatization, discrimination), if misapplied?		✓
Does your project have the potential for terrorist or criminal abuse e.g. infrastructural vulnerability studies, cybersecurity related project?		✓
SECTION 8: LEGAL ISSUES		
Will your project use or produce software for which there are copyright licensing implications?		✓
Will your project use or produce goods or information for which there are data protection, or other legal implications?		✓
SECTION 9: OTHER ETHICS ISSUES		
Are there any other ethics issues that should be taken into consideration?		✓

Chapter 2

Background

2.1 Denoising Diffusion Probabilistic Models

Much of the theory presented in this section will refer to the technical explanation in "Denoising Diffusion Probabilistic Models" authored by Ho et al. [7], as well as "Diffusion Models Beat GANs on Image Synthesis" authored by Prafulla Dhariwal and Alex Nichol [6].

A Denoising Diffusion Probabilistic Model (DDPM), or simply diffusion model, is a generative model that learns how to gradually remove noise from an image in a series of time steps. This, trained on a dataset of images, will learn what parts of an image are noise and what parts of an image should be restored and retained.

Since diffusion models are trained to remove a small amount of noise from images of various different prior levels of noise, they can be applied in sequence on an initial image that has no distinguishing features and consists of only noise. The model infers details from this noisy image and generates a new random image from the noise. This is called the reverse diffusion process.

The forward diffusion process is a Markov chain that adds some amount of noise to the image in a number of time steps. This is trivially defined and no training is required. Each image in the forward diffusion process is represented as $x_1, ..., x_T$, where T is the number of time steps in the process, and x_T is an image that is indistinguishable from a completely noisy image. x_0 is the original input image with no noise applied. The noise added from image x_{t-1} to image x_t is modelled by q, a Gaussian distribution, and the amount of noise depends on the variance schedule of the model, $\beta_1, ..., \beta_T$:

$$q(x_t|x_{t-1}) \triangleq \mathcal{N}(x_t, \mu = \sqrt{1-\beta_t}x_{t-1}, \sigma^2 = \beta_t I) \tag{2.1}$$

At time step t, the variance schedule is β_t, which is usually defined as a hyperparameter of the model [7]. To provide some context as to the values used in practice, Ho et al. [7] set the number of time steps $T = 1000$ and values for $\beta_1, ..., \beta_T$ to linearly increase between 10^{-4} and 0.02.

Ho et al. [7] show it is possible to rewrite 2.1 in closed form for the purposes of efficient training:

$$q(x_t|x_0) = \mathcal{N}(x_t, \mu = \sqrt{\bar{\alpha}_t}x_0, \sigma^2 = (1-\bar{\alpha}_t)I) \tag{2.2}$$

Where $\alpha_t \triangleq 1 - \beta_t$ and $\bar{\alpha}_t \triangleq \prod_{s=1}^{t} \alpha_s$. Using the closed form, we can train the model with various input images at random time steps, without first calculating all previous less noisy images, allowing more diverse training images in each batch. This lets us train on x_t without previously calculating $x_1, ..., x_{t-1}$.

To learn how to remove noise in the reverse diffusion process, we need to model $p_\theta(x_{t-1}|x_t)$ with parameters θ. This is a Gaussian distribution that produces the image at the previous time step in the forward diffusion process:

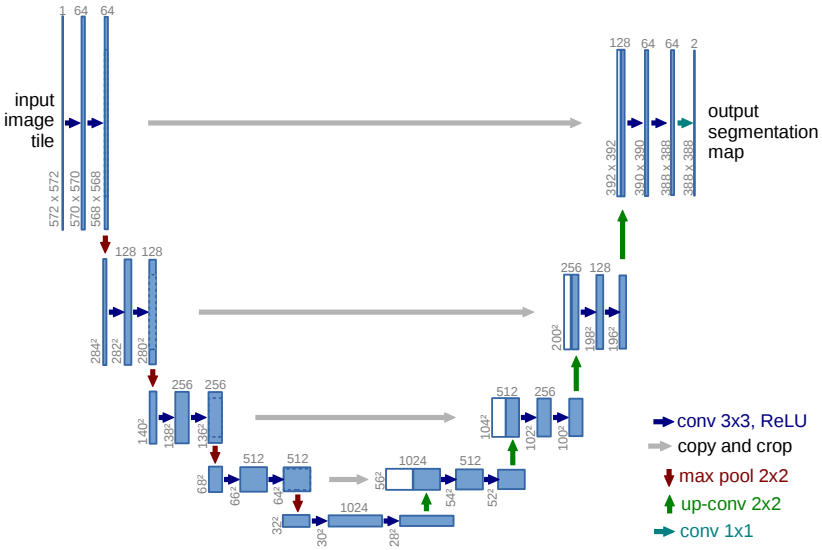


Figure 2.1: Classical U-Net architecture presented by Olaf Ronneberger et al. [14]

$$p_\theta(x_{t-1}|x_t) \triangleq \mathcal{N}(x_{t-1}, \mu = \mu_\theta(x_t, t), \sigma^2 = \Sigma_\theta(x_t, t)) \tag{2.3}$$

Whilst we could train a model for μ_θ and Σ_θ, Ho et al. [7] find that you can instead predict the noise added to the image instead at a time step. This is represented by ϵ_θ, a model to predict the noise ϵ added to x_{t-1} to produce x_t. With the predicted noise at time step t, $\epsilon_\theta(x_t, t)$, we can trivially produce an estimate of x_{t-1}:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) \tag{2.4}$$

With these equations in mind, Ho et al. [7] formulate a simple loss function for evaluating the quality of the reverse diffusion process model:

$$\mathcal{L}_{simple}(\theta) \triangleq \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2\right] \tag{2.5}$$

All diffusion models use the fundamental theory presented by Ho et al. [7], and later examples build extensions such as super-resolution networks that upscale the low-resolution image generated [8, 9], and perform conditional image generation on text [10, 11, 12] or clinical data [13, 2].

2.2 U-Net

Now that we know we need to model $\epsilon_\theta(x_t, t)$, the predicted noise at time step t, and we know how to optimise this using 2.5, we need to choose an appropriate model architecture for the reverse diffusion process. Ho et al. [7] use a model based on a U-Net [14] convolutional neural network architecture for this purpose. U-Nets have proven useful in image-based machine learning applications such as in segmentation [14], making them a natural choice for predicting the noise in an image.

Figure 2.1 shows the contracting and expanding paths of a U-Net. The first contracting path takes a high-resolution input, i.e. an image, and applies several rounds of convolution which increases the number of channels, followed by a pooling stage that performs downsampling. The contracting stage aims to capture the context of the input image [14]. The expanding path is mostly symmetric, with transposed convolution used instead of pooling. The expanding path increases the resolution, and also takes as input the cropped feature map of the output of

the respective stage in the contracting path [14]. For diffusion models, the input and output dimensions of the U-Net are the same since the U-Net predicts the noise portion of the input image.

2.3 Conditioning U-Nets on the time step, t, and other parameters

The same U-Net model is used regardless of the time step t, meaning parameters are shared for different values of t. Therefore, t must be specified as a parameter to each block of the U-Net somehow. Ho et al. [7] use the Transformer sinusoidal position encoding [15] to condition the input with t.

The sinusoidal position encoding used in the Transformer architecture [15] is defined as follows:

$$\mathrm{PE}_{(t,2i)} = sin\left(\frac{t}{10000^{2i/d}}\right) \tag{2.6}$$

$$\mathrm{PE}_{(t,2i+1)} = cos\left(\frac{t}{10000^{2i/d}}\right) \tag{2.7}$$

where d is the chosen dimension of the encoding, and i is the index of the current dimension. Vaswani et al. [15] hypothesise that this function would allow the model to learn the relative positions between inputs, as PE_{t+k} can be represented as a linear function of PE_t for some offset k.

Importantly, this allows one to generate an encoding for any number of dimensions; something that is important when conditioning the U-Net on the positional encoding since the dimensions of the U-Net's input and the encoding must match.

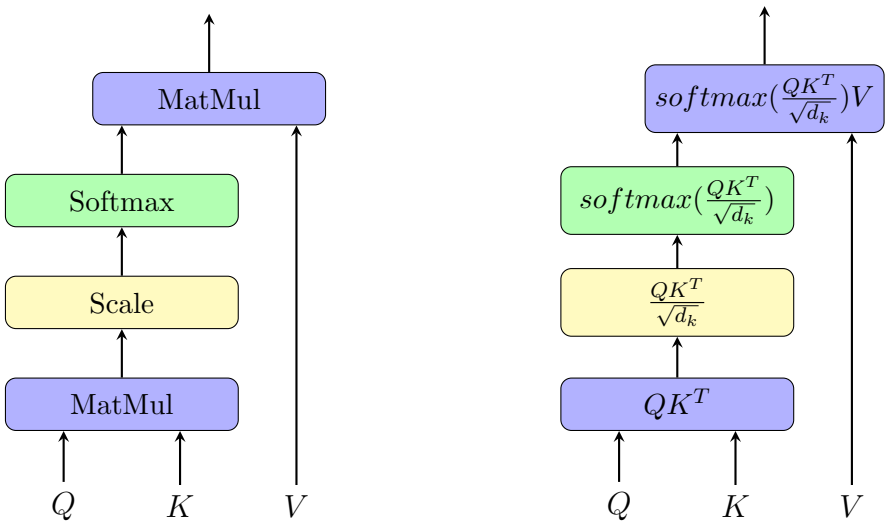
Conditioning can be performed by simply adding the encodings to the input [7], hence why the dimension of the positional encoding must be the same as the input. Alternatively, you could apply a method such as FiLM [16] that uses a separate neural network that learns, using the positional encodings, how to condition the input by scaling and shifting the input. Currently, implementations of state-of-the-art text-to-image generators, DALL-E 2 [11] and Imagen [12], such as `imagen-pytorch` [17] use cross-attention [15] to combine the encodings with the input. This is explained in detail in Section 2.4.

2.4 Attention

Intuitively, attention can be thought of as a mechanism to give parts of the input different weights depending on how important they are, whilst decreasing the weight of less important parts. This enables a model to pay closer 'attention' to certain parts of the input and put less focus on others. The aim is to have a model that puts more effort into learning important things.

In the context of diffusion models, the most important parts of the input are high-frequency details that change in colour or shape quickly. For example, we don't care about the detail in the graininess of the sky taken at night as much as we do the facial features and edges of the subject of the image, so we should pay stronger attention to that and try to focus our learning efforts.

Ho et al. [7] use attention mechanisms between each convolutional block based on those utilised in the Transformer architecture [15]. These increase the quality of the image by paying closer attention to high-frequency details in the image [18]. Attention is also utilised in diffusion models when conditioning U-Nets with text embeddings in implementations such as in the



(a) Operations involved

(b) Matrix output

Figure 2.2: Scaled Dot-Product Attention as presented in "Attention Is All You Need" [15]

`imagen-pytorch` library [17]. Whilst initially utilised for textual information, aspects from the Transformer architecture such as attention are also very effective for image generation [18].

The attention mechanism described by Vaswani et al. [15] is Scaled-Dot-Product Attention, as seen in Figure 2.2. The attention function has three inputs, Q, K, and V, which are matrices of queries, keys, and values. Each individual query and key has the same dimension, d_k. Vaswani et al. [15] introduce a few different applications of this: namely self-attention, and 'encoder-decoder attention', otherwise called cross-attention.

In self-attention, Q, K, and V are all generated from the input embeddings by multiplying the input embeddings with each of the corresponding weight matrices W^Q, W^K, and W^V which are learned during training. However, in cross-attention, one embedding is used to generate Q, and another embedding is used to generate both K and V. Both applications have the same underlying attention function, as in Figure 2.2. Importantly, when using multiple different embeddings in cross-attention, they must have the same dimension.

Self-attention is employed in diffusion models between the convolutional blocks of the U-Net [7], whereas cross-attention is useful for conditioning the input embeddings into the U-Net on some conditioning or contextual embeddings (e.g. age of a patient, or time since biopsy). This works by generating Q with the input embeddings and K and V with the contextual embeddings.

In practice, the scaled dot-product attention is simple to implement, and the attention function is defined as follows:

$$\text{Attention}(Q, K, V) \triangleq \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.8}$$

where Q, K, and V are generated by multiplying embeddings by W^Q, W^K, and W^V respectively, and d_k is the dimension of an individual query/key.

Vaswani et al. [15] find it beneficial to perform this multiple times with many different weight matrices. They describe each application of the attention function as a 'head', with h heads in total, and hence call this Multi-Head Attention. To achieve this, there are now weight matrices W_i^Q, W_i^K, and W_i^V for all $i \in [0, h)$ used to generate Q_i, K_i, and V_i, and calculate $\text{Attention}(Q_i, K_i, V_i) = Z_i$. Each output Z_i can then be concatenated to form a matrix $Z_{concat} = [Z_0, ..., Z_{h-1}]$. Since Z_{concat} is not the same dimension as the output in single-head attention, a final weight matrix W^O is needed to project Z_{concat} into the same dimensions as before.

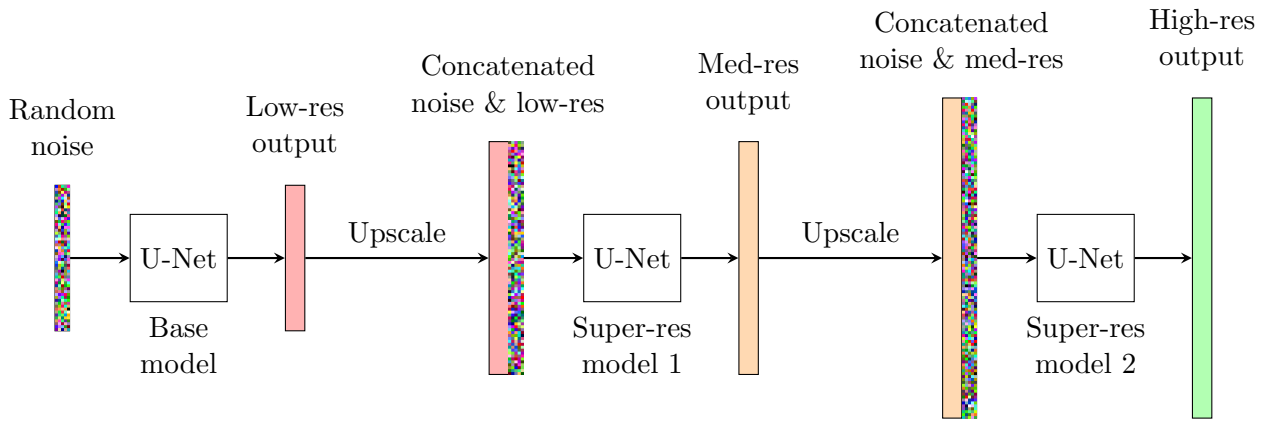


Figure 2.3: Cascaded Diffusion Model architecture for high-resolution image synthesis [9]

Whilst offering better performance, multi-head attention has a similar computational cost to single-head attention since the dimension of Q, K, and V are reduced by a factor of h, the number of heads [15].

Whilst described here as weight matrices, attention functions consist of multiple fully-connected neural networks that produce Q, K, and V for each head. Therefore, they are trained in the way you would expect with gradient descent, allowing them to learn what inputs to best pay attention to.

2.5 High-resolution image synthesis

Naturally, when generating images, there is a motivation to increase the quality of the image by generating higher resolution images. The first obvious thing to try is just to increase the input resolution and output resolution so that you are performing the same diffusion process on a high-resolution image and then denoising a high-resolution noise image at inference time. However, as the resolution increases, the number of iterative refinement steps must increase for the quality of the image to remain the same [8]. This leads to a single very large diffusion model.

Saharia et al. [9] propose cascading the diffusion models in several 'super-resolution' stages in order to independently train several small models, rather than one large model. For example, you could start with a diffusion model that generates 64×64 pixel images, and then use two super-resolution diffusion models to upscale this from $64 \times 64 \rightarrow 256 \times 256$ and then from $256 \times 256 \rightarrow 1024 \times 1024$. This leads to faster sampling as fewer iterative refinement steps are needed per model, and allows you to easily train each model in parallel [9].

