

3D MedDiffusion: A 3D Medical Latent Diffusion Model for Controllable and High-Quality Medical Image Generation

Haoshen Wang^{ID}, Zhentao Liu^{ID}, Kaicong Sun^{ID}, Xiaodong Wang,
Dinggang Shen^{ID}, *Fellow, IEEE*, and Zhiming Cui^{ID}

Abstract—The generation of medical images presents significant challenges due to their high-resolution and three-dimensional nature. Existing methods often yield sub-optimal performance in generating high-quality 3D medical images, and there is currently no universal generative framework for medical imaging. In this paper, we introduce a 3D Medical Latent Diffusion (3D MedDiffusion) model for controllable, high-quality 3D medical image generation. 3D MedDiffusion incorporates a novel, highly efficient Patch-Volume Autoencoder that compresses medical images into latent space through patch-wise encoding and recovers back into image space through volume-wise decoding. Additionally, we design a new noise estimator to capture both local details and global structural information during diffusion denoising process. 3D MedDiffusion can generate fine-detailed, high-resolution images (up to $512 \times 512 \times 512$) and effectively adapt to various downstream tasks as it is trained on large-scale datasets covering CT and MRI modalities and different anatomical regions (from head to leg). Experimental results demonstrate that 3D MedDiffusion surpasses state-of-the-art methods in generative quality and exhibits strong generalizability across tasks such as sparse-view CT reconstruction, fast MRI reconstruction, and data augmentation for segmentation and classification. Source code and checkpoints are available at <https://github.com/ShanghaiTech-IMPACT/3D-MedDiffusion>

Received 26 December 2024; revised 18 March 2025 and 21 May 2025; accepted 27 June 2025. Date of publication 2 July 2025; date of current version 2 December 2025. This work was supported in part by NSFC under Grant 6230012077, in part by Shanghai Municipal Central Guided Local Science and Technology Development Fund under Project YDZX20233100001001 and in part by the High Performance Computing (HPC) Platform of ShanghaiTech University. Recommended by Associate Editor O. Colliot. (*Corresponding author: Zhiming Cui.*)

Haoshen Wang, Zhentao Liu, Kaicong Sun, and Zhiming Cui are with the School of Biomedical Engineering and the State Key Laboratory of Advanced Medical Materials and Devices, ShanghaiTech University, Shanghai 201210, China (e-mail: wanghsh2022@shanghaitech.edu.cn; liuzt2022@shanghaitech.edu.cn; sunc@shanghaitech.edu.cn; cuizhm@shanghaitech.edu.cn).

Xiaodong Wang is with United Imaging Healthcare Company Ltd., Shanghai 201807, China (e-mail: xiaodong.wang@united-imaging.com).

Dinggang Shen is with the School of Biomedical Engineering and the State Key Laboratory of Advanced Medical Materials and Devices, ShanghaiTech University, Shanghai 201210, China, also with Shanghai United Imaging Intelligence Company Ltd., Shanghai 200230, China, and also with Shanghai Clinical Research and Trial Center, Shanghai 201210, China (e-mail: dgshen@shanghaitech.edu.cn).

Digital Object Identifier 10.1109/TMI.2025.3585372

Index Terms—3D medical image generation, stable diffusion model, Patch-Volume Autoencoder, noise estimator.

I. INTRODUCTION

GENERATIVE artificial intelligence (AI) has led to remarkable breakthroughs in recent years, driving substantial advancements across various domains and applications. Specifically, it has significantly contributed to progress in medical imaging and healthcare, with immense potential for future innovations. Generative AI have shown strong capabilities in various medical applications including image translation [1], segmentation [2], and the reconstruction of Computed Tomography (CT) [3] and Magnetic Resonance (MR) images [4].

Different kinds of generative models have shaped such rapid advancement including Variational Autoencoders (VAE) [5], normalizing flows [6], Generative Adversarial Networks (GAN) [7], and Denoising Diffusion Probabilistic Model (DDPM) [8]. VAEs and normalizing flows are likelihood-based models, which explicitly model the data distribution and are theoretically well-founded. However, they often produce relatively modest generative results. On the other hand, GANs are generative models that employ an adversarial training approach to learn the underlying probability distribution of the data. This method can result in training instability, characterized by issues such as mode collapse [9]. DDPM addresses these limitations by utilizing a diffusion process, modeled as a Markov chain [8], [10]. Noise is incrementally added to the data, and a model is trained to reverse this process to generate samples. This method ensures stable training, producing high-quality, diverse outputs while offering a interpretable generative process. Large-scale models like stable diffusion [11] are widely applied in natural image domains and easily adapted for tasks such as image inpainting [12], and super-resolution [13].

Despite recent advancements, several challenges continue to impede the broader application of generative models in medical imaging. First, adapting computer vision algorithms to medical imaging is challenging due to its three-dimensional nature, which introduces significant computational overhead. Existing generative models in the medical domain are often

limited in both resolution and quality [14], [15]. Some works rely on slice-wise 2D generation [16], [17], [18], while clinical practice needs 3D volumetric imaging. Second, the lack of efficient and controllable generation mechanisms forces the development of task-specific models, thereby increasing development costs.

We propose 3D Medical Latent Diffusion (3D MedDiffusion), a large-scale 3D generative model designed for high-fidelity generation of medical images. First, 3D MedDiffusion introduces a novel Patch-Volume Autoencoder that compresses high-resolution images into low-resolution latent representations via a patch-wise encoder and recovers the entire image with a volume-wise decoder. Patch-wise encoding reduces the computational overhead in the training stage by exploiting the local properties in 3D medical images, while volume-wise decoding ensures artifact-free reconstruction results. Second, we design a novel noise estimator (named BiFlowNet) to replace the standard U-Net [19] in diffusion model. This module combines intra-patch and inter-patch flows to process both local (patch-wise) and global (volume-wise) information. The dual flow structure is crucial for reconstructing local details from noise while preserving global structure information. Furthermore, 3D MedDiffusion is trained on a comprehensive and diverse collection of 3D medical image datasets, covering both CT and MRI modalities and multiple anatomical regions from head to leg. Notably, ControlNet [20], a widely used method for injecting conditions into diffusion models, can be seamlessly integrated with the pre-trained 3D MedDiffusion, facilitating efficient fine-tuning for various downstream tasks.

In this study, we evaluate 3D MedDiffusion on its generative performance and its effectiveness in downstream tasks: sparse-view CT reconstruction, fast MRI reconstruction, and data augmentation for segmentation and classification. 3D MedDiffusion outperforms existing methods, demonstrating its superior performance.

II. RELATED WORK

A. Medical Image Generation

Medical image generation is promising to address several clinical tasks, such as modality translation, CT enhancement, and MR acceleration. Existing research has explored the application of GAN-based methods for generating 3D medical images. Jin et al. [21] introduced an autoencoding GAN to generate 3D brain MRI images. Cirillo et al. [22] proposed a 3D model conditioned on multi-channel 3D brain MR images to generate tumor masks for segmentation. Sun et al. [23] developed a GAN with a hierarchical design to address memory demands of generating high-resolution 3D images.

Nevertheless, the unstable training process inherent to GAN framework often results in mode collapse and low quality generation. Recently, many researchers have shifted their focus to diffusion models (DMs) due to their high fidelity generation and stable training [8]. Müller-Franzes et al. [24] proposed a conditional latent DM, demonstrating its promise for generating 2D medical images. Moghadam et al. [25] showed that DMs were capable of generating a wide range of

histopathological images. However, 3D imaging is more prevalent in medical contexts. This task poses challenges due to high computational complexity of DMs for three-dimensional data. Consequently, many researchers explored efficient methods for generating 3D medical images using DMs. Zhu et al. [16] developed 3D image generators using pseudo-3D architectures, addressing volumetric inconsistencies through refiners or tuning. Friedrich et al. [14] applied a diffusion model to wavelet-decomposed images for efficiency. Khader et al. [15] employed a latent DM combining an autoencoder and video DM to generate high-quality 3D CT and MR images. Guo et al. [26] introduced MAISI, a concurrent latent DM based framework that improved autoencoder efficiency via tensor-splitting parallelism. This method divides feature maps into smaller segments, enabling parallel processing across multiple GPUs or sequential processing on a single GPU. It resembles sliding window inference on feature space, so it produces less artifacts than directly applying it on image space. In contrast, our Patch-Volume Autoencoder follows a two-stage patch-wise and volume-wise training strategy, ensuring no boundary artifacts during single-GPU inference. Besides, we can also integrate tensor parallelism [27] for multi-GPU inference to accelerate the process.

B. Medical Generative Model for Downstream Tasks

Medical generative models hold great promise for diverse downstream tasks [28], [29], [30]. Recent studies have explored the data augmentation with DMs, such as training conditional DMs to generate synthetic data from segmentation masks [26]. Other works have applied DMs to solve inverse problems, including CT reconstruction [31] and image-to-image translation [32]. However, most of these methods require dedicated training for specific tasks. In contrast, our method leverages ControlNet [20] to eliminate task-specific training, enabling efficient fine-tuning for diverse downstream tasks.

III. METHOD

Our approach consists of three key components: a Patch-Volume Autoencoder that compresses a 3D medical image into a compact latent space (Sec. III-A), a noise estimator architecture named BiFlowNet for the 3D medical diffusion model (Sec. III-B), and ControlNet for conditional generation in downstream tasks (Sec. III-C).

