MSc Electrical Engineering
Final Project

Fast and Precise Lung CT Image Generation via Diffusion Models

Xinrui Zu

Committee:
dr. ir. C. O. Tan
dr. ir. M. Abayazid
dr. M. Poel
E. I. S. Hofmeijer MSc

June, 2023

Department of Electrical Engineering
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Contents

Abstract

The insufficiency and inadequate diversity of medical datasets have been significant problems in computer-aided diagnosis (CAD) systems and medical imaging research. Consequently, medical image synthesis has emerged as one of the most dynamic machine learning research areas presently. We present a novel approach for fast and precise semantic image synthesis using diffusion models. In particular, our method generates high-quality Lung computed tomography (CT) images with precise lung nodule areas by utilizing the semantic labels and key features of the nodules in LIDC-IDRI dataset. We use a diffusion model to learn the conditional distribution of the image pixels given the semantic label map, which enables us to generate realistic and diverse images by sampling from the learned distribution. To further improve the quality of the synthesis results, we evaluate several key pathology features of the lung nodules in LIDC-IDRI dataset and use classifier guidance with these key features to optimize the generated nodule areas. We evaluate our method with various kinds of synthetic image quality metrics and show that it outperforms state-of-the-art methods in terms of visual quality, diversity, and fidelity to the input semantic labels. Our method has the potential to enable new applications in medical imaging, such as medical data augmentation, anomaly in-painting, and diagnosis training.

Keywords: machine learning, CT, medical imaging, deep generative models, semantic image synthesis, diffusion models, classifier guidance

Chapter 1

Introduction

1.1 Motivation

Medical research has greatly benefited from the advancement of machine learning techniques in recent years [37, 14, 46, 50, 25, 2, 42, 18]. With the ability of statistical and inductive analysis, deep learning methods have revolutionary impact on diagnosis and treatment of diseases and have shown remarkable performance in a variety of medical imaging tasks, including segmentation, detection, and classification. The success of these approaches is highly dependent on the availability and quality of training data. Undersized and imbalanced datasets can lead to biased and inaccurate models, which may have serious implications for patient care [51].

There are several challenges in acquiring medical image datasets. Firstly, the collection of medical images is often limited by ethical and legal issues [51]. Secondly, manual annotation and labeling of medical images is time-consuming and requires expertise. Finally, collecting and processing medical images can be difficult and expensive, due to limited availability and costs concerned with using medical imaging equipment such as CT or Magnetic resonance imaging (MRI) scanner. Consequently, the difficulty of acquiring medical image data pose a major challenge for the development of CAD systems and medical imaging research [35]. To address this issue, generating synthetic medical images with accurate pathology features is supposed to be a promising solution [41], which potentially allows for the generation of unlimited virtual medical data. In particular, semantic image synthesis (SIS), which utilizes a segmentation label map as conditional information, has proven to be an efficient approach to generate high-quality images [15, 32].

However, the state-of-the-art SIS methods have inevitable limitations when dealing with tiny semantic areas in the map. For example, existing methods often struggle to faithfully generate and preserve the fine-grained details of tiny semantic areas such as human eyes in the facial dataset, traffic lights in the scene dataset, and furniture details in LSUN bedroom dataset [61].

For medical image synthesis, such limitations can result in misleading information and significantly weaken the value of the synthetic medical data. Specifically, inaccurate information of small regions of interest (ROIs) can lead to serious misdiagnosis and severe fault in treatment planning. Recent research has focused on the accuracy of generation based on complex semantic maps [43, 60]. However, the improvements and systematic analysis of the generated results with respect to certain small ROIs, such as nodules in lung CT scans, have received less attention and lack evaluation.

To address this problem, we propose the latent semantic diffusion model (LSDM), which better utilizes the semantic maps of LIDC-IDRI dataset as the input of adaptive

normalization layers and control the diffusion process in the latent space. Existing methods have limitations in precisely generating small areas such as lung nodules in the image. In contrast, our method uses several nodule pathology features provided by LIDC-IDRI dataset to describe the requirements of the generation, and explicitly control the characteristics of the generated lung nodule areas via multi-classifier guidance based on these pathology features. We also use a balancing method (discussed in section 4.2) in dataset pre-processing to improve the performances of the classifiers.

We evaluate our method with various kinds of synthetic image quality metrics. Our experimental results demonstrate the superiority of our method over state-of-the-art techniques in terms of visual quality and quantitative metrics such as diversity and fidelity. To further evaluate the multi-classifier guidance, we also do ablation studies with respect to the classifiers.

Our method has the potential to enable better utilization of the information in the datasets and generate small segmentation areas precisely. We believe that this work can contribute to the development of new medical image synthesis methods and benefit clinical applications and patient care.

1.2 Contributions

To overcome the challenge of generating precise image characteristics in detailed semantic areas such as lung nodules, we specifically make two contributions. Firstly, we utilize spatial-adaptive normalization layers aligned with concatenated semantic channels in the blocks of the diffusion model to leverage the information contained in the semantic labels. Secondly, in order to quantitatively control the generation results on small ROIs in an explicit approach, we propose in-painting style multi-classifier guidance for our reverse diffusion process based on the key pathological characteristics of the nodules. Combined with the above, we also implement the diffusion process in the latent space for fast inference and use a re-sampling trick to solve the divergence problem with severe imbalanced distribution of the key pathology features in the dataset.

According to the requirement of precise generation in small ROIs, the proposed training pipeline consists of 3 stages: training the autoencoder for latent space, training the key feature classifiers, and training the latent semantic diffusion model (LSDM). The stages are disentangled and trained separately, enabling flexible implementation and efficient deployment of the model. During the inference procedure, our method first takes the semantic labels of the images in pixel space as inputs and encodes them into latent space, which filters out unrelated redundant information. subsequently, the method generates Gaussian noises in the latent space and concatenates them with the encoded semantic labels. Aligned with the spatial-adaptive normalization layers with respect to the semantic labels in the diffusion model and the guidance of the key feature classifiers, our method then goes through the reverse diffusion process to generate target images in latent space. Finally, the method decode the target image into the real pixel space. A detailed illustration is shown in Fig. 3.2 and Chapter 3.

1.3 Report Outline

The rest of the thesis is organized as follows: Chapter 2 presents a detailed overview of related work. Chapter 3 describes the proposed method in detail, including the diffusion model, in-painting style classifier guidance, and training procedure. Chapter 4 introduces the pre-processing methods of the utilized data, followed by the experimental results and

comparisons with state-of-the-art methods in Chapter 5. In addition, we discuss the interesting connections between diffusion models, which is the basis of our method, and particle filtering/reinforcement learning in Chapter 6. Furthermore, Chapter 7 presents several discussions and illustrates the limitations of our methods. Finally, Chapter 8 points out the possible directions of future work.

Chapter 2

Related Work

In this chapter, we first have a brief review of generative adversarial nets (GAN) and their variants. Then we discuss the conditional image synthesis using GAN-based models. After that, we discuss the recent research efforts focused on image generation using diffusion models and their variants, illustrating the advantages of them with respect to the GAN-based models. Furthermore, we concentrate on the discussion of conditioning in diffusion models and introduce the guidance mechanism to control the reverse diffusion process. Specifically, we discuss the challenge of small semantic area generation, as it requires the generation of precise and detailed images in a limited area. Finally, we focus on the recent research efforts of generative models for this purpose.

2.1 Generative Adversarial Nets

GAN has emerged as one of the most promising approaches for generating high-quality synthetic images for years. GAN consists of a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$, which are trained in an adversarial manner [8, 1, 20]. The generator $\mathcal{G}$ tries to generate images that mislead the discriminator $\mathcal{D}$, while the discriminator $\mathcal{D}$ tries to correctly identify the generated images. If converged, the generator $\mathcal{G}$ learns to generate images that are similar to the training data, and the discriminator $\mathcal{D}$ learns to estimate the probability of a sample from the training dataset. Both structures are parameterized with neural networks.

The original GAN training pipeline works as follows: Given a training dataset with distribution p_{data} and a noise vector z with distribution p_z in the latent space, the generator $\mathcal{G}$ takes z as input and generate image $x' = \mathcal{G}(z)$, based on which the discriminator $\mathcal{D}$ outputs the probability of a sample (x or x') belongs to the training dataset. The objective function of this pipeline is a two-player min-max game which can be described as follows:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \{\mathbb{E}_{x \sim p_{data}} \log(\mathcal{D}(x)) + \mathbb{E}_{z \sim p_z} \log(1 - \mathcal{D}(\mathcal{G}(z)))\} \tag{2.1}$$

Where p_{data} is the distribution of the training data and p_z is the distribution of the latent noisy vector. The two networks play a zero-sum game, in which the generator's objective is to synthesize samples that minimize the objective which indicates the accuracy of the discriminator, while the discriminator's goal is to maximize its accuracy in distinguishing real from fake samples.

GANs can be used to generate synthetic data that significantly resemble real data. This makes it a powerful tool for generating data with complex information such as high-resolution images [8, 1]. In addition, GANs can be used for unsupervised learning without labelled data, leveraging the vast amount of real-world data. However, there are several

disadvantages of GANs including the instability of adversarial training, mode collapse and the distribution bias of the generated data with respect to the real data.

Several variants of the original GAN algorithm have been proposed to address its limitations and improve its performance in various applications. Wasserstein GANs (WGANs)[1] replace the original adversarial objective with the Wasserstein distance, resulting in a more stable training process that partially avoids mode collapse and difficulty in hyper-parameter tuning encountered by GAN. Specifically, WGANs adjust the discriminator network with 1-Lipschitz constraints to formulate objective as the estimation of Wasserstein distance, solving the derivative vanishing problem of the original objective when the distributions of the real data and the generated data have little overlap. Deep Convolutional GAN (DCGAN) [33] uses a convolutional neural network (CNN) to capture spatial correlations between pixels of the images and evolves several minor adjustments. Cycle-Consistent GAN (CycleGAN) [63] use a cycle-consistency loss to enforce that the reconstruction of a generated data should match the original data, and vice versa. This mechanism allows for domain transfer without the need for paired training data. StyleGAN [17] introduces a style vector to the generator, allowing for fine-grained control of the generated image's style. StyleGAN also employs progressive growth during training and hierarchical latent spaces for the generator and the mapper.

2.2 Conditional Image Synthesis using GAN

Conditional image synthesis aims to generate realistic images based on the input conditions including text, labels, sketches, semantic maps or other modalities. Conditional image synthesis is supervised learning with the conditioning information which can also improve the generation quality. Leveraging from the interaction-style input of the conditioning information, GAN-based conditional image synthesis can also help to understand the underlying structure and intrinsic distribution of natural images.

There are various research efforts on this supervised-style generation technique based on GAN-like models. Conditional GAN (cGAN) [27] is an extension of the original GAN model. The objective function of cGAN is:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \{\mathbb{E}_{x\sim p_{data}} \log(\mathcal{D}(x|y)) + \mathbb{E}_{z\sim p_z} \log(1 - \mathcal{D}(\mathcal{G}(z|y)))\} \tag{2.2}$$

Where y is the conditional information encoded in both generator and discriminator, changing $\mathcal{D}(x|y)$ into a conditional probability. The cGAN algorithm enlightened the controllability study of GAN and its variants, providing significant improvements in generation fidelity. Derived from cGAN, Auxiliary Classifier GAN (ACGAN) [31] uses an auxiliary classifier in the discriminator to enforce the correspondence between the input conditions and the generated images. It also uses a projection mechanism to match the conditional distributions of real and fake images. To improve the utilization of image conditions and translate an image from one domain to another, Image-to-Image Translation with Conditional Adversarial Networks (pix2pix) [15] uses a U-Net based generator and a PatchGAN discriminator, which only penalizes structure at the scale of patches, to perform image-to-image translation tasks, such as edges to the photo, day to night, semantic map to the street scene, etc. It also uses an L1 loss to preserve the low-frequency information of the input images. For multi-modal conditional information, Product-of-Experts GAN (PoE-GAN) [13] consists of a product-of-experts generator and a multi-modal multi-scale projection discriminator. With the carefully designed network and training scheme, PoE-GAN learns to synthesize high-quality and diverse images.

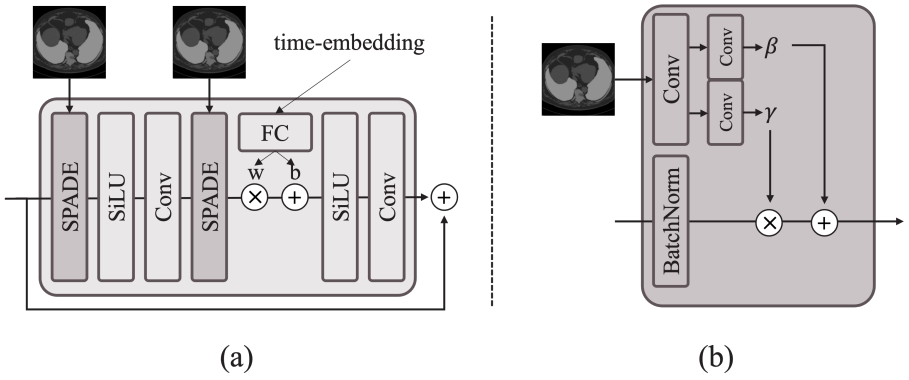


FIGURE 2.1: (a). SPADE ResNet [9] block with time embedding used in our method. (b). The SPADE normalization layer illustration. The semantic map is first projected onto an embedding space and convolved to produce γ and β, which are the normalization parameters acting pixel-wise on the hidden layers.

Recently, Spatially-Adaptive Denormalization (SPADE) [32] aims to address the limited ability of generative models to capture finer spatial details in image synthesis by directly adapting to the input data's spatial structure. The main idea behind SPADE is to implement spatially-adaptive denormalization layers in the generator network to adjust the normalization statistics in the hidden layers. SPADE shows that directly feeding semantic map to the generator is sub-optimal as the normalization layers in the generator tend to "wash away" semantic information. SPADE bypasses the need for instance normalization and generates images with more fidelity. The normalization process in SPADE is divided into two steps. First, the input semantic maps are projected onto an embedding space and convoluted to produce normalization parameters γ and β, which have the same spatial dimensions as the normalized activation. Second, the produced γ and β are multiplied and added to aforementioned activation element-wise. The illustration of this procedure is shown in Fig. 2.1. SPADE's effectiveness in generating high-quality images has been demonstrated in various image-to-image translation tasks, including scene generation [32, 54], style transfer [48], and facial image synthesis [55].

With the above research efforts concentrating on overcoming the disadvantages of the original GAN structure, the performances has been impressively improved in many downstream generation tasks, yet the nature of mode collapse and training instability has not been completely eliminated in GAN-based models [5].

2.3 Diffusion Models

Diffusion models have been wildly accepted as another dominant technique on image synthesis recently. The idea behind the diffusion process, however, can be traced back to 2015, when [44] proposed an unsupervised learning method under the concept of non-equilibrium thermodynamics, which is named as diffusion probabilistic models (DPM) enlightened from the field of statistical physics. In general, the diffusion process can be formulated as stochastic differential equations (SDEs) that describe the evolution of the target data over time [47, 46]:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w} \tag{2.3}$$

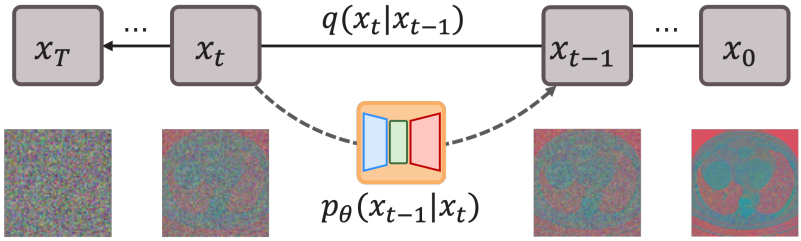


FIGURE 2.2: Forward and reverse diffusion processes. The forward diffusion process (described in equation 2.5) is a Markov chain that gradually adds Gaussian noise to the data. The reverse diffusion process (equation 2.6), on the other hand, de-noise the data by a parameterized generative model $p_\theta(x_{t-1}|x_t)$.

Where $\mathbf{x}$ is the target data we aim to describe, dw represents a Brownian motion process, which can be intuitively understood as an infinitesimal Gaussian noise. In image synthesis, this continuous process can be simplified and explained as follows [11]:

$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}) \tag{2.4}$$

Where $\mathbf{x}(t)$ is the latent representation data over time t, $\alpha(t)$ and $\sigma(t)$ are scalar functions that describe the magnitudes and $\mathbf{z}$ is the Gaussian noise added to the original data with $t = 0$. To parameterize the reverse diffusion process as the generation procedure. The above continuous process is modeled as a Markov chain with the transition probabilities q over discrete time steps t shown as follows[11]:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{2.5}$$

Where $\beta_1, ..., \beta_t$ are fixed variance schedule of the Gaussian distribution. As is dictated in equation **??**, a noisy sample $\mathbf{x}_t$ can be obtained directly from the clean sample $\mathbf{x}_0$ as $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}$ with $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ and $\alpha_t := 1 - \beta_t, \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. The parameterized generative model then learns to reverse this transition over time steps and gradually produce realistic images from Gaussian noise. Specifically, denoising diffusion probabilistic model (DDPM) [11] formulate the following reverse transitions:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\sigma}_t) \tag{2.6}$$

In DDPM implementation, the estimated mean of the Gaussian distribution $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ is achieved by predicting the added noise of the forward diffusion process $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ by a U-Net [37] neural network with:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)) \tag{2.7}$$

To summarize, DDPM learns the reverse diffusion process mean function $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ by predicting the added noise $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ in the forward diffusion process. DDPM simplifies the diffusion model's variational bound to an objective as follows:

$$\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2 \tag{2.8}$$

The objective is derived from minimizing the variational bound of the negative log-likelihood of the data distribution, which is equivalent to minimize the KL divergence

between the real and the estimated posterior probability distribution of the data, enabling diffusion models to formulate a more accurate distribution assumption than the GAN-based models whose goal is to cheat the discriminator.