Cascaded Diffusion Models (CDMs) work by first generating a low-resolution image with a base model, and using the output from this to condition the denoising process of a U-Net used for super-resolution. Conditioning can be performed in various ways, as discussed in Section 2.3, but for image conditioning Ho et al. [9] first upsample the low-resolution image to match the resolution of the input and then perform channel-wise concatenation with the input to the super-resolution U-Net. Further super-resolution stages condition using the lower-resolution image generated in the previous stage in the same way. An overview of this architecture is presented in Figure 2.3.

Since CDMs have separate stages for initial image generation and super-resolution, the model capacity of each stage can be tailored for the best performance. CDMs tend to perform best when most of the model capacity is focused on the base low-resolution model, as they are most important for image quality, and are quicker to train [9].

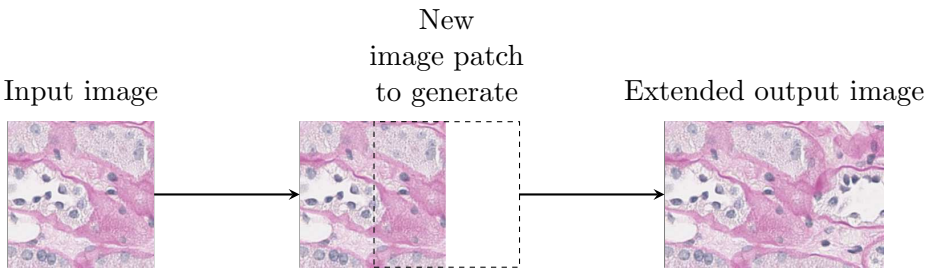


Figure 2.4: Outpainting [4] process by using inpainting [3] on part of a previously-generated image to extend the bounds of the original

Ho et al. [9] propose 'conditioning augmentation' to improve the quality of the super-resolution networks. This is the process of augmenting the lower-resolution output before providing it as input into the super-resolution networks, by the way of blurring or distorting the image in some way. This might seem counter-intuitive at first - better results are obtained by making inputs to the super-resolution networks lower quality - but Ho et al. [9] explain that these augmentations prevent the super-resolution networks from overfitting to inaccuracies or imperfections in the lower-resolution output. The base model might not be good enough quality compared with the images they are trained on, so applying conditioning augmentation tries to obscure any of these quality issues so that the super-resolution networks 'gloss over' them.

Conditioning augmentation is applied in a few different ways. These can be summarised by either applying some Gaussian blur to the low-resolution image or adding some Gaussian noise. Ho et al. [9] find that blurring the image works best at higher resolutions, whereas adding noise to the image is best at lower resolutions. Blurring augmentation is simply implemented by convolving a Gaussian filter with a 3×3 kernel over the image [9].

For lower resolutions, it is better to add some form of Gaussian noise, either by stopping the reverse diffusion process early (meaning that not all of the noise is removed from the image), or adding Gaussian noise to the denoised image after all time steps. This is referred to as 'truncated conditioning augmentation' and 'non-truncated conditioning augmentation' respectively [9]. For a trained model, there is little to no performance benefit of using truncated over non-truncated augmentation, and there are practical benefits of only needing to store the final denoised image when using non-truncated augmentation, so this is used in practice [9].

2.6 Inpainting

Inpainting is the process of generating parts of an image that are masked out using a diffusion model [3]. This means that you provide a real image and ask the diffusion model to regenerate part of the image, taking the context of the surrounding image into account. Applications of this could be changing the facial expression of a person by masking out their mouth or restoring distorted images.

Whilst there could be some utility in inpainting for this project, for example, to fine-tune a kidney biopsy image to make it look more realistic, 'outpainting' is likely to be much more applicable. Outpainting precedes diffusion models [19], but the word has been popularised more recently by OpenAI's DALL-E [4] in the context of diffusion models. It refers to using inpainting to extend the bounds of the original image by generating a new patch of an image that partially overlaps with the original, and masking out any part that doesn't overlap so that it is filled in by the diffusion model. Figure 2.4 demonstrates this process.

Inpainting is achievable without any fine-tuning, retraining, or consideration prior to training, meaning it can be performed on any off-the-shelf diffusion model [3]. Lugmayr et al. [3] describe

the method below for inpainting.

At time step t we have the original image, x_0, that is to be inpainted, so we can add noise according to the variance schedule as described in Equation 2.2 to get x_{t-1}^{known}. If we sample from the diffusion model, like in Equation 2.4, we can generate $x_{t-1}^{unknown}$ from x_t. Given a mask, m, that indicates which pixels in x_{t-1}^{known} should be kept, we can then generate x_{t-1} as follows:

$$x_{t-1} = m \odot x_{t-1}^{known} + (1 - m) \odot x_{t-1}^{unknown} \tag{2.9}$$

where $m \odot x$ are the pixels that we want to keep from the original image, and $(1 - m) \odot x$ are the pixels that should be regenerated by the model.

This method is very intuitive, and applying it directly in a standard forward-pass through a diffusion model might match the general shape and colour of surrounding pixels, but it will yield semantically incorrect results [3]. Lugmayr et al. [3] speculate that as Equation 2.9 generates $x_{t-1}^{unknown}$ without any consideration of x_{t-1}^{known}, it does not harmonise well with x_{t-1}^{known}. Although this harmonisation will improve in each time step, since sampling uses x_t as input which is the inpainted image from the previous time step, it will not harmonise before the final time step. Combining this with the decreasing variance of noise according to the variance schedule, there is limited time for the inpainted region to harmonise.

Lugmayr et al. [3] use a technique called resampling to increase the harmonisation at each time step. This repeats reverse and forward diffusion steps r times at each time step, meaning that noise is added to an inharmonious image, distorting it, and then the diffusion model denoises the image slightly. Since diffusion models are trained to generate images in the same distribution as the data they were trained on, denoising multiple times in a time step will bring the inharmonious image closer to a more semantically correct image. Lugmayr et al. [3] find that $r = 10$ (i.e. removing and then adding noise 10 times) works well in practice.

On top of resampling, the idea of a 'jump length' is added by Lugmayr et al. [3]. This applies the resampling described above over an increased number of forward and reverse diffusion steps at a time. At the initial time step T in the reverse diffusion process, j reverse diffusion steps will be applied (removing noise), followed by j forward diffusion steps (adding noise). This is then repeated, as before, r times. Lugmayr et al. [3] demonstrate that this further increases harmonisation in the image. Unsurprisingly, adding repeated reverse diffusion steps by resampling or increasing the jump length can significantly increase the time taken to perform inpainting.

2.7 Classifier-Free Guidance

Ho et al. [20] introduce classifier-free guidance as a simple way to trade-off the diversity in types of images generated by the diffusion process for image quality. Prior generative models such as GANs can make this trade-off simply by changing the range or variance of the noise input into the model at sample time, but this decreases the image quality in diffusion models [20]. Dhariwal et al. [6] demonstrate that 'classifier guidance' is achievable by training an additional classifier alongside the diffusion model, and partially using the gradient of the classifier during training. Mixing the gradient to varying degrees allows you to make the trade-off between diversity and quality. However, training an additional classifier adds significant complexity; hence why classifier-free guidance is desired.

Classifier-free guidance works by training both a conditional and unconditional diffusion model at the same time. In practice, only a single model is used for both, saving on training time and complexity, by specifying a null token as input for the condition when training unconditionally. Formally, this gives us two models: $\epsilon_\theta(x_t, \mathbf{c})$ where $\mathbf{c}$ is what we are conditioning the image on (e.g. clinical parameters), and $\epsilon_\theta(x_t)$ which is an unconditional model. In reality, $\epsilon_\theta(x_t) = \epsilon_\theta(x_t, \mathbf{c} = \emptyset)$, where $\emptyset$ is the null token, as we can reuse the same model for both tasks.

At sample time the classifier guidance weight, w, is specified to control this trade-off. This gives the new equation for sampling the noise present in an image:

$$\tilde{\epsilon_\theta}(x_t, \mathbf{c}) = (1 + w)\epsilon_\theta(x_t, \mathbf{c}) - w\epsilon_\theta(x_t) \tag{2.10}$$

Intuitively, the diversity comes from the unconditional model, whereas the quality and alignment with the conditions come from the conditional model. Ho et al. [20] propose training the unconditional model (i.e. inputting a null token) with probability p_{uncond}, which they find gives the best performance when $p_{uncond} \in \{0.1, 0.2\}$. At sample time, small weights ($w = 0.1$ or $w = 0.3$) provide the best results for diversity, whereas large weights ($w \geq 4$) give the highest quality results [20].

Very large guidance weights further increase alignment between the image and the condition, but harm image quality, resulting in over-saturated and unnatural images [20]. Imagen [12] fixes this by thresholding the image dynamically, normalising the image to whatever range is present in the predicted image. This significantly improves photorealism when using large weights [12].

2.8 Text to Image Diffusion Models

Whilst not strictly relevant to generating biopsy images conditioning on non-textual data, I will very briefly mention the Imagen [12] text-to-image diffusion model as it culminates all the background presented. `imagen-pytorch` [17], the library I will be using, is based on the architecture in this paper.

Imagen [12] uses a set of three cascaded diffusion models, starting with a base model that generates 64x64 images, and then using two super-resolution models to upscale to 256x256 and then 1024x1024. Conditioning augmentation is used in super-resolution networks to improve sample quality.

`imagen-pytorch` [17] is an unofficial open-source implementation of the architecture and techniques used to create Imagen [12] with the groundwork laid out to create these models. It offers most of what is required to make this project possible, including an interface to create U-Nets, support for cascaded diffusion models with multiple U-Nets, and basic inpainting. Since it is based on Imagen [12], it expects textual conditioning in the form of text embeddings. This input can be trivially changed into any arbitrary embedding, or even removed for unconditional image generation, making this a non-issue.

2.9 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) [21] were introduced as a new framework for data generation where two models are trained at the same time, one that generates fake data, G, and another that classifies images as either generated or real, D. These are called the generator and discriminator respectively. During training, G tries to maximise the chance that D makes a mistake, leading to a two-player minimax game between the models [21].

One of many improvements to the original architecture presented was the progressive growth of GANs to improve their quality, stability, and variation [22]. This works by growing both the generator and discriminator from smaller models to larger models - new layers are added to each, increasing their resolution during training [22].

StyleGAN [23] builds on top of this progressive architecture by improving the generator model. StyleGAN removes the traditional random latent code used as the input into the generator and replaces it with a vector representing the style of the image. This style vector is generated by sampling a latent code, like before, but then passing it through a fully-connected neural

network. The style is then combined at each stage of the progressive GAN to incorporate it into the generation of the image. Since the style is used at each stage, it can learn representations at different scales - for example skin colour at a coarse scale and eye colour at a finer scale [23]. Noise is also injected into each layer to increase variety. This along with other smaller changes, led to state-of-the-art performance for StyleGAN [23].

Incremental improvements between StyleGAN [23], StyleGAN2 [24], and StyleGAN3 [25] led to state-of-the-art image synthesis performance at the time, with diffusion models then achieving state-of-the-art later [6]. Because of this, using StyleGAN3 is sensible as a baseline model architecture to compare image quality and diversity against using the FID metric [26]. This is needed given that the kidney dataset described in Section 5.1.1 used in this project is private and has no baseline results.

2.10 Fréchet Inception Distance

The Fréchet Inception Distance [26] (FID) is a metric to evaluate the image quality and diversity of generated images when compared to their real counterparts. This is used as a standard in evaluating new generative models such as Imagen [12] and DALL-E 2 [11].

Calculating the FID is most useful when it is compared to benchmarks of other generative models on a public dataset, as it allows for a direct quantitative comparison of the quality and diversity of images generated by different models. Once an FID score is calculated for a baseline generative model, improvements in different models can be quantified.

The FID compares two Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$, and $\mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$, where r and g represent real and generated images respectively. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are found by fitting to latent-space feature vectors of real images and generated images. The feature vectors are typically generated by inputting the real and generated images into a pre-trained Inception v3 [27] model trained on the ImageNet [28] dataset. The Fréchet Inception Distance [26] is therefore defined as follows:

$$\text{FID}(\mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r), \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)) = \|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\|^2 - \text{tr}(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r \boldsymbol{\Sigma}_g)^{\frac{1}{2}}) \tag{2.11}$$

For two identical sets of images, i.e. comparing real images with themselves, the FID score will be 0. This means the closer to 0 that the FID is, the better the image quality is.

2.11 Diffusion Models for Medical Image Synthesis

There has been success using diffusion models to generate synthetic medical images, for example in generating brain images [13], and in histopathology [2], which is more relevant to this project. Both point to the lack of required medical datasets for training complex models. Specifically, Öttl et al. [2] mention the class imbalance present when segmenting tumours in histopathology images, something relevant to renal histopathology as cell structures such as glomeruli are significantly rarer than tubules in the kidney dataset used in this project. Öttl et al. [2] condition the image generation on a segmentation mask, allowing the segmentation masks for underrepresented features to be used to generate more images for that class. Using this to create a more balanced dataset and re-training improved performance in their segmentation model. In addition, Pinaya et al. [13] successfully condition the image generation on various clinical parameters, including age, sex, and brain structure volumes.

Chapter 3

Unconditional and Segmentation-Mask-Conditioned Image Generation

This chapter details a novel method of generating realistic synthetic images and corresponding segmentation masks to enrich existing datasets. The goal is to improve downstream performance by training a segmentation model on both real and synthetic data. We start with generating images that are not conditioned on any segmentation mask (but may or may not be conditioned on clinical parameters, such as the type of tissue present), as this is a necessary first step. Then, we proceed by utilising a pre-trained baseline segmentation model to segment these images. Finally, using both the segmentation masks generated, we can generate unseen variations of images using the same segmentation mask, including by changing the clinical parameters used to generate the image.

For clarity, much of this chapter details prior work completed as part of this project by myself and Sarah Cechnicka [29]. I worked on developing and training the diffusion models, and Sarah worked on the segmentation models and evaluated the diffusion models on the downstream segmentation task using the KUMAR dataset [30]. For the purposes of this report, I will be presenting much of what is in this paper that was completed as part of this project, with a focus on diffusion models.

3.1 Problem Statement

Annotating large Whole Slide Images (WSI) is an extremely laborious task carried out by expert pathologists, motivating the automatic annotation of histopathology images using segmentation models such as U-Nets [14]. Such models benefit from large datasets, which geometric image augmentation can try and emulate but cannot replace. Geometric image augmentation is typically limited to basic operations such as flipping and rotating, with the assumption that the image is still sensible after augmentation. Our hypothesis is that enriching data-limited datasets with synthetic images generated using diffusion models conditioned on segmentation masks will provide better performance than traditional image augmentation methods.

There are two separate measures of success for this chapter:

Realistic-looking images generated. The images generated by both the unconditional and segmentation-mask-conditioned models should be close in realism to real images. This will be evaluated by having expert pathologists compare generated images to real images, as well as by comparing baseline results from a StyleGAN3 [25] model.

Improved performance on downstream segmentation task. Using synthetic image

and segmentation mask pairs to supplement an existing real dataset must provide a performance advantage for this to be successful. This will be evaluated by training a baseline segmentation model trained on real data, and then generating images and retraining the segmentation model using synthetic data as well as real data and comparing performance metrics.

3.2 Proposed Method

Let us denote the image space as $\mathcal{X}$ and the segmentation mask space as $\mathcal{Y}$. The goal is to generate realistic but new pairings within the joint space $\mathcal{X} \times \mathcal{Y}$. First, a baseline segmentation model is considered $M_\theta : \mathcal{X} \to \mathcal{Y}$ that performs pixel-wise classification.

Ideally, we would perform an inversion of this baseline model, but practically inverting the segmentation model to $p(\boldsymbol{x}|\boldsymbol{y}, \theta)$ is not possible, as the transformation M_θ is not bijective, and inverting it would yield many plausible samples from $\mathcal{X}$. However, the inversion can be modelled by sampling single realistic images $\hat{\boldsymbol{x}} \in \hat{\mathcal{X}}$ given $\boldsymbol{y} \in \mathcal{Y}$ and additional random noise $z \sim \mathcal{N}(0, \sigma)$ holding the random state of our generative process. Modelling this approach can be achieved through diffusion probabilistic models [7].