A. Compact Latent Space for 3D Medical Image

To enable efficient 3D medical image generation, we develop a novel compression framework that maps high-resolution 3D volumes into a compact, semantically meaningful latent space. It facilitates more efficient diffusion process. Building upon VQ-VAE [33], we introduce Patch-Volume Autoencoder with a two-stage training strategy to address the challenges of 3D medical image compression (Fig. 1), offering three key advantages. **(1) Memory efficient.** The first stage compresses and reconstructs small, fine-detailed patches, overcoming memory limitations of processing entire volumes. **(2) Artifact-free reconstruction.** The second stage

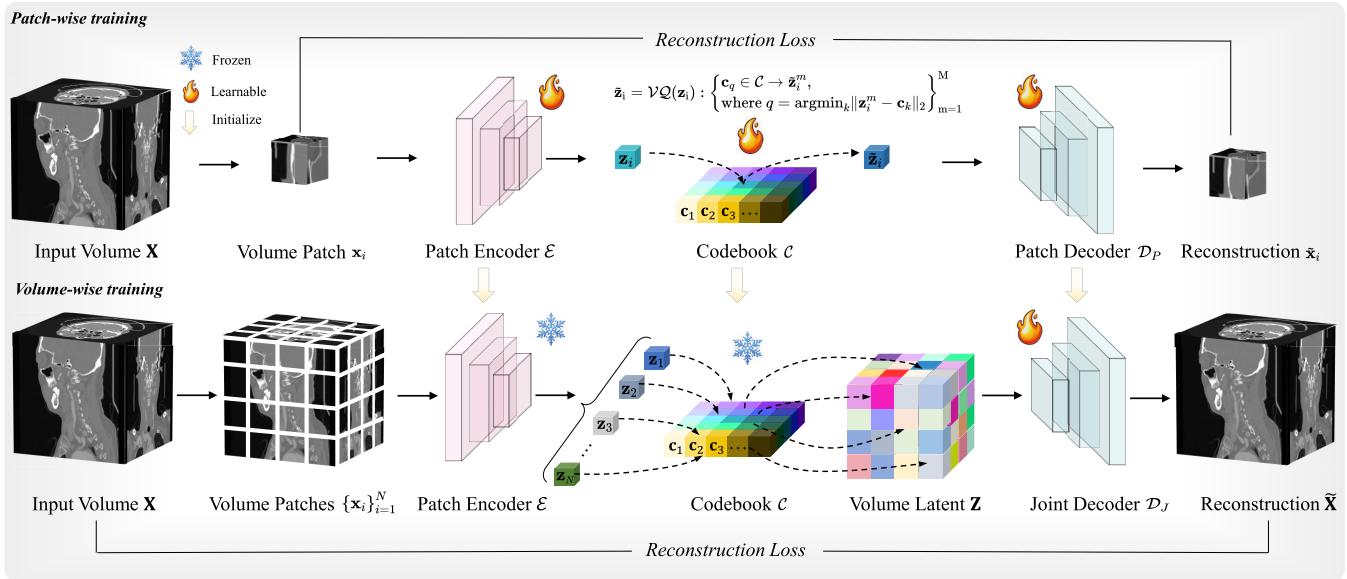


Fig. 1. Patch-Volume Autoencoder with a two-stage training strategy. In the first stage, the model is trained solely to compress and reconstruct small patches from high-resolution volumes. In the second stage, all parameters are fixed except for the decoder, which is fine-tuned on high-resolution volumes to become a joint decoder.

fine-tunes the joint decoder, enabling seamless full-volume reconstruction without producing patch boundary artifacts. (3) **Meaningful and compact representation.** The Vector Quantization (VQ) mechanism maps continuous representations to discrete codes in a learned codebook, capturing anatomical patterns and handling representation variations across anatomical regions.

1) Patch-Wise Training: In the first stage, the 3D volume $\mathbf{X} \in \mathbb{R}^{H \times W \times D}$ is divided into smaller patches $\{\mathbf{x}_i \in \mathbb{R}^{h \times w \times d}\}_{i=1}^N$, where N denotes the total number of patches. We process these patches in a patch-wise manner using a patch encoder \mathcal{E} , a patch decoder \mathcal{D}_P , and a discrete and learnable codebook $\mathcal{C} = \{\mathbf{c}_k \in \mathbb{R}^C\}_{k=1}^K$ with K denoting the codebook size

First, for each patch \mathbf{x}_i , the patch encoder \mathcal{E} extracts the corresponding patch feature $\mathbf{z}_i = \mathcal{E}(\mathbf{x}_i) \in \mathbb{R}^{M \times C}$, where M denotes the number of codes. Second, \mathbf{z}_i would then be transformed into quantized patch feature $\tilde{\mathbf{z}}_i \in \mathbb{R}^{M \times C}$ via vector quantization \mathcal{VQ} :

$$\tilde{\mathbf{z}}_i = \mathcal{VQ}(\mathbf{z}_i) : \left\{ \begin{array}{l} \mathbf{c}_q \in \mathcal{C} \rightarrow \tilde{\mathbf{z}}_i^m, \\ \text{where } q = \operatorname{argmin}_k \|\mathbf{z}_i^m - \mathbf{c}_k\|_2 \end{array} \right\}_{m=1}^M. \quad (1)$$

Third, the patch decoder \mathcal{D}_P produces the patch reconstruction $\tilde{\mathbf{x}}_i = \mathcal{D}_P(\tilde{\mathbf{z}}_i) \in \mathbb{R}^{h \times w \times d}$ from the quantized patch feature $\tilde{\mathbf{z}}_i$.

2) Volume-Wise Training: We fine-tune the patch decoder \mathcal{D}_P into a joint decoder \mathcal{D}_J , avoiding the need for individual patch processing and concatenation. It enables high-resolution reconstruction devoid of boundary artifacts.

As shown in Fig. 1, in this stage, the entire volume is first partitioned into smaller patches and each patch is then encoded and quantized to quantized patch features $\{\tilde{\mathbf{z}}_i\}_{i=1}^N$, similarly to prior patch-wise training. Afterward, these quantized features are concatenated into a latent volume $\tilde{\mathbf{Z}} \in \mathbb{R}^{N \cdot M \times C}$, which is then decoded by the joint decoder \mathcal{D}_J to produce the final reconstruction $\tilde{\mathbf{X}} = \mathcal{D}_J(\tilde{\mathbf{Z}}) \in \mathbb{R}^{H \times W \times D}$ at once. Notably,

TABLE I
COMPARISON OF MEMORY USAGE AND MAXIMUM BATCH SIZE FOR
DIFFERENT IMAGE SIZES WITH RESPECT TO TWO TRAINING
STRATEGIES

Image size	Naive		Efficient	
	Memory (GB)	Batch size	Memory (GB)	Batch size
$64 \times 64 \times 64$	3.56	28	2.88	41
$128 \times 128 \times 128$	24.10	3	16.21	5
$192 \times 192 \times 192$	/	/	52.48	1

the parameters of the patch encoder \mathcal{E} and the codebook \mathcal{C} are initialized from patch-wise training stage and kept frozen during this stage, while the joint decoder \mathcal{D}_J is learnable. This approach is memory-efficient, as it only requires storing the backward gradients for the joint decoder, \mathcal{D}_J . To demonstrate this, we report memory usage in Tab. I, comparing “Naive” volume-wise training, where gradients for all components are stored, with our “Efficient” volume-wise training, where only gradients for the joint decoder \mathcal{D}_J are stored. We tested three input sizes: $(64 \times 64 \times 64)$, $(128 \times 128 \times 128)$, and $(192 \times 192 \times 192)$, and present the **memory usage per image** and **maximum batch size** that can be processed on a single 80GB GPU. Note that we did not report results for “Naive” training with a $(192 \times 192 \times 192)$ input size, as its memory consumption exceeds the available 80GB. Our two-stage training (patch-wise encoding followed by volume-wise integration learning) specifically enables the model to learn how to coherently assemble patch latent representations into larger volumes, even if full volumes of that final size (e.g., 256^3 or 512^3) were not trained end-to-end during the volume-wise training stage.

3) Training Loss: We use the same loss functions for both training stages. In this section, we use \mathbf{x} to replace either \mathbf{x}_i

or \mathbf{X} and omit the dimensionality for simplicity. The full loss function for the Autoencoder, denoted as \mathcal{L}_{AE} , is formulated as follows:

$$\mathcal{L}_{AE} = \mathcal{L}_{VQ} + \lambda_{Adv} \mathcal{L}_{Adv} + \lambda_{TP} \mathcal{L}_{TP}, \quad (2)$$

where \mathcal{L}_{VQ} , \mathcal{L}_{Adv} , and \mathcal{L}_{TP} are vector quantization loss, adversarial loss and tri-plane loss, respectively. λ_{Adv} and λ_{TP} are loss weights for \mathcal{L}_{Adv} and \mathcal{L}_{TP} , respectively. Each loss function is detailed in the following.

\mathcal{L}_{VQ} is the fundamental loss for Patch-Volume Autoencoder training [33], defined as:

$$\begin{aligned} \mathcal{L}_{VQ} = & \| \mathbf{x} - \tilde{\mathbf{x}} \|_2 + \| \text{sg}[\mathcal{E}(\mathbf{x})] - \tilde{\mathbf{z}} \|_2 \\ & + \| \text{sg}[\tilde{\mathbf{z}}] - \mathcal{E}(\mathbf{x}) \|_2, \end{aligned} \quad (3)$$

where $\tilde{\mathbf{x}}$ is the reconstruction counterpart of \mathbf{x} , $\tilde{\mathbf{z}}$ is the quantized feature, and $\text{sg}[\cdot]$ stands for stop-gradient operation. This loss function minimizes the disparity between input data \mathbf{x} and its reconstruction $\tilde{\mathbf{x}}$, while aligning encoded features $\mathcal{E}(\mathbf{x})$ and feature codes $\tilde{\mathbf{z}}$ searched from codebook \mathcal{C} .

\mathcal{L}_{Adv} incorporates a discriminator \mathfrak{D} to encourage more realistic reconstructions [34]:

$$\mathcal{L}_{Adv} = \log \mathfrak{D}(\mathbf{x}) + \log(1 - \mathfrak{D}(\tilde{\mathbf{x}})). \quad (4)$$

\mathcal{L}_{TP} is employed to maintain high perceptual quality during vector quantization with help of perceptual loss [35], [36], [37]. In order to leverage existing 2D pre-trained feature extractor, we randomly select three orthogonal 2D planes from input data $\{\mathbf{p}_n\}_{n=1}^3 \subset \mathbf{x}$ and reconstruct the planes $\{\tilde{\mathbf{p}}_n\}_{n=1}^3 \subset \tilde{\mathbf{x}}$ at the same position to compute perceptual loss:

$$\mathcal{L}_{TP} = \sum_{n=1}^3 \| \phi(\mathbf{p}_n) - \phi(\tilde{\mathbf{p}}_n) \|_2, \quad (5)$$

where ϕ is the pre-trained VGG-16 network [38], [39].