Several variants of DDPM have been proposed with different modifications and improvements, including denoising diffusion implicit models (DDIM) [45], improved DDPM [30], cold diffusion [3], etc. Unlike DDPM, DDIM uses a non-Markovian diffusion process for more efficient and flexible sampling, which is optionally deterministic without adding random noise during generation. Derived from the original DDPM forward diffusion process, the modified non-Markovian reverse diffusion process of DDIM is:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \alpha_t^2\mathbf{I}) \tag{2.9}$$

Where $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$ is the variance of the posterior conditioned on both current state $\mathbf{x}_t$ and the initial state $\mathbf{x}_0$. When $\sigma_t^2 = 0$, the sampling process is deterministic and can be skipped during the inference, and only a subset of the time steps has reverse diffusion sampling, leading to faster generation using a much fewer number of steps. DDIM also has consistency property according to the optional deterministic generation process, meaning that the generated samples conditioned on the same initial noise latent have similar high-level features. Improved DDPM [30] parameterizes the variances of the reverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to improve the scales of the added noises during the process. It also uses a cosine schedule on the variance magnitudes instead of a linear one, enabling more accurate sampling at the end of the reverse diffusion process. Cold diffusion [3] is a most recent diffusion-based method which uses a series of deterministic processes that corrupt the input data via variant operations including blurring, masking, down-sampling, etc. It also proposes a sampling mechanism based on bi-direction degradation. The main claim of cold diffusion is that for diffusion-based models, it is not mandatory to inject noise during the diffusion process.

Instead of describing the diffusion model as a denoiser of the preturbed data with a finite number of noise distributions, score-based models [47, 46] start from considering the SDEs of the data distribution envolved over continuous time, which is illustrated in 2.3. By reversing the SDE's process in 2.3, a reverse-time SDE [46] is satisfied:

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}}\log p_t(\mathbf{x})]dt + g(t)d\bar{\mathbf{w}} \tag{2.10}$$

Where $\bar{\mathbf{w}}$ is a Brownian motion process when time flows backwards from T to 0 and $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$ is the score function which is estimated by the models:

$$\mathbf{s}_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}}\log p_t(\mathbf{x}) \tag{2.11}$$

The estimation can be optimized by score-matching methods [47] as is described in Score-matching with Langevin Dynamics (SMLD) and Score-based generative modeling [46]. The generation process of the original score-based model then plug the score function into the following Langevin Markov chain Monte Carlo (MCMC) method:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \epsilon\mathbf{s}_\theta(\mathbf{x_i}, i) + \sqrt{2\epsilon}\mathbf{z}_i, \quad i = 0, 1, ..., K \tag{2.12}$$

Where $\mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I})$. Other score-based models follow the similar sampling procedure with minor changes in the equation above. Mathematically, the score function update is equivalent to the denoising update in 2.6 as is discussed in [46]:

$$\mathbf{s}_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}}\log p_t(\mathbf{x}) = \frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}} \tag{2.13}$$

To summarize, diffusion models generate data with iterative sampling processes under a more systematic formulation of matching the data distribution than GAN-based models. Diffusion models have made significant breakthrough in image, audio and video generation [30, 16, 36, 39]. On the other hand, several disadvantages of them (discussed in the first paragraph of section 3.1) which limit the further applications have been discussed in recent research and are currently being overcame by the fast-developing community.

2.4 Conditional Information in Diffusion Models

One key limitation of the generative models is the trade-off between sample fidelity and diversity [26]. Leveraging conditional information, GAN-based models are able to trade off diversity for fidelity to produce high-quality samples [63, 5], while vanilla diffusion-based models can't. To address this problem, [5] developed guided diffusion models with independent classifiers of the intermediate generated data, namely classifier guidance. In particular, the classifier is trained on noisy images $\mathbf{x}_t$ and used for guiding the diffusion sampling process in 2.6 with its gradient:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, y) := \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_\theta + s\boldsymbol{\Sigma}_\theta \nabla_{\mathbf{x}_t} \log p_\phi(y|\mathbf{x}_t), \boldsymbol{\Sigma}_\theta) \tag{2.14}$$

Where $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ are predicted mean and variance of the reverse diffusion process in Improved DDPM [30], $\nabla_{\mathbf{x}_t} \log p_\phi(y|\mathbf{x}_t)$ is the gradient of the classifier given the class label y of the data. With the shifted mean of the reverse diffusion process, classifier guidance improves image quality with additional conditioning information, while decreasing the divergence of the generated data as cost.

Classifier guidance requires training a separate classifier in addition to the diffusion model, leading to extra computational cost and higher complexity. To alleviate the computational cost, Classifier-free guidance [12] jointly trains a conditional and an unconditional diffusion model with conditional information. Instead of training a separate classifier model, classifier guidance use a single neural network $\epsilon_\theta(\mathbf{x}_t, t, y)$ to parameterize both conditional and unconditional diffusion models, where for the unconditional input a null token $y = \emptyset$ is plugged. During training, the model is fed with a mixture of the conditional and unconditional inputs y, the sampling procedure then uses the following linear combination of the conditional and unconditional estimates:

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, y) = (1 + w)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \emptyset) \tag{2.15}$$

Where w is the guidance strength. Classifier-free guidance provides another approach to increase sample quality while decreasing diversity in diffusion models. However, in the case of using other types of guidance such as regression model guidance and local in-painting style guidance, classifier-free guidance is hard to implement and loses guarantees on performance, which are the reasons why we use explicit classifiers for the model illustrated in the next chapter.

Another key problem for the diffusion-based models is the iterative slow sampling procedure, leading to limited applications on real-time scenarios. To enable diffusion models training on limited computational resources while retaining their advantages, latent diffusion models (LDMs) [36] implement the forward and reverse diffusion processes in a information-preserved latent space of powerful pre-trained autoencoders, considerably speeding up the training and inference procedures while preserving the quality and flexibility of diffusion models. Furthermore, LDMs achieve state-of-the-art score on various tasks such as image in-painting and class-conditional image synthesis. Consequently, LDMs have became the backbone structures of the dominant text-to-image generation model series Stable Diffusion [36].

2.5 Small Semantic Area Generation

Zooming out to the overall picture of generative models including GAN-based structures, a general concern highly related to our nodule generation task is how to precisely synthesize small segmentation areas given a complex semantic map. In general, vanilla generative methods can feed the semantic map as input to the neural network, enabling implicit awareness of the semantic information. Spatially-adaptive normalization (SPADE) [32] introduces a simple but effective layer in GAN-based models for synthesizing photorealistic images given a semantic map. Instead of directly feeding the semantic layout as input to the neural network like the previous methods, which is shown to be suboptimal as the normalization layers tends to eliminate semantic information, SPADE proposes using the semantic layout for modulating the activations in normalization layers through a spatially-adaptive, learned transformation. SPADE is shown to better preserve semantic information than common normalization layers, even for small semantic areas in the layout.

Inspired by the spatially-adaptive normalization, semantic diffusion models (SDMs) [54] adapts the SPADE layer into the U-Net [37] neural network structure, enabling SDMs to benefit from both precise semantic awareness and diffusion models' flexibility.

To summarize, recent research has shown incredible generation ability of diffusion-based models in various fields. Combining the semantic information utilization and the flexibility of the diffusion process, generative models can synthesize realistic images with high fidelity to the small semantic areas. With the benefit of all the related research, we present latent semantic diffusion models (LSDMs) with local in-painting style classifiers to explicitly control the characteristics of the small semantic areas. Our method is illustrated in the next chapter.

Chapter 3

Method

In this chapter, we first give the rationale why we use diffusion models in the latent space instead of the original pixel space. Then we introduce the method to train the autoencoders to obtain such latent space. After that, we present the main model structure of LSDMs with implementation details. In addition, we focus on the design of local in-painting style classifiers and the mechanism of them interacting with the reverse diffusion process.

3.1 Latent Space

A widely-accepted hypothesis of the real-world data is that the data tend to concentrate on low dimensional manifolds embedded in a high dimensional space (a.k.a., the ambient space). This manifold hypothesis empirically holds for many real-world datasets, and has become one of the fundamental ideas of manifold learning [50, 49, 25, 29, 64]. Under this hypothesis, a natural connection is that the proportion of effective information with respect to the pixel bits of an image is small in most of the real-world data. In other words, most bits of a digital image correspond to the imperceptible details [36]. On the other hand, diffusion models suffer from iterative slow sampling procedure and expensive computational cost. Previous diffusion-based models evaluate the neural network backbone (both in training and inference) on all pixels, leading to unnecessarily heavy optimization and inference in an iterative manner [36].

Aiming at reducing the redundant computation mentioned above, we implement diffusion models in the latent space, which effectively eliminates the unrelated pixel bits of the original ambient space. In order to extract a reasonable latent space, an autoencoder is trained on the original dataset. With carefully designed loss objective and image-specific inductive bias, the autoencoder filters the pixel bits and the U-Net backbone of the diffusion model can further focusing on the perceptually most relevant information in the latent space:

$$\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2 \tag{3.1}$$

Where $\mathbf{z}_0 \in \mathbb{R}^{h\times w\times c}$ is the encoded image-like data in the latent space from the trained encoder $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0), \mathbf{x}_0 \in \mathbb{R}^{H\times W\times 3}$. Specifically, the encoder downsamples the image $\mathbf{x}_0$ by a downsampling factor $f = H/h = W/w$, the diffusion model then implement training and inference pipeline in the encoded latent space. We set $f = 4, c = 3$ in our final method. We explain the choice of downsampling factor in Chapter 7.

Following the implementation of LDM [36], the loss objective of training the autoencoder in our method contains three functional parts: reconstruction term, adversarial term

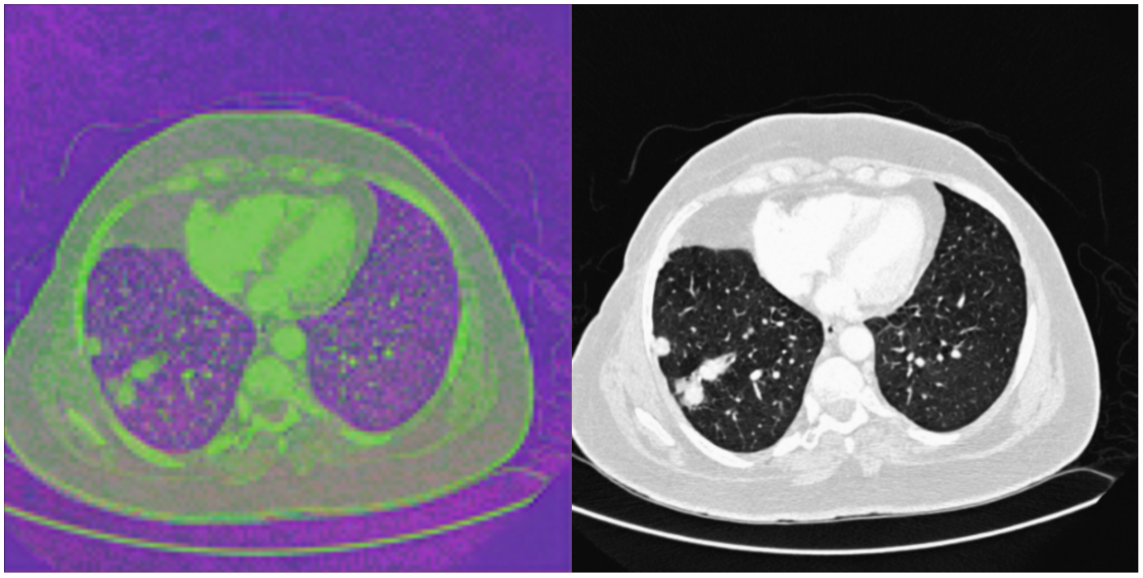


FIGURE 3.1: Visualization of the generated latent code (left) after the reverse diffusion process and the final synthetic CT image (right). Due to the usage of 3-channels latent space $\mathbf{z} \in \mathbb{R}^{128\times128\times3}$, we can visualize the latent code in RGB channel setting. The detailed structure of the original CT scan is preserved in the latent data even for some small speckled-like area. Furthermore, the latent data tends to distill different patterns of the small speckled-like areas in different values of the latent pixels, which reflects considerable expressive capability.

and regularization term:

$$\mathcal{L}_{autoencoder} = \min_{\mathcal{D},\mathcal{E}} \max_{\phi} \Big(\mathcal{L}_{rec}\big(\mathbf{x}, \mathcal{D}(\mathcal{E}(\mathbf{x}))\big) - \mathcal{L}_{adv}\big(\mathcal{D}(\mathcal{E}(\mathbf{x}))\big) + \log \mathbf{D}_{\phi}(\mathbf{x}) + \mathcal{L}_{reg}(x; \mathcal{E}, \mathcal{D}) \Big) \tag{3.2}$$

Where $\mathcal{L}_{rec}\big(\mathbf{x}, \mathcal{D}(\mathcal{E}(\mathbf{x}))\big)$ is the reconstruction term to minimize the difference between original image and the reconstructed image, $\log \mathbf{D}_{\phi}(\mathbf{x}) - \mathcal{L}_{adv}\big(\mathcal{D}(\mathcal{E}(\mathbf{x}))\big)$ is the adversarial term optimizing the discriminator $\mathbf{D}_{\phi}$ to differentiate original images from reconstructions $\mathcal{D}(\mathcal{E}(\mathbf{x}))$ while enabling the autoencoder to cheat the discriminator $\mathbf{D}_{\phi}$. The last term $\mathcal{L}_{reg}(x; \mathcal{E}, \mathcal{D})$ is the regularization term avoiding arbitrary scaled latent spaces with a very small weight factor to obtain high-fidelity reconstructions. Note the different notations between the decoder $\mathcal{D}$ and the additional discriminator $\mathbf{D}_{\phi}$.

Fig. 3.1 shows the autoencoder results with the original image $\mathbf{x} \in \mathbb{R}^{512\times512\times3}$ and the latent $\mathbf{z} \in \mathbb{R}^{128\times128\times3}$. The detailed structure of the original CT scan is preserved in the latent data even for some small speckled-like area. Furthermore, the latent data tends to distill different patterns of the small speckled-like areas in different values of the latent pixels, which reflects considerable expressive capability.

In general, instead of directly implementing forward and reverse diffusion process in the original pixel space, we use an autoencoder trained with a well-suited loss objective and transfer the original pixel space into an expressive latent space, in which we manipulate diffusion and denoise processes with an U-Net network. The latent code preserves most of the perceptual information in the image while considerably decrease the computational

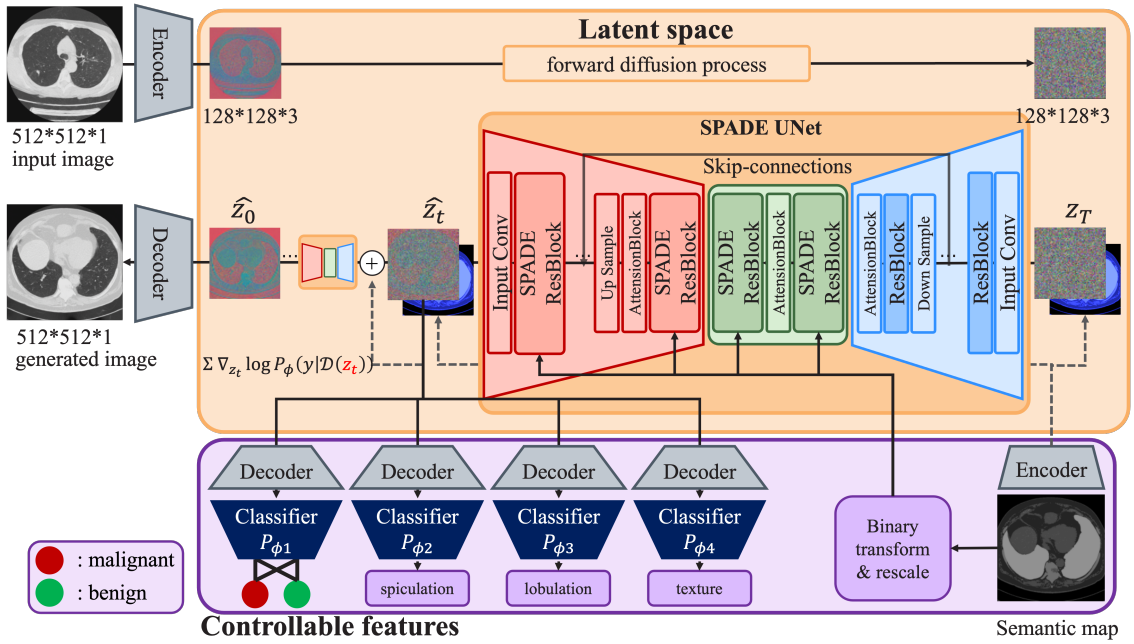


FIGURE 3.2: The overall LSDM model illustration. With the semantic map and several controllable pathology features (bottom purple panel), we explicitly control the generated image with desired feature values of the lung nodules.

cost of the diffusion models. In particular, we choose the size of the latent space to be $\mathbf{z} \in \mathrm{R}^{128\times128\times3}$ when the original image size of LIDC-IDRI dataset is $\mathbf{x} \in \mathrm{R}^{512\times512\times3}$. We train the autoencoder using selected training set of LIDC-IDRI dataset, which will be introduced in Chapter 4.

3.2 Diffusion Model

In this section, we introduce the detailed structure of our diffusion model and present the reasons of our particular choices. We start with the overall picture of our model, following with the model implementation and training pipeline, and finally end up with the inference procedure.

As is discussed in the last section, we express the diffusion process in the latent space with image-specific inductive bias. The high-level structure of our proposed model named latent semantic diffusion model (LSDM) is shown in Fig. 3.2. LSDM consists of an autoencoder model with encoder $\mathcal{E}$, decoder $\mathcal{D}$ and reconstruction discriminator $\mathbf{D}_\phi$, an U-Net network as the core structure predicting the Gaussian noise in the latent $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)$, and a series of pathology features classifiers ${}^i\mathbf{p}_\psi(y_i|\mathbf{x}_t)$, where y_i is the i-th pathology feature label of the CT scan evaluated from four radiologists in LIDC-IDRI dataset.