Therefore, we can define $D_\phi : \mathcal{Z} \to \hat{\mathcal{X}}$, where $\mathcal{Z}$ is a set of Gaussian noise samples, as our unconditional diffusion model that generates realistic images without a segmentation mask. A further model can then be conditioned on segmentation masks $\boldsymbol{y}$ to produce matching elements to the joint space $\mathcal{X} \times \mathcal{Y}$ yielding $D_\xi : \mathcal{Z} \times \mathcal{Y} \to \hat{\mathcal{X}}$.

The first step of our approach, shown in Figure 3.1, is to generate a set of images $X_1 = \{\boldsymbol{x}_n^{(1)} | \boldsymbol{x}_n^{(1)} = D_\theta(z), z \sim \mathcal{N}(0, \sigma)\} \subset \hat{\mathcal{X}}$ where D_θ is an unconditional diffusion model trained on real data samples. We then map all samples $\boldsymbol{x}_n^{(1)}$ to the corresponding elements in the set of predicted label masks $Y_1 = \{\boldsymbol{y}_n^{(1)} | \boldsymbol{y}_n^{(1)} = M_\theta(\boldsymbol{x}_n^{(1)}), \boldsymbol{x}_n^{(1)} \in X_1\} \subset \hat{\mathcal{Y}}$, where M_θ is the baseline segmentation model trained on real data pairs. This creates a dataset denoted d_1.

The second step is to generate a dataset d_2 of segmentation masks and images, using a diffusion model conditioned on the synthetic segmentation masks, Y_1. This diffusion model is D_ξ, which is trained on real images and segmentation masks and applied to the data pairs in d_1, such that $X_2 = \{\boldsymbol{x}_n^{(2)} | \boldsymbol{x}_n^{(2)} = D_\xi(\boldsymbol{y}_n^{(1)}, z), \boldsymbol{y}_n^{(1)} \in Y_1, z \sim \mathcal{N}(0, \sigma)\}$. This lets us generate a much larger and more diverse dataset of image-label pairs, where the images are generated from the labels.

Our final step is to use this dataset to train a new segmentation model M_ζ that largely outperforms M_θ. To do so, we evaluate performance through various training strategies, including first training M_ζ on the synthetic dataset d_2 and fine-tuning it on the real dataset.

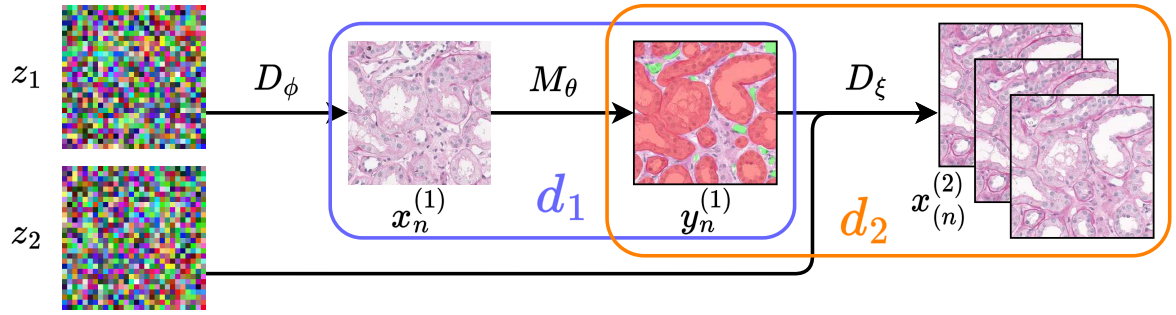


Figure 3.1: Pipeline for generating a synthetic dataset with paired images and segmentation masks as described in Section 3.2. Diffusion model D_ϕ generates images unconditionally, M_θ segments them, and D_ξ creates multiple images from these segmentations. Dataset d_2 is the one used to train our final model M_ζ. (Cechnicka et al. [29])

3.3 Implementation

3.3.1 Dataset Pre-processing

The first dataset we trained and evaluated our models on is the public KUMAR dataset [30] detailed in Section 5.1.2. Patches in this dataset are randomly cropped to 256×256, so only a base 64×64 model and $64 \times 64 \rightarrow 256 \times 256$ model are used for this dataset. Along with random crops, the images are flipped and rotated when training the diffusion model. When training the segmentation model, rotation, flipping, colour shift, random cropping, and elastic transformations are all used to augment the images. It is worth noting that the baseline methods [31] only use 16 of the 30 images available for training. The labels for tissue and cancer type are additionally used to condition image generation in the diffusion model.

The second dataset is a private dataset of kidney whole slide images, detailed in Section 5.1.1. When training the diffusion model, patches of 1024×1024 pixels are taken from the whole slide image. Each patch is flipped and rotated, and also shifted meaning that some patches in the dataset overlap. Whole slide images have a lot of whitespace which is filtered out in a pre-processing step so that the model learns only to synthesise patches that are valuable to downstream tasks. Similar augmentations are made to the segmentation masks when training the segmentation-mask-conditioned models.

For both datasets, the images are resized down to the dimension of the model being trained. For example, for the base model, the patches are resized to 64×64 pixels and for the $64 \times 64 \rightarrow 256 \times 256$ model, the patches are resized to 64×64 for the low-resolution conditioning image, and 256×256 for the output.

3.3.2 Models and Training

We use a cascaded diffusion model similar to Imagen [12] with three stages, starting with a base model that generates 64×64 images, and then two super-resolution models to upscale to resolutions 256×256 and 1024×1024. The model trained on the KUMAR dataset [30] does not have the last super-resolution model since it generates 256×256 images. Conditioning augmentation [9] is used in super-resolution networks to improve sample quality.

In contrast to [12], we experimented with using v-parameterization [32] to train the super-resolution models. This is instead of a typical noise-parameterized model [7]. The super-resolution models, particularly the $256 \times 256 \rightarrow 1024 \times 1024$ model, are much more computationally demanding to train and sample. I found that using v-parameterization allowed as few as 256 sampling steps instead of 1024 in the noise prediction setting and trained faster. Despite this, when experimenting further with noise prediction, although early convergence was significantly worse, performance using either seemed to converge when training for longer periods of time even at 256 sampling steps. The unconditional and segmentation-mask-conditioned results presented in Section 5.5 are trained using v-parameterization, whereas the unconditional image generation in Section 5.4 is evaluated on both v-parameterization in the first diffusion model and noise-parameterization in the final better-performing diffusion model.

We use PyTorch with three Nvidia A5000 GPUs to initially train and evaluate our diffusion and segmentation models, along with three Nvidia A100 GPUs to improve upon the unconditional model later on. The kidney study segmentation models were trained for 200 epochs and fine-tuned for 25, the KUMAR study used 800 epochs and was fine-tuned for 100. Training takes about 10 days with this setup. Where real data was used for fine-tuning this was restricted to 30% of the original dataset. Diffusion models were trained with a learning rate of $1e{-}4$ and segmentation models were pre-trained with a learning rate of $1e{-}3$ which dropped to $3e{-}6$ when no change was observed on the validation set in 15 epochs. All models used Adam optimiser.

When training the segmentation-mask-conditioned model, there was much less training data available since annotating whole slide images is very time-consuming and must be carried out by expert pathologists. As a result, we first pre-trained a model using the unannotated dataset with blank segmentation masks. This allows the model to take advantage of a much bigger dataset. After pre-training, the model is fine-tuned on the smaller annotated dataset so that the model learns to associate the segmentation masks with structures it learns how to generate in pre-training.

Two unconditional models were trained for the kidney dataset. Both are evaluated in Section 5.4, but only the first model is evaluated on the downstream segmentation task. The first unconditional model uses all of the hyperparameters listed in Tables 3.1 and 3.2, but instead with U-Net 1 `dim_mults` as (1, 2, 3, 4), and `pred_objectives` as (noise, v, v). The first model is also conditioned on patient information, such as the patient outcome, but this led to poor diversity even when generating images unconditionally. The same hyperparameters were used when training using the KUMAR dataset [30] as in the first model, with the removal of U-Net 3 since only 256×256 pixel images were generated. The segmentation-mask-conditioned models were trained by fine-tuning the first unconditional models for both the kidney and KUMAR datasets. The second and final unconditional model has hyperparameters listed in Tables 3.1 and 3.2.

Hyperparameter	**Value**
`dim`	256
`dim_mults`	(1, 2, 4, 8)
`cond_dim`	512
`num_resnet_blocks`	3
`layer_attns`	(F, T, T, T)
`layer_cross_attns`	(F, T, T, T)
`init_conv_to_final _conv_residual`	False

(a) U-Net 1 Hyperparameters

Hyperparameter	**Value**
`dim`	128
`dim_mults`	(1, 2, 4, 8)
`cond_dim`	512
`num_resnet_blocks`	2
`layer_attns`	(F, F, F, T)
`layer_cross_attns`	(F, F, T, T)
`init_conv_to_final _conv_residual`	True

(b) U-Net 2 Hyperparameters

Hyperparameter	**Value**
`dim`	128
`dim_mults`	(1, 2, 4, 8)
`cond_dim`	512
`num_resnet_blocks`	(2, 4, 4, 4)
`layer_attns`	False
`layer_cross_attns`	(F, F, F, T)
`init_conv_to_final _conv_residual`	True

(c) U-Net 3 Hyperparameters

Table 3.1: Hyperparameters used for U-Nets in the final unconditional diffusion model. Hyperparameter names are the same as those used in `imagen_pytorch` [17] for clarity. Any hyperparameters not mentioned are set to defaults. F and T mean False and True respectively.

Hyperparameter	Value
`image_sizes`	(64, 256, 1024)
`timesteps`	(1024, 256, 256)
`pred_objectives`	(noise, noise, noise)
`random_crop_sizes`	(None, None, 256)

Table 3.2: Global hyperparameters/settings used for the final unconditional diffusion model. Names are the same as those used in `imagen_pytorch` [17] for clarity. Any hyperparameters not mentioned are set to defaults.

For the kidney dataset, the first unconditional diffusion model was trained for 173,000 steps on U-Net 1, 348,000 steps on U-Net 2, and 185,000 steps on U-Net 3. The fine-tuned segmentation-mask-conditioned models were trained for a further 145,000 steps on U-Net 1, 143,000 steps on U-Net 2, and 43,000 steps on U-Net 3. The second and final unconditional diffusion model was trained for significantly longer at 1,734,000 steps on U-Net 1, 1,671,000 steps on U-Net 2, and 1,398,000 steps on U-Net 3. The models trained on the KUMAR dataset were trained for 171,000 steps on U-Net 1, and 286,000 steps on U-Net 2.

For the kidney dataset where there are three U-Nets and sampling at 1024×1024 pixels, the unconditional and segmentation-mask-conditioned models take around 1 minute and 10 seconds to sample a single image on a single A100 GPU. This is much longer than other diffusion models, such as Stable Diffusion [10], but this is to be expected given that no model distillation is used here. Much faster sampling should be possible without any noticeable loss in performance using distillation [32], but this is not a focus of this project.

Chapter 4

Ultra-Resolution Cascaded Diffusion Models (URCDMs)

4.1 Problem Statement

Generating high-resolution images is already a challenge, with 1024×1024 being the typical resolution that a high-resolution generative model will stop at [9]. This is due to several constraints, such as having less training data at very high resolutions, GPU memory, sampling time, and model complexity. This makes generating ultra-resolution images, images that are over a gigapixel in size, dramatically more complicated: a $10,000 \times 10,000$ gigapixel image with $1,000,000,000$ pixels has approximately 1000 times the number of pixels as a high-resolution 1024×1024 image.

Section 5.5 shows that smaller datasets can exhibit great improvements in performance on downstream tasks when enriching the datasets with synthetic images generated using diffusion models. Given that very high-resolution image datasets naturally have fewer images, there is reason to believe that they would also benefit from being enriched with synthetic images to improve performance.

Prior work from Chai et al. [5] achieves a similar goal using GANs without using additional layers in the generator to iteratively upscale the image. However, Ho et al. [9] show that better results are achieved for diffusion models when using separate U-Nets for each increase in resolution. Inspired by this approach, it makes sense to experiment with a variant of the work by Chai et al. [5] and Ho et al. [9] and apply it to the even higher-resolution domain of histopathology whole slide images, which are discussed more in Section 5.1.1. Given the limited prior work in synthetic gigapixel imagery, especially when using diffusion models, this work is much more experimental and will assess whether this is possible at a proof-of-concept level, whilst offering many ways that it could be improved in future work in Section 6.2.

Ultra-resolution histopathology images are usually split into patches to increase the amount of training data and computational complexity. However, pathologists typically analyse Whole Slide Images in their entirety; zooming in and out at multiple scales. Synthesising ultra-resolution imagery could allow for more complex downstream algorithms that operate on the entire image at different scales due to the additional training data it provides. Algorithms that operate on very high-resolution images at different scales, such as You Only Look Twice [1] which is applied to satellite imagery, will benefit greatly from both the long-distance spatial coherency and high-quality fine details that ultra-high resolution synthetic images can provide.

There are two separate measures of success for this chapter:

Long-distance spatial coherency. The main goal of this chapter is to show that coherent image generation at this scale is possible so that it can be feasibly used to enrich

datasets in the same way as the previous chapter. Due to a lack of real ultra-resolution images, and long sampling times for synthetic images, this will largely be evaluated qualitatively by comparing real and synthetic images visually. Patch-FID [5], explained in Section 5.2 will also partially evaluate this since it computes the FID at multiple scales, including full scale at a low resolution.

Realistic-looking at multiple scales. Generated ultra-resolution images must look realistic both when 'zoomed-out' and when 'zoomed-in'. Again, qualitative evaluation is most important here, but Patch-FID [5] emulates this well by computing the FID at multiple scales and positions.

Both of these measures of success can additionally be compared against outpainting as a baseline, which is possible with a standard unconditional diffusion model. Furthermore, expert pathologist evaluation will help assess the realism of the synthetic images.

4.2 Proposed Method

Ultra-Resolution Cascaded Diffusion Models (URCDMs) are based on similar principles as Cascaded Diffusion Models (CDMs) [9], in which there are three stages of increasing resolution. Figure 4.1 shows the general architecture of a URCDM, in which multiple CDMs generate high-resolution images at different magnification levels.

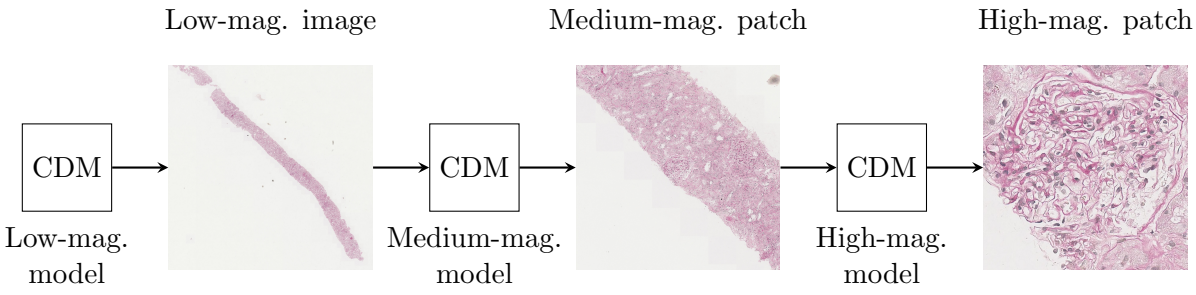


Figure 4.1: Ultra-Resolution Cascaded Diffusion Model (URCDM) architecture for generating a single patch of an ultra-resolution image. Each image generated conditions the generation of the higher magnification patch.

URCDMs start with a low-magnification model that unconditionally generates the entire image at a resolution much lower than the full-resolution image. Using this image, the next medium-magnification model is conditioned on the low-magnification image. This model is responsible for generating a medium-magnification patch of the centre of the low-magnification image. Then, we can stitch many patches together to form a low-magnification, but much higher-resolution, image.

To achieve this, the lower-magnification conditioning image is shifted in a way such that the patch that we want to generate is in the centre of the higher-magnification image. This means that if we want to generate a medium-magnification patch of the top left of the low-magnification image, we must shift the image that conditions generation such that the top-left is in the centre, and pad or rollover any of the gaps left by shifting the image.

Once a higher-resolution image has been stitched together using the low and medium-magnification models, we can use the high-magnification model to repeat a similar process and generate high-magnification patches at each point in the higher-resolution image. Once all high-magnification patches have been generated using the corresponding conditioning images, they can be stitched together in the same way as before. This creates an ultra-resolution image with long-distance coherency.