B. 3D Medical Diffusion Model

1) *Diffusion Model*: The diffusion model in 3D MedDiffusion operates in the compact latent space, either patch-wise (\mathbf{x}_i) or volume-wise (\mathbf{X}). In this section, we use \mathbf{x} to denote both and omit dimensionality for simplicity. Diffusion model first defines a forward noising process that incrementally adds noise to the initial latent $\mathbf{z}^0 = \mathcal{E}(\mathbf{x})$, formulated as:

$$q(\mathbf{z}^{1:T} | \mathbf{z}^0) := \prod_{t=1}^T q(\mathbf{z}^t | \mathbf{z}^{t-1}),$$

$$\text{where } q(\mathbf{z}^t | \mathbf{z}^{t-1}) = \mathcal{N}(\mathbf{z}^t; \sqrt{1 - \beta^t} \mathbf{z}^{t-1}, \beta^t \mathbf{I}). \quad (6)$$

Here $\sqrt{1 - \beta^t} \mathbf{z}^{t-1}$ is the mean and $\beta^t \mathbf{I}$ is the covariance of the distribution at timestep $t \in \{1, 2, \dots, T\}$, with T denoting the total number of timesteps. \mathbf{I} is the identity matrix, and $\beta_t \in (0, 1)$ is a noise-level hyperparameter.

Using the reparameterization trick [6], one can sample \mathbf{z}^t as follows:

$$\mathbf{z}^t = \sqrt{\alpha^t} \mathbf{z}^0 + \sqrt{1 - \alpha^t} \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad (7)$$

where $\alpha^t = 1 - \beta^t$ and $\bar{\alpha}^t = \prod_{i=1}^t \alpha^i$. For the reverse process, the diffusion model is trained to learn a denoising network μ_θ , expressed as:

$$p_\theta(\mathbf{z}^{t-1} | \mathbf{z}^t) = \mathcal{N}(\mathbf{z}^{t-1}; \mu_\theta(\mathbf{z}^t, t), \sigma^t \mathbf{I}), \quad (8)$$

Here $\mu_\theta(\mathbf{z}^t, t)$ and $\sigma^t \mathbf{I}$ are the mean and covariance. By combining Eq. (7) and Eq. (8), we can derive:

$$\mu_\theta(\mathbf{z}^t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}^t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}^t}} \epsilon_\theta(\mathbf{z}^t, t) \right). \quad (9)$$

The training objective is to train a noise estimator ϵ_θ to predict the added Gaussian noise ϵ , described by the loss function:

$$\mathcal{L}_{Diff} = \mathbb{E}_{\mathbf{z}^0, \epsilon, t, \mathbf{c}} \left[\| \epsilon - \epsilon_\theta(\mathbf{z}^t, t, \mathbf{c}) \|_2 \right], \quad (10)$$

where \mathbf{c} is the class condition, indicating the data modality and anatomical regions. Once $p_\theta(\mathbf{z}^{t-1} | \mathbf{z}^t)$ is trained, new latents can be generated by progressively sampling $\mathbf{z}^{t-1} \sim p_\theta(\mathbf{z}^{t-1} | \mathbf{z}^t)$.

2) *BiFlowNet Architecture*: We propose a novel noise estimator called BiFlowNet, which integrates both intra-patch flow and inter-patch flow to generate latent representations. Intra-patch flow captures fine-grained details, whereas inter-patch flow maintains global structural consistency. The dual-flow latent features are integrated to achieve both objectives simultaneously.

a) *Intra-patch flow*: In our approach, high-resolution 3D medical images are encoded into latent space on patch-wise basis, resulting in each patch following an independent distribution. Intra-patch flow focuses on denoising each patch and recovering fine-grained local details. We adopt diffusion transformer (DiT) [40] as the backbone for intra-patch flow, leveraging its exceptional performance in complex generative tasks on large datasets. Class embedding is also incorporated as input to enable multi-class training for different regions and modalities. The DiT block architecture follows the design in [40], with adaptations for 3D inputs such as expanded spatial dimensions, as shown in Fig. 2.

b) *Inter-patch flow*: However, generating only patch-wise regions fails to preserve global structural consistency when reconstructing the entire 3D volume. In response to this challenge, we design inter-patch flow to capture and reconstruct the global structures across the entire volume. We opt a standard 3D U-Net [19] as the backbone to eliminate computational overhead raised from high-resolution 3D images.

c) *Dual flow integration*: As shown in Fig. 2, to avoid isolated optimization, we combine local and global features from dual flows with element-wise addition at each timestep. Features from DiT's first and last two blocks are merged with U-Net's corresponding blocks. The latent volume is then processed by the U-Net and divided into patches for the DiT at the next timestep.

C. ControlNet for Downstream Tasks

To improve controllability and adaptability for diverse downstream tasks, we integrate ControlNet [20] into the diffusion model. ControlNet is specifically designed to inject

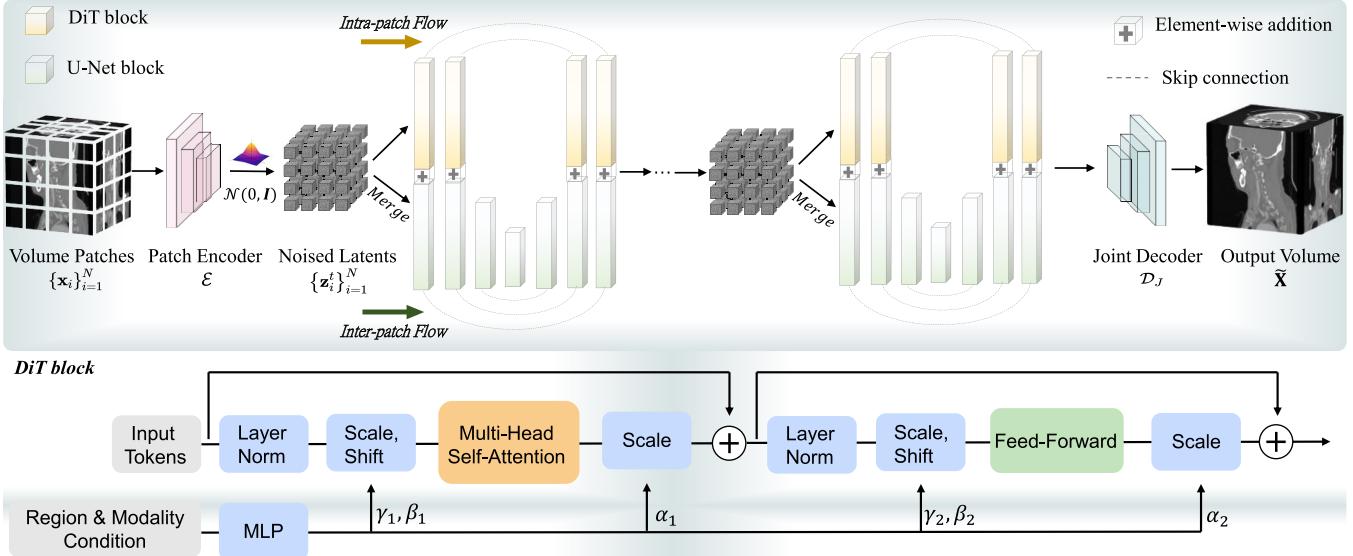


Fig. 2. BiFlowNet noise estimator. The intra-patch flow focuses on denoising each patch and recovering fine-grained local details, while the inter-patch flow is designed to capture and reconstruct the global structures across the entire volume.

additional task-specific conditions into the diffusion model via fine-tuning. In this process, the parameters of the pre-trained diffusion model are frozen, while a cloned, trainable encoder of BiFlowNet is introduced. This trainable copy accepts task-specific condition as input and its output is connected to the pre-trained diffusion model via zero convolutions [20]. The network architecture is shown in Fig. 3. Since the diffusion model in 3D MedDiffusion operates within the latent space, it is essential to encode the task-specific condition into this space, denoted as \mathbf{c}_{task} . The loss function for the fine-tuning process is defined as:

$$L_{Con} = \mathbb{E}_{\mathbf{z}^0, \epsilon, t, \mathbf{c}, \mathbf{c}_{task}} \left[\|\epsilon - \epsilon_\theta(\mathbf{z}^t, t, \mathbf{c}, \mathbf{c}_{task})\|_2 \right]. \quad (11)$$

This control mechanism allows the 3D MedDiffusion to evolve from a generic generative model into a task-specific model by leveraging the pre-trained model's strong prior knowledge.

Moreover, the fine-tuning process is notably time-efficient, as only a small set of parameters require training.

IV. EXPERIMENTS

In this section, we begin by detailing the datasets (Sec. IV-A.1) and implementation details (Sec. IV-A.4). Next, we compare our 3D MedDiffusion with competing methods (Sec. IV-B). Finally, we assess the generalizability of 3D MedDiffusion across four downstream tasks: sparse-view CT reconstruction, MRI reconstruction, data augmentation for segmentation and data augmentation for classification (Sec. IV-E).