The autoencoder $(\mathcal{E}/\mathcal{D}/\mathbf{D}_\phi)$ is based on a residual neural network (ResNet) [9] with attention mechanism plugged in the intermediate representations [52]. The U-Net backbone $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)$ also formulated with multi-resolution ResNet blocks, which contain skip connections to better enhance spatial fidelity [37] and spatial-adaptive normalization layers to better preserve the information of the input semantic label. The inputs of the U-Net backbone are the image latent code concatenated with the encoded semantic map $\mathcal{E}(\mathbf{m})$, which

double the number of channels to $2c$:

$$\textbf{Concat}(\mathbf{z}_t, \mathcal{E}(\mathbf{m})) \in \mathrm{R}^{h \times w \times 2c} \tag{3.3}$$

It should be noted that the autoencoder's training data doesn't include the segmentation maps. Yet we found that the autoencoder possesses the capability of extracting sufficient information from the segmentation maps.

The training procedure of the U-Net backbone is as follows: Firstly, we sample pure Gaussian noise in the latent space $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I}), \quad \mathbf{z}_T \in \mathbb{R}^{h \times w \times c}$. Secondly, we encode the original CT scan into the latent space $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0) \in \mathbb{R}^{h \times w \times c}$ and obtain the noisy input of the U-Net backbone given a random timestep t: $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}_T$ following equation 2.5. Then we concatenate the noisy input $\mathbf{z}_t$ with the encoded segmentation map $\mathcal{E}(\mathbf{m})$ and feed into the U-Net backbone parameterized with θ. The inputs also include the time embedding information t. Finally, we compute the mean squared error (MSE) loss of the output with respect to the true noise $\|\mathbf{z}_T - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)\|^2$ and back-propagate the loss to update the network parameters θ. The overall training procedure is summarized in Algorithm 1.

Algorithm 1 LSDM Training Procedure

Require: dataset $\{\mathbf{x}_0\}$, trained autoencoder $\mathcal{E}$ and $\mathcal{D}$, diffusion schedule $\bar{\alpha}_t$
  repeat
    randomly select a batch of data $\mathbf{x}_0 \sim \{\mathbf{x}_0\}$
    $\mathbf{z}_0 \leftarrow \mathcal{E}(\mathbf{x}_0)$
    $t \sim \text{Uniform}(1, ..., T)$
    $\mathbf{z}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$
    Take the gradient descent step on $\nabla_\theta \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)\|^2$
  until convergence

With a well-trained LSDM backbone model $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)$, we can generate samples in the latent space $\hat{\mathbf{z}}_0$ by evaluating the U-Net backbone network iteratively using full time schedule of the reverse diffusion process as in 2.6 or, alternatively, selected time schedule as in 2.9. After generating the final sample $\hat{\mathbf{z}}_0$, we can obtain the generated CT scan by passing the sample through the trained decoder $\hat{\mathbf{x}}_0 = \mathcal{D}(\hat{\mathbf{z}}_0)$. The overall inference pipeline is illustrated in Algorithm 2. In addition, the detailed inference procedure also includes conditional information (segmentation maps) and classifier guidance, which will be discussed in the next two sections.

Algorithm 2 LSDM Inference Pipeline

Require: trained autoencoder $\mathcal{E}$ and $\mathcal{D}$, U-Net backbone $\boldsymbol{\epsilon}_\theta$, trained four classifiers $\mathbf{p}_{\psi i}(y_i|\text{Crop}(\mathcal{D}(\mathbf{z}_t)))$, $i = 1, 2, 3, 4$, diffusion schedule $\bar{\alpha}_t$, i-th classifier's label y_i, gradient scale s
  $\mathbf{z}_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
  for all t from T to 1 **do**
    $\mathbf{z}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)\right) + s\Sigma_{i=1}^{4}\sigma_t \log \mathbf{p}_{\psi_i}(y_i|\text{Crop}(\mathcal{D}(\mathbf{z}_t))) + \sigma_t\boldsymbol{\epsilon}', \quad \boldsymbol{\epsilon}' \sim \mathcal{N}(0, \mathbf{I})$
  end for
  $\mathbf{x}_0 \leftarrow \mathcal{D}(\mathbf{z}_0)$
  return $\mathbf{x}_0$

3.3 Diffusion Model with Spatial-adaptive Normalization

In the last section, we utilize the segmentation maps by concatenating them with the noisy input $\mathbf{z}_t$ and feed into the U-Net backbone. To further preserve the spatial semantic

information inspired by SPADE model [32], we insert the SPADE layers in the U-Net backbone network with pre-processed segmentation maps as the guided information of the SPADE layers. Fig. 3.2 shows the SPADE U-Net backbone with adjusted ResNet blocks in each part of the network. The pre-processing of the segmentation maps follows the procedure in the original SPADE model [32], which downsamples the segmentation maps into multi-resolution masks corresponding to the input size of a series of SPADE ResNet blocks.

In particular, given a semantic map with the same spatial size of the original image $m \in \mathbb{R}^{H\times W\times 1}$, where the pixel value of m is the segmentation class label of the corresponding pixel in the original image, we downsample the map using max-pooling and transfer each semantic label into a separate channel with binary values. The pre-processed mask is a binary mask $\tilde{m} \in \mathbb{Z}_2^{h_i\times w_i\times k}$, where h_i and w_i corresponding to the input size of the i-th ResNet blocks and k is the number of segmentation class. Fig. 2.1 shows the normalization method of the SPADE layer, with the input binary mask $\tilde{m}$ in different downsampling sizes. In Chapter 5, we compare the generation results of the models with and without SPADE layers in the U-Net backbone, which illustrate the effectiveness of the modulation.

3.4 Classifier Guidance

In order to improve the fidelity of the generated images, which is critical in medical image synthesis, we exploit a series of nodule classifiers to improve the accuracy and controllability of the nodule area synthesis. Each classifier is trained on noisy latent codes $\mathbf{z}_t$ with a selected pathology feature of the nodules provided by 4 radiologists in LIDC-IDRI dataset. The network structure of the classifiers is shown in Fig. 3.3, most of whose structures follow the encoder part of U-Net backbone with an additional pooling layer grafted before the classifier's output.

In the normal classifier guidance of diffusion models, the gradients $\nabla_{\mathbf{x}_t}\log\mathbf{p}_\psi(y|\mathbf{x}_t,t)$ are used to guide the diffusion inference process towards a selected class label y. The log derivative $\nabla_{\mathbf{x}_t}\log\mathbf{p}_\psi(y|\mathbf{x}_t,t)$ is obtained by conditioning the probability of the reverse diffusion process 2.6:

$$\mathbf{p}_{\theta,\psi}(\mathbf{x}_t|\mathbf{x}_{t+1},y) = \frac{\mathbf{p}_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})\mathbf{p}_\psi(y|\mathbf{x}_t)}{\mathbf{q}(y|\mathbf{x}_{t+1})} \tag{3.4}$$

Where $\mathbf{q}(y|\mathbf{x}_{t+1})$ is a constant since it doesn't depend on variable $\mathbf{x}_t$. Using Taylor expansion approximation, the conditional log probability of the reverse diffusion process can be estimated as:

$$\log\big(\mathbf{p}_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})\mathbf{p}_\psi(y|\mathbf{x}_t)\big) \approx \log\mathbf{p}(\mathbf{x}'_t), \quad \mathbf{x}'_t \sim \mathcal{N}(\boldsymbol{\mu}+\boldsymbol{\Sigma}\mathbf{g},\boldsymbol{\Sigma}) \tag{3.5}$$

Where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and variance of the unconditional reverse diffusion probability $\mathbf{x}_t \sim \mathbf{p}_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}) = \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$, $\mathbf{g}$ is the log derivative of the classifier $\nabla_{\mathbf{x}_t}\log\mathbf{p}_\psi(y|\mathbf{x}_t,t)$. This property enables us to directly add the log derivative $\mathbf{g}$ of the classifier to the estimated mean of the reverse diffusion process $\boldsymbol{\mu}$ while preserving the conditional probability distribution $\mathbf{p}_{\theta,\psi}(\mathbf{x}_t|\mathbf{x}_{t+1},y)$. The inference step of the classifier guided diffusion then becomes the following equation according to 2.7:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}}\big(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}(\mathbf{x}_t,t)\big) + s\sigma_t\log\mathbf{p}_\psi(y|\mathbf{x}_t) + \sigma_t\mathbf{z}, \quad \mathbf{z}\sim\mathcal{N}(0,\mathbf{I}) \tag{3.6}$$

Where s is the gradient scale as the guidance weight of the classifier, σ_t and β_t are the diffusion variance schedule in 2.5 and 2.6. In LSDM, we fix the variance schedule β_t as a

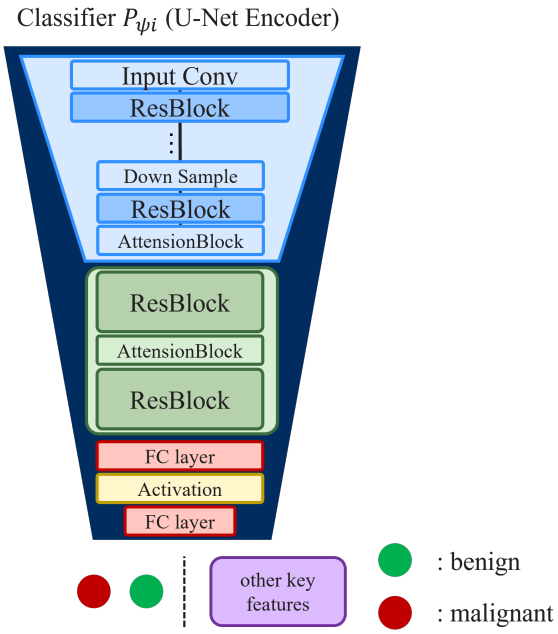


FIGURE 3.3: Classifier structure illustration. We utilize the encoder block (blue) and the middle block (green) of the LSDM U-Net backbone without SPADE layers to ensure the simplicity and consistency of the model. The output of the classifier is either the binary label (used in malignancy) with two neurons in the last fully-connected (FC) layer, or the regression label (used in other key features) with one neuron as the last FC layer.

linear function with respect to the timestep t. The gradient scale s is a key hyper-parameter in LSDM, on which we do an ablation study in Chapter 5.

3.4.1 Local In-painting Multi-classifier guidance

Instead of adding the whole gradient to the intermediate output $\mathbf{x}_{t-1} \in \mathbb{R}^{H\times W\times C}$ pixel-wise, we present local in-painting multi-classifier guidance onto the intermediate output $\mathbf{z}_{t-1} \in \mathbb{R}^{h\times w\times c}$ in the latent space. To introduce the LSDM classifier guidance step-by-step, we first illustrate the meaning of local guidance, then present the in-painting style classifier guidance with a nodule mask, finally we show the pipeline of multi-classifier guidance.

Local classifier guidance

We train the classifier on noisy latent codes $\mathbf{z}_t$ with a selected pathology feature of the nodules provided by 4 radiologists in LIDC-IDRI dataset. In particular, we decode the noisy latent codes into the original pixel space $\mathbf{x}_t = \mathcal{D}(\mathbf{z}_t)$ and crop them around the nodule areas to obtain the inputs of the classifiers $\tilde{\mathbf{x}}_t$. As is shown in Fig. 3.4, we frozen the trained decoder $\mathcal{D}$ and plug it into the classifier training pipeline before the classifier network parameterized with ψ:

$$\mathbf{p}_\psi(y|\mathbf{x}_t, t) = \mathbf{p}_\psi\big(y|\text{Crop}(\mathcal{D}(\mathbf{z}_t)), t\big) \tag{3.7}$$

Where y is the pathology feature value of the nodules, which will be introduced in Chapter 4. We call this local classifier guidance for the cropped inputs of the classifiers.

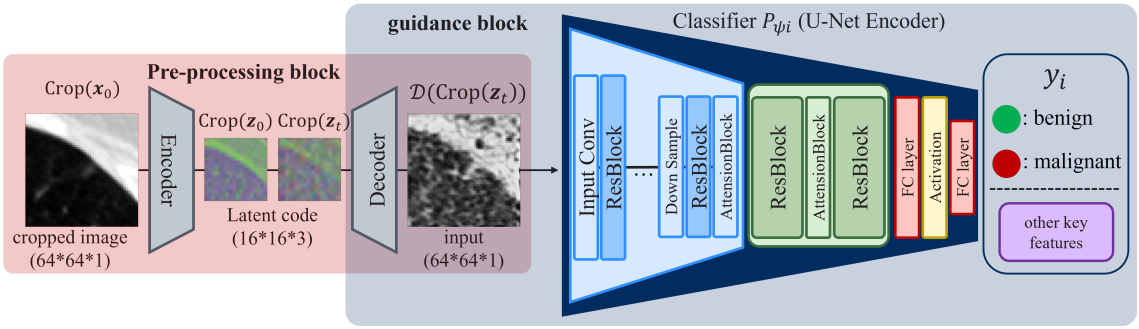


FIGURE 3.4: The training pipeline of the classifiers. In the pre-processing block (red box), we first crop the original CT image centering at the nodule and encode it into the latent space. Then we add Gaussian noise to the cropped latent code following the schedule of the forward diffusion process in equation 2.5 with a random time step t. We complete the pre-processing step by decoding the noisy latent code Crop($\mathbf{z}_t$) into the original pixel space, which is the training data of the classifiers.

In-painting style classifier guidance

Iteratively adding the log derivative locally may introduce undesirable artifacts around the boundary of the cropped areas. To eliminate this, we use in-painting guidance only concentrated on the nodule areas. In particular, we generate a binary mask $\mathbf{M}$ around the nodule semantic area using max-pooling operation in the latent space, then implement the guidance only on unmasked region of the cropped input:

$$\tilde{\mathbf{g}} = (1 - \mathbf{M})\nabla_{\tilde{\mathbf{z}}_t} \log \mathbf{p}_\psi\big(y|\text{Crop}(\mathcal{D}(\mathbf{z}_t)), t\big) \tag{3.8}$$

Where $\tilde{\mathbf{g}}$ is the masked gradient in the latent space. In LSDM, we choose the size of the cropped image to be $\tilde{\mathbf{x}}_t = \text{Crop}(\mathcal{D}(\mathbf{z}_t)) \in \mathbb{R}^{64\times64\times C}$. Consequently, the size of the cropped latent code is $\tilde{\mathbf{z}}_t \in \mathbb{R}^{16\times16\times3}$. Fig. 3.5 illustrates the in-painting guidance during the inference process.

The in-painting guidance prevents the generated image from unnatural artifacts of the cropping boundary. Furthermore, diffusion models in the latent space have shown sufficient capability in in-painting tasks [36], allowing us to safely implement the masked gradients.

Multi-classifier guidance

To explicitly control the pathology characteristics of the nodules, we use multi-classifier guidance concentrated on the following key features of the nodules: malignancy, lobulation, spiculation and texture, which are evaluated by 4 radiologists in LIDC-IDRI dataset and will be further described in Chapter 4. In general, we train a binary classifier on malignancy and regression classifiers on other features, because malignancy is rated from a comprehensive judgment, while others are evaluated as the degrees of certain morphology characteristics.

In particular, the malignancy of a nodule has an integer value $y_{mal} \in [1, 5]$, and we consider the nodule with $y_{mal} \leq 2$ to be benign and $y_{mal} \geq 3$ to be malignant, which emphasizes the significance of true positive (TP) and avoids false negative (FN). The lobulation, spiculation and texture of a nodule also have integer values $y_{lob}, y_{spi}, y_{tex} \in [1, 5]$, and we directly utilize the values to be the labels of our regression classifiers.

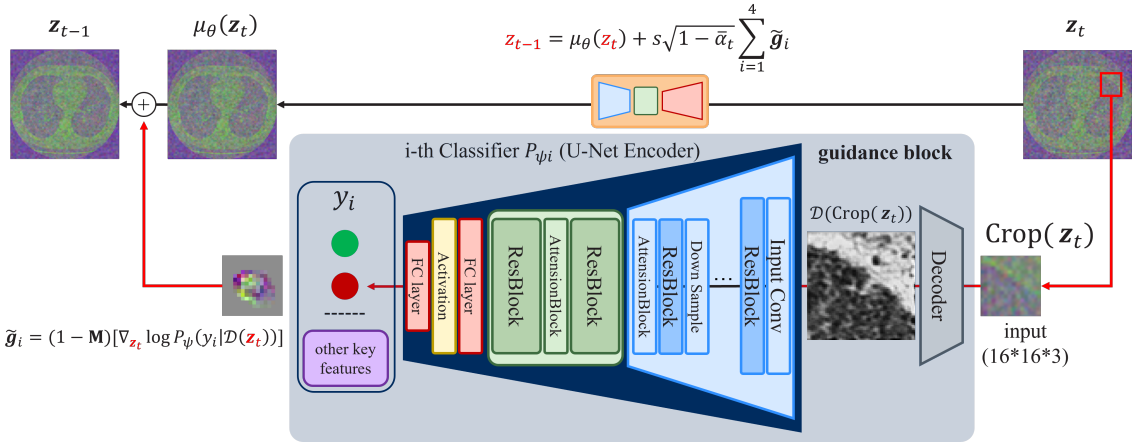


FIGURE 3.5: The guidance pipeline of the classifiers. From the intermediate latent code $\mathbf{z}_t$, we crop it centering at the nodule $Crop(\mathbf{z}_t)$. The guidance block (gray box) contains a frozen decoder $\mathcal{D}$ followed by the selected classifier $\mathbf{p}_{\psi_i}$, whose output forms the loss $\mathcal{L}(y_i - y_i')$. Then we calculate the derivative of the loss with respect to the cropped latent code and add the mask to the derivative to obtain $\tilde{\mathbf{g}}_i$ in the figure. Finally, the iteration is taken by adding the estimated reverse diffusion process mean $\mu_\theta(\mathbf{z}_t)$ with the masked guidance $\tilde{\mathbf{g}}_i$, which can be optionally combined to formulate multi-classifier guidance.