Intuitively, since higher-magnification patches are generated using the wider context of lower-magnification patches, ultra-resolution images that are contextually accurate over many pixels should be possible. This is in stark contrast to achieving the same result with outpainting [4] where there is no context or knowledge of the image outside of the area the patch is being generated. Using the kidney transplant dataset in Section 5.1.1 as an example, this means that the overall structure of the whole slide image remains coherent, and it also looks coherent when zoomed in. Using outpainting would only allow for the image to look coherent when zoomed in.

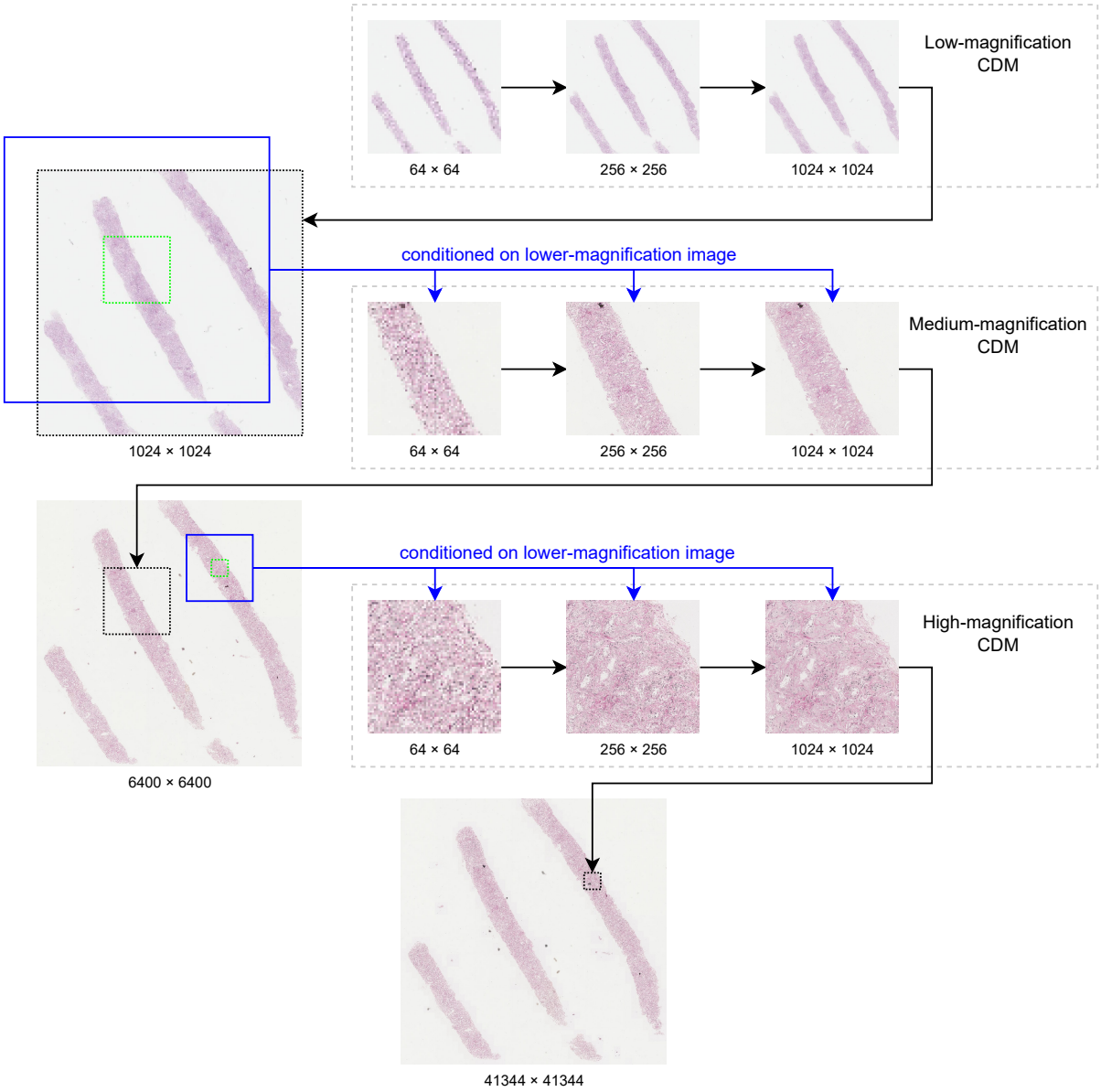


Figure 4.2: Detailed overview of the Ultra-Resolution Cascaded Diffusion Model (URCDM) image generation process. The medium and high-magnification CDMs are sampled many times, and the patches generated are stitched together. This diagram only shows one patch from the medium and high-magnification images being generated. A blue outline indicates the lower-magnification conditioning image. A green outline indicates the resultant patch that will be 'zoomed in' on and generated. Not to scale.

Figure 4.2 gives a detailed look at the image generation process in practice, showing each of the three CDMs, each with three diffusion models. This highlights one difficulty in image generation,

which is when patches are generated near the bounds of the lower-magnification image. The lower-magnification conditioning image can go outside the bounds of the original image, which means those pixels must be padded. In practice, the value that you pad this with is dataset-dependent. The datasets we are working with have lots of cream-coloured whitespace making it a sensible colour to pad with.

4.3 Implementation

4.3.1 Models and Training

In the Ultra-Resolution Cascaded Diffusion Model, there are 9 diffusion models in total, from three separate Cascaded Diffusion Models (CDMs). None of the models are dependent on another so they are trained in parallel on 9 separate Nvidia A100 GPUs. The model architecture and training are heavily based on Imagen [12] using the `imagen-pytorch` [17] library. I am using PyTorch version 2.0.0 and `imagen-pytorch` version 1.18.5.

Each CDM targets a different magnification of the overall image. The first CDM is a full-scale but low-resolution image, taking a $40,000 \times 40,000$ pixel crop of the kidney image and resizing this to 1024×1024. The second model takes 6500×6500 crops and resizes this to 1024×1024. Finally, the last model takes full-magnification images and doesn't resize them, taking 1024×1024 crops. 6500×6500 was arbitrarily chosen as being approximately 6 times larger than 1024, and 6 times smaller than 40,000, placing it in the middle of both in terms of magnification. The URCDM is not restricted to these values though, and they can be trivially changed to suit the dataset.

I also experimented with removing the base 64×64 U-Net for the medium and high magnification models and instead using a centre crop of the lower-magnification image to serve as the low-resolution input into the super-resolution networks. This would be beneficial as it could increase coherency between the CDMs and speed up sampling. Unfortunately, this resulted in blurry images that were much lower quality. This could be because the base model is much larger and so it has a better capacity for generating high-quality images than the super-resolution models in the previous layer, or because previous CDMs are generating images that are out-of-distribution when compared to what the higher magnification CDMs have been trained on.

Much work went into finding the best set of hyperparameters to use when training these models. Training in general was much less stable when compared with the unconditional diffusion models in Section 3, and spurious infinite or NaN losses would halt training after training for a few hundred thousand steps. I only noticed this behaviour on the larger $256 \times 256 \rightarrow 1024 \times 1024$ models, and it was dramatically worse when training this model using noise-parameterisation instead of v-parameterisation. To combat this and stabilise training, I employed gradient clipping, setting the max gradient norm to 1. In addition to stability issues, URCDM models trained using noise-parameterisation in the super-resolution networks tend to have much lower quality fine details, and would often blur or heavily distort lower magnification images when 'zooming in'. This led to v-parameterisation being the better choice for training super-resolution networks in URCDMs.

Hyperparameter	Value
dim	256
dim_mults	(1, 2, 3, 4)
cond_dim	512
num_resnet_blocks	3
layer_attns	(F, T, T, T)
layer_cross_attns	(F, T, T, T)
init_conv_to_final _conv_residual	False
cond_images_channels	3 (0 for low mag)

(a) U-Net 1 Hyperparameters

Hyperparameter	Value
dim	128
dim_mults	(1, 2, 4, 8)
cond_dim	512
num_resnet_blocks	2
layer_attns	(F, F, F, T)
layer_cross_attns	(F, F, T, T)
init_conv_to_final _conv_residual	True
cond_images_channels	3 (0 for low mag)

(b) U-Net 2 Hyperparameters

Hyperparameter	Value
dim	128
dim_mults	(1, 2, 4, 8)
cond_dim	512
num_resnet_blocks	(2, 4, 4, 4)
layer_attns	False
layer_cross_attns	(F, F, F, T)
init_conv_to_final _conv_residual	True
cond_images_channels	3 (0 for low mag)

(c) U-Net 3 Hyperparameters

Table 4.1: Hyperparameters used for U-Nets in the final URCDM trained on kidney data. Hyperparameter names are the same as those used in `imagen_pytorch` [17] for clarity. Any hyperparameters not mentioned are set to defaults. F and T mean False and True respectively.

Hyperparameter	Value
image_sizes	(64, 256, 1024)
timesteps	(1024, 256, 256)
pred_objectives	(noise, v, v)
random_crop_sizes	(None, None, 256)
max_grad_norm	1

Table 4.2: Global hyperparameters/settings used for the final URCDM trained on kidney data. Names are the same as those used in `imagen_pytorch` [17] for clarity. Any hyperparameters not mentioned are set to defaults.

Table 4.3 shows the number of steps taken to train each U-Net within the URCDM. The low-magnification model was trained for the least amount of time, as it was being trained on very few low-magnification images, so the distribution of images was much less broad. The most important models to train for a long period of time were the first U-Nets in the medium and high-magnification models. I speculate as to why this is important in Section 5.6.2.

U-Net	Training Steps
1	475,000
2	450,000
3	375,000

(a) Low-Magnification Model

U-Net	Training Steps
1	2,450,000
2	450,000
3	200,000

(b) Medium-Magnification Model

U-Net	Training Steps
1	1,125,000
2	1,550,000
3	325,000

(c) High-Magnification Model

Table 4.3: Number of steps that each network was trained for during training of the kidney URCDM.

4.3.2 Outpainting and Parallel Processing

Outpainting is used to smoothly merge generated patches together with minimal seams between patches, so there is a dependency between neighbouring patches as each patch must slightly overlap with each other so that they can see the edges of their neighbours when sampling. Figure 4.3 shows this dependency, along with the lower-magnification condition images that the model is 'zooming-in' on, and the inpainting images showing the overlapping neighbouring patches used to generate each patch.

The first patch in Figure 4.3 is the top-left of the lower-magnification image. The second and third patches are below and next to the first patch, respectively, and both require the first patch to be generated before them since they use the first patch for the inpainting image. Since the first three patches are on the corner or edges of the image, they don't have any other neighbouring patches. Finally, since the fourth patch is not on an edge, it has three neighbouring patches in the top-left direction, so all of these must be generated before the fourth patch's inpainting image is known. This dependency between patches is especially important when batch-processing images or sampling on multiple GPUs as it determines when patches can be generated in parallel, something essential for reasonable sampling speeds.

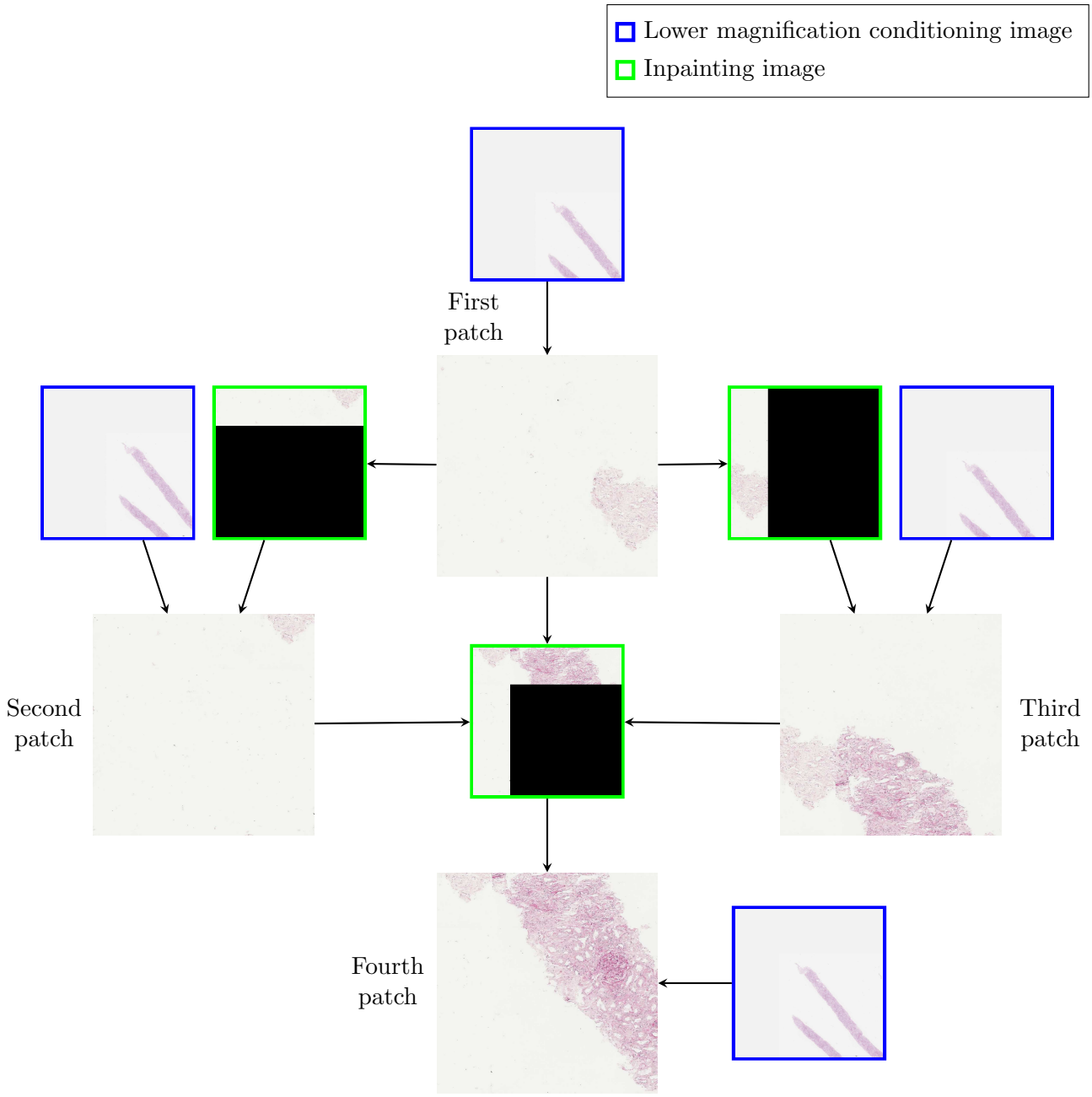


Figure 4.3: Order that patches need to be generated when using outpainting with an Ultra-Resolution Cascaded Diffusion Model (URCDM). To merge images without visible seams between patches, the diffusion model has access to the edges of neighbouring patches, shown in the 'inpainting image' for each patch. Every patch is conditioned on a lower magnification conditioning image that gives the model a wider context during image generation. The model outputs a 'zoomed-in' view of each conditioning image in the centre. Patches in this example have 25% overlap.

Figure 4.4 shows the section of each patch generated that is 'fresh' and not overlapping with previously generated patches. The first patch generated has no neighbouring patches so the entire patch is new, whereas all other patches overlap with one or many other patches during inpainting, so a smaller portion of the generated patch is new. The result of this is that the more that patches overlap, the smaller the region that is newly generated is, so more patches must be generated.

When choosing how much the patches overlap, there is a trade-off between the time taken to

generate an ultra-resolution image and patching artifacts. With no overlap, patching artifacts are obvious and you can clearly see the boundary between generated patches. In practice, I found that when 12.5% of the patches overlap with their neighbours (both in the vertical and horizontal direction), artifacts were minimal and the number of patches was not dramatically increased. I saw little to no increase in quality when increasing the overlap further.