A. Experimental Setting

1) Dataset: To develop a large-scale 3D generative model, we train the 3D MedDiffusion on an extensive collection of high-quality 3D medical image datasets. These datasets include CT scans of head and neck, chest and abdomen, and lower limb, as well as MR images of brain, chest, abdomen, and knee, covering the major anatomical regions of the human

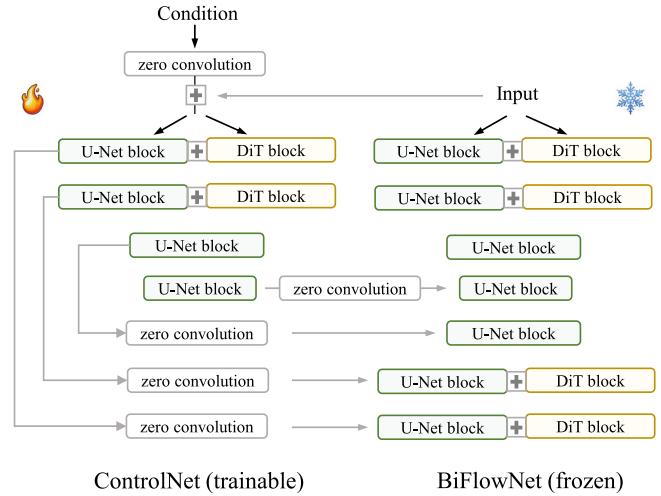


Fig. 3. The network architecture of ControlNet.

body. We organize these multi-modality and region-mixed datasets into six sub-datasets: **CTHeadNeck**, **CTChestAbdomen**, **CTLowerLimb**, **MRBrain**, **MRChestAbdomen**, **MRKnee**.

a) CTHeadNeck dataset: For head and neck region of CT modality, we collect one public dataset and two in-house datasets. The public dataset is the CQ500 dataset [41] with 491 CT scans. The in-house datasets consist of a CTA dataset with 256 images and a CBCT dataset with 600 images. In total, we collect 1,347 images with image size ranging from $256 \times 256 \times 128$ to $256 \times 256 \times 256$ and voxel spacing below 1.25mm along all three axes.

b) CTChestAbdomen dataset: For chest and abdomen region of CT modality, we involve four public CT datasets: The Pulmonary Airway dataset [42], [43], [44], [45] with 300 CT scans. The LIDC-IDRI dataset [46] with 1,010 CT scans. The CTSpline1K dataset [47] with 1,005 CT scans. The

AbdomenCT-1K dataset [48] with 1,112 CT scans. In all these four datasets, we gather 3,000 images after excluding low-quality ones with image size ranging from $256 \times 256 \times 256$ to $512 \times 512 \times 512$ and voxel spacing below 1.25mm along all three axes.

c) *CTLowerLimb dataset*: For lower limb region of CT modality, we develop an in-house dataset with 400 CT scans, covering the area from the thigh to the foot. The image size ranges from $256 \times 256 \times 256$ to $256 \times 256 \times 512$, with voxel spacing below 1.25mm along three axes.

d) *MRBrain dataset*: For brain region of MR modality, we collect 3,000 MR images from the UK Biobank [49], comprising 1,500 T1-weighted images and 1,500 T2-weighted FLAIR images. Non-brain regions have been excluded to enhance model focus. All images have a size of $192 \times 192 \times 192$ and a voxel spacing of 1mm along all three axes.

e) *MRChestAbdomen dataset*: For chest and abdomen region of MR modality, we create an in-house dataset comprising 1,826 MR images, including both T1-weighted and T2-weighted data. All images have a size of $256 \times 256 \times 128$ and a voxel spacing of 1.25mm along the x - and y -axes, and 2mm along the z -axis.

f) *MRKnee dataset*: For knee region of MR modality, we use the publicly available fastMRI [50] dataset, focusing exclusively on single-coil data for model development. This dataset comprises 1172 volumes, each consisting of approximately 40 slices with size of 320×320 . We interpolate the data along the z -axis, standardizing each volume to a size of $320 \times 320 \times 64$. All images have a voxel spacing of 1mm along the x - and y -axes, and 5mm along the z -axis.

During the data collection process, for some public datasets, we perform random selection to choose an appropriate amount of data from the original dataset to ensure class balance. For example, we use only 3,000 MR brain images from the original dataset. Additionally, we exclude images of low quality, such as those with very large spacing or abnormal FOV (Field of View). We perform the max-min normalization, scaling the voxel values to the range of $[-1, 1]$.

2) Competing Methods: We select several generative methods as baselines, including both GAN-based and diffusion-based approaches, as detailed below. HA-GAN [23] is a hierarchical amortized GAN which generates low-resolution images and high-resolution sub-volumes simultaneously to address memory constraints. Medical Diffusion [15] is a video diffusion model trained in the learned latent space of a 3D VQ-GAN. WDM [14] is a diffusion-based framework for medical image synthesis using wavelet-decomposed images. MAISI [26] is a 3D U-Net trained in the learned latent space of a 3D VAE. MedSyn [51] is a diffusion-based framework designed for generating high-fidelity 3D CT images with textual guidance.

3) Comparison Settings: To ensure a fair comparison, we trained three models for our method: Ours (CT), trained exclusively on **CTChestAbdomen** dataset, Ours (MR), trained exclusively on **MRBrain**, and Ours (universal) trained on all of the six dataset mentioned above. Then we retrained all of the competing methods listed in Sec. IV-A.2 on

TABLE II
TRAINING AND INFERENCE TIME OF EACH COMPARISON METHODS ON THE **CTChestAbdomen** DATASET

	Training Time	Inference Time
HA-GAN	30h	<1s
MedicalDiffusion	79h	83s
WDM	210h	251s
MAISI	81h	82s
MedSyn	71h	255s
Ours (CT)	83h	128s

CTChestAbdomen dataset for comparison with Ours (CT), and also on **MRBrain** for comparison with Ours (MR).

4) Implementation Details: For Patch-Volume Autoencoder, during the patch-wise training phase, we set the patch size to $64 \times 64 \times 64$, with a compression ratio of $4 \times 4 \times 4$. The codebook contains 8192 codes with a dimensionality of 8. The learning rate is 3×10^{-4} . We start to train the discriminator after 20,000 iterations. In the volume-wise training phase, only the decoder and discriminator are trainable. The learning rate is lowered to 3×10^{-5} . We set $\lambda_{Adv} = 2$ and $\lambda_{TP} = 4$ for both phases.

For BiFlowNet noise estimator, we employ the cosine noise schedule [52] with $T = 1000$ timesteps, and the learning rate of 1×10^{-4} with polynomial learning rate decay.

In the comparison experiments, we trained all of the comparison methods from scratch including Ours (CT) and Ours (MR) under the same hardware condition, i.e., a single NVIDIA A100 80 GB GPU. The training and inference times for each competing method on the **CTChestAbdomen** dataset are reported in Tab. II. For training time, all methods were trained for over 10 days. We selected the best checkpoint to report both training time and evaluation metrics. For the inference time, we performed inference for all methods on generating a single image at $256 \times 256 \times 256$ resolution.

Moreover, for Ours (universal), the training of the Patch-Volume Autoencoder for this model was performed on a single NVIDIA A100 80 GB GPU, requiring 72 GPU hours. The training for the BiFlowNet noise estimator, 8 NVIDIA A100 80 GB GPUs were used, requiring a total of 120 GPU hours.

Regarding resolution, all competing methods, Ours (CT), and Ours (MR) were trained using a uniform resolution by resizing all images in the **CTChestAbdomen** dataset to $256 \times 256 \times 256$, while images in the **MRBrain** dataset were not resized, as they are originally at $192 \times 192 \times 192$ size. However, for Ours (universal), uniform resolution was not applied.

5) Evaluation Metrics: We evaluate the fidelity and diversity of generated images using Fréchet Inception Distance (FID) [53] and Maximum Mean Discrepancy (MMD) [54], which assess the similarity between the distributions of real and generated images. For feature extraction in FID and MMD calculations, we use a 3D ResNet model pre-trained on 3D medical images [55]. Lower FID and MMD values indicate higher realism of the generated images. Besides, we use the Multi-Scale Structural Similarity Index (MS-SSIM) to assess the diversity of the generated images. MS-SSIM measures the

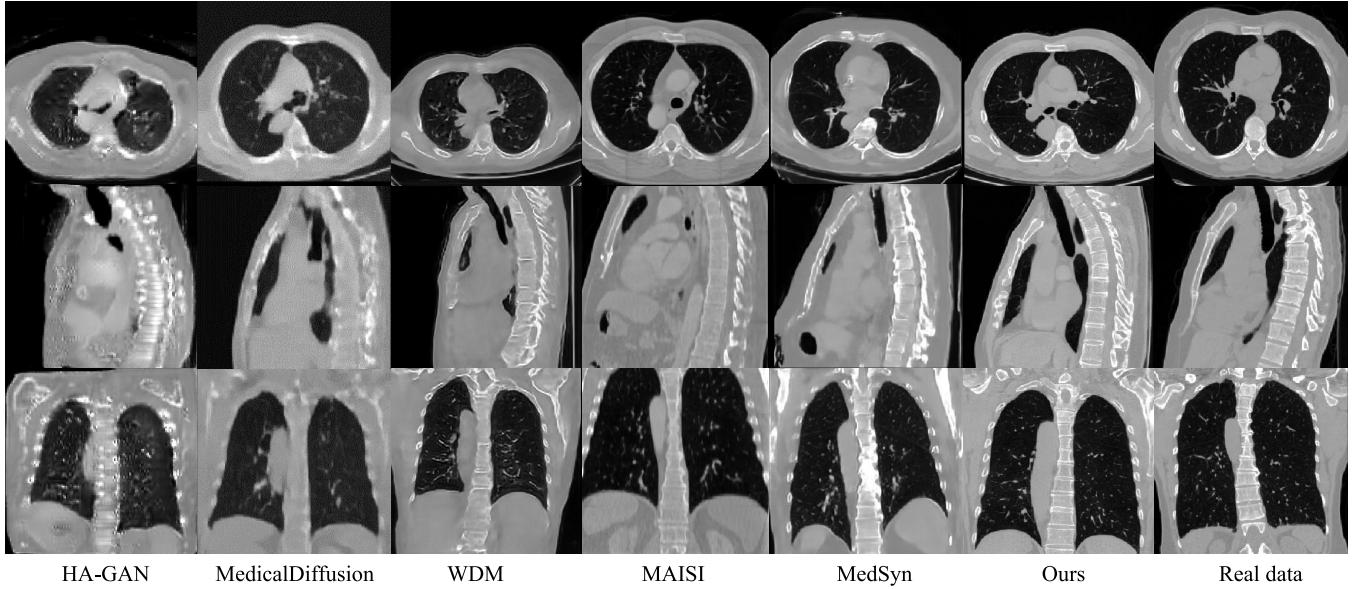


Fig. 4. Qualitative results on **CTChestAbdomen**. Window: $[-1000, 1000]$ HU.