Finally, we discuss the details of the inference pipeline. We follow the classifier guidance rules in [5] and develop our own local in-painting classifier guidance pipeline. In Fig. 3.5, we illustrate the process of one iteration step $\mathbf{z}_{t-1} \leftarrow \mathbf{z}_t$ with the classifier guidance, where we first crop the latent code $\mathbf{z}_t$ centering at the nodule with size 16×16. The cropped latent code $\text{Crop}(\mathbf{z}_t)$ is subsequently the input of the frozen decoder $\mathcal{D}$, whose output is fed into the selected classifier(s). We then calculate the derivative of the classifier's loss (cross-entropy loss for binary malignancy classifier and mean squared error (MSE) loss for other regression feature classifiers) $\mathcal{L}(y - y')$ with respect to $\text{Crop}(\mathbf{z}_t)$ to obtain the masked log-derivative $\tilde{\mathbf{g}}$ and add it to the full-size latent code to obtain $\mathbf{z}_{t-1}$.

For the multi-classifier case, when we exclude the malignancy classifier, other classifiers equally use the original latent code to calculate the gradients $\tilde{\mathbf{g}}_{spic}, \tilde{\mathbf{g}}_{lobu}, \tilde{\mathbf{g}}_{text}$ and sum them up as the guided result. When the malignancy classifier is included, we first execute the previous step and then calculate the gradient of the malignancy classifier $\tilde{\mathbf{g}}_{mali}$ using the guided result. We compare the generation results of different classifier combinations in Chapter 5.

Chapter 4

Dataset

In this chapter, we introduce LIDC-IDRI dataset which we use to train and evaluate the LSDM model. We first briefly describe the properties of the CT scan dataset and present our pre-processing method. Then we explain the selection of the key features of the lung nodules. After that, we discuss our method to obtain the semantic maps of the CT scans and finally make a conclusion about the difference between the CT scans and the images with other modalities.

4.1 Properties

The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI) established the lung CT scan dataset, which contains 1018 cases from 1010 patients with clinical thoracic CT scans and the results of a two-phase image annotation process performed by four experienced thoracic radiologists [24]. The first annotation phase is blinded-read with each radiologist independently reviewed the CT scans and classified the lesions into three categories: nodule $\geq$ 3 mm, nodule $\leq$ 3 mm and non-nodule $\leq$ 3 mm. The second phase is unblinded-read with each radiologist reviewed their marks along with other three radiologists' opinions to render a final assessment. The nodules with diameter $\geq$ 3 mm are further annotated with contours and include subjective characteristic ratings, from which we consider malignancy, lobulation, spiculation and texture and train classifiers on them.

We select 20% of slices from each scan with or without nodules. For the non-nodule slices, we select them uniformly to prevent data imbalance. We also exclude CT scans whose slice thickness is greater than 2.5 mm for preserving the image quality, resulting in 49992 slices in total.

We split these slices into training and testing set patient-wise, 42952 and 5616 slices respectively. All of the slices are stored as single-channel PNG images with 512×512 resolution. We clip the Hounsfield Units (HU) values of each pixel within $[-1024, 800]$ following the settings in [23], which indicates the meaningful pixel value for nodule classification. The clipped pixel values are subsequently normalized into integers within $[0, 255]$.

For the contours of the nodules, we use the annotations based on a 50% consensus criterion, which implies that the given pixels have been included in the nodule contour by at least 50% of the radiologists who annotated that nodule.

To conclude, we pre-process LIDC-IDRI dataset into 49992 normalized CT scan images with or without nodules. The annotated nodules possess of subjective characteristic ratings, some of which are used for the classifier guidance. The selection of the nodule features is illustrated in the next section.

4.2 Key Features Determination

The annotated nodules have subjective characteristic ratings in eight aspects, namely subtlety, internal structure, spiculation, lobulation, sphericity, solidity, margin, and malignancy. These features are subjective and highly related to each other, leading to difficulties in training the classifiers upon every feature separately. We choose appropriate features with clear influence on the appearance of the nodules, which is determined by the convergence of the classifiers' training. Among all the eight features, four of them succeed to have a converged classifier trained on them: malignancy, lobulation, spiculation, and texture. Empirically, the convergence of the classifier training indicates the relativeness between the corresponding feature and the appearance of the nodule expressed in pixel values, which is crucial for the guidance in pixel values based on that feature. For the training pipeline, we use noisy cropped inputs as discussed in chapter 3 and develop data balancing and exponential moving average techniques to stabilize the training process. In particular, we balance the label distribution of the key features (sample the nodules of different labels with equal probability when training classifiers), and also employ exponential moving average (EMA) when updating the network parameters of the classifiers. That is, given a decay rate $0 \leq \gamma < 1$, we perform the following update after each optimization step:

$$\theta \leftarrow \gamma\theta + (1-\gamma)\theta' \tag{4.1}$$

Where θ' is the updated parameters after back-propagation and we set the decay rate $\gamma = 0.9999$. EMA can greatly stabilize the training process of the classifiers and improve the final performance of the classifier guidance.

As an empirical analysis, we argue that for a noisy input of the classifier, the features with a converged classifier succeed in preserving their relativeness to the appearances of the nodules when mixing with the Gaussian noise, while other features failed because the Gaussian noise eradicated this correlation and thus unable to converge during classifiers' training. As an empirical example, the texture of a nodule indicates the pixel value distribution to some extent, whose relationship with the appearance is hard to be eliminated by mixing with the Gaussian noise, while the margin of a nodule is sensitive to the additional noise and tends to be influenced.

Consequently, we finally train the classifiers on the four selected features, with malignancy to be binary classified and others for regression.

4.3 Semantic Maps

For semantic image synthesis, the semantic map is crucial to generate appropriate images. The semantic maps of the dataset are obtained following [59], which uses two different methods: non-lung segmentation and lung segmentation.

For non-lung segmentation in the CT scan, we classify the areas into three categories based on their HU values: body, soft tissues, and high-dense tissues. The body part is segmented with the HU values in $[-400, 0]$. The soft tissues refer to the substances with HU values in $[0, 200]$, which are higher than skin and fat, while lower than bone and heart. Most likely, they are organs, muscles, connective tissues, and others. The high-dense tissues, whose HU values range from 200 to 800, include the bones and some muscular tissues such as the cardiac muscle.

For lung segmentation, the naive classification by HU values fails. We instead use K-Means classification and marching square algorithm to segment the lung. Compared to

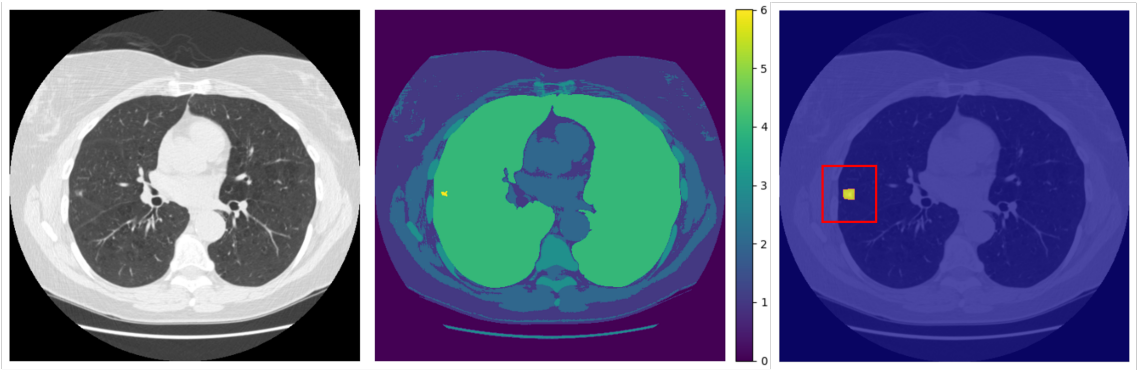


FIGURE 4.1: Pre-processing of a LIDC-IDRI CT scan slice. Left figure: the processed slice image with pixel values normalized into $[-1, 1]$ and the resolution 512×512. Middle figure: the processed semantic map of the slice with eight labels representing the pixel value from 0 to 7, where $0, 1, 2, 3, 4$ are the non-nodule labels and $5, 6, 7$ respectively represent the malignancies of the nodule $y_{mal} < 3, y_{mal} = 3, y_{mal} > 3$. The middle figure contains a nodule with semantic pixel value 6, which indicates the malignancy of it is $y_{mal} = 3$. As discussed in section 3.4.1, we consider the binary label of this nodule to be malignant. Right figure: the in-painting mask M illustrated in equation 3.8 (semi-transparent blue mask). The mask is obtained by implementing max-pooling step to the nodule semantic area. We only employ the classifier guidance in the unmasked areas (yellow). The red box represents the crop box for the nodule classifiers.

non-lung regions, pixel values of lung area are usually lower. We used K-Means method to classify the non-lung and lung pixel with several morphology operations, which include erosion and dilation. The background was excluded by setting up a maximum area threshold. Combined with K-Means method which robustly locate the lung region, we utilize marching square algorithm to find the explicit contour and render a precise boundary between lung and non-lung regions. The selected contours will then be stored as the semantic labels for lung.

Fig. 4.1 shows the processed semantic map of a LIDC-IRDI CT scan slice with a nodule, which has eight semantic labels including background, body, soft tissues, high-dense tissues, lung, and nodules with three stages of malignancy. The pixel values of the semantic maps are illustrated in Fig. 4.1. During the training procedure of LSDM, we normalize the pixel values into $[-1, 1]$ to unify with the pixel range of the image inputs.

Chapter 5

Experiments

In this chapter, we systematically analyze the qualitative and quantitative results for our method, including the image reconstruction capability of the autoencoder, the fidelity of the generated CT scan, and the diversity of the generated nodule areas with respect to the different settings of the controllable key features. We show that our method is capable for generating high-quality lung CT images with the segmentation maps and the features. Furthermore, we can generate diverse nodules using different classifier guidances.

5.1 Qualitative Results

The goal of our method is to synthesize high-quality CT images with controllable characteristics of the nodules. To evaluate the quality, we first visualize the reconstruction results to examine the effectiveness of utilizing latent space. We also compare the generation of the full version of our method (with all four classifiers under the true nodule feature values) to the baseline method (SPADE). We then concentrate on the controllability of the nodule characteristics.

The reconstruction ability of the autoencoder in LSDM is shown in Fig. 3.1, where we zoom in to the details of the lung tissue and distinguish the differences between the ground truth and the reconstructed ones. The reconstruction is implemented by passing the CT images only through the trained encoder and the decoder of LSDM, which are the gray blocks shown in Fig. 3.1 and Fig. 3.2. In addition, the reconstruction procedure doesn't include the concatenation of the semantic maps as inputs. In Fig. 3.1, the high-level structure

Then we compare the synthesis results between the baseline (SPADE) and our method (with all four classifiers combined). We use the true values of the nodule features for the corresponding classifiers to guide the reverse diffusion process of LSDM. In general, our method better utilizes the semantic maps to generate results with higher fidelity and preserves the details of the thoracic structures.

In Fig. 5.2, we select three representative cases. the top row introduces a case with multiple malignant nodules, two of which have relatively blurred boundaries and are actually the parts of one nodule according to the dataset. In this case, SPADE generates an unexpected low-pixel-value area while LSDM doesn't in the red box (a), which is often the situation where the regions of other organs such as the stomach or intestines appear as the scan section moves downwards. In fact, LSDM rarely generates such unexpected areas but SPADE occasionally does. The medium row indicates a case with an extremely small nodule semantic area in the green box (b). SPADE generates a vague pattern dissimilar to the ground truth, while LSDM adheres to the tiny semantic area and generates a more

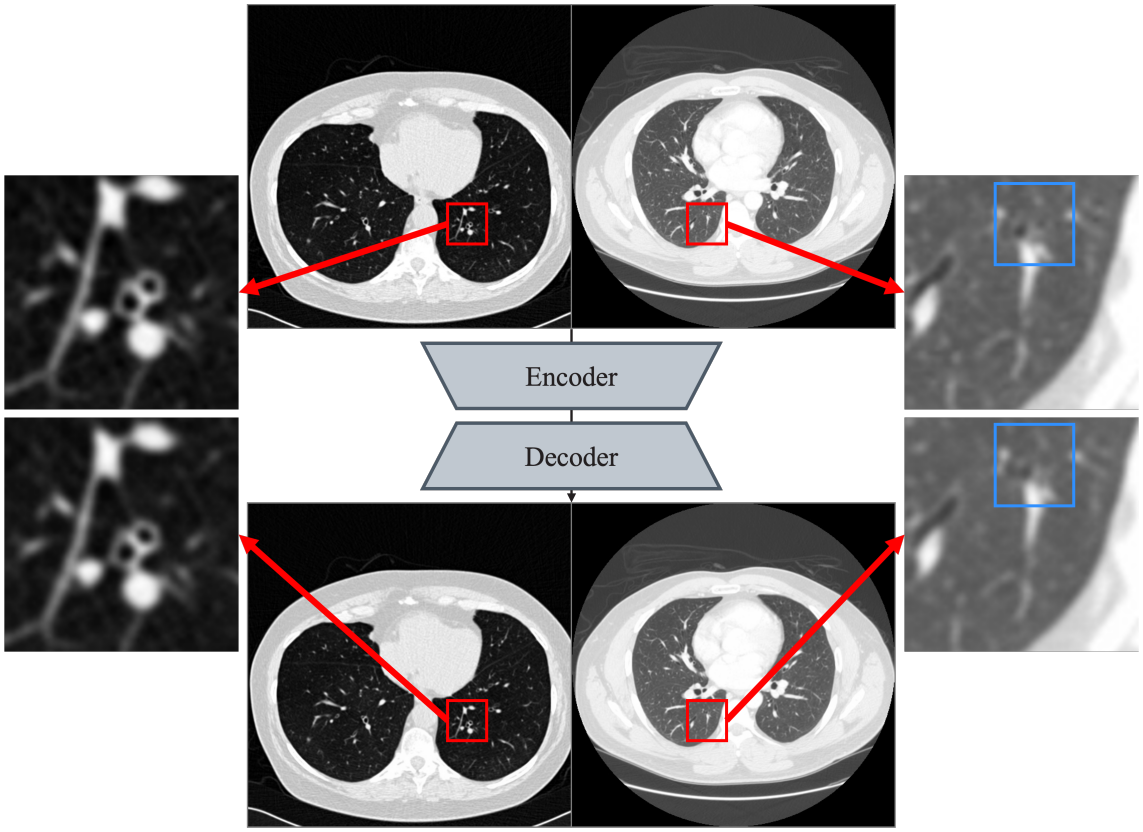


FIGURE 5.1: Comparison between the reconstruction results (top row) and the original CT scans (bottom row). The information are nearly flawlessly preserved, where we zoom in to the red boxes as the left and right side pictures. We can only discern almost imperceptible differences in detail in the blue box of the zoomed picture on the right. As is discussed in Chapter 3, we use the downsampling factor $f = 4$, which leads the latent space to be $\mathbf{z} \in \mathbb{R}^{128\times128\times3}$

.

accurate nodule. Finally, the bottom row presents a case with rich details of the lung areas (i.e. the boundary lines of lung lobes, and cross-sections of bronchi) as shown in the blue box (c). It is worth mentioning that these details are not reflected in the input semantic map. SPADE fails to synthesize such unlabeled details while LSDM tends to express them even around the correct location, indicating the fidelity of the LSDM CT image generations. Fig. 5.3 shows additional comparison results concentrated on the unlabeled lung lobe boundaries, further substantiating the ability of LSDM to synthesize CT images with high fidelity.

To illustrate the effectiveness of the classifier guidance, we first visualize the log derivatives of the classifier with respect to the reconstructed images cropped around the nodule areas. Specifically, the inputs of the classifier are the cropped areas of the reconstructed latent code mixed with Gaussian noise $\text{Crop}(\mathcal{D}(\mathbf{z}_t)) \in \mathbb{R}^{64\times64\times1}$ as shown in the second column of Fig. 5.4. We emphasize that the training data of the classifiers should not be the original cropped images $\text{Crop}(\mathbf{x}_t)$, because during the inference process, these classifiers only take reconstructed inputs before which are mixed with Gaussian noise.

