Hierarchical Amortized GAN for 3D High Resolution Medical Image Synthesis

Li Sun, *Member, IEEE*, Junxiang Chen, Yanwu Xu, Mingming Gong, Ke Yu, and Kayhan Batmanghelich, *Member, IEEE*

***Abstract*—Generative Adversarial Networks (GAN) have many potential medical imaging applications, including data augmentation, domain adaptation, and model explanation. Due to the limited memory of Graphical Processing Units (GPUs), most current 3D GAN models are trained on low-resolution medical images, these models either cannot scale to high-resolution or are prone to patchy artifacts. In this work, we propose a novel end-to-end GAN architecture that can generate high-resolution 3D images. We achieve this goal by using different configurations between training and inference. During training, we adopt a hierarchical structure that simultaneously generates a low-resolution version of the image and a randomly selected sub-volume of the high-resolution image. The hierarchical design has two advantages: First, the memory demand for training on high-resolution images is amortized among sub-volumes. Furthermore, anchoring the high-resolution sub-volumes to a single low-resolution image ensures anatomical consistency between sub-volumes. During inference, our model can directly generate full high-resolution images. We also incorporate an encoder with a similar hierarchical structure into the model to extract features from the images. Experiments on 3D thorax CT and brain MRI demonstrate that our approach outperforms state of the art in image generation. We also demonstrate clinical applications of the proposed model in data augmentation and clinical-relevant feature extraction.**

***Index Terms*—3D image synthesis, generative adversarial networks, high resolution.**

Manuscript received 28 November 2021; revised 27 March 2022; accepted 26 April 2022. Date of publication 6 May 2022; date of current version 9 August 2022. This work was supported in part by the National Institutes of Health (NIH), Bethesda, MD, USA under Grant 1R01HL141813-01, in part by the National Science Foundation (NSF), Alexandria, VA, USA under Grant 1839332 Tripod+X, SAP SE, and in part by the Pennsylvania Department of Health. The computational resources in this work was provided by Pittsburgh SuperComputing under Grant TG-ASC170024. *(Corresponding author: Kayhan Batmanghelich.)*

Li Sun, Yanwu Xu, and Ke Yu are with the School of Computing and Information, University of Pittsburgh, Pittsburgh, PA 15206 USA (e-mail: lis118@pitt.edu; yanwuxu@pitt.edu; key44@pitt.edu).

Junxiang Chen and Kayhan Batmanghelich are with the Department of Biomedical Informatics, University of Pittsburgh, Pittsburgh, PA 15206 USA (e-mail: juc91@pitt.edu; kayhan@pitt.edu).

Mingming Gong is with the School of Mathematics and Statistics, University of Melbourne, Parkville, VIC 3010, Australia (e-mail: mingming.gong@unimelb.edu.au).

This article has supplementary downloadable material available at https://doi.org/10.1109/JBHI.2022.3172976, provided by the authors.

Digital Object Identifier 10.1109/JBHI.2022.3172976

I. Introduction

GENERATIVE Adversarial Networks (GANs) have succeeded in generating realistic-looking natural images [1], [2]. It has shown potential in medical imaging for augmentation [3], [4], image reconstruction [5] and image-to-image translation [6], [7]. The prevalence of 3D images in the radiology domain renders the real-world application of GANs in the medical domain even more challenging than the natural image domain. In this paper, we propose an efficient method for generating and extracting features from high-resolution volumetric images.

The training procedure of GANs corresponds to a min-max game between two players: a generator and a discriminator. While the generator aims to generate realistic-looking images