TABLE III
QUANTITATIVE RESULTS ON **CTCHESTABDOMEN**

	FID ↓	MS-SSIM ↓	MMD ↓
HA-GAN	0.0367	0.5723	1.0387
MedicalDiffusion	0.0284	0.2227	0.6500
WDM	0.0203	0.2495	0.4656
MAISI	0.0135	0.2713	0.2782
MedSyn	0.0174	0.2725	0.3122
Ours(CT)	0.0055	0.2238	0.1049
Ours (universal)	0.0041	0.2173	0.1023

TABLE IV
QUANTITATIVE RESULTS ON **MRBRAIN**

	FID ↓	MS-SSIM ↓	MMD ↓
HA-GAN	0.1284	0.7443	4.4638
MedicalDiffusion	0.0291	0.7847	1.9492
WDM	0.0057	0.7844	0.8742
MAISI	0.0075	0.7380	0.6957
MedSyn	0.0087	0.7778	0.7760
Ours(MR)	0.0044	0.7036	0.6372
Ours (universal)	0.0039	0.7027	0.6199

structural similarity between generated images across multiple scales, with lower values reflecting higher diversity.

B. Comparison

1) Quantitative Evaluation: We generated 1,000 images for each competing method for quantitative evaluation. The quantitative experimental results for **CTChestAbdomen** dataset are provided in Tab. III. For fairness, we exclude the results of Ours (universal) from the following analysis, as it is trained on more extensive resources and possesses a stronger capability, although its results are still presented. Ours (CT) achieves the lowest FID and MMD scores compared with other competing methods, demonstrating superior generative fidelity and realism. Compared to the second-best method, MAISI, FID and MMD scores for Ours (CT) are over two times lower, highlighting a significant improvement in generative fidelity. While MedicalDiffusion achieves the best performance in MS-SSIM metric, our method obtains comparable results. The quantitative experimental results for the **MRbrain** dataset are summarized in Tab. IV. Our method also achieves the best performance among all the metrics by a considerable margin.

2) Qualitative Evaluation: To qualitatively evaluate the generation capabilities of all methods, we present some generated samples from **CTChestAbdomen** dataset (Fig. 4) and **MRBrain** dataset (Fig. 5). It is evident that HA-GAN, MedicalDiffusion, and WDM struggle to produce high-resolution CT images, resulting in noisy and blurry outputs with insufficient details. MAISI and MedSyn produce satisfactory results but are slightly inferior to our method. In contrast, our proposed method excels in capturing fine details like vertebrae in CT images. It also produces sharp edges, such as the brain surface of MR images.

To further evaluate the authenticity of the generated images, we project the images into a high-dimensional embedding space and apply a dimensionality reduction technique to visualize the distributions. For comparison, we first employ a 3D pre-trained ResNet model [55] to extract features from 750 generated CT images for each method. We then apply t-SNE [56] to reduce the dimensionality to 2. This enables us to evaluate whether the generated distributions align well with real distributions in latent space. The resulting two-dimensional space is visualized in Fig. 6. We observe that our

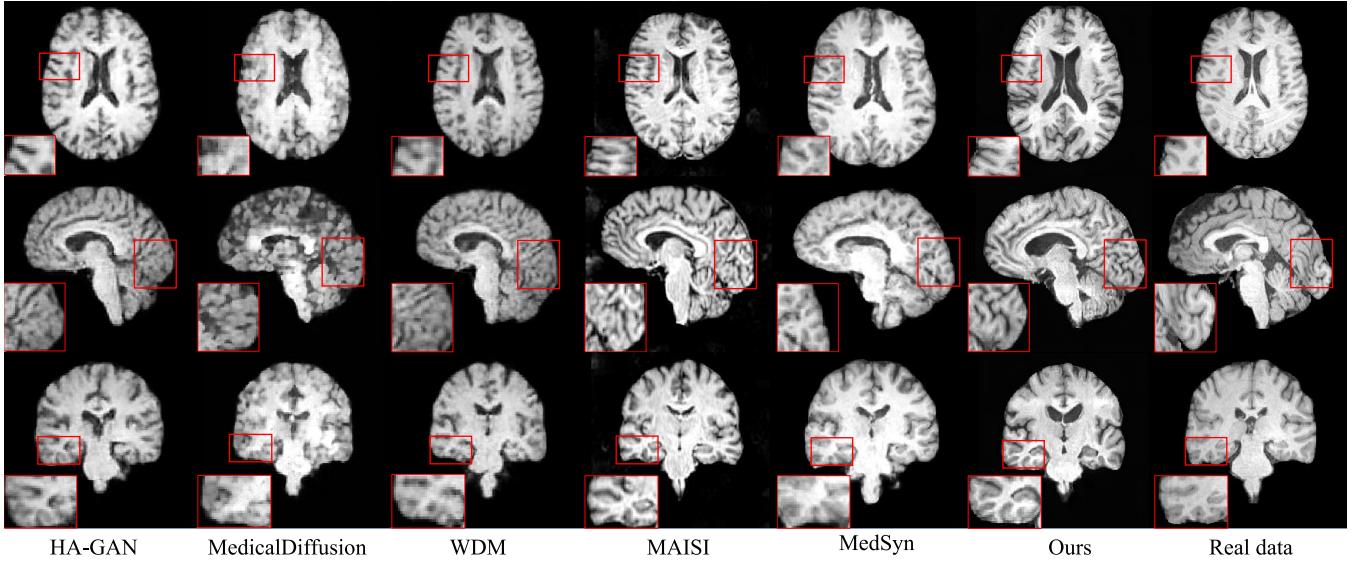


Fig. 5. Qualitative results on **MRBrain**.

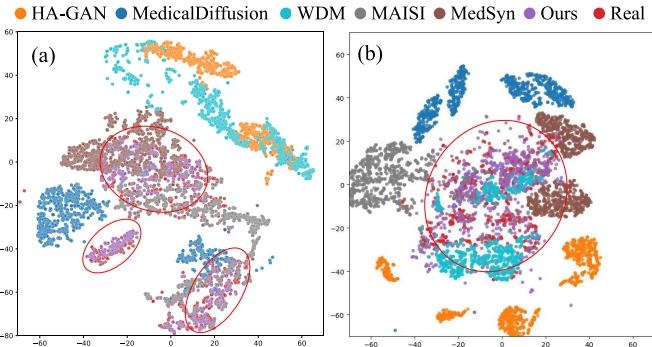


Fig. 6. t-SNE visualization on (a) **CTChestAbdomen** dataset and (b) **MRBrain** dataset. Features extracted from the generated images are embedded into a 2-dimensional space.

generated distribution (purple dots) aligns more closely and compactly with the real distribution (red dots), as highlighted by the red ellipses. This demonstrates the superiority of our method in generating realistic images.

C. Ablation Study

In this section, we validate the effectiveness of two main components employed in our method, including Patch-Volume Autoencoder and BiFlowNet noise estimator.

1) Patch-Volume Autoencoder: Our method uses a two-stage training strategy for Autoencoder. In the first stage, entire network is trained in a patch-wise manner. In the second stage, the patch decoder is fine-tuned into a joint decoder in a volume-wise manner. To evaluate the effectiveness of our joint decoder and the used losses, we conducted two ablation studies.

In **Study a.**, we assess the effectiveness of our joint decoder. Since training the model in a purely volume-wise manner is not feasible due to memory constraints, we focus on evaluating the joint decoder by comparing it to the patch-wise decoder for image compression and reconstruction.

TABLE V
QUANTITATIVE RESULTS OF ABLATION STUDY ON JOINT DECODER

	PSNR (dB) \uparrow	SSIM \uparrow
w/o joint decoder	34.2307 (1.2492)	0.9273 (0.0925)
w/ joint decoder	35.7916 (1.2187)	0.9314 (0.0963)

In **Study b.**, we investigate how the adversarial loss and tri-plane loss affect self-reconstruction performance. We compare four different settings: (1) removing the adversarial loss, \mathcal{L}_{Adv} , (2) removing the tri-plane loss, \mathcal{L}_{TP} , (3) removing both the adversarial and tri-plane losses, and (4) using the full loss.

Both **Study a.** and **Study b.** were conducted using the **CTChestAbdomen** dataset, with all settings controlled under the same hardware conditions. We evaluated performance using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [57] between the original and reconstructed images.

The results of **Study a.** are provided in **Tab. V**. Numbers outside parentheses represent the mean values, while those inside denote the standard deviations. The results indicate that incorporating the joint decoder significantly improves reconstruction quality across both metrics. Specifically, as shown in **Fig. 7**, the joint decoder effectively eliminates patch boundary artifacts, which are commonly introduced by patch-wise decoding.

The results of **Study b.** are presented in **Tab. VI**. Numbers outside parentheses represent the mean values, while those inside denote the standard deviations. When either \mathcal{L}_{Adv} or \mathcal{L}_{TP} is removed, performance drops significantly. Notably, the tri-plane loss, \mathcal{L}_{TP} , plays a critical role in improving performance, highlighting that the perceptual loss is highly effective for the self-reconstruction task.

2) BiFlowNet: To verify the effectiveness of our BiFlowNet noise estimator, we conducted two ablation studies: **Study a.**, focusing on the effectiveness of the architecture design. And

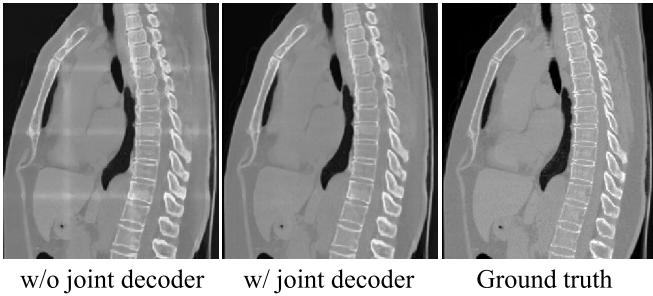


Fig. 7. Self-reconstruction results of ablation study on joint decoder. Window: [−1000, 1000] HU.