In Fig. 5.4, we visualize the log derivative $\nabla_{\text{Crop}(\mathcal{D}(\mathbf{z}_t))} \log \mathbf{p}_\psi(y|\text{Crop}(\mathcal{D}(\mathbf{z}_t)))$ of the binary malignancy classifier with respect to the reconstructed cropped image $\text{Crop}(\mathcal{D}(\mathbf{z}_t))$

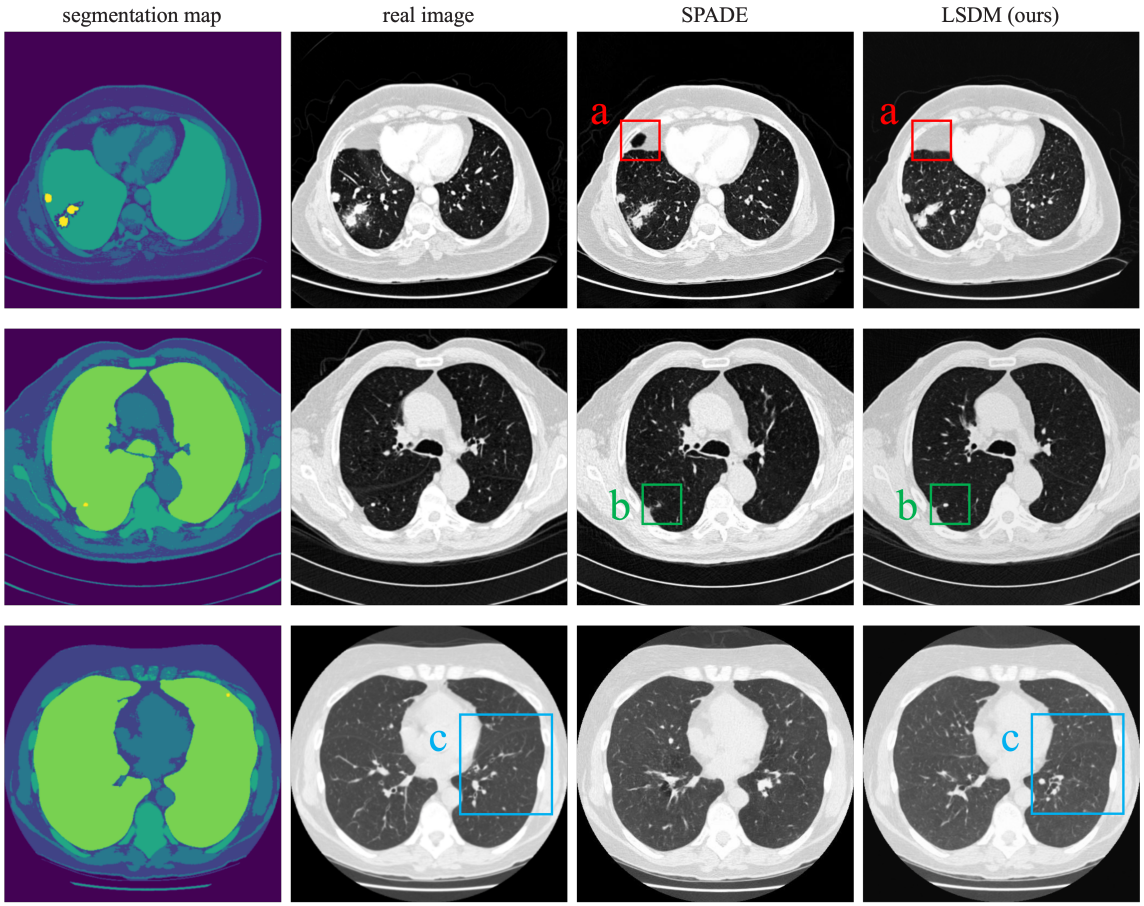


FIGURE 5.2: The qualitative results compared to the real images and the baseline (SPADE) results. We exhibit various situations such as large nodules, tiny nodules, and detailed lung structures. Top raw: SPADE occasionally generates unexpected structures (a), while LSDM does not. Medium row: for tiny nodules (b), SPADE may generate vague pixels while LSDM preserves the clarity. Bottom row: LSDM even possesses the ability to synthesize intricate details of the tissues (i.e. the demarcation lines of lung lobes, and cross-sections of bronchi in (c)).

(we replace the expression $\text{Crop}(\mathcal{D}(\mathbf{z}_t))$ with $\mathbf{x}_t$ in Fig. 5.4 due to the spacial limitation). When we set the class label $y =' \textit{benign}'$ in the third column, we can observe positive derivatives at the center of the nodules and negative derivatives around the boundaries, which is an indication of the shrinkage of the nodule areas and the sharpening of the nodule boundaries. When we set the class label $y =' \textit{malignant}'$ in the last column, the results indicate the opposite effect to the ones in the third column, illustrating reasonable guidance of the binary classifier.

In addition, we fix the guidance scale $s = 1$. Unless specified, we set the same value in all subsequent results.

During the inference process of LSDM, we actually use a slightly different derivative form $(1-M)\nabla_{\text{Crop}(\mathbf{z}_t)} \log \mathbf{p}_\psi(y|\text{Crop}(\mathcal{D}(\mathbf{z}_t)))$ instead of $\nabla_{\text{Crop}(\mathcal{D}(\mathbf{z}_t))} \log \mathbf{p}_\psi(y|\text{Crop}(\mathcal{D}(\mathbf{z}_t)))$. Notice the difference in the denominator of the partial differentiation. M indicates the mask of the derivative generated using the max-pooling method to slightly increase the considered area. In Fig. 5.5, we visualize the real derivative calculated during the inference

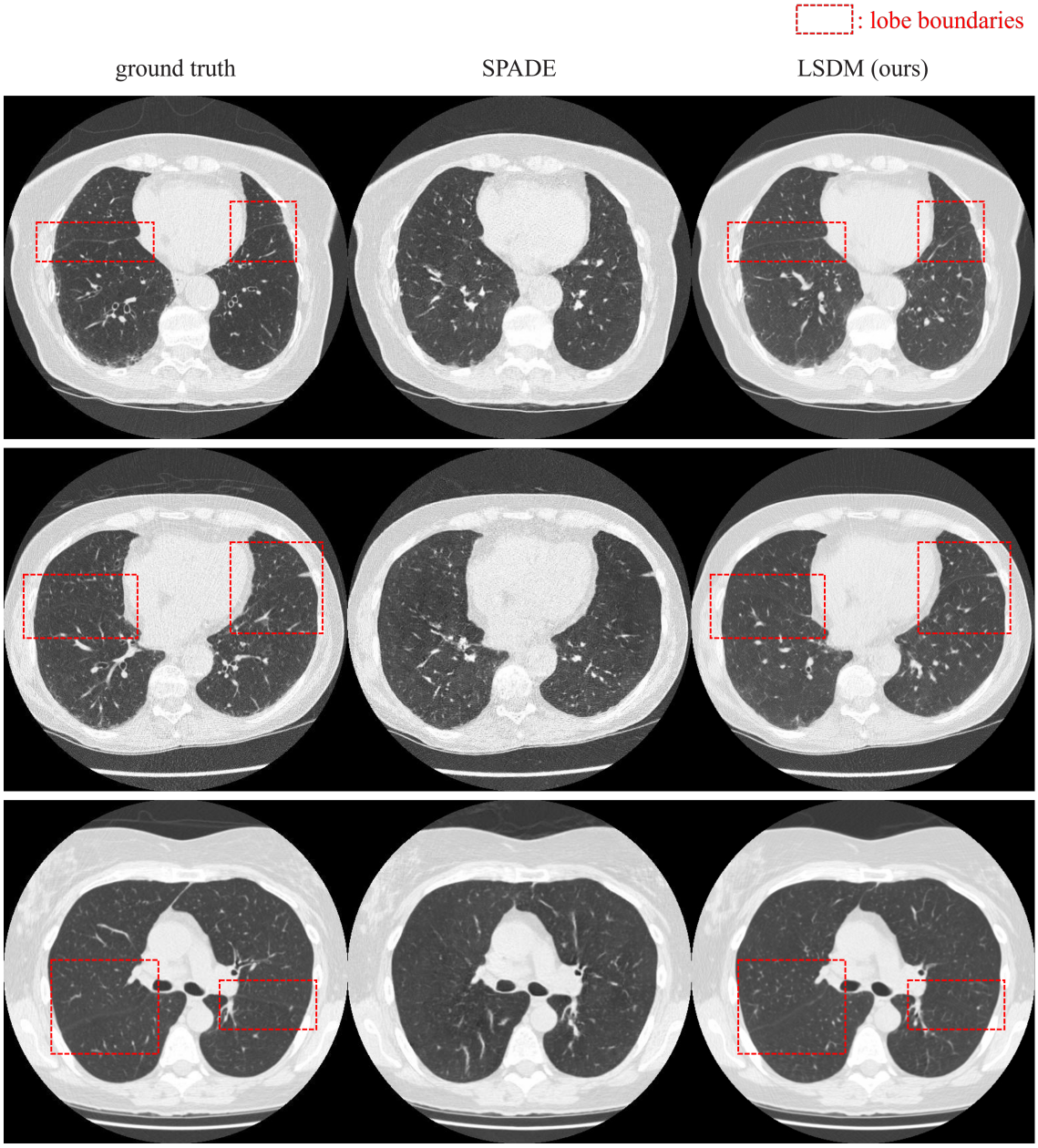


FIGURE 5.3: Unlabeled lung structures generation. The red boxes in the left column represent the lung lobe boundaries in the ground truth images and are not included in the semantic maps. SPADE generations ignore such unlabeled structures, while LSDM tends to reconstruct the hidden information as shown in the right column. Observing the whole generated test dataset, we find that LSDM is able to preserve such details in most cases, demonstrating the ability of LSDM to synthesize CT images with high fidelity.

procedure, where the bottom row shows the different guidance with respect to "benign" and "malignant" class labels. To conclude, Fig, 5.4 and Fig. 5.5 elaborate the rationale of using binary malignancy classifier in local in-painting style, enabling further indication of the multi-classifier guidance.

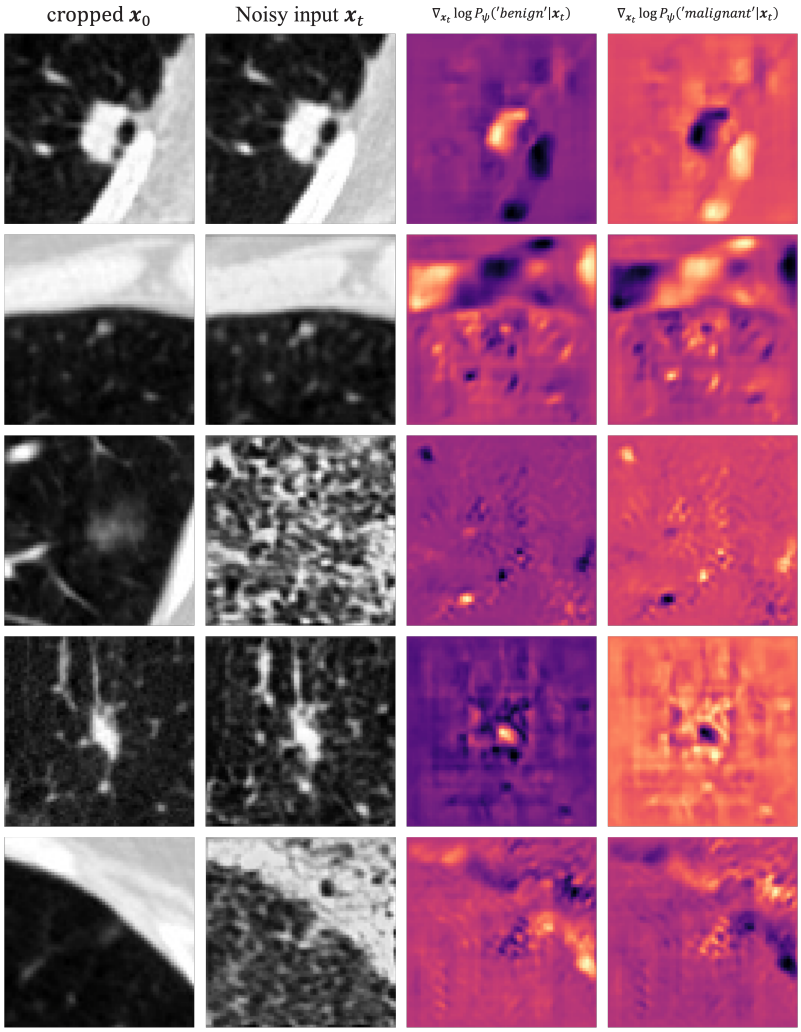


FIGURE 5.4: Visualization of the classifier's log derivatives. We use the nodule malignancy classifier trained on the cropped CT images centralized with the nodules. With the desired label to be "benign", the derivatives tend to be positive in the central region of nodules and negative in the boundary region, resulting in the shrinking of nodules under such guidance. Furthermore, in some cases of blurred or spiky nodules, benign labels lead the classifier to eliminate the fuzzy regions as much as possible. Additionally, even with a low signal-to-noise ratio, the classifier can still generate meaningful derivatives for central nodules. In the most extreme scenario of a significantly low signal-to-noise ratio, the classifier may produce minimal derivatives except for a few areas with significant error, which can be addressed through in-painting masks.

For the multi-classifier guidance, we observe that it is challenging to assess the individual influence of the other classifiers by visualization because these classifiers operate on more abstract and ambiguous concepts (spiculation, lobulation, and texture), which are hard to visually classify even for experienced radiologists. Consequently, we simply compare the results between the guiding with all available classifiers and the others, which is shown in Fig 5.6. We use the ground truth labels of the nodule features to guide the generation in order to evaluate the fidelity of the generated results. As illustrated in the

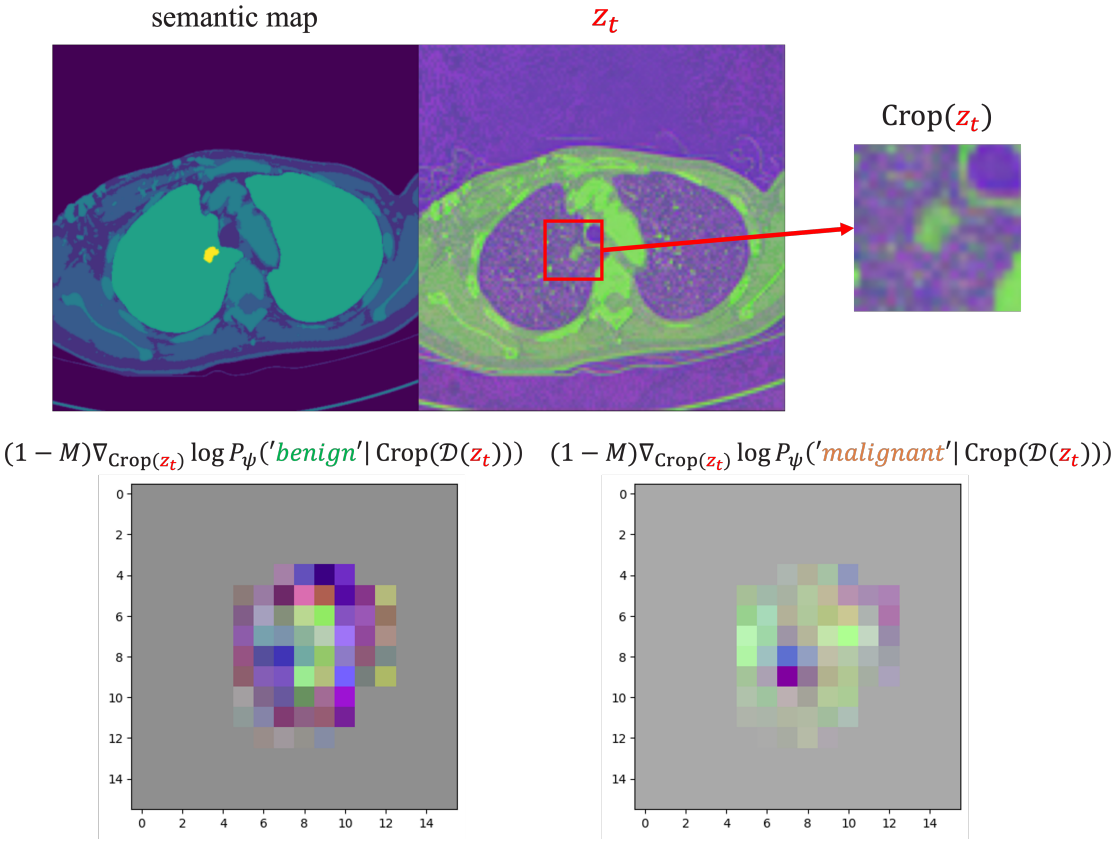


FIGURE 5.5: Visualization of the malignancy classifier derivatives with respect to the latent codes $\mathbf{z}_t$. As is discussed in Fig. 5.4, the label "benign" results in negative derivatives around the boundary of the nodule, while "malignant" leads to the extension. The derivatives are visualized using RGB color channels normalized to $[0, 1]$.

previous paragraphs, LSDM tends to generate high-fidelity results that strictly adhere to the semantic maps (the second column of Fig. 5.6). With the benefit of the classifiers, the generated results are able to contain additional details at the boundaries, leading to a more realistic synthesis.

Furthermore, we alter each nodule feature's value to evaluate the visual effects of all the features in Fig. 5.7. We observe minor visual effects with the features except for malignancy, indicating the ambiguity of visualizing the guidance of these features individually, which has been discussed in the last paragraph. For the malignancy results, we observe the diversity of the nodules under different classifier labels ("benign" or "malignant"), indicating the ability of the classifier guidance to generate diverse results under controllable features.