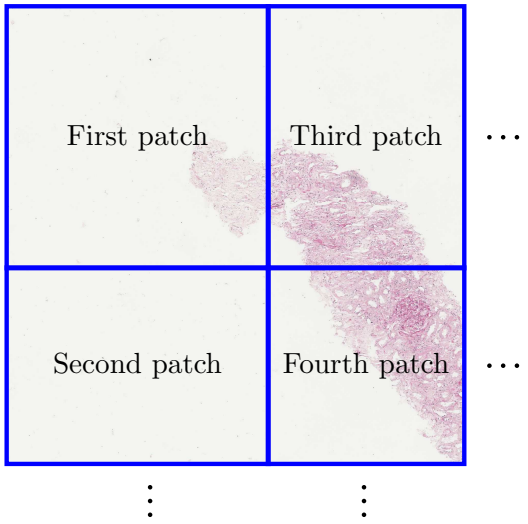


Figure 4.4: Small section of higher-magnification image generation, showing the portion of each patch generated that is new and doesn't overlap with previously-generated patches as a result of outpainting. The first patch in the top-left is in the corner of the image, and has no neighbouring patches to the left and above, so the entire patch is newly generated. The second and third patches are on the edges, so overlap above and to the left with the first patch respectively. Finally, the fourth patch overlaps with all three other patches so it contributes the least to the overall image. Patches in this example have 25% overlap.

When sampling on multiple GPUs, one process is started for each GPU, and each process is given access to a shared dictionary in which all generated patches are stored, along with a queue of patches that are still left to generate. Because access to neighbouring patches is required for outpainting, patches can only be generated once the patch next to, above, and above and next to, are all generated *or* there is no patch in that direction because it is on the edge. Therefore, each process dequeues a patch position to process and checks whether the patch can be generated. If the patch can be generated, it is generated and then added to the shared dictionary so that other processes can access it. If the patch cannot be generated, it is added back to the queue and the process dequeues the next patch position. At the start, only one process can run as all patches depend on the first patch directly or indirectly, but quickly as a few patches have been generated, parallelism/batch-processing becomes possible.

Due to the added complexity, I did not implement sampling patches in batches as sampling times were acceptable without this. However, this is something that is likely to considerably improve sampling speeds as generating several images on a single GPU at once is significantly faster overall than generating one image at a time.

4.3.3 Dataset Pre-processing

Since histopathology whole slide images are mostly whitespace, there is little reason to generate patches at very high magnification for areas with lots of whitespace. Patches at higher magnification take up less of the overall image, so there are more patches and therefore more potential for patching artifacts. As a result, high-magnification patches of the whole slide image that are

mostly white are ignored in both training and when sampling. Instead, any white patches are replaced with an upscaled version of the medium-magnification image. This is a practical step to speed up sampling that is specific to this dataset that could be skipped for other datasets where the entire image is useful, such as satellite imagery.

When generating whole slide images, this creates minor patching artifacts near the edges of patches without whitespace since the medium-magnification and high-magnification models have slightly different 'white' colours. You can see this near the edges of the kidney tissue in Figure 5.9a.

Whole slide images vary significantly in size and are non-square, unlike the output of this model. Therefore, another pre-processing step made was to crop images to $40,000 \times 40,000$ pixels. The images in the dataset vary between around 10,000 and 70,000 pixels in each dimension, with most being less than 40,000, hence this choice. For images that are smaller than $40,000 \times 40,000$ and cannot be cropped to this, I instead padded the images with the background colour of the whole slide image and moved the images to the centre of the modified $40,000 \times 40,000$ image. The scale of each image must be kept the same, which is why resizing is not an option. This pre-processing step is unlikely to be necessary in domains such as satellite imagery where images are usually square patches, like in the AIRS dataset [33].

Chapter 5

Evaluation

5.1 Datasets

5.1.1 Renal Histopathology Whole Slide Image Dataset

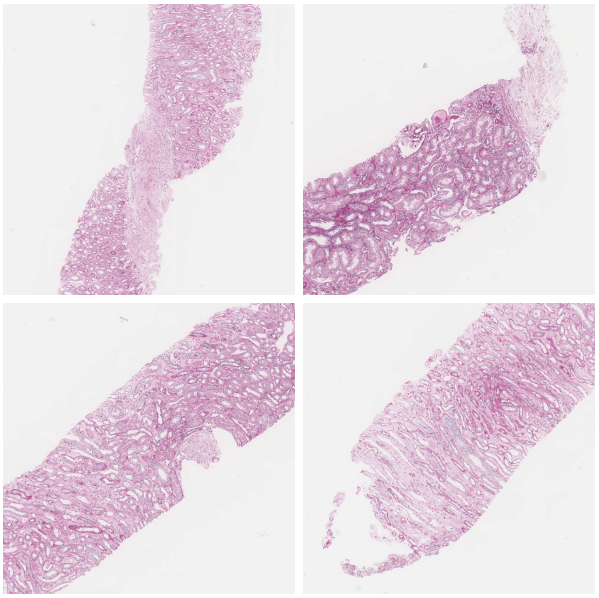
Renal histopathology is the analysis of kidney cells and tissue for signs of non-cancerous disease. Pathologists traditionally analyse slides containing tissue under a digital microscope, but more recently use digital slide scanners to take high-resolution and high-magnification images of tissue.

The kidney histopathology dataset is a proprietary dataset of around 400 patients who have received kidney transplants, each with one or more Whole Slide Images that have been taken from a biopsy of their kidney. Patients also have an outcome that indicates whether their kidney transplant has been successful or not to varying degrees.

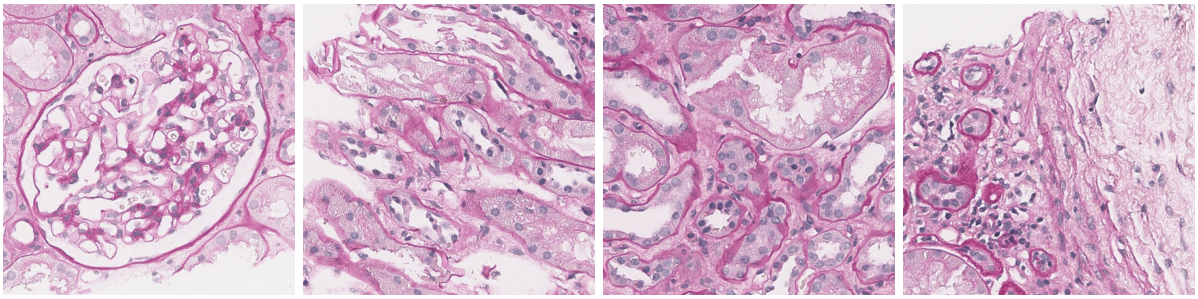
This dataset contains 428 Whole Slide Images. Resolution and file size can vary significantly from image to image, but this dataset contains images with widths and heights ranging from approximately 10,000 pixels to 70,000 pixels, and file sizes ranging from 75MB to 850MB. In this case, the tissue has been extracted from a kidney as part of a biopsy. Due to the extreme size of a WSI, it makes sense to split the image into smaller resolution and fixed-size patches, like in Figure 5.1c. A patch size of 1024×1024 pixels has been chosen as this was used when pathologists made the annotations for the WSIs, and so it is compatible with the annotation dataset, as well as being large enough for pathologists to make out distinct features in the image. Splitting the image into patches also provides significantly more data to generate images with, as each image can be split into hundreds of patches. Much of a WSI is white space where the tissue isn't present. This is uninteresting from an image synthesis perspective, so these are filtered out during training. After filtering, this dataset has a train/valid/test split of 54359/1394/759 patches. The unconditional models discussed in Section 5.4 are trained on a smaller set of patches with more aggressive whitespace filtering, having a train/valid/test split of 42779/1097/634 patches. Additionally, since WSIs don't have an orientation and are rotation-invariant, the patches can be flipped and rotated to maximise the utility of each patch.

Finally, there are annotations of some WSI patches that have been made by pathologists. These are stored on a service called Labelbox [34] and were originally used for a prior segmentation model trained using this data, but can also be repurposed for this project. 1024×1024 pixel patches were taken from the same dataset of WSIs and then given to pathologists to annotate, indicating types of structures like glomeruli and tubules, as well as regions of the kidney like the cortex and medulla. This dataset contains 1459 1024×1024 pixel patches with corresponding segmentation masks, with a train/valid/test split of 1295/34/130 patches.

(a) Full-scale whole slide image



(b) Partially zoomed-in crops of the image in 5.1a



(c) Highly zoomed-in crops of the image in 5.1a

Figure 5.1: Real whole slide image from kidney dataset

5.1.2 KUMAR

KUMAR is a public histopathology dataset [30] consisting of 30 WSI training images and 14 test images of 1000 × 1000 pixels. Each image has corresponding labels for tissue and cancer type (`Breast`, `Kidney`, `Liver`, `Prostate`, `Bladder`, `Colon`, and `Stomach`). Some sample images from this dataset can be seen in Figure 5.2

This dataset is used only to evaluate the segmentation-mask-conditioned diffusion models. It allows us to compare our method with the state-of-the-art directly. It is also a much smaller dataset where generating synthetic training data is likely to be more beneficial.

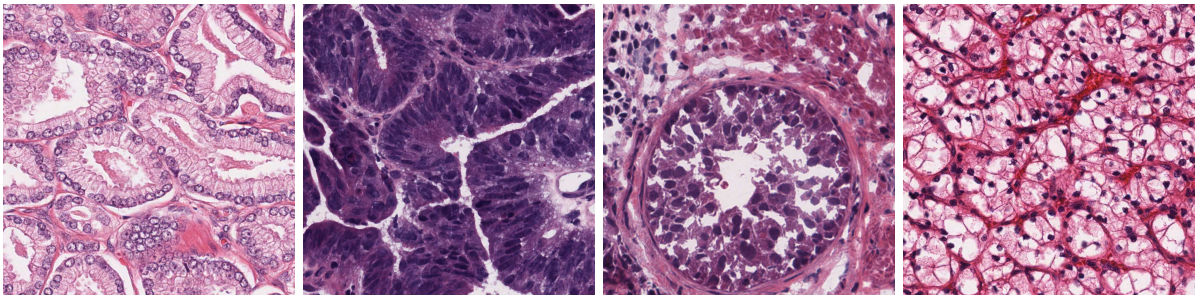


Figure 5.2: Real sample images from KUMAR [30] dataset.

5.2 Evaluation Plan

Overall, we evaluate this project in four separate ways:

- Unconditional diffusion models are evaluated by calculating FID scores and comparing them with a baseline StyleGAN3 [25] model.
- Segmentation-mask-conditioned diffusion models are evaluated by assessing the performance of an enriched dataset on a downstream segmentation task.
- URCDMs are evaluated by calculating the Patch-FID and comparing it against outpainting as a baseline.
- Unconditional diffusion models and URCDMs are evaluated in a user study carried out by expert pathologists to rate the realism of synthetic images.

Evaluating the quality of generated images is not a trivial task. Whilst there are quantitative ways of evaluating the similarity and quality when compared to real images such as Fréchet Inception Distance [26], these are only valuable as comparisons to the same quantitative measurements performed on the same real images. Where baseline models are easily trainable, such as for unconditional image generation, we compare our results against a previously state-of-the-art model. Specifically, we use StyleGAN3 [25] as a baseline model to compare FID scores.

To evaluate the performance of segmentation-mask-conditioned diffusion models for dataset enrichment, we can evaluate downstream segmentation performance by training segmentation models using a public dataset that we have enriched. This allows us to compare with the previous state-of-the-art in segmentation on the same dataset.

Qualitative evaluation of ultra-resolution images is more difficult, given typical FID only evaluates images at low resolutions. Patch-FID (pFID), introduced by Chai et al. [5] is a variation of the FID metric [26] for very high-resolution images. Standard FID downsamples to 299×299; dramatically smaller than the over $40,000 \times 40,000$ images that we are generating here. This means that only the low-resolution global structure is evaluated [5]. Furthermore, we would need to compute the FID with several thousand ultra-resolution images, something that is completely infeasible given the sampling times.

Patch-FID works by taking random crops at random scales from the image and computing the FID between the real and generated patches by sampling different patches at matching scales and positions [5]. This makes it trivial to sample several thousand samples from a much smaller dataset of real and generated images, whilst also evaluating the fine details of the generated image significantly more than the regular FID would.

When evaluating Ultra-Resolution Cascaded Diffusion Models, we quantitatively compare results against simple outpainting on an unconditional diffusion model as a baseline. This is both due to the novelty of URCDMs making it tricky to train a prior baseline model that has been shown to generate gigapixel images and because URCDMs directly build upon outpainting as a more advanced method of generating ultra-resolution images with diffusion models.

Arguably more important than model-based evaluation such as FID and Patch-FID, a large part of evaluating image quality should be conducted by renal histopathologists, given that they are experts in the field. Pathologists will know what to look for in a generated histopathology image to confirm whether it looks realistic or not. As a result, a qualitative comparison between real images and generated images should be performed. This is discussed more in Section 5.3. The FID metric is also not a replacement for human evaluation, given that compressing an image can drastically worsen its FID even if it looks the same to a human [35], and evaluation by a human can lead to different conclusions than using an FID score [12].

5.3 Kidney Diffusion Evaluation Platform

To evaluate whether expert pathologists think that these images look realistic or not, I created a platform that compares real and fake images side-by-side without telling the pathologist which is which. The pathologist must then choose the image they think is real. If the images generated using diffusion models look perfectly realistic, then pathologists will be choosing an image essentially at random and get 50% of classifications correct. Conversely, if the images are clearly fake, the pathologists will get 100% of classifications correct. Therefore, the realism of the images can be quantified by the number of mistakes pathologists make in classification.

The website is publicly accessible at kidney-diffusion.doc.ic.ac.uk but requires a login to identify the person classifying the image because real medical imagery is being shown. Please get in touch to try this for yourself.

Figure 5.3: `kidney-diffusion` evaluation platform for pathologists to choose the image they think is a real image of kidney tissue.

Figure 5.3 shows the main page that pathologists will interact with on the evaluation platform. Typically, pathologists took under 10 seconds to evaluate each pair of images by clicking on the button under the image they think is real.

Pairings of real and fake images are completely random, along with the position of the real and fake images. Both the real and fake images are sampled at random without repeats and are consistent along different pathologists allowing direct analysis of which fake images are hardest to identify. Some pre-processing was carried out on both the real and fake images to remove any that were mostly whitespace as these were artificially hard to classify as they had no identifying information.

There are five people evaluating the realism of generated images in total, with four expert medical renal pathologists that have years of formal training, and an additional non-expert that has close familiarity with the images but no formal training. Each person involved has very generously donated their time in evaluating this project. The pathologists involved are Dr

Candice Roufosse, Dr Catherine Hordfield, Dr Andrew Smith, and Dr Naomi Simmonds. The non-expert involved is Sarah Cechnicka. Both Candice and Sarah have some familiarity with what the synthetic images look like as they have been shown early images, and have worked directly on parts of the project respectively. This could mean that they may find it easier to tell when an image is fake. All other pathologists have never seen a fake image before evaluating it.

In results tables such as Table 5.2 I have kept some anonymity in order to keep the performance of each pathologist private by numbering them randomly. The non-expert user has been labelled as such to keep a distinction between results from users with and without formal training.

Furthermore, where times are presented, such as in Table 5.4, these have been calculated as the difference between consecutive classifications during a single session of using the evaluation platform. Any times that are greater than five minutes have been ignored when calculating the median as these are likely to be when the user takes a break or goes for several days without using the platform.

5.3.1 Implementation

The evaluation platform is built as a statically-generated website using Flask as a backend. I am using SQLAlchemy to interface between Flask and an SQLite database that stores all patch and user information. Each user simply consists of an id, username, and hashed password. Patches have an id, a boolean flag to specify whether it is a real patch or not, and a version integer. The version is used to differentiate between generates images from different models, allowing me to evaluate them separately. Finally, classifications are stored as the id of the real patch and fake patch that was classified, the user id that classified it, the time it was classified, and whether it was correctly classified or not. Users are served a consistent pairing of real and fake patches by querying the first real and fake patch with the lowest id and latest version.

Login sessions were handled with Flask-Login, allowing users to log in and stay logged in for an extended amount of time. Before being shown patches to classify, users are shown a basic login screen which refreshes the page after logging in and loads the main page shown in Figure 5.3. I wanted this interface to be as simple to use as possible to ensure pathologists were happy to use it for longer periods of time. To enable this I made sure that after logging in once, they would rarely need to log in again, and using the website was as clicking a single button to classify an image.