TABLE VI
QUANTITATIVE RESULTS OF THE ABLATION STUDY ON THE LOSS FUNCTION OF PATCH-VOLUME AUTOENCODER

\mathcal{L}_{VQ}	\mathcal{L}_{Adv}	\mathcal{L}_{TP}	PSNR (dB) \uparrow	SSIM \uparrow
✓			32.8993 (1.4932)	0.9172 (0.1092)
✓	✓		33.2713 (1.2983)	0.9166 (0.0991)
✓		✓	34.2783 (1.0423)	0.9184 (0.0954)
✓	✓	✓	35.7916 (1.2187)	0.9314 (0.0963)

Study b., focusing on the compatibility with our proposed Autoencoder.

In **Study a.**, we compare two different settings: (1) removing the intra-patch flow from the noise estimator, and (2) replacing the DiT block of the intra-patch flow with a standard U-Net block. Note that the model cannot be trained with intra-patch flow alone, as the inter-patch flow is essential for the model to produce meaningful reconstructions across the entire volume.

In **Study b.**, to demonstrate that BiFlowNet is more suitable for patch-wise encoded latents and fully compatible with our proposed Autoencoder, we compare two settings: (1) training BiFlowNet on volume-wise encoded latents and (2) training BiFlowNet on patch-wise encoded latents. Our Patch-Volume Autoencoder employs patch-wise encoding and volume-wise decoding during training. Due to memory constraints, volume-wise encoding is not feasible in the training phase. However, during inference, we can perform volume-wise encoding to generate latents, as storing gradients is no longer necessary.

Both **Study a.** and **Study b.** were conducted using the **CTChestAbdomen** dataset, and all settings were controlled under the same hardware conditions. We evaluated performance using FID and MMD metrics.

The results of **Study a.** are presented in Tab. VII and Fig. 8. These findings clearly demonstrate that capturing intra-patch information is crucial for generating fine-grained local details. The DiT block outperforms the U-Net block, especially for generative models trained on large-scale datasets.

The results of **Study b.** are presented in Tab. VIII. The findings indicate that, when training the noise estimator on patch-wise encoded latents, its performance is significantly better than the case when trained on volume-wise encoded latents. This highlights the perfect compatibility of our BiFlowNet noise estimator with our proposed Patch-Volume Autoencoder.

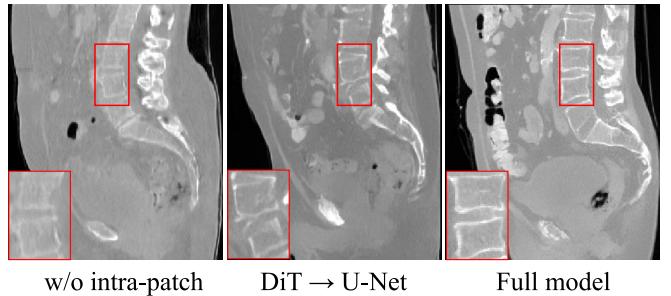


Fig. 8. Qualitative results of ablation study on BiFlowNet. Window: [−1000, 1000] HU.

TABLE VII
QUANTITATIVE RESULTS OF THE ABLATION STUDY ON THE BIFLOWNET ARCHITECTURE

	w/o intra-patch	DiT → U-Net	Full model
FID \downarrow	0.0117	0.0087	0.0055
MMD \downarrow	0.2392	0.1253	0.1049

TABLE VIII
QUANTITATIVE RESULTS OF ABLATION STUDY ON THE ENCODING STRATEGIES

	Volume-wise encoding	Patch-wise encoding
FID \downarrow	0.0073	0.0055
MMD \downarrow	0.1147	0.1049

D. Human Study

To further evaluate the realism and quality of the generated images, we conducted a human study involving radiologists. This study was designed to assess both the perceived image quality and realism of the generated images from our proposed method, competing methods, and real images. This evaluation specifically focused on the chest and abdomen regions, which are critical areas in medical imaging. A total of four radiologists participated in the evaluation process. Each radiologist was required to rank the images based on their overall visual quality and the degree to which they resembled real clinical images. The images used in this human study were carefully selected to ensure that they provided a representative sample of typical clinical scenarios. Following the experimental setup described in [51], we designed an online survey that allowed the radiologists to rank the images in terms of quality and realism. The ranking scale used in the survey ranged from 1 to 7, with 1 representing the highest realism/quality and 7 representing the lowest. The mean ranks for each method are summarized in Fig. 9, which reveal that the mean rank of the images generated by our method closely aligns with the real images, outperforming the other methods in both quality and realism.

E. Downstream Tasks

The pre-trained generative model can be adapted to a variety of downstream tasks through integration with ControlNet. In this section, we assess its performance across four tasks: sparse-view CT reconstruction, fast MRI reconstruction, data

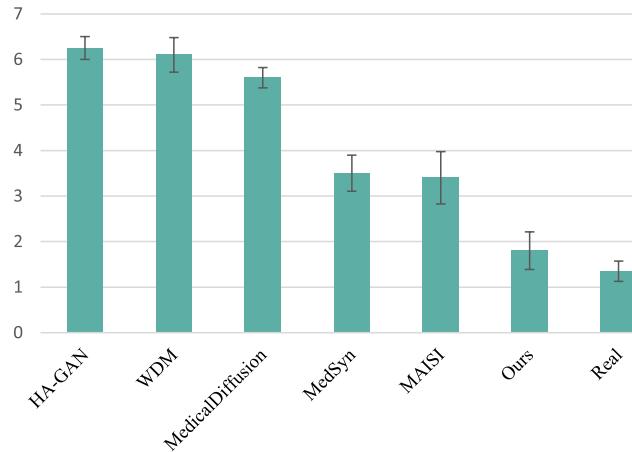


Fig. 9. The ranks for each method in human study.

TABLE XI
QUANTITATIVE RESULTS OF DATA AUGMENTATION FOR CLASSIFICATION

	Baseline	Augmentation
Accuracy(%) \uparrow	75.00	79.17
F1 \uparrow	0.7368	0.7863

2) Fast MRI Reconstruction: In this experiment, we aim to accelerate MRI reconstruction by leveraging pre-trained 3D MedDiffusion to map the k-space downsampled reconstructions to the fully-sampled reconstructions. We conduct experiment on **MRKnee** dataset and compare our method against zero-filling, U-Net [61], and DDS [31]. PSNR and SSIM are used as evaluation metrics. We adopt two types of downsampling masks including 1D Gaussian mask and Poisson mask, and the downsampling ratio is $8\times$. The quantitative results are summarized in Tab. X (with the values in parentheses denoting the standard deviations), with two examples of different downsampling masks shown in Fig. 11. Notably, our method achieves the highest metric performance in both downsampling masks, and obtain superior reconstructions. Our volume-wise generation ensures better 3D consistency compared to the slice-wise generation of competing methods.

3) Data Augmentation for Segmentation: Deep learning-based medical image segmentation is often limited by insufficient training data. In this experiment, we leverage the strong generative capabilities of 3D MedDiffusion to augment the data and enhance segmentation performance. We use a standard nnU-Net [62] for segmentation on the **KiTs19** dataset [58], which includes kidney and tumor regions. The testing set consists of 40 CT images and their segmentation masks, with the remaining 170 cases available for training. We define the following five training settings: **(1) 100% Synthesis:** Generate new CT images using 3D MedDiffusion conditioned on the segmentation masks from the training set. **(2) 50% Real + 50% Synthesis:** Randomly select 50% of the real data from the training set, and then generate new CT images from the remaining 50% using 3D MedDiffusion conditioned on their segmentation masks. **(3) 100% Real:** Use all 170 cases. **(4) 100% Real + 50% Synthesis:** Use all 170 cases and generate new CT images for 50% of the selected cases. **(5) 100% Real + 100% Synthesis:** Use all 170 cases and generate new CT images for 100% of the selected cases.

For evaluation, we conducted 5-fold cross-validation and used the 95% Hausdorff Distance (95HD) and Dice Similarity Coefficient (DSC) as comparative metrics. The overall quantitative results are presented in Fig. 12 (with the values in parentheses denoting the standard deviations).

For settings (1), (2), and (3), when the number of training samples is controlled, synthetic data can effectively replace real data. Specifically, when comparing 100% synthetic data with a mix of 50% synthetic and 50% real data (experiments (1) and (2)), the Dice score for tumor and kidney segmentation decreased by just 0.13% and 0.16%, respectively. Even when comparing 100% synthetic data to 100% real data (experiments (1) and (3)), the performance degradation remained

TABLE IX
QUANTITATIVE RESULTS OF SPARSE-VIEW CT RECONSTRUCTION

	FBP	FBPConv	DOLCE	DDS	Ours
PSNR (dB) \uparrow	16.5327 (2.3140)	24.8138 (2.5968)	18.3710 (1.2488)	24.8110 (3.0709)	27.9169 (4.1454)
SSIM \uparrow	0.6643 (0.1352)	0.7756 (0.1571)	0.7091 (0.1396)	0.6488 (0.1270)	0.9297 (0.0313)

TABLE X
QUANTITATIVE RESULTS OF MRI RECONSTRUCTION

	Zero-filling	U-Net	DDS	Ours
PSNR (dB) \uparrow	29.4335 (1.8688)	33.5105 (2.2259)	30.4892 (1.8758)	34.5374 (3.2673)
SSIM \uparrow	0.8717 (0.0308)	0.9107 (0.0453)	0.8847 (0.0374)	0.9130 (0.0569)

augmentation for segmentation and data augmentation for classification.