In the previous results, we use the classifier guidance scale $s = 1$. To further test the classifier guidance, we alter the malignancy classifier guidance with $s = 0.5, 1, 2, 4, 8, 16, 32$ under both binary labels "benign" and "malignant" in Fig. 5.8. The corresponding results elaborate that as s increases, the generated nodule enhances the characteristics of the guided label (benign or malignant). For instance, as s increases with the label "benign", the generated nodule becomes progressively lighter and divided into several parts with clearer boundaries. In the final two stages with $s = 16, 32$, the nodule even tends to blend

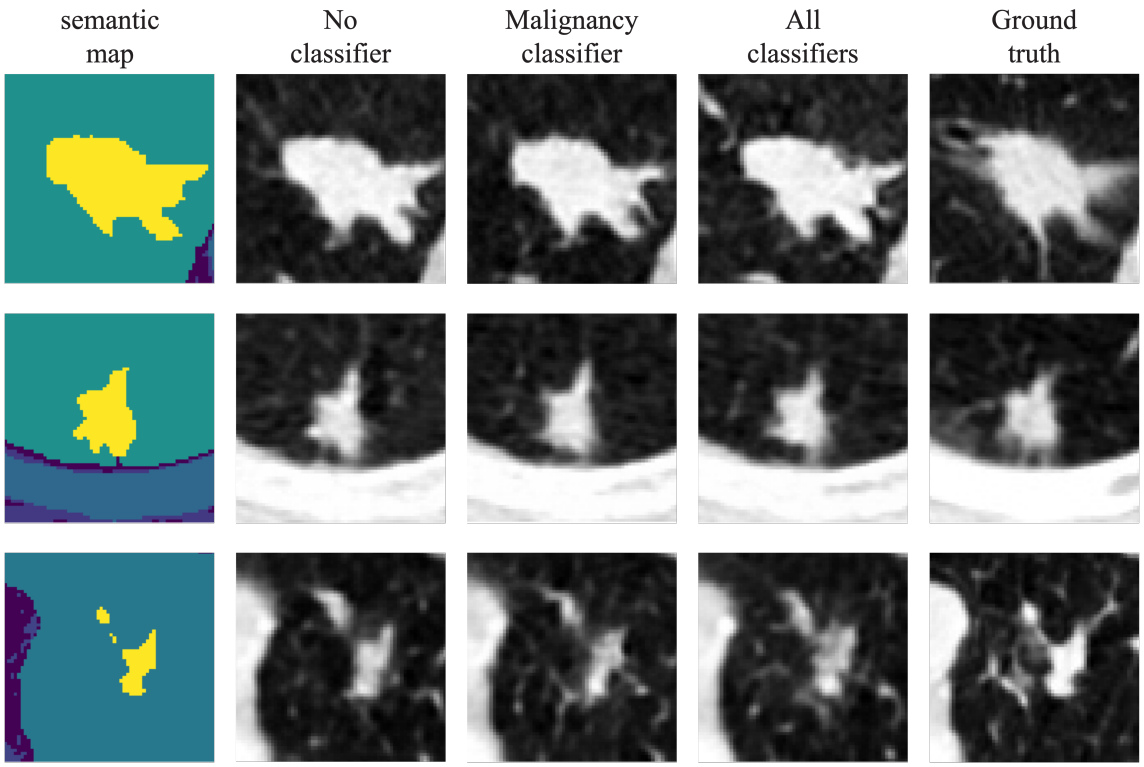


FIGURE 5.6: Qualitative results of the nodule synthesis with different guidance settings. For a large nodule semantic area (top row), LSDM generates high-fidelity results with respect to the input semantic map, yet have differences compared to the ground truth especially around the semantic boundary. With the guidance of the classifiers (third and fourth column), the generated results tend to express additional details at the boundaries, thereby approaching the realism of the ground truth more closely.

with the surroundings as if it is normal tissue as shown in Fig. 5.9.

To conclude, local in-painting multi-classifier guidance in LSDM presents effectiveness in visualization results. We show the reasons for using local in-painting style guidance and multi-classifier guidance. When generating with the ground truth labels of the classifiers, our method is able to synthesize high-fidelity results with robustness. When generating with alternative labels and guidance scales, our method achieves high diversity of nodule characteristics.

5.2 Quantitative Results

In this section, we present the quantitative results of our proposed method compared to the baselines using commonly used evaluation metrics. We conduct experiments on the split test dataset of LIDC-IDRI dataset illustrated in Chapter 4 with nodules, resulting in 770 total CT images. TO evaluate the quality of the generated images, we employ the following evaluation metrics:

1. Fréchet Inception Distance (FID): FID measures the similarity between the distribution of real and generated images based on features extracted from a pre-trained

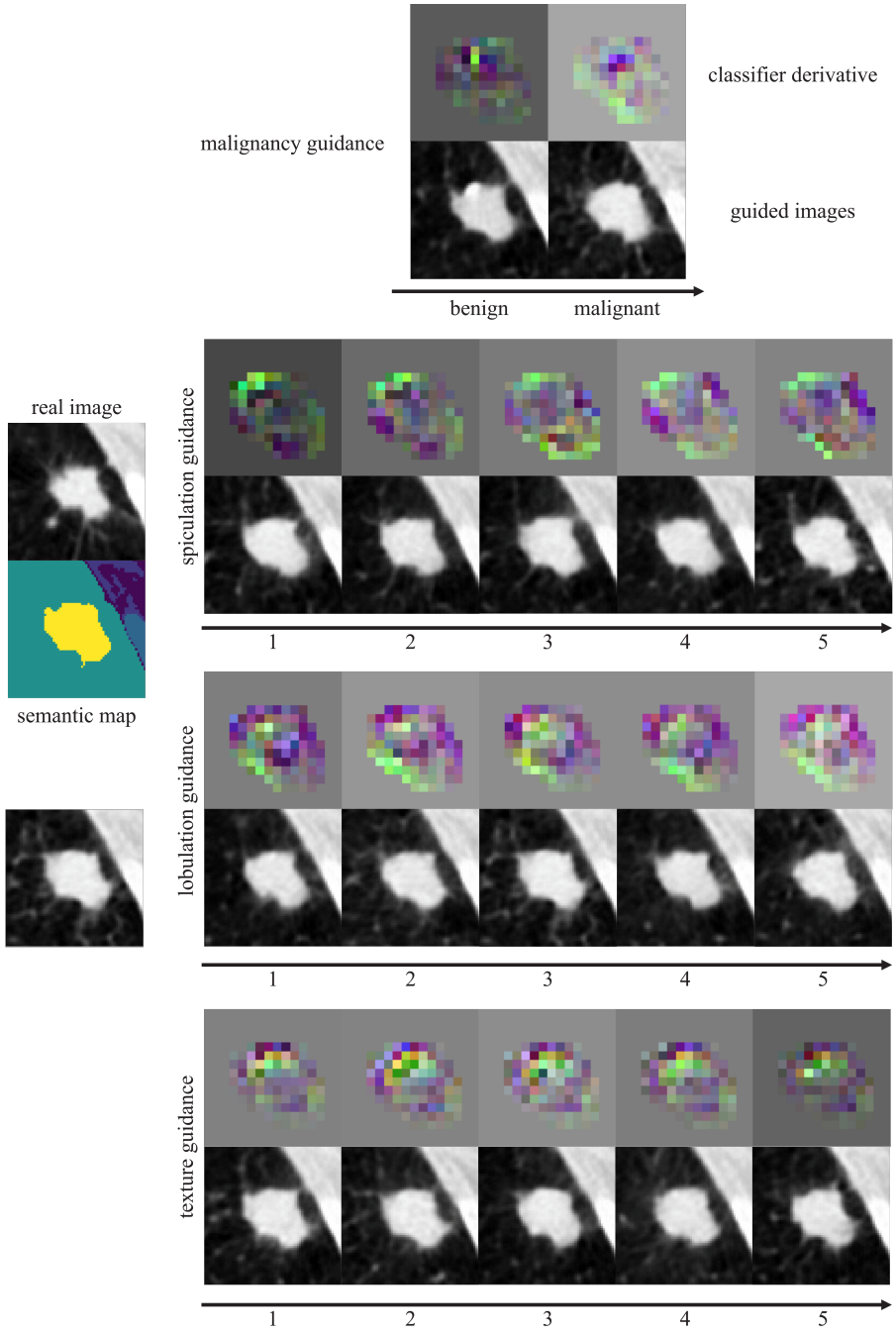


FIGURE 5.7: Generated results under different controllable key features. We can observe the diversity of the nodule synthesis under the binary classifier guidance of malignancy (top row). The other key features have a minor effect on the visualization results.

Inception-v3 network [10]. Lower FID scores indicate better similarity.

2. Structural Similarity Index (SSIM): SSIM measures the similarity in terms of luminance, contrast, and structural information between the generated and ground truth images [56]. Higher SSIM scores indicate better structural similarity.

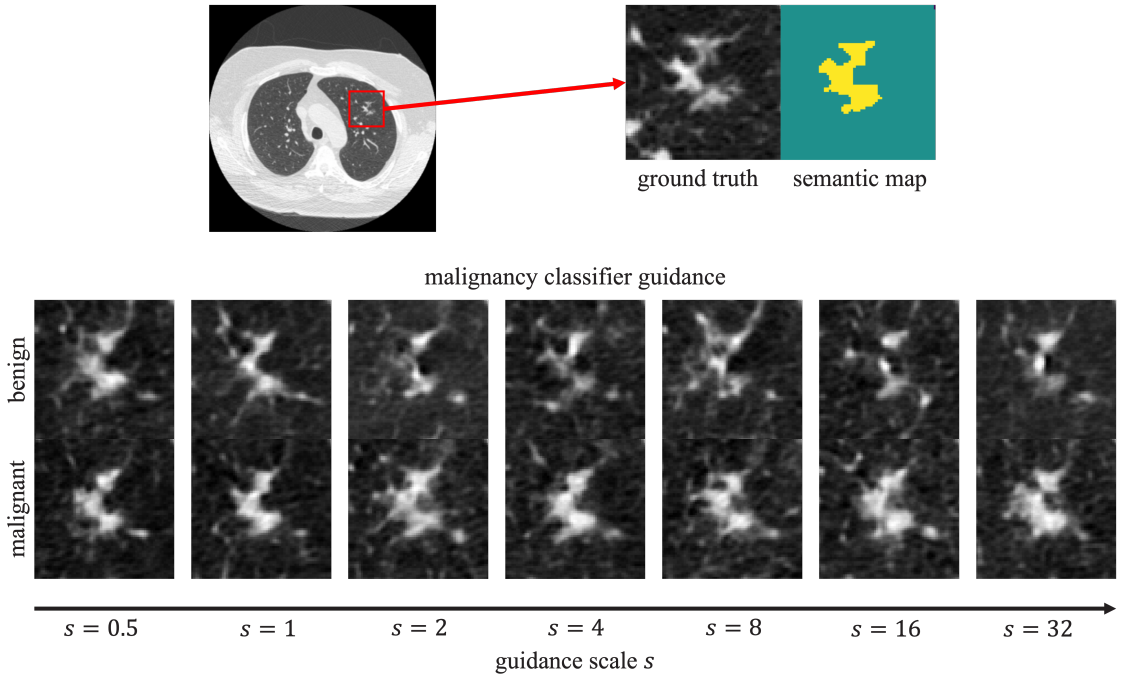


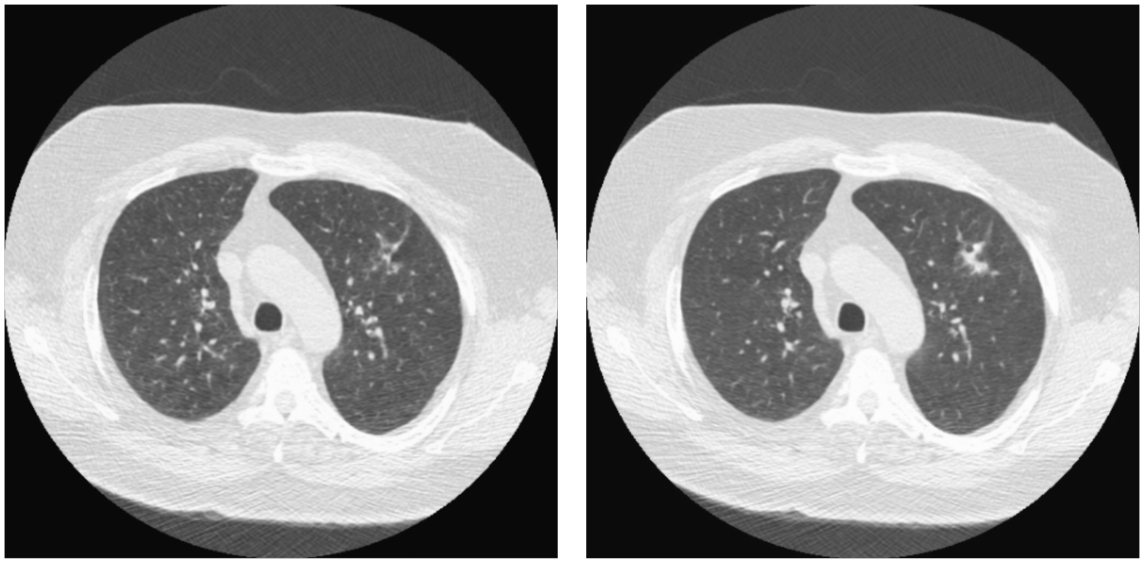
FIGURE 5.8: Ablation study of the classifier guidance scale s. During the reverse diffusion process, we alter the guidance scale s of the malignant classifier under both binary labels "benign" and "malignant" (top row and bottom row). The results indicate that as the guiding scale increases, the generated results tend to enhance the corresponding characteristics of the guided class (benign or malignant), such as the nodule area and boundary clarity. We can clearly observe that with the increase of the guidance scale s, the generated results under the benign label gradually transform into multiple small nodules, while the results under the malignant label connect to form larger regions and tend to expand outward.

3. Multi-Scale Structural Similarity Index (MS-SSIM): MS-SSIM is an extension of SSIM that considers structural similarities at multiple scales [57]. Higher MS-SSIM scores indicate better structural preservation.

4. Learned Perceptual Image Patch Similarity (LPIPS): LPIPS measures the perceptual similarity between the generated and real images using a deep neural network [62]. Lower LPIPS scores indicate better perceptual similarity.

We compare our method against the baselines including SPADE and LDM. We also evaluate an alternative without the classifier guidance as an ablation study. We present the quantitative evaluation results in Table 5.1. Our proposed methods (LSDM with/without classifier guidance) outperform the compared approaches across the evaluation metrics except for SSIM, which is concentrated on the low-level structural similarities where our method still performs the second best, demonstrating its effectiveness in generating high-quality CT images.

We further present the generation quality focusing on the nodule areas to illustrate the advantage of using our method when dealing with small semantic areas. We crop the generated images with the size of 64×64 centering at the nodules and calculate the FID score for all the methods. Our methods outperform the baseline methods but the classifier



benign, $s = 8$ malignant, $s = 8$

FIGURE 5.9: Comparison of the generative results with "benign" and "malignant" classifier guidance under the guidance scale $s = 8$. The generated result under the "benign" label tends to blend with the surroundings as if it is normal tissue. On the other hand, the result under the "malignant" label has a distinguishable affected area which is actually much larger than the ground truth.

guidance has a negative effect on the FID score performance, which is explainable since the guidance can be considered as a switch of the nodule data distribution to the ground truth, leading to inevitably decreasing the performances concentrating on the data distribution similarities.

In conclusion, our method exhibits superior performance compared to the baseline methods in terms of fidelity, structural preservation, perceptual similarity, and semantic consistency.

TABLE 5.1: Quantitative results of **generation fidelity and diversity comparison** with four metrics: Fréchet inception distance (FID), Structural Similarity Index (SSIM), multi-scale SSIM (MS-SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). For FID, lower value is seen as an indicator of a low distance between the real data distribution and the generated results. For SSIM and MS-SSIM, higher value means better structural accuracy. For LPIPS, lower value is considered to indicate better perceptual preservation quality.

Generation quality comparison

Metrics	SPADE	LDM	LSDM	LSDM with guidance
FID	29.86±1.73	26.05±1.89	25.71±1.66	**24.49±1.70**
SSIM	**0.42±0.04**	0.28±0.08	0.41±0.05	0.34±0.04
MS-SSIM	0.82±0.03	0.83±0.06	**0.86±0.07**	0.83±0.10
LPIPS	0.09±0.02	0.13±0.05	**0.08±0.04**	0.09±0.03

TABLE 5.2: Evaluation of the **fidelity of the generated results** concentrated on nodule areas. We use Fréchet inception distance (FID) to indicate the distribution distance between the generated nodules and the ground truth. For the names of the methods (top row), "LSDM+mali/spic/lobu/text" respectively denotes the LSDM model with malignancy/spiculation/lobulation/texture classifier guidance. We can observe that adding classifier guidance actually violets the fidelity of the generated nodules with respect to the ground truth, because we introduce extra guidances which are apparently different from the ground truth's distribution. Such effects need further study to accurately evaluate the guidance. All the results are calculated on the cropped images with the box size $64 * 64$ located at the nodules. We use the fixed guidance scale $s = 1$.

Fidelity of the generated nodules

Metrics	SPADE	LDM	LSDM	LSDM+mali	LSDM+spic	LSDM+lobu	LSDM+text
FID	49.83±3.41	47.62±2.95	**42.30±4.02**	61.77±8.36	65.59±6.58	67.09±9.94	68.07±7.59

Chapter 6

Diffusion Model VS Particle Filtering and Reinforcement Learning

In this chapter, providing a new perspective to understand the iterative generation methods such as diffusion models, normalizing flows, and neural ordinary differential equations (Neural ODEs) [11, 4], we present interesting connections between diffusion models and other two methods, namely particle filtering [6] and reinforcement learning [28, 58, 40, 22]. We first explain why diffusion models gathered much more attention than other iterative generation methods recently. Then we discuss the shared mathematical basis (Markov chain Monte Carlo) among diffusion models, particle filtering, and reinforcement learning. We further analyse the conceptual similarity among these methods, and finally provide some insights for the future research.

6.1 Why Diffusion Models?

In general, diffusion models are not the first series of generative methods with iterative sampling from the target data distributions, which have been developed with different intuitive initializations in variant research fields [6, 34, 4, 19]. A simple but significant question is: why diffusion models succeed?