5.4 Unconditional Image Generation

5.4.1 Results

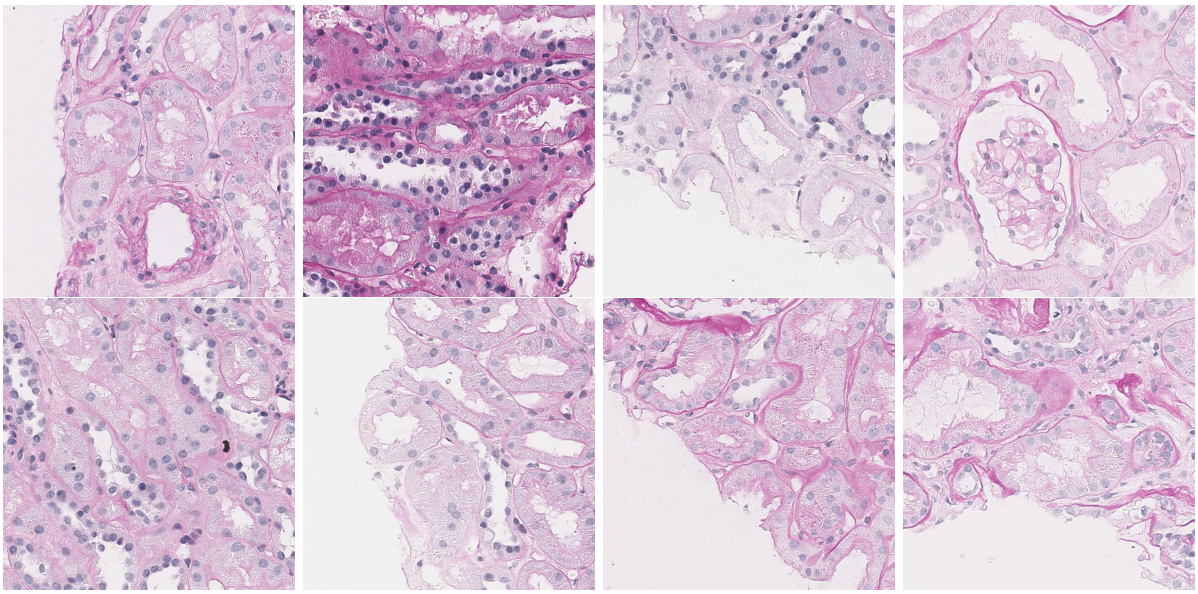


Figure 5.4: Random synthetic images generated from the final best-performing unconditional diffusion model trained on the kidney dataset.

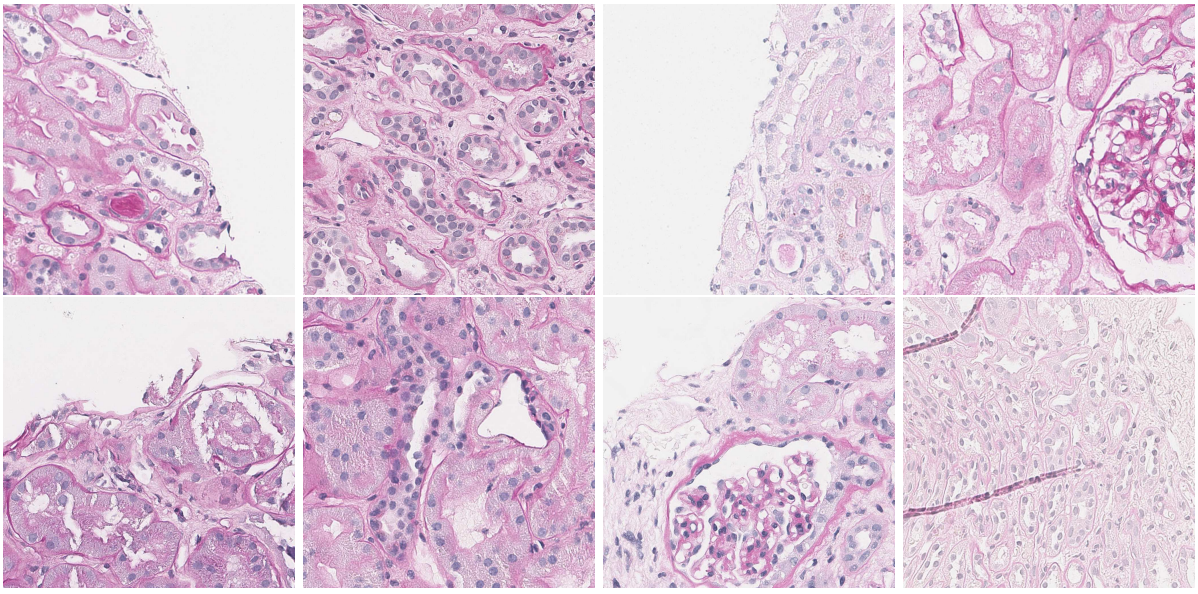


Figure 5.5: Randomly chosen real patches from the kidney dataset.

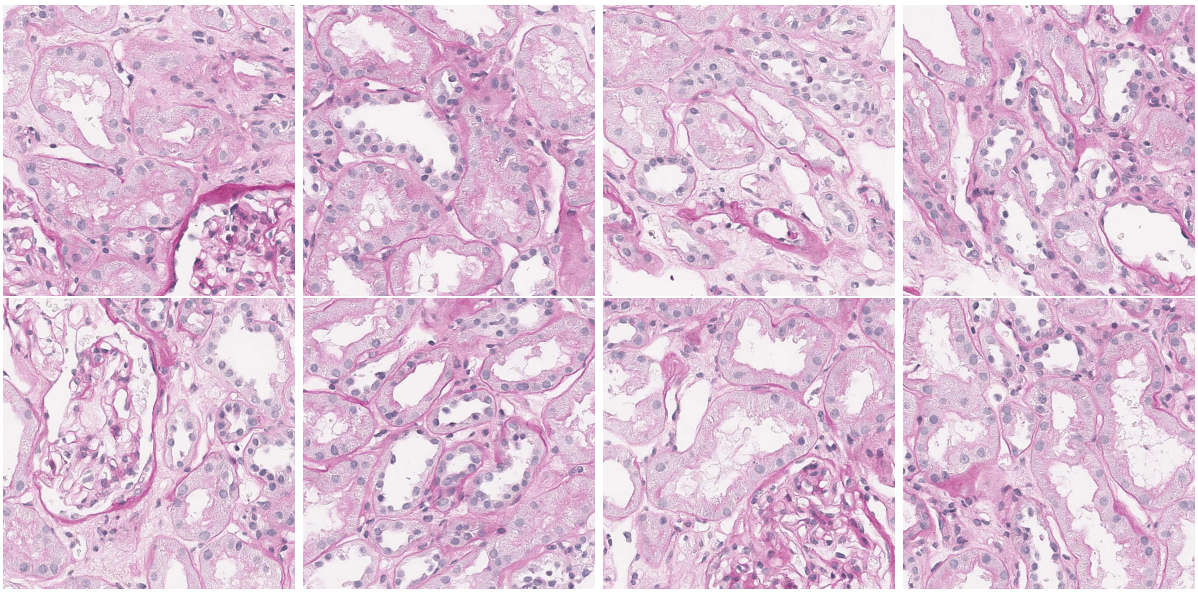
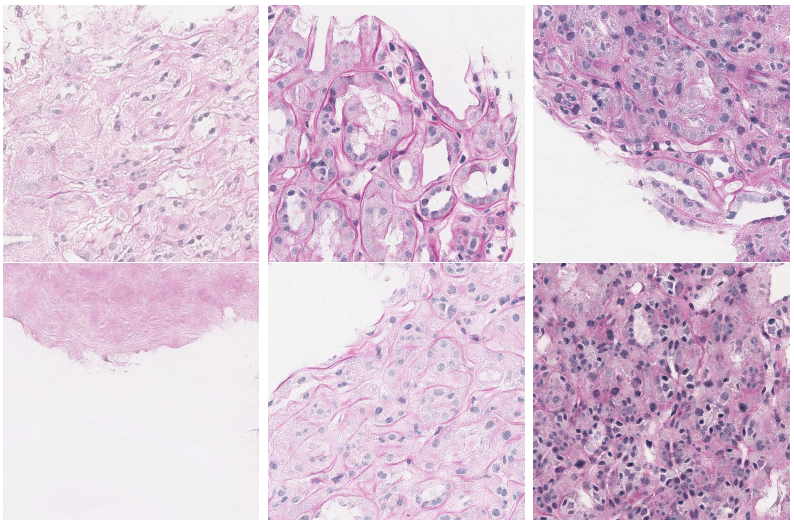
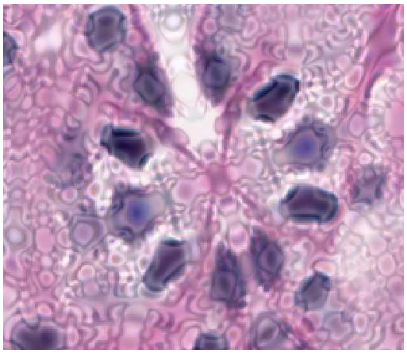


Figure 5.6: Random synthetic images generated from the first unconditional diffusion model trained on the kidney dataset.



(a) Random synthetic images generated from a baseline unconditional StyleGAN3 [25] model trained on the kidney dataset.



(b) Zoomed-in crop of the last image on the left. This highlights the symmetries and ringing artifacts present in the StyleGAN3 [25] when looking in detail.

Figure 5.7: Results from images generated using a StyleGAN3 [25] baseline model. Left shows randomly generated images. Right shows issues with those images.

Model	**FID-10k (lower is better)**
Baseline StyleGAN3 [25] model	38.62
First diffusion model conditioned on patient info	73.80
Final unconditional diffusion model	**10.35**

Table 5.1: Results of FID-10k metric on each unconditional generative model trained on the kidney dataset.

5.4.2 Discussion

Figure 5.4 shows the results of generating images using the final unconditional diffusion model. This can be qualitatively compared with the real images in Figure 5.5, as well as the baseline results from StyleGAN3 [25]. By eye, you can tell that the images generated by the diffusion model look natural and don't have any noticeable image artifacts as a result of the diffusion process. This is in contrast to the baseline results from StyleGAN3 [25] where unnatural symmetries appear in the image, clearly giving away that the image is not real. These symmetries are highlighted in Figure 5.7. On top of looking unnatural, the baseline results have ringing artifacts when zooming into the image, which are not present in images generated by the diffusion model.

Compared with both the real images in Figure 5.5, and the StyleGAN3 [25] images, the diffusion model images from the first unconditional model shown in Figure 5.6 (evaluated in the downstream segmentation task) are less diverse and don't generate a wide range of kidney tissue. Of the changes made to this first model, described in Section 3.3.2, I suspect that removing any conditioning patient information and training for much longer were the main factors that lead to improved diversity. The patient information used in the first model was very noisy and we were not evaluating this aspect of the model as it was not very useful, so it was appropriate to remove it.

The final unconditional diffusion model has the benefits of both the StyleGAN3 [25] and the first unconditional model since it is both diverse and looks very realistic with no visible artifacts.

Table 5.1 shows the quantitative results of each model trained on the kidney dataset. Interestingly, the StyleGAN3 [25] model has better performance than the first diffusion model trained. This is because the images generated by the first diffusion model, whilst being of much higher visible quality with fewer artifacts, were not diverse. Since FID measures both diversity and quality, the baseline model performs better. Additionally, since FID resizes the image to 299 pixels [5], artifacts that are only visible when zooming in will be less visible. The final unconditional diffusion model performs significantly better than both models.

5.4.3 Expert Pathologist User Study

User	**Correctly Classified as Real**	**Incorrectly Classified as Real**	**Proportion Incorrect (p)**	$\lvert p - 0.5 \rvert$
Pathologist 1	51	36	0.4138	0.0862
Pathologist 2	210	100	0.3226	0.1774
Pathologist 3	14	92	0.8679	0.3679
Pathologist 4	32	74	0.6981	0.1981
Non-expert	14	22	0.6111	0.1111
Total	**321**	**324**	**0.5023**	**0.1961 (weighted MAE)**

Table 5.2: Results of human evaluation of the realism of synthetic images generated by the first diffusion model conditioned on patient information.

User	Correctly Classified as Real	Incorrectly Classified as Real	Proportion Incorrect (p)	$\lvert p - 0.5 \rvert$
Pathologist 1	250	179	0.4172	0.0823
Pathologist 2	106	145	0.5777	0.0777
Pathologist 3	29	99	0.7734	0.2734
Pathologist 4	–	–	–	–
Non-expert	110	162	0.5956	0.0956
Total	**495**	**585**	**0.5417**	**0.1074 (weighted MAE)**

Table 5.3: Results of human evaluation of the realism of synthetic images generated by the final unconditional diffusion model.

Tables 5.2 and 5.3 show the results of human evaluation using the kidney diffusion evaluation platform. Here, we look at how many times for the initial and improved diffusion model a user chooses the real image correctly, and incorrectly chooses the synthetic image as the real image. We also look at the proportion, p, of images chosen that were synthetic, meaning the user thinks a fake image is real. If the diffusion model perfectly captures the distribution of images seen in the real images, then we expect $p = 0.5$ since it is essentially a coin flip as to which image is chosen. Therefore, we also consider the absolute error of p from 0.5. The larger this error is, the further the distribution of fake images is from the real images. When showing overall scores, we consider the weighted Mean Absolute Error, by weighting each user's absolute error by the proportion of overall classifications that they have made. This prevents one user with very few classifications from disproportionately affecting the mean error.

Importantly, however, a distinction should be made between very low values of p and very high values. If $p = 0$, then the user can perfectly tell when an image is fake. Conversely, if $p = 1$ then the user thinks the fake images look more real than the actual images and prefers the fake images. Either indicates the distribution of images is different, however, $p > 0.5$ is preferable over $p < 0.5$ as it shows that the user believes the generated images are more real.

We can see from Table 5.2 that p varies quite a lot between users, with some users having a strong preference for real images, and others having a strong preference for fake images. This balances out with the total proportion being close to 0.5. However, this variability leads to a high weighted MAE in the first model. Table 5.3 for the final model shows a much lower variability of p overall, with each user's individual absolute error being strictly less than for the first model. Overall, there is even a slight preference for the synthetic images generated by the final model, confirming that synthetic images generated by this model look very realistic.

User	Median Time to Classify (s)	Median Time to Classify When Correct (s)	Median Time to Classify When Incorrect (s)
Pathologist 1	10.0	9.0	11.0
Pathologist 2	10.0	10.0	10.0
Pathologist 3	25.0	23.0	25.5
Pathologist 4	5.0	6.0	5.0
Non-expert	2.0	2.0	2.0
Overall	**10.0**	**9.0**	**10.0**

Table 5.4: Time spent by pathologists for each classification of a synthetic image generated by the first unconditional diffusion model.

User	Median Time to Classify (s)	Median Time to Classify When Correct (s)	Median Time to Classify When Incorrect (s)
Pathologist 1	9.0	9.0	9.0
Pathologist 2	9.0	8.0	9.0
Pathologist 3	26.5	23.0	27.0
Pathologist 4	–	–	–
Non-expert	2.0	2.0	2.0
Overall	**7.0**	**7.0**	**7.0**

Table 5.5: Time spent by pathologists for each classification of a synthetic image generated by the final unconditional diffusion model.

Table 5.4 indicates that, in general, pathologists took slightly longer to classify images generated by the first model when they get the answer wrong (i.e. they believe the synthetic image is actually real). This makes sense as images generated by this model are less diverse, so on the rarer occasion that real and fake images look very similar, it will be harder to tell them apart and will take longer to classify. The difference in time to classify is another indication that images generated by the first model do not model the real images perfectly.