1) Sparse-View CT Reconstruction: In this experiment, we demonstrate the feasibility of leveraging the powerful generative prior knowledge learned by 3D MedDiffusion to address the sparse-view CT reconstruction. We conduct experiment on a public dataset **KiTs19** [58], which includes 210 Kidney CT scans. For each 3D CT volume, we uniformly render 40 views in a full circle with fan-beam geometry based on ASTRA-toolbox [59] and then perform reconstruction. Our pre-trained model is fine-tuned with ControlNet. The fine-tuning objective is defined in Eq. (11). The Filtered Back-Projection (FBP) [60] reconstruction results used as the condition input with the full-view reconstruction results serving as the ground truth.

We compare our method with several existing approaches, including FBPCov [61], DOLCE [3], and DDS [31], and report both PSNR and SSIM metrics in Tab. IX (with the values in parentheses denoting the standard deviations) to evaluate performance. As shown in Tab. IX, our method achieves the highest scores across both metrics, demonstrating our effectiveness. Additionally, two representative cases are presented in Fig. 10, where our reconstruction results deliver superior image quality and enhanced clarity of details.

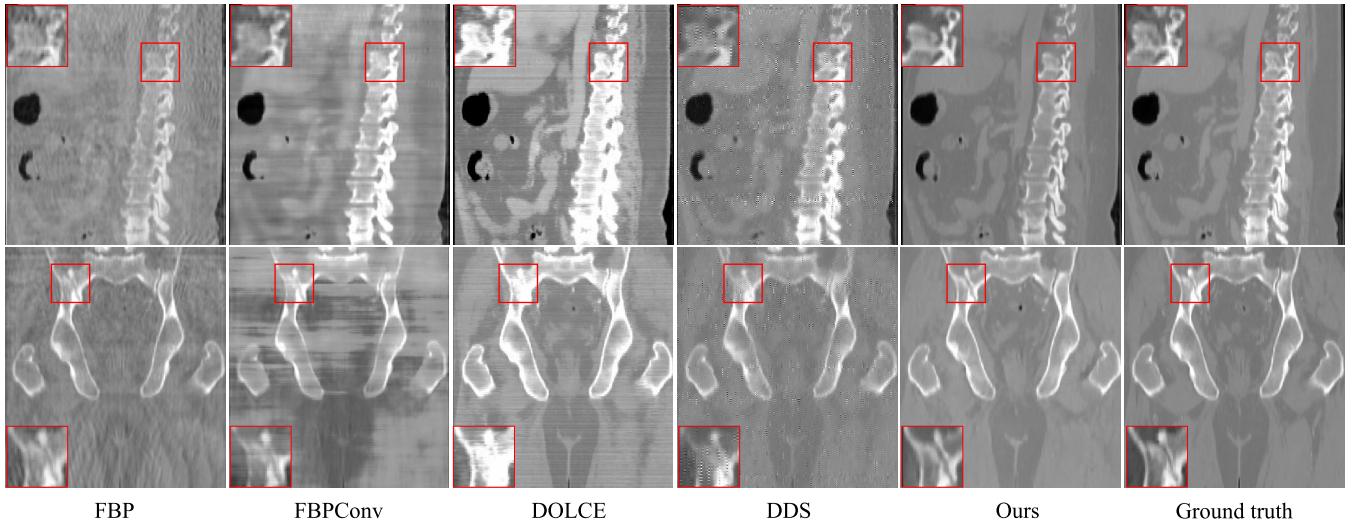


Fig. 10. Qualitative results of sparse-view CT reconstruction. Window: $[-1000, 1000]$ HU.

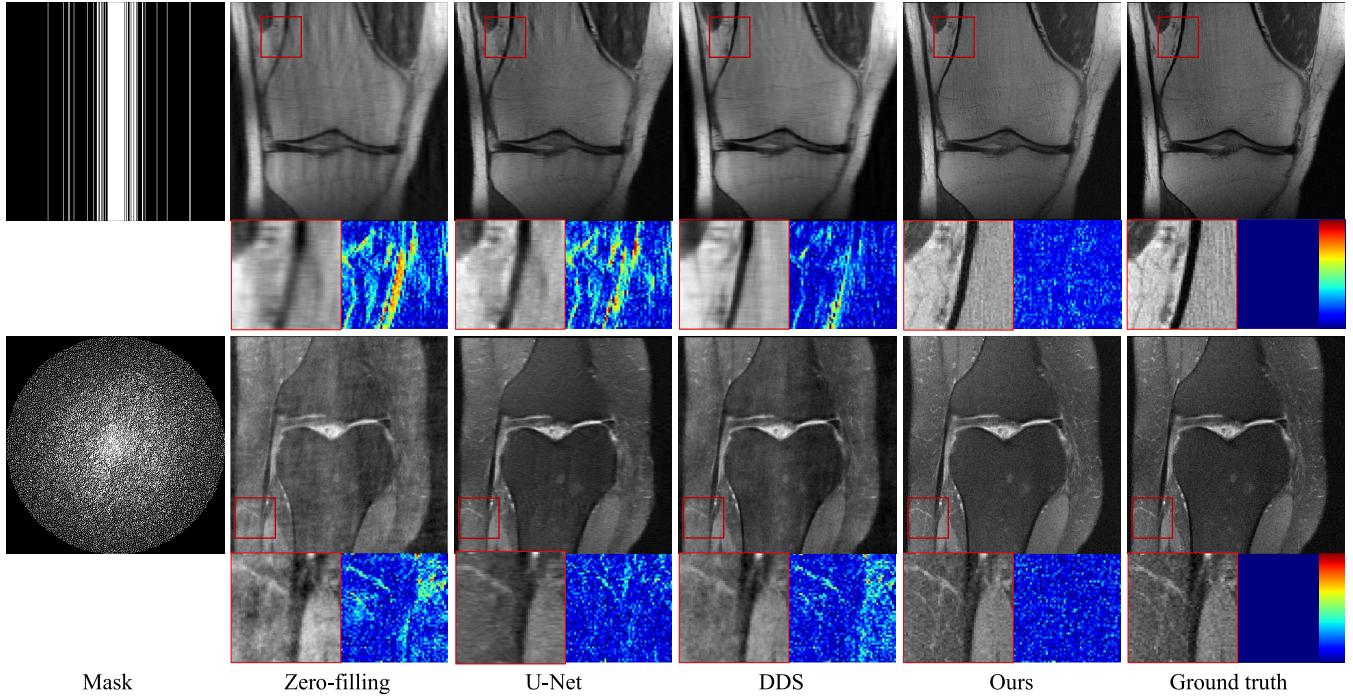


Fig. 11. Qualitative results of fast MRI reconstruction.

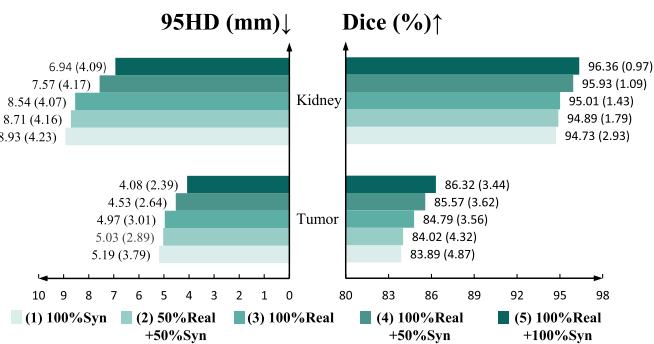


Fig. 12. Quantitative comparison on data augmentation.

minimal, with Dice score differences within 1% and aslo 95HD variations within 1 mm for both tumor and kidney segmentation.

For settings (3), (4) and (5), even when the real data is relatively sufficient (100%), the addition of synthetic data still improves the network's performance. Specifically, when we introduce additional 100% synthetic data, we observe a 1.6 mm decrease in 95HD for kidney segmentation and a 1.53% increase in Dice score for tumor segmentation.

4) Data Augmentation for Classification: In this experiment, we demonstrate the feasibility of leveraging the synthesized samples from generative model to augment the training dataset for a classification task. Different from the three downstream tasks above, in this task, we do not use the ControlNet but fine-tune our generative model on the MosMedData [63], which consists of 1,110 studies, including both those with COVID-19 related findings and those without. The experiment was designed as follows: first, we fine-tuned our model on this dataset with minor modifications, using two class labels:

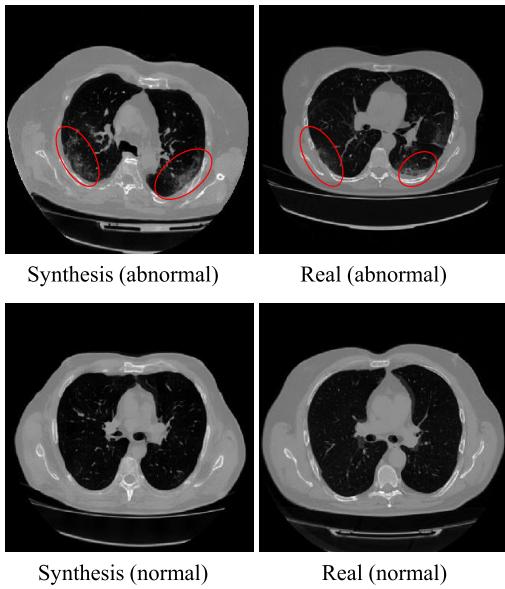


Fig. 13. Synthetic and real images, with (abnormal) and without (normal) COVID-19 related findings.

$\mathbf{c} = 0$ for studies with COVID-19 related findings (abnormal) and $\mathbf{c} = 1$ for studies without COVID-19 findings (normal). Next, we used the trained model to generate new images with corresponding class labels (200 images for each of the two classes). Finally, we combined the original training dataset with the synthetic images to train a binary-class classifier as the augmented classifier, and used the original dataset without synthetic images to train another binary-class classifier as the baseline classifier. We then evaluated the performance of both classifiers on the test set. The dataset was split into training, validation, and test sets, with the test set containing 120 images. The model and classifiers were trained and evaluated on a single NVIDIA A100 GPU with 80GB of memory. We report the quantitative results in Tab. XI. Additionally, we show some synthetic images with and without COVID-19 related findings in Fig. 13. It can be observed that our proposed method generates realistic normal and abnormal lung CT images. Moreover, the classifier trained with synthetic augmented data outperforms the baseline model, which was trained only on real images from the training set.