In a word, the significant breakthrough of diffusion models in image, audio, and video generation mainly comes from the suprisingly simple training objective of diffusion models [11], which is crucial for efficient deep learning. The simple objective derives from the evidence lower bound (ELBO) of the target data:

$$\log p_\theta(\mathbf{x}_0) \geq \mathbb{E}_{\mathbf{x}_{0:T}\sim q(\mathbf{x}_{0:T})}\left[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = -\mathcal{L}_{ELBO} \tag{6.1}$$

Where $p_\theta(\mathbf{x}_{0:T})$ is the joint distribution of the whole trajectory in diffusion process, $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ is the distribution of intermediate points $\mathbf{x}_{1:T}$ given the initial state $\mathbf{x}_0$, which we consider them as observed latent variables. In variational Bayesian methods, the evidence lower bound of a variable x have the following equation:

$$\log p_\theta(x) = \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] + D_{KL}\big(q_\phi(z|x)\|p(z|x)\big) \tag{6.2}$$

Consequently, given a fixed distribution $p_\theta(x)$, maximizing ELBO is equivalent to minimizing the difference between the real conditional distribution $p(z|x)$ and the estimated

conditional distribution $q_\phi(z|x)$. The objective of diffusion models is then derived from $\mathcal{L}_{ELBO}$ in equation 6.1:

$$\begin{aligned}\mathcal{L}_{ELBO} &= \mathbb{E}_{\mathbf{x}_{0:T}\sim q(\mathbf{x}_{0:T})}\left[\log\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] \\ &= \mathbb{E}_q[\underbrace{D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)\|p_\theta(\mathbf{x}_T))}_{\mathcal{L}_T} + \underbrace{\sum_{t=2}^{T} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{\mathcal{L}_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{\mathcal{L}_0}]\end{aligned}$$

The KL-divergence terms $\mathcal{L}_{t-1}$ all compare two Gaussian distributions and therefore they are simplified into closed form:

$$\mathcal{L}_{t-1} = D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) = \frac{1}{2\sigma^2(t)}\|\boldsymbol{\mu}_\theta - \boldsymbol{\mu}\|^2 \tag{6.3}$$

Where $\boldsymbol{\mu}_\theta$ is the estimated mean of $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ and $\boldsymbol{\mu}$ is the mean of $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$. In DDPM [11], $\mathcal{L}_{t-1}$ is further simplified into the diffusion models' objective:

$$\mathcal{L}_{simple} = \|\boldsymbol{\epsilon}_\theta - \boldsymbol{\epsilon}\|^2 \tag{6.4}$$

Where minimizing the simple mean squared error (MSE) objective is equivalent to maximizing the ELBO of the target data $\mathbf{x}_0$ in equation 6.1, and subsequently minimizing the difference between the target distribution and the sample distribution. Following such simplification, diffusion models formulate a closed-form estimation of the target distribution using an MSE loss term which is supremely capable for the neural networks to train on, leading to emergence of the diffusion-based generative models.

6.2 Markov Chain Monte Carlo

A common property of the mentioned three methods is that they model a Markov process $p(x_{t+1}|x_t)$ to simulate the state x_t of a dynamical system. With Markov property, the probability of moving to the next state depends only on the current state without the previous states:

$$p(x_{t+1}|x_{0:t}) = p(x_{t+1}|x_t) \tag{6.5}$$

In diffusion models, the states are the intermediate generation $\mathbf{x}_t$, which are denoised iteratively using neural network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$. In particle filtering, the states are the particles $x_t^{(i)}$ sampled from the target distribution $p(x_t|x_{t-1})$. In reinforcement learning, the states are the environment states s_t of the agent, which are decided by the last actions and the corresponding states $p(s_t|s_{t-1}, a_{t-1})$.

This common hypothesis enables the three methods to efficiently estimate the desired distributions using Markov chain Monte Carlo (MCMC) method. For diffusion models, the perturbed data distribution $q_\sigma(\mathbf{x_0})$ with σ being the variance of the Gaussian noise is estimated using Langevin dynamics as the Markov chain Monte Carlo process under the score-based model settings:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \frac{\alpha}{2}\mathbf{s}_\theta(\mathbf{x}_{t-1}, \alpha) + \sqrt{\alpha}\boldsymbol{\epsilon}_t \tag{6.6}$$

Where $\mathbf{s}_\theta(\mathbf{x}_{t-1}, \alpha) \approx \nabla_{\mathbf{x}_0}\log q_\alpha(\mathbf{x}_0)$ is the estimated score function of perturbed data distribution $q_\alpha(\mathbf{x}_0)$. Following Langevin dynamics, MCMC sample the intermediate noisy

images from $q_\alpha(\mathbf{x}_0)$ with step size α, enabling sampling from untraceable complicated data distributions.

For particle filtering (PF), the particles used for estimating the state distribution are obtained through sequential importance sampling:

$$x_t^{(i)} \sim p(x_t|x_{t-1}), \quad w_t^{(i)} = p(y_t|x_t)w_{t-1}^{(i)} \tag{6.7}$$

Where $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ are the particles and their corresponding weights for estimating the distribution, and y_t is the observed variable of the dynamical system. This MCMC sampling method allows PF to approximate the non-linear, non-Gaussian distributions of the state dynamics.

For reinforcement learning (RL), the purpose of an agent is to execute an action under a certain policy π in a certain state s_t and eventually maximize the total reward G in an interactive environment. With the premise of a Markov process in the series of states $p(s_t|s_{t-1}, ..., s_0) = p(s_t|s_{t-1})$, RL formulate a Markov decision process with an additional variable action a_t in the series:

$$p(s_t|s_{t-1}, a_{t-1}, ..., s_0, a_0) = p(s_t|s_{t-1}, a_{t-1}) \tag{6.8}$$

Where the action a_t is manipulated by the agent in the environment, given the current state s_t. After executing the action a_t, the environment return an immediate reward r_{t+1} and transfer the state into s_{t+1}, establishing a trajectory of the Markov decision process:

$$\tau = s_0, a_0, r_1, s_1, a_1, r_2, ..., s_{T-1}, a_{T-1}, r_T, s_T \tag{6.9}$$

The total return $G(\tau) = \sum_{t=0}^{T-1} \gamma^t r_{t+1}$ consider every immediate reward r_t in the trajectory with a discount factor $\gamma \in [0, 1]$. In order to explicitly evaluate the significance of a state, RL introduces state value function $V^\pi(s_t)$ to guide the policy π of the agent. The value function can be defined as:

$$V^\pi(s_t) = \mathbb{E}_{\tau_{t:T} \sim p(\tau_{t:T})}[G(\tau_{t:T})] \tag{6.10}$$

The value function provides the estimation of the total reward started from state s_t given the subsequent trajectory distribution $p(\tau_{t:T})$, which is obtained by MCMC sampling method:

$$V^\pi(s_t) \approx \frac{1}{N} \sum_{n=1}^{N} G(\tau_{s_0=s_t}^{(n)}) \tag{6.11}$$

In particular, given the initial state $s_0 = s_t$, the agent uses the current policy $\pi = \pi_\theta$ to sample a batch of trajectories in a Monte Carlo manner, on which the average of total reward is calculated to estimate the value function of the initial state $V^\pi(s_t)$.

To conclude, MCMC sampling plays a significant role in all three methods, focusing on the estimation of complex distributions, which are perturbed data distribution $\mathbf{x}_t \sim q_\alpha(\mathbf{x}_0)$ in diffusion models, state distribution $x_t^{(i)} \sim p(x_t|x_{t-1})$ in particle filtering, and trajectory distribution $\tau_{t:T} \sim p(\tau_{t:T})$ in reinforcement learning. Fig. 6.1 illustrates the series structure of the three methods and the MCMC sampling targets of them.

6.3 Predict and Correct

As is discussed in the last section, there exist pivotal connections among the three various methods. In this section, we additionally provide a structural perspective to re-exam the

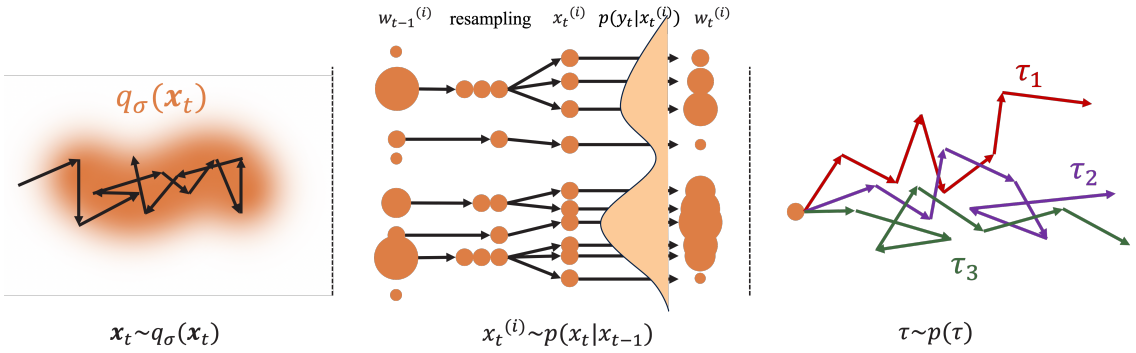


FIGURE 6.1: The similar Markov chain Monte Carlo (MCMC) processes in all three methods. Left panel: Langevin dynamics of the diffusion models under score-based model settings. The blurred orange area represents the perturbed distribution of the data $q_\sigma(\mathbf{x}_0)$. The series of black arrows indicate the Langevin Monte Carlo sampling process, under which the samples will adhere to the distribution $\mathbf{x}_t \sim q_\sigma(\mathbf{x}_0)$. Middle panel: particle filtering (PF) iteration rules. With the benefit of importance sampling and resampling trick, the sampled particles of PF follow the MCMC procedure while updating the weights, estimating the distribution $x_t^{(i)} \sim p(x_t|x_{t-1})$. Right panel: MCMC sampling of different trajectories (different colors of the arrows) in reinforcement learning (RL). RL utilizes the estimation of the trajectory distribution to maximize the total trajectory reward $G(\tau)$. This estimation is sampled using MCMC.

interplay of the methods, presenting the strong connection in terms of the fundamental principles of the methods.

Specifically, diffusion models, PF, and RL all adopt a similar approach within their high-level framework, employing a two-part alternating optimization process. In this process, one step plays a dominant role, primarily responsible for the direct evolution of the states ($\mathbf{x}_t/x_t/s_t$), while the other step focuses on correcting and refining the predicted outcomes from the first step. To simplify the nomenclature, we use "predictor" to elucidate the first step, and "corrector" for the other step in all three methods following the name in [46].

Before the comparison among these methods, it is worthy to mention that either the diffusion-based models are named under score-based models or diffusion models with the corresponding algorithmic pipelines, they can all be considered as learning a non-homogeneous differential equation with time embedding in 2.3 and evaluating the trained neural network using a certain ODE solver with or without noises [46, 16]. Consequently, all the diffusion-based methods share the same high-level framework, which is significant to the subsequent comparison.

In general, the predictor serves as the backbone of the algorithm, transforming the state $\mathbf{x}_t/x_t/s_t$ from the previous iteration into the state at the next time step $\mathbf{x}_{t+1}/x_{t+1}/s_{t+1}$. In diffusion models, the predictor updates the intermediate results using $\mathbf{s}_\theta(\mathbf{x}_t, t)$ following PC sampling in score-based models [46]. In particle filtering, the predictor generates the particles of the next time step. In reinforcement learning, the predictor utilizes the value function $V^{(\pi)}$ to determine the action a_t in state s_t.

The corrector performs refinement of the predictor's step, improving the accuracy of the predictor's outcome and aligning the results with the temporal evolution of the system

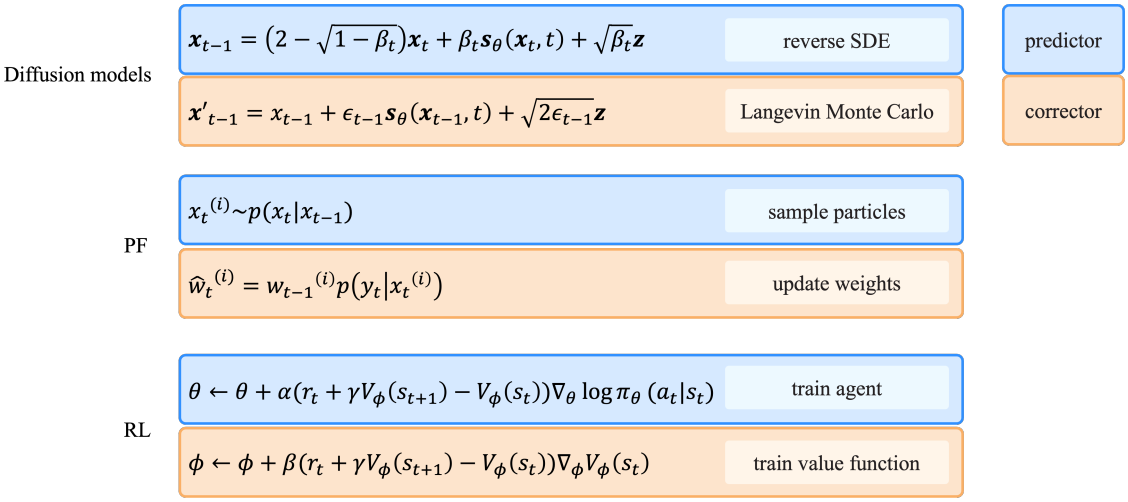


FIGURE 6.2: Illustration of the similar high-level structures of the three methods. In general, all three methods consist of a "predictor" step (blue panels) to evolute the corresponding dynamical systems with the auxiliary of a "corrector" step (orange panels) to update the estimation and improve the accuracy of the states.

dynamics. In diffusion models, the corrector can be described as the Langevin Monte Carlo procedure in PC sampling. In particle filtering, the corrector updates the weights of the particles to align with the distribution in the next time step. In reinforcement learning, the corrector can be considered as optimizing the parameters ϕ value function $V_\phi^\pi(s_t)$ to better evaluate the importance of the state s_t. Fig. 6.2 emphasizes and compares the two steps in each method with pseudo codes, illustrating the relationship between the three iterative algorithms.

There are other minor similarities among the three, such as the same log-likelihood formulations of the Markov processes as is shown in Fig. 6.3, and the coincident importance sampling trick in both particle filtering and Proximal Policy Optimization (PPO) [40], which is one of the breakthroughs in RL achieving stable convergences in various complex tasks.

The findings of similar structures in all three methods highlight the common sense in iterative updating methods for dynamical systems and are actually novel lines of thought considering that people often invent similar algorithms from different perspectives. It is crucial to emphasize the distinctions between these algorithms and why one algorithm may be superior to others in some tasks. In addition, by comprehensively understanding these methods in a contrastive approach, we can present research proposals such as: Can we improve the diffusion models by inserting candidate mechanism as is in particle filtering? Can we treat the guidance of the diffusion models as the value function $V^\pi(s_t)$ or Q-function $Q^\pi(s_t, a_t)$ in RL?

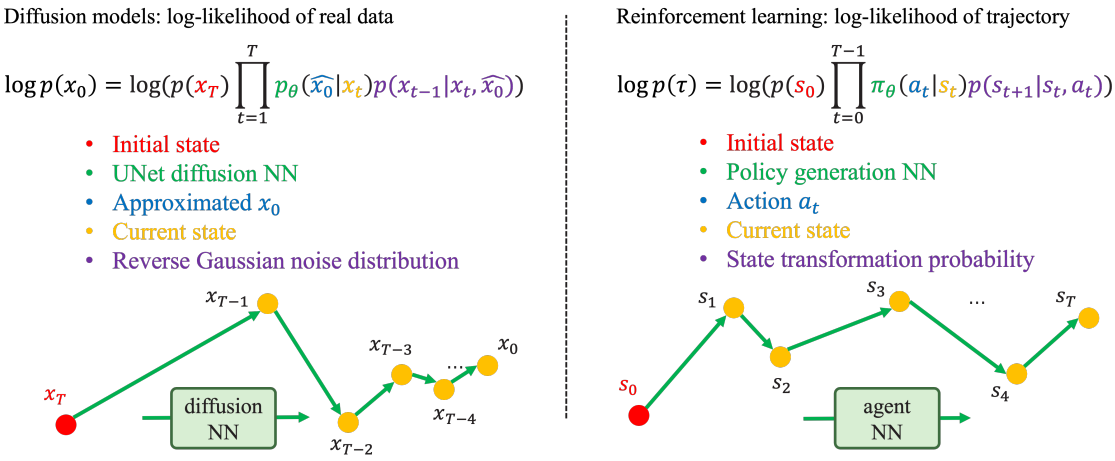


FIGURE 6.3: The same log-likelihood formulations of the Markov processes in diffusion models and reinforcement learning (RL). The green arrows indicate the evaluation of the neural network representing the dynamical change given a certain state observation $\mathbf{x}_t$ or s_t. Notice the different lengths of the arrows in diffusion models and RL.

Chapter 7

Discussions and Conclusions

In this chapter, we provide comprehensive discussions and conclusions of our method. We reflect on the choices of hyper-parameters, address the limitations of our method, and draw meaningful conclusions based on the findings of our study. This work contributes to the growing body of literature on medical image synthesis, where the use of diffusion models has attracted increasing attention, showing promising potential for improving the accuracy of the synthesis and solving inverse problems in medical imaging [45].

7.1 Hyper-parameter Choices

During the implementation of our method, several key hyper-parameters needed to be discussed to ensure optimal performance, one of which has been mentioned in Chapter 5 (guidance scale s). In this section, we explain the choices of these hyper-parameters empirically.

In Chapter 5, we choose the default guidance scale to be $s = 1.0$, because only under such choice can we achieve a better FID score illustrated in Table 5.1. We have discussed the negative effect of the classifier guidance in Section 5.2, which shows deceased performance with classifier guidance except for the results in Table 5.1. Furthermore, [5] also suggests using the guidance scale $s = 1.0$ for ideally trade-off diversity for fidelity.

The down-sampling factor, denoted by f, plays a crucial role in trading-off reconstruction quality for sample efficiency. A lower down-sampling factor f, which indicates a higher dimensional latent space, allows for increased expressiveness and reconstruction ability, potentially enabling the model to generate more accurate images. However, an excessively low down-sampling factor leads to expensive computation during training and inference [36]. At the early stage of our research, We empirically explored different down-sampling factors $f = 4$, $f = 8$, and $f = 16$, which correspondingly lead the size of the latent space to be $128 \times 128 \times 3$, $64 \times 64 \times 3$, and $32 \times 32 \times 3$. We roughly evaluated the generation quality at the early stage and decided to use $f = 4$, which has almost no perceptual information loss during the reconstruction while still possesses of efficient sampling.

For other hyper-parameters such as the schedule of the diffusion process α_t, we follow the implementation of LDM [36].

7.2 Limitations

Despite the promising results of our method in Chapter 5, several limitations need to be acknowledged, some of which arise from the fundamental nature of the task or the underlying assumptions of the model, while others come from the limitation of time.