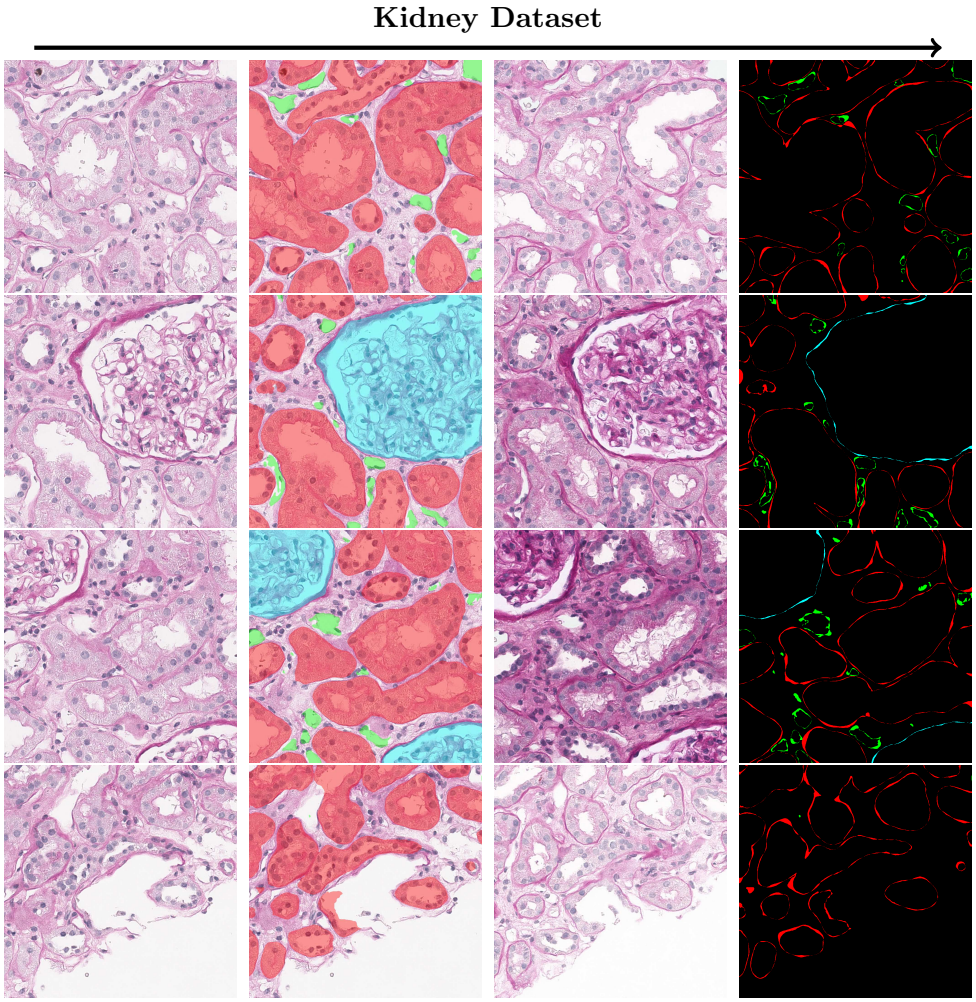
In Table 5.5, the time to classify is much more consistent between making correct and incorrect classifications. This suggests decisions made by pathologists are closer to a 'coin flip' with this final model as you would expect these times to be equal if the synthetic images perfectly modelled the real images.

Both Table 5.1 and Table 5.3 confirm that the final unconditional diffusion model generates highly realistic images that closely match the real distribution of kidney whole slide image patches. Table 5.1 shows this by having the lowest FID when compared to earlier less diverse models and a StyleGAN3 [25] baseline, and Table 5.3 shows this by having a low weighted MAE meaning that each user was choosing between real and fake images nearly at random. These results are further confirmed in Table 5.4 by having little to no difference in the time it takes a pathologist to classify an image when their classification is incorrect versus when it is correct. I speculate that the final unconditional diffusion model has $p > 0.5$ because I have personally noticed that there are very few 'edge-case' images generated by the diffusion model. This means that if there is some distortion, smudge, or blemish on the real slide image, users may have a preference for the more 'normal' looking fake image, leading to a slight preference for the images it generates.

5.5 Segmentation-Mask-Conditioned Diffusion Models

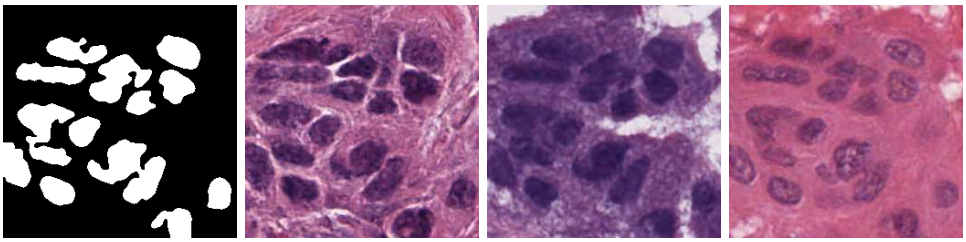
To evaluate whether enriching the dataset is beneficial for a downstream segmentation task, we evaluate the performance of an nnU-Net [36] trained on the data enriched by our method. Table 5.6 shows our results after training in various combinations of training sets. First, we train a base nnU-Net solely on real image data, (1), before fine-tuning it, independently, twice: once with a mixture of real and synthetic images as (2), and once exclusively with synthetic images as (3). The 4^{th} and 5^{th} models correspond to nnU-Nets retrained from scratch using exclusively synthetic images as (4), and one further fine-tuned on real images as (5) in Table 5.6.

5.5.1 Results



(a) D_ϕ sample (b) M_θ seg. mask (c) D_ξ sample (d) M_ζ -M_θ

KUMAR [30] Dataset



(e) (f) (g) (h)

Figure 5.8: Above shows outputs from the model trained on the kidney dataset. From left to right: (a) sample from D_ϕ, (b) overlaid segmentation from M_θ, (c) sample from D_ξ, (d) difference map of segmentation from M_ζ and M_θ. Segmentation colours are: red: Tubuli, blue: Glomeruli, and green: Vessels. Below shows outputs from the model conditioned on different tissue types in KUMAR [30], using the same label mask (e). Generated images are from Liver (f), Bladder (g), and Breast (h) tissues. This shows that our conditioning seems to allow a plausible mask to produce any kind of tissue. (Cechnicka et al. [29])

Table 5.6: Comparison with the state-of-the-art methods on the KUMAR dataset (top) and the KIDNEY transplant dataset (bottom). Metrics are chosen as suggested in [31]: Dice and AJI. Variant (1-5) show performance on the full as well as limited (1a) KIDNEY data. Best values in bold. (Cechnicka et al. [29])

		Method	Dice (%)			AJI (%)		
			Seen	Unseen	All	Seen	Unseen	All
KUMAR		CNN3 [37]	82.26	83.22	82.67	51.54	49.89	50.83
		DIST [38]	-	-	-	55.91	56.01	55.95
		NB-Net [39]	79.88	80.24	80.03	59.25	53.68	56.86
		Mask R-CNN [40]	81.07	82.91	81.86	59.78	55.31	57.86
		HoVer-Net [41] (*Res50)	80.60	80.41	80.52	59.35	56.27	58.03
		TAFE [42] (*Dense121)	80.81	83.72	82.06	61.51	61.54	61.52
		HoVer-Net + InsMix [31]	80.33	81.93	81.02	59.40	57.67	58.66
		TAFE + InsMix [31]	81.18	84.40	82.56	**61.98**	**65.07**	**63.31**
	Ours	(1) trained on real	80.30	80.58	80.38	48.35	49.62	50.81
		(2) fine-tuned by synthetic+real	85.65	86.96	86.03	56.60	57.70	56.91
		(3) fine-tuned by synthetic	75.83	82.28	77.67	32.29	40.06	34.51
		(4) trained on synthetic	84.52	**89.37**	85.90	48.12	57.52	50.80
		(5) trained on synthetic, fine-tuned on real	**86.13**	88.29	**86.75**	56.80	58.44	57.27
KIDNEY	Ours	(1) trained on real (100% data)			94.22			77.45
		(1a) trained on real (30% data)			88.01			62.05
		(2) fine-tuned by synthetic+real			92.25			69.11
		(3) fine-tuned by synthetic			89.65			58.59
		(4) trained on synthetic			82.00			42.40
		(5) trained on synthetic, fine-tuned on real			92.74			71.55

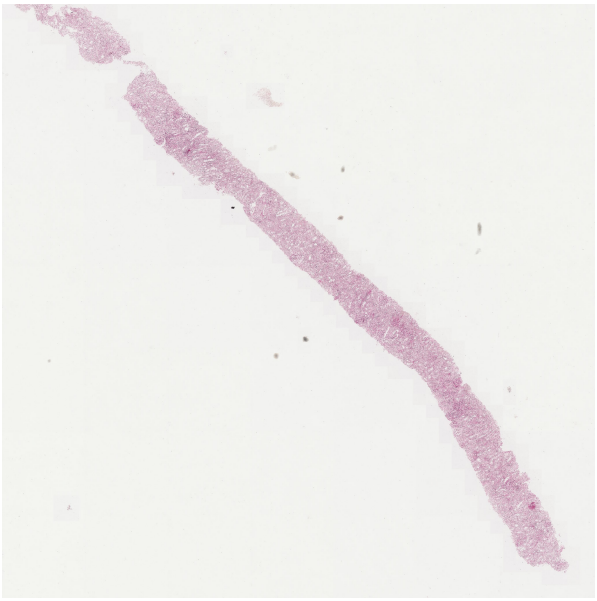
5.5.2 Discussion

Our quantitative results are summarised and compared to the state-of-the-art in Table 5.6 using the Dice coefficient (Dice) and AJI as suggested by [31]. Qualitative examples are provided in Figure 5.8 (left), which illustrates that our model can convert a given label mask into a number of different tissue types and in Figure 5.8, where we compare synthetic enrichment images of various tissue types from our kidney transplant data.

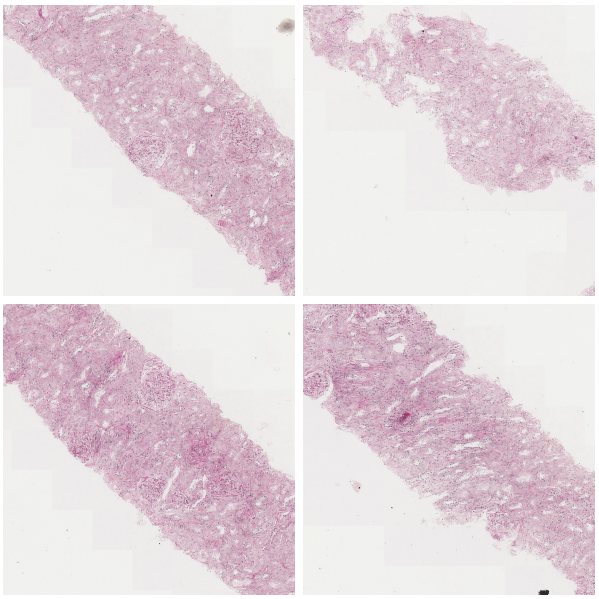
Out of our 5 models relying on additional synthetic data in the KUMAR dataset experiments, 4 of them outperform all previous SOTA on the Dice score. AJI over-penalizes overlapping regions [41], and our method does not optimise for this metric, explaining its performance. Table 5.6 shows that, for the KIDNEY dataset, we can reach high performance (88% Dice) while training on 30% (500 samples) of the real KIDNEY data (1a), and training on the full training set (1) yields a Dice score of 94%. We also observe that the model pre-trained on synthetic data and fine-tuned on 500 real images (5), outperforms the one only trained on 500 real images (1a). Additionally, we discover that training the model on real data before fine-tuning it on synthetic samples (3) does not work as well as the opposite approach. These results seem to indicate that pre-training a segmentation model on generated data gives it a strong prior on large dataset distributions and alleviates the need for many real samples in order to learn the final, exact, decision boundaries, making the learning procedure more data efficient. (Cechnicka et al. [29]).

5.6 Ultra-Resolution Cascaded Diffusion Model

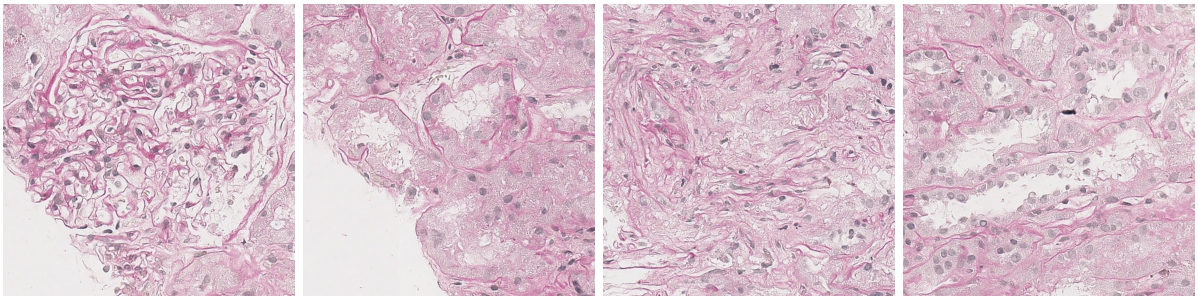
5.6.1 Results



(a) Full-scale ultra-resolution image
(40000 × 40000 pixels)

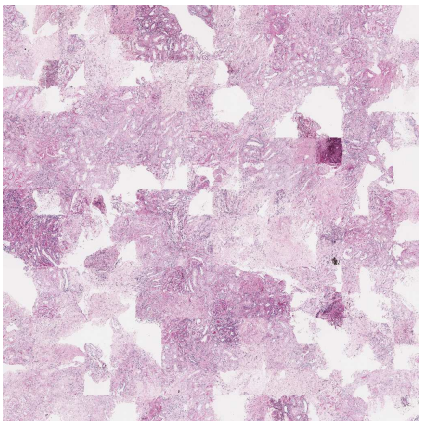


(b) Partially zoomed-in crops of the image in 5.9a
(6500 × 6500 pixels)

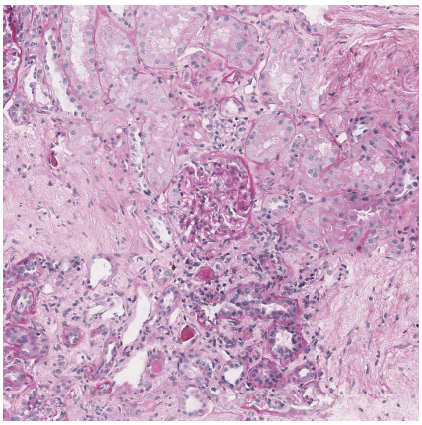


(c) Highly zoomed-in crops of the image in 5.9a
(1024 × 1024 pixels)

Figure 5.9: Sample of an ultra-resolution whole slide image generated using a URCDM.



(a) Full-scale ultra-resolution synthetic image. There is no spatial coherency over large distances.



(b) Zoomed-in crop of a smaller region of several patches, showing good quality fine details, but poor spatial coherency even over smaller distances.

Figure 5.10: Results from ultra-resolution images generated using outpainting with a standard unconditional diffusion model instead of a URCDM. 5.10a shows the full image generated to evaluate spatial coherency, and 5.10b shows a zoomed-in crop to evaluate the quality of fine details.

Model	**pFID-50k (lower is better)**
Baseline unconditional model using outpainting	150.15
Ultra-Resolution Cascaded Diffusion Model	**39.52**

Table 5.7: Results of pFID-50k metric on different methods of generating ultra-resolution images using diffusion models.

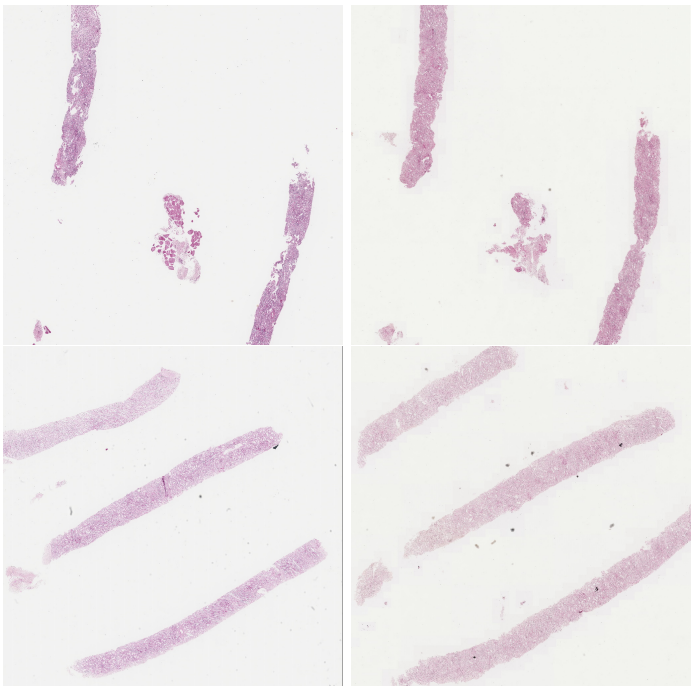


Figure 5.11: Overfitting to the real images experienced when generating synthetic images using a URCDM. Real images are shown on the left of each row, and synthetic images are shown on the right.