V. CONCLUSION

In this paper, we propose a 3D Medical Latent Diffusion (3D MedDiffusion) model to generate high-quality 3D medical images. 3D MedDiffusion achieves its performance through two key design components: 1) the Patch-Volume Autoencoder and 2) the BiFlowNet noise estimator. Additionally, it can be integrated with ControlNet, enabling efficient adaptation to a large range of downstream tasks in medical scenarios. Our experiments also demonstrate superior generative performance and strong generalizability across diverse tasks.

Although achieving promising results, our method still has limitations. First, it lacks the ability to produce images at arbitrary resolutions. Arbitrary-size generation could potentially be achieved with implicit neural encoder [64], which will be our future work. Second, our method does not incorporate age

and gender as conditioning factors for synthesizing medical images. Future work could explore the inclusion of additional conditions to enhance clinical applicability. Third, diffusion models require significant time and resources for generating high-resolution 3D medical images. We aim to further improve its time and memory efficiency in the future.

REFERENCES

- [1] M. Özbeý et al., “Unsupervised medical image translation with adversarial diffusion models,” *IEEE Trans. Med. Imag.*, vol. 42, no. 12, pp. 3524–3539, Dec. 2023.
- [2] T. Amit, T. Shahrbany, E. Nachmani, and L. Wolf, “SegDiff: Image segmentation with diffusion probabilistic models,” 2021, *arXiv:2112.00390*.
- [3] J. Liu et al., “DOLCE: A model-based probabilistic diffusion framework for limited-angle CT reconstruction,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 10464–10474.
- [4] C. Cao et al., “High-frequency space diffusion model for accelerated MRI,” *IEEE Trans. Med. Imag.*, vol. 43, no. 5, pp. 1853–1865, May 2024.
- [5] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” 2013, *arXiv:1312.6114*.
- [6] D. J. Rezende and S. Mohamed, “Variational inference with normalizing flows,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 37, 2015, pp. 1530–1538.
- [7] I. J. Goodfellow et al., “Generative adversarial networks,” *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [8] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.
- [9] T. Che, Y. Li, A. Paul Jacob, Y. Bengio, and W. Li, “Mode regularized generative adversarial networks,” 2016, *arXiv:1612.02136*.
- [10] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.
- [11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 10684–10695.
- [12] C. Saharia et al., “Palette: Image-to-image diffusion models,” in *Proc. ACM SIGGRAPH Conf.*, 2022, pp. 1–10.
- [13] J. Wang, Z. Yue, S. Zhou, K. C. K. Chan, and C. C. Loy, “Exploiting diffusion prior for real-world image super-resolution,” *Int. J. Comput. Vis.*, vol. 132, no. 12, pp. 5929–5949, Dec. 2024.
- [14] P. Friedrich, J. Wolleb, F. Bieder, A. Durrer, and P. C. Cattin, “WDM: 3D wavelet diffusion models for high-resolution medical image synthesis,” 2024, *arXiv:2402.19043*.
- [15] F. Khader et al., “Medical diffusion: Denoising diffusion probabilistic models for 3D medical image generation,” 2022, *arXiv:2211.03364*.
- [16] L. Zhu et al., “Make-a-volume: Leveraging latent diffusion models for cross-modality 3D brain MRI synthesis,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Jan. 2023, pp. 592–601.
- [17] A. Volokitin et al., “Modelling the distribution of 3D brain MRI using a 2D slice VAE,” in *Proc. Med. Image Comput. Comput. Assist. Intervent.-MICCAI*, Jan. 2020, pp. 657–666.
- [18] K. Han et al., “MedGen3D: A deep generative framework for paired 3D image and mask generation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Jan. 2023, pp. 759–769.
- [19] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.
- [20] L. Zhang and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 3836–3847.
- [21] W. Jin, M. Fatehi, K. Abhishek, M. Mallya, B. Toyota, and G. Hamarneh, “Applying artificial intelligence to glioma imaging: Advances and challenges,” 2019, *arXiv:1911.12886*.
- [22] M. D. Cirillo, D. Abramian, and A. Eklund, “Vox2Vox: 3D-GAN for brain tumour segmentation,” in *Proc. Int. MICCAI Brainlesion Workshop*, 2021, pp. 274–284.
- [23] L. Sun, J. Chen, Y. Xu, M. Gong, K. Yu, and K. Batmanghelich, “Hierarchical amortized GAN for 3D high resolution medical image synthesis,” *IEEE J. Biomed. Health Informat.*, vol. 26, no. 8, pp. 3966–3975, Aug. 2022.
- [24] G. Müller-Franzes et al., “A multimodal comparison of latent denoising diffusion probabilistic models and generative adversarial networks for medical image synthesis,” *Sci. Rep.*, vol. 13, no. 1, p. 12098, Jul. 2023.

- [25] P. A. Moghadam et al., "A morphology focused diffusion probabilistic model for synthesis of histopathology images," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 1999–2008.
- [26] P. Guo et al., "MAISI: Medical AI for synthetic imaging," 2024, *arXiv:2409.11169*.
- [27] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-LM: Training multi-billion parameter language models using model parallelism," 2019, *arXiv:1909.08053*.
- [28] P. Friedrich, Y. Frisch, and P. C. Cattin, "Deep generative models for 3D medical image synthesis," 2024, *arXiv:2410.17664*.
- [29] J. Fan, X. Cao, Q. Wang, P.-T. Yap, and D. Shen, "Adversarial learning for mono- or multi-modal registration," *Med. Image Anal.*, vol. 58, Dec. 2019, Art. no. 101545.
- [30] L. Xiang et al., "Deep-learning-based multi-modal fusion for fast MR reconstruction," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 7, pp. 2105–2114, Jul. 2018.
- [31] H. Chung, S. Lee, and J. Chul Ye, "Decomposed diffusion sampler for accelerating large-scale inverse problems," 2023, *arXiv:2303.05754*.
- [32] J. Kim and H. Park, "Adaptive latent diffusion model for 3D medical image to image translation: Multi-modal magnetic resonance imaging study," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 7589–7598.
- [33] A. Van Den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
- [34] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12873–12883.
- [35] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.
- [36] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. 33rd Int. Conf. Mach. Learn.*, Feb. 2016, pp. 1558–1566.
- [37] A. Lamb, V. Dumoulin, and A. Courville, "Discriminative regularization for generative models," 2016, *arXiv:1602.03220*.
- [38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [40] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 4195–4205.
- [41] S. Chilamkurthy et al., "Development and validation of deep learning algorithms for detection of critical findings in head CT scans," 2018, *arXiv:1803.05854*.
- [42] L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, in *Proc. 24th Int. Conf. Med. Image Comput. Comput. Assist. Intervent.*, vol. 12901, 2021, pp. 109–119.
- [43] H. Zheng et al., "Alleviating class-wise gradient imbalance for pulmonary airway segmentation," *IEEE Trans. Med. Imag.*, vol. 40, no. 9, pp. 2452–2462, Sep. 2021.
- [44] W. Yu, H. Zheng, M. Zhang, H. Zhang, J. Sun, and J. Yang, "Break: Bronchi reconstruction by geodesic transformation and skeleton embedding," in *Proc. IEEE 19th Int. Symp. Biomed. Imag. (ISBI)*, Mar. 2022, pp. 1–5.
- [45] Y. Qin et al., "AirwayNet: A voxel-connectivity aware approach for accurate airway segmentation using convolutional neural networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2019, pp. 212–220.
- [46] S. G. Armato et al., "The lung image database consortium (LIDC) and image database resource initiative (IDRI): A completed reference database of lung nodules on CT scans," *Med. Phys.*, vol. 38, no. 2, pp. 915–931, Jan. 2011.
- [47] Y. Deng et al., "CTSpine1K: A large-scale dataset for spinal vertebrae segmentation in computed tomography," 2021, *arXiv:2105.14711*.
- [48] J. Ma et al., "AbdomenCT-1K: Is abdominal organ segmentation a solved problem?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6695–6714, Oct. 2022.
- [49] C. Sudlow et al., "U.K. biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age," *PLOS Med.*, vol. 12, no. 3, Mar. 2015, Art. no. e1001779.
- [50] J. Zbontar et al., "FastMRI: An open dataset and benchmarks for accelerated MRI," 2018, *arXiv:1811.08839*.
- [51] Y. Xu et al., "MedSyn: Text-guided anatomy-aware synthesis of high-fidelity 3-D CT images," *IEEE Trans. Med. Imag.*, vol. 43, no. 10, pp. 3648–3660, Oct. 2024.
- [52] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 8162–8171.
- [53] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2017, pp. 6629–6640.
- [54] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, Jan. 2012.
- [55] S. Chen, K. Ma, and Y. Zheng, "Med3D: Transfer learning for 3D medical image analysis," 2019, *arXiv:1904.00625*.
- [56] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–27, 2008.
- [57] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [58] N. Heller et al., "The KITS19 challenge data: 300 kidney tumor cases with clinical context, CT semantic segmentations, and surgical outcomes," 2019, *arXiv:1904.00445*.
- [59] W. van Aarle et al., "Fast and flexible X-ray tomography using the ASTRA toolbox," *Opt. Exp.*, vol. 24, no. 22, pp. 25129–25147, 2016.
- [60] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. Philadelphia, PA, USA: SIAM, 2001.
- [61] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4509–4522, Sep. 2017.
- [62] F. Isensee et al., "NnU-net: Self-adapting framework for U-net-based medical image segmentation," 2018, *arXiv:1809.10486*.
- [63] S. P. Morozov et al., "MosMedData: Chest CT scans with COVID-19 related findings dataset," 2020, *arXiv:2005.06465*.
- [64] J. Kim and T.-K. Kim, "Arbitrary-scale image generation and upsampling using latent diffusion model and implicit neural decoder," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 9202–9211.