First, the effectiveness of our method significantly relies on the availability of accurate and reliable ground truth segmentations. Inaccurate segmentation maps may lead to biased results and affect the performance of our model.

Second, guiding the generation process with several additional classifiers has unknown effects on diagnostic accuracy. While we show the advantages of the classifier guidance, it is challenging to assess the diagnostic accuracy of the synthesized image especially when we explicitly control the generated nodules with extra features.

Third, we observe possibly contradictory results about the FID scores in Table 5.1 and Table 5.2, where LSDM with classifier guidance outperforms others under the full-size generation while has an apparent disadvantage under the nodule generation. A possible explanation is that during the FID score evaluation, the Inception V3 neural network encodes the inputs with different levels of perceptual scales under full-size and local cropped settings, resulting in different encoding concentrations on scales.

Lastly, while we have primarily focused on lung CT scan analysis, the applicability of our method to other medical imaging modalities and pathology detection tasks remains an open research question. Exploring and adapting our approach to different medical imaging domains will pave the way for more widespread adoption and validation.

7.3 Conclusions

In conclusion, our research presents a novel approach combining newly-designed diffusion models with local in-painting style multi-classifier guidance for precise and accurate image generation, especially for small semantic areas. Through the integration of semantic image synthesis and classifier guidance, we have demonstrated the potential to explicitly control the characteristics of the lung nodules. Our method leverages the strengths of diffusion models in explicitly and resolvably guiding the generation and SPADE layers preserving spatial information to enhance the fidelity and diversity of the generated lung CT images. The results obtained from our experiments on the LIDC-IDRI lung CT dataset show promising performance in terms of visual quality and quantitative evaluation metrics.

By leveraging the explicit and resolvable guidance of lung nodule generation, we can improve semantic image synthesis for small areas, leading to better synthetic medical dataset and potentially training of the new radiologists. However, it is crucial to address the limitations and challenges discussed in this chapter to ensure the practical applicability of our method in real-world clinical settings.

Chapter 8

Future Work

In this chapter, we discuss potential future directions and extensions of our research on diffusion models in medical image synthesis. Building upon the insights gained from our current study, we outline three areas that hold significant promise for further exploration: software deployment considerations, classifier-free guidance [12], and integration with optimal transport theory [53]. Software deployment will be the first extension of our work, with the goal to develop a user-friendly online web page for generating synthetic lung CT images with tunable control parameters. By delving into the aspects of classifier-free guidance and optimal transport, we aim to advance the performance and capability of our current method.

8.1 Software Deployment

As our method progresses towards practical applications, it is crucial to address software deployment considerations. The aim of developing such user-friendly software is to efficiently generate controllable lung CT images and possibly train new radiologists based on the labeled synthetic data. In addition, the implementation of our model should be made accessible to the wider medical imaging community.

Developing efficient and scalable software solutions, along with intuitive user interfaces, would facilitate the adoption and replication of our approach across different institutions and research settings. Additionally, comprehensive documentation should be provided to promote collaboration within the scientific community. A representative example would be the widely popular Stable Diffusion [36] webpage that gained significant attention in the field of image generation last year, sparking extensive discussions. Fig. 8.1 illustrates the basic layout of the webpage, which contains a sketch board / preview box for drawing the semantic map, a file uploading panel to utilize the existing semantic maps, a primary generated image display window, and a control panel of tunable hyper-parameters such as the nodule features, the number of the generated batch, and the resolution of the generated CT image.

By providing an interface under such principles of the webpage design, trainees can manipulate various parameters, such as nodule size, shape, and location, to simulate different clinical scenarios and pathologies. This allows the platform to generate considerable synthetic CT images and can serve as a valuable tool for radiologists to enhance their diagnostic skills, improve their understanding of pulmonary nodules, and gain exposure to rare or challenging cases that are difficult to encounter in real-world clinical practice.

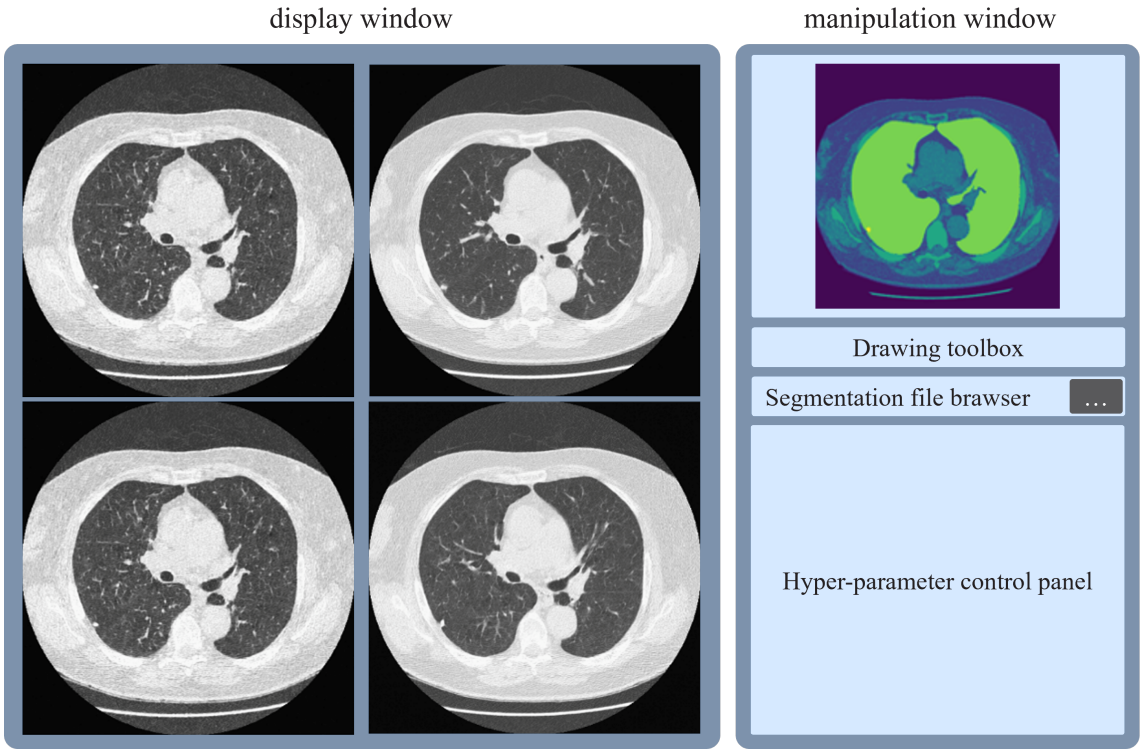


FIGURE 8.1: User interface of the lung CT image synthesis webpage.

8.2 Classifier-free Guidance

While our current approach leverages classifier guidance to enhance the fidelity of generated lung CT images, especially in nodule areas, an interesting avenue for future work lies in exploring classifier-free guidance techniques with explicit controllable nodule features. By eliminating the reliance on pre-trained classifiers, classifier-free guidance has shown advantages in multiple tasks. By developing a suitable classifier-free guidance pipeline, we can potentially overcome limitations associated with classifier biases and additional hyper-parameters such as crop size. we can also investigate the use of unsupervised learning methods, such as contrastive learning methods, to guide the diffusion model in capturing salient features and spatial dependencies within lung CT scans. This approach may enable more robust and flexible image synthesis, particularly in medical imaging scenarios where annotated training data is scarce or unavailable.

8.3 Optimal Transport

Another promising direction for future work is the integration of optimal transport (OT) theories into diffusion models in medical imaging synthesis. OT [53, 1, 38] provides a powerful framework for measuring the dissimilarity between probability distributions. By definition in OT, the optimal transport cost between two distributions μ and ν in the same space is:

$$\text{Cost}(\mu, \nu) = \inf_{T_{\#}\mu=\nu} \int_{\mathcal{X}} c(x, T(x)) d\mu(x) \tag{8.1}$$

Where $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ defines the cost of transportation between two points $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, which for instance can be the quadratic cost $c(x, y) = \|x - y\|^2$. $T : \mathcal{X} \to \mathcal{Y}$ is the transport map $T(x) \in \mathcal{Y}$ pushing μ to ν. As an intuitive yet non-rigorous example, there exists an OT map $T(x) = y$ between two Gaussian distributions $x \sim \mathcal{N}(0, 1)$ and $y \sim \mathcal{N}(10, 10)$, which minimize the total transportation cost $c(x, y)$. This OT map can be considered as the optimal projection between the samples of the two distributions.

OT theory has been successfully applied in image synthesis and domain adaptation tasks [1, 21, 7]. By incorporating OT principles into our method, we can further enhance the fidelity of the generated lung CT images and improve the alignment between the semantic maps and the synthetic images.

For example, we can calculate the OT map between the pure Gaussian noise and the real CT scans and formulate OT trajectories during the diffusion process, which will possibly speed up the iterative generation while preserving image quality. We can also possess an OT map between the nodule semantic maps and the real nodule images, leveraging the minimal transport cost to implement direct style transfer between the semantic maps and the real images. This integration could potentially lead to more realistic and accurate lung nodule representations, benefiting subsequent analysis and data augmentation.

Bibliography

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. doi:10.48550/ARXIV.1701.07875.

[2] Mehdi Astaraki, Yousuf Zakko, Iuliana Toma Dasu, Örjan Smedby, and Chunliang Wang. Benign-malignant pulmonary nodule classification in low-dose CT with convolutional features. *Physica Medica*, 83:146–153, mar 2021. doi:10.1016/j.ejmp.2021.03.013.

[3] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S. Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise, 2022. doi:10.48550/ARXIV.2208.09392.

[4] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6572–6583, Red Hook, NY, USA, 2018. Curran Associates Inc.

[5] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. doi:10.48550/ARXIV.2105.05233.

[6] P.M. Djuric, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, and J. Miguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, sep 2003. doi:10.1109/msp.2003.1236770.

[7] Jiaojiao Fan, Shu Liu, Shaojun Ma, Haomin Zhou, and Yongxin Chen. Neural monge map estimation and its applications, 2021. doi:10.48550/ARXIV.2106.03812.

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. doi:10.48550/ARXIV.1406.2661.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016. doi:10.1109/cvpr.2016.90.

[10] 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. doi:10.48550/ARXIV.1706.08500.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. doi:10.48550/ARXIV.2006.11239.

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. doi:10.48550/ARXIV.2207.12598.

[13] Xun Huang, Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu. Multimodal conditional image synthesis with product-of-experts gans, 2021. doi:10.48550/ARXIV.2112.05130.

[14] Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, dec 2020. doi:10.1038/s41592-020-01008-z.

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017. doi:10.1109/cvpr.2017.632.

[16] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022. doi:10.48550/ARXIV.2206.00364.

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018. doi:10.48550/ARXIV.1812.04948.

[18] Amirhossein Kazerouni, Ehsan Khodapanah Aghdam, Moein Heidari, Reza Azad, Mohsen Fayyaz, Ilker Hacihaliloglu, and Dorit Merhof. Diffusion models in medical imaging: A comprehensive survey. *Medical Image Analysis*, page 102846, may 2023. doi:10.1016/j.media.2023.102846.

[19] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series, 2020. doi:10.48550/ARXIV.2005.08926.

[20] Alexander Korotin, Vage Egiazarian, Arip Asadulaev, Alexander Safin, and Evgeny Burnaev. Wasserstein-2 generative networks, 2019. doi:10.48550/ARXIV.1909.13082.

[21] Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. 2022. doi:10.48550/ARXIV.2201.12220.

[22] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2015. doi:10.48550/ARXIV.1509.02971.

[23] Menglu Liu, Junyu Dong, Xinghui Dong, Hui Yu, and Lin Qi. Segmentation of lung nodule in CT images based on mask r-CNN. In *2018 9th International Conference on Awareness Science and Technology (iCAST)*. IEEE, sep 2018. doi:10.1109/icawst.2018.8517248.

[24] K Scott Mader. The lung image database consortium image collection (lidc-idri), 2021. URL: https://dx.doi.org/10.21227/zce3-jp96, doi:10.21227/zce3-jp96.

[25] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, sep 2018. doi:10.21105/joss.00861.

[26] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik P. Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models, 2022. doi:10.48550/ARXIV.2210.03142.

[27] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. doi:10.48550/ARXIV.1411.1784.

[28] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. 2016. doi:10.48550/ARXIV.1602.01783.

[29] Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12):1482–1492, dec 2019. doi:10.1038/s41587-019-0336-3.

[30] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. doi:10.48550/ARXIV.2102.09672.

[31] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans, 2016. doi:10.48550/ARXIV.1610.09585.

[32] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. 2019. doi:10.48550/ARXIV.1903.07291.

[33] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015. doi:10.48550/ARXIV.1511.06434.

[34] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2015. doi:10.48550/ARXIV.1505.05770.

[35] Michael Roberts, Derek Driggs, Matthew Thorpe, Julian Gilbey, Michael Yeung, Stephan Ursprung, Angelica I. Aviles-Rivero, Christian Etmann, Cathal McCague, Lucian Beer, Jonathan R. Weir-McCall, Zhongzhao Teng, Effrossyni Gkrania-Klotsas, Alessandro Ruggiero, Anna Korhonen, Emily Jefferson, Emmanuel Ako, Georg Langs, Ghassem Gozaliasl, Guang Yang, Helmut Prosch, Jacobus Preller, Jan Stanczuk, Jing Tang, Johannes Hofmanninger, Judith Babar, Lorena Escudero Sánchez, Muhunthan Thillai, Paula Martin Gonzalez, Philip Teare, Xiaoxiang Zhu, Mishal Patel, Conor Cafolla, Hojjat Azadbakht, Joseph Jacob, Josh Lowe, Kang Zhang, Kyle Bradley, Marcel Wassin, Markus Holzer, Kangyu Ji, Maria Delgado Ortet, Tao Ai, Nicholas Walton, Pietro Lio, Samuel Stranks, Tolou Shadbahr, Weizhe Lin, Yunfei Zha, Zhangming Niu, James H. F. Rudd, Evis Sala, and Carola-Bibiane Schönlieb and. Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans. *Nature Machine Intelligence*, 3(3):199–217, mar 2021. doi:10.1038/s42256-021-00307-0.

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. doi:10.48550/ARXIV.2112.10752.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, pages 234–241. Springer International Publishing, 2015. doi:10.1007/978-3-319-24574-4_28.

[38] Litu Rout, Alexander Korotin, and Evgeny Burnaev. Generative modeling with optimal transport maps, 2021. doi:10.48550/ARXIV.2110.02999.

[39] Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation, 2022. doi:10.48550/ARXIV.2212.09478.

[40] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. doi:10.48550/ARXIV.1707.06347.

[41] August DuMont Schütte, Jürgen Hetzel, Sergios Gatidis, Tobias Hepp, Benedikt Dietz, Stefan Bauer, and Patrick Schwab. Overcoming barriers to data sharing with medical image generation: a comprehensive evaluation. *npj Digital Medicine*, 4(1), sep 2021. doi:10.1038/s41746-021-00507-3.

[42] Mohammad Shehab, Laith Abualigah, Qusai Shambour, Muhannad A. Abu-Hashem, Mohd Khaled Yousef Shambour, Ahmed Izzat Alsalibi, and Amir H. Gandomi. Machine learning in medical applications: A review of state-of-the-art methods. *Computers in Biology and Medicine*, 145:105458, jun 2022. doi:10.1016/j.compbiomed.2022.105458.

[43] Youssef Skandarani, Pierre-Marc Jodoin, and Alain Lalande. Gans for medical image synthesis: An empirical study, 2021. doi:10.48550/ARXIV.2105.05318.

[44] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL: https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2020. doi:10.48550/ARXIV.2010.02502.

[46] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models, 2021. doi:10.48550/ARXIV.2111.08005.

[47] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2020. doi:10.48550/ARXIV.2011.13456.

[48] Zhentao Tan, Dongdong Chen, Qi Chu, Menglei Chai, Jing Liao, Mingming He, Lu Yuan, Gang Hua, and Nenghai Yu. Efficient semantic image synthesis via class-adaptive normalization, 2020. doi:10.48550/ARXIV.2012.04644.

[49] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, dec 2000. doi:10.1126/science.290.5500.2319.

[50] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

[51] Gaël Varoquaux and Veronika Cheplygina. Machine learning for medical imaging: methodological failures and recommendations for the future. *npj Digital Medicine*, 5(1), apr 2022. doi:10.1038/s41746-022-00592-y.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. doi:10.48550/ARXIV.1706.03762.

[53] Cédric Villani. *Optimal Transport*. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-540-71050-9.

[54] Weilun Wang, Jianmin Bao, Wengang Zhou, Dongdong Chen, Dong Chen, Lu Yuan, and Houqiang Li. Semantic image synthesis via diffusion models, 2022. doi:10.48550/ARXIV.2207.00050.

[55] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior, 2021. doi:10.48550/ARXIV.2101.04061.

[56] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, apr 2004. doi:10.1109/tip.2003.819861.

[57] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems &amp Computers, 2003*. IEEE. doi:10.1109/acssc.2003.1292216.

[58] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, may 1992. doi:10.1007/bf00992696.

[59] S. Wu. Generation of lung ct images using semantic layouts. Technical report, 2021 [Unpublished].

[60] Yan Yang, Jun Yu, Jian Zhang, Weidong Han, Hanliang Jiang, and Qingming Huang. Joint embedding of deep visual and semantic features for medical image report generation. *IEEE Transactions on Multimedia*, 25:167–178, 2023. doi:10.1109/tmm.2021.3122542.

[61] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, 2016. arXiv:1506.03365.

[62] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. doi:10.48550/ARXIV.1801.03924.

[63] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2017. doi:10.1109/iccv.2017.244.

[64] Xinrui Zu and Qian Tao. SpaceMAP: Visualizing high-dimensional data by space expansion. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference*

on Machine Learning, volume 162 of *Proceedings of Machine Learning Research*, pages 27707–27723. PMLR, 17–23 Jul 2022. URL: https://proceedings.mlr.press/v162/zu22a.html.