5.6.2 Discussion

Figure 5.9 shows a sample image generated by the Ultra-Resolution Cascaded Diffusion Model. When compared with a randomly-chosen real whole slide image in Figure 5.1, it's clear that the synthetic image generated with the URCDM is coherent over long distances. To further exemplify this, Figure 5.10 shows how a regular unconditional model would perform outpainting. Figure 5.10a shows no sense of coherency or shape at full-scale, and whilst Figure 5.10b is slightly more coherent, it still generates parts of kidney tissue directly next to each other, like the cortex and medulla, without blending them in a natural way. With a URCDM, the high-magnification model uses the lower-magnification conditioning image to prevent this from happening.

The texture of the tissue in Figure 5.9 is realistically wispy in regions of the kidney further from the cortex, like in the real whole slide image in Figure 5.1. Additionally, structures that are only visible at high magnification are of high quality, like the glomerulus and tubules in Figure 5.9c. In general, at high magnifications, there is a lot of diversity between different regions, whilst maintaining a good level of quality.

Despite successful results, the fine details in Figure 5.9c are visibly poorer than those in the real whole slide image, and those in the unconditional model in Figure 5.4. I suspect this is because the URCDM has to maintain consistency through all nine stages of the model, and the role of the base U-Net at the medium and high magnification levels is *much* more important than in a regular unconditional diffusion model. The base U-Nets are required to consistently and accurately 'zoom in' the lower magnification image, and failure to do so will result in neighbouring patches looking very different, like in the outpainting example in Figure 5.10. Further optimising these base U-Nets is an important area to explore in future work, perhaps by increasing their capacity or by conditioning them on more images.

In addition to the qualitative results, Table 5.7 shows the Patch-FID (pFID) for both outpainting using a regular unconditional diffusion model, and when using a URCDM. The pFID for the images generated using the URCDM is much lower than the images using outpainting, as expected. Whilst when zoomed in, fine details of the images generated using outpainting are of higher quality when analysed qualitatively, the lack of spatial coherency leads to a much poorer pFID score for the images generated using outpainting.

Unsurprisingly, since the number of ultra-high resolution images being trained on is very low, the low-magnification model tends to overfit the high-level structure of the training data. Figure 5.11 shows this clearly with real whole slide images alongside synthetic counterparts. Upon closer inspection, it is immediately clear that only the high-level shape of training data is reflected in the generated images. Finer details and even subtleties in the shape of the slide images are not reflected. For example, in the second pair in Figure 5.11, you can see that the tissue in the top-left of the real image is curved upwards slightly, unlike the synthetic image. This means that whilst the URCDM is memorising some aspects of the training data, the synthetic images will be increasingly novel at anything other than very low magnification.

5.6.3 Expert Pathologist User Study

User	Correctly Classified as Real	Incorrectly Classified as Real	Proportion Incorrect (p)	$\lvert p - 0.5 \rvert$
Pathologist 1	61	66	0.5197	0.0197
Pathologist 2	152	6	0.0380	0.4620
Pathologist 3	28	33	0.5410	0.0410
Pathologist 4	47	10	0.1754	0.3246
Non-expert	29	21	0.4200	0.0800
Total	**317**	**136**	**0.3002**	**0.2219 (weighted MAE)**

Table 5.8: Results of human evaluation of the realism of random crops of synthetic images generated by the Ultra-Resolution Cascaded Diffusion Model.

Expert pathologist evaluation was also carried out on ultra-resolution images, with results shown in Table 5.8. Interpretation of these results is elaborated on in Section 5.4.3. Results here are quite inconsistent among pathologists, with some pathologists very reliably identifying the fake image, and others choosing fake and real images fairly equally. Due to the limited number of ultra-resolution images generated, they are more difficult to evaluate. To combat this, pathologists were shown a subset of the crops taken to calculate the pFID. This means pathologists compare real and fake images at different scales. Unfortunately, this is quite unnatural for pathologists as they usually have access to the entire slide image and can zoom around throughout the image. At these lower-resolution crops, pathologists may struggle to analyse whether the morphology of images is realistic, which could explain the inconsistencies between pathologists. Despite this, if some pathologists are consistently correct, this should be a clear indication that the generated images are not as realistic as real images. p is lower than the unconditional diffusion models, and the weighted MAE is higher, confirming the qualitative discussion prior.

User	Median Time to Classify (s)	Median Time to Classify When Correct (s)	Median Time to Classify When Incorrect (s)
Pathologist 1	9.0	10.0	9.0
Pathologist 2	3.0	3.0	6.0
Pathologist 3	10.0	12.0	9.0
Pathologist 4	7.0	6.0	10.0
Non-expert	2.0	2.0	2.0
Overall	**5.0**	**4.0**	**8.0**

Table 5.9: Time spent by pathologists for each classification of a random crop of a synthetic image generated by the Ultra-Resolution Cascaded Diffusion Model.

Table 5.9 mirrors similar results to the earlier worse-performing model in Section 5.4.3. Overall, the median time to classify is significantly longer when the pathologist makes a mistake. This suggests that most of the time the pathologist can quickly identify the real from the fake, and there are rare circumstances where the real and fake images take a while to distinguish and result in a misclassification. These results further confirm that the generated ultra-resolution images are not as realistic as real images or the images generated from the earlier unconditional models.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

I have demonstrated in Section 3 that my novel method of generating synthetic images and matching segmentation masks for training downstream histopathology tasks leads to state-of-the-art performance in data-limited scenarios, such as the KUMAR [30] dataset. When dataset size is not as limited, such as the kidney dataset, there are fewer reasons to enrich the dataset with synthetic images as the performance of downstream tasks is likely already saturated. In addition to improving the downstream performance of segmentation tasks, the synthetic images are perceived by expert pathologist ts as looking extremely realistic, with pathologists actually having a slight preference for synthetic images over real images.

Furthermore, Ultra-Resolution Cascaded Diffusion Models are a novel and promising way of generating images with more than $1,000,000,000$ pixels using diffusion models. The results and discussion show that images generated by URCDMs are very spatially coherent over long distances, meaning that the images are plausible when zoomed out. Furthermore, fine details are retained well when zoomed in. Although the fine details are not as high quality as the unconditional diffusion models developed earlier in this project, they are still of decent quality, serving as a great starting point for images that were not possible to generate before.

Importantly, URCDMs generate much more coherent images than outpainting; the popular method of generating larger-resolution images that have no regard for spatial coherency. This is extremely important for applications of ultra-resolution imagery where multiple scales of the image are used at inference [1], rather than when splitting the image into equal-scale patches.

Overall, this project first successfully utilises diffusion models to generate synthetic pairs of histopathology images and segmentation masks. This is a novel method that can lead to state-of-the-art performance on downstream machine learning tasks. These images are indistinguishable from real images as evaluated by expert pathologists. Furthermore, this project introduces a novel way to successfully generate ultra-resolution images with long-distance spatial coherency that look more realistic at multiple scales than outpainting. Whilst there is room for improvement in stabilising training, and for higher-quality fine details, this is a big step towards generating extremely high-resolution images that are indistinguishable from their real counterparts.

6.2 Future Work

6.2.1 Ultra-Resolution Cascaded Diffusion Model Improvements

Follow-up work could look to make architectural or algorithmic changes to URCDMs to improve both training stability and to get higher-quality fine details. Below are some suggestions as to

areas that would be worth exploring:

Improved batching and parallel processing. Currently, when sampling from a URCDM on multiple GPUs, each GPU samples a single image from a U-Net at a time. If this batch size is increased, sampling is much more efficient, but due to the dependency between patches as neighbouring patches overlap, it is not trivial to sample in batches. By having a separate producer process that manages the images that can be generated, this work could be sent in batches to the GPU consumer processes to improve sampling speeds.

Removing the base model for higher-magnification models. In theory, given that high-magnification models 'zoom in' on low-magnification images, the area in the centre of the low-magnification image is a low-resolution image of the image that the high-magnification model will generate. You could take advantage of this fact and remove the base models entirely for the higher-magnification models and take a centre crop of the low-magnification image as the 64×64 pixel base image. This would further improve sampling speeds.

Conditioning the higher-magnification models on extra information. Currently, training stability and quality of higher-magnification models are heavily limited on how well the base model 'zooms in' the low-magnification image. The base model often struggles with this task, so adding extra information to condition image generation could make this task easier. Such information could include an upscaled centre crop of the low-magnification conditioning image so that the model doesn't have to learn the bounds of the region it is zooming in on.

Use the same Cascaded Diffusion Model for all magnifications. Whilst this project explores a 'cascaded' approach similar to CDMs [9], it would be interesting to compare it with a model that only has a single CDM that is trained at all three magnifications. This would be much more similar to Anyres-GAN [5] and would be interesting to apply to diffusion models. This allows common information to be shared between each magnification level and could stabilise training as weights are shared.

Some of these improvements would hopefully lead to the generalisability of URCDMs to other applications such as satellite imagery. Figure 6.1 shows much poorer performance on the AIRS dataset [33]. Whilst the image from the low-magnification model looks very high quality, the medium-magnification image doesn't capture any high-frequency details and this error is compounded in the high-magnification image.

Images from AIRS don't have any whitespace, unlike the kidney dataset, and also don't have a sensible colour to pad conditioning images with when they go out of bounds. I opted to pad with black pixels, which for some models will not be observed at training time. To combat this, I suspect augmenting conditioning images to add regions of black pixels at training time could significantly improve performance here so that real and fake conditioning images come from more similar distributions.

Figure 6.1: Results after training a URCDM on the AIRS dataset [33]. A real image from AIRS is above, with a synthetic sample underneath with increasing resolutions.

6.2.2 Model Distillation

A major factor in the usability of these models for enriching datasets, particularly for the Ultra-Resolution Cascaded Diffusion Model, is the sampling speed. Sampling a URCDM can take multiple hours when running on eight Nvidia A100 GPUs, which is not fast enough if many images are needed, and not accessible enough if high-performance computation is unavailable. This was not a significant issue in this project for evaluation due to evaluation metrics such as Patch-FID [5] that can work on fewer ultra-resolution images, and because the compute needed was available, but it could be a reason to avoid using this method in another context.

The most effective way of improving this aspect of URCDMs would be to introduce model distillation. This would make it possible to reduce the number of sampling steps, and therefore sampling time, dramatically with little to no performance decrease [32, 43]. We have already explored the use of v-parameterisation [32] in this project, one approach of creating a diffusion model that can be distilled during training. Taking this a step further and implementing distillation as presented in these papers would make this method much more accessible.

Alternatively, it would be interesting to explore the possibility of fine-tuning an existing fast-to-sample and open-source model such as Stable Diffusion [10]. This already uses model distillation and sampling speeds are dramatically faster than the models used in this project. By taking three Stable Diffusion models and fine-tuning them, you could potentially create the different magnification models required in URCDMs.

Chapter 7

References

[1] Adam Van Etten. You only look twice: Rapid multi-scale object detection in satellite imagery, 2018.

[2] Mathias Öttl, Jana Mönius, Matthias Rübner, Carol I. Geppert, Jingna Qiu, Frauke Wilm, Arndt Hartmann, Matthias W. Beckmann, Peter A. Fasching, Andreas Maier, Ramona Erber, and Katharina Breininger. Improved her2 tumor segmentation with subtype balancing using deep generative networks, 2022. URL https://arxiv.org/abs/2211.06150.

[3] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022. URL https://arxiv.org/abs/2201.09865.

[4] DALL·E: Introducing outpainting, Aug 2022. URL https://openai.com/blog/dall-e-introducing-outpainting/.

[5] Lucy Chai, Michael Gharbi, Eli Shechtman, Phillip Isola, and Richard Zhang. Any-resolution training for high-resolution image synthesis. In *European Conference on Computer Vision*, 2022.

[6] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

[8] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement, 2021. URL https://arxiv.org/abs/2104.07636.

[9] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *arXiv preprint arXiv:2106.15282*, 2021.

[10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. URL https://arxiv.org/abs/2112.10752.

[11] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125.

[12] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. URL https://arxiv.org/abs/2205.11487.

[13] Walter H. L. Pinaya, Petru-Daniel Tudosiu, Jessica Dafflon, Pedro F da Costa, Virginia Fernandez, Parashkev Nachev, Sebastien Ourselin, and M. Jorge Cardoso. Brain imaging generation with latent diffusion models, 2022. URL https://arxiv.org/abs/2209.07162.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

[16] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017. URL https://arxiv.org/abs/1709.07871.

[17] Phil Wang. lucidrains/imagen-pytorch: Implementation of Imagen, Google's Text-to-Image Neural Network, in Pytorch — github.com. https://github.com/lucidrains/imagen-pytorch, 2022. [Accessed 12-Nov-2022].

[18] Susung Hong, Gyuseong Lee, Wooseok Jang, and Seungryong Kim. Improving sample quality of diffusion models using self-attention guidance, 2022. URL https://arxiv.org/abs/2210.00939.

[19] Mark Sabini and Gili Rusak. Painting outside the box: Image outpainting with gans, 2018. URL https://arxiv.org/abs/1808.08483.

[20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

[21] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018.

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020.

[25] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021.

[26] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2017. URL https://arxiv.org/abs/1706.08500.

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. URL https://arxiv.org/abs/1409.4842.

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[29] Sarah Cechnicka, James Ball, Callum Arthurs, Candice Roufosse, and Bernhard Kainz. Realistic data enrichment for robust image segmentation in histopathology, 2023.

[30] Neeraj Kumar, Ruchika Verma, Sanuj Sharma, Surabhi Bhargava, Abhishek Vahadane, and Amit Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE Transactions on Medical Imaging*, 36(7):1550–1560, 2017. doi: 10.1109/TMI.2017.2677499.

[31] Yi Lin, Zeyu Wang, Kwang-Ting Cheng, and Hao Chen. Insmix: Towards realistic generative data augmentation for nuclei instance segmentation. In *MICCAI 2022, Part II*, pages 140–149. Springer, 2022.

[32] Tim Salimans and Jonathan Ho. Progressive Distillation for Fast Sampling of Diffusion Models, June 2022. arXiv:2202.00512.

[33] Qi Chen, Lei Wang, Yifan Wu, Guangming Wu, Zhiling Guo, and Steven Waslander. Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings, 07 2018.

[34] Labelbox: The leading ai data engine platform. URL https://labelbox.com/.

[35] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation, 2021. URL https://arxiv.org/abs/2104.11222.

[36] Fabian Isensee and Paul F Jaeger. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation, 2018.

[37] Neeraj Kumar, Ruchika Verma, Deepak Anand, Yanning Zhou, Omer Fahri Onder, Efstratios Tsougenis, Hao Chen, Pheng-Ann Heng, Jiahui Li, Zhiqiang Hu, et al. A multi-organ nucleus segmentation challenge. *IEEE transactions on medical imaging*, 39(5):1380–1391, 2019.

[38] Peter Naylor, Marick Laé, Fabien Reyal, and Thomas Walter. Segmentation of nuclei in histopathology images by deep regression of the distance map. *IEEE transactions on medical imaging*, 38(2):448–459, 2018.

[39] Yuxin Cui, Guiying Zhang, Zhonghao Liu, Zheng Xiong, and Jianjun Hu. A deep learning algorithm for one-step contour aware nuclei segmentation of histopathology images. *Medical & biological engineering & computing*, 57:2027–2043, 2019.

[40] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE ICCV'17*, pages 2961–2969, 2017.

[41] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical Image Analysis*, 58:101563, 2019.

[42] Shengcong Chen, Changxing Ding, and Dacheng Tao. Boundary-assisted region proposal networks for nucleus segmentation. In *MICCAI 2020, Part V 23*, pages 279–288. Springer, 2020.

[43] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models, 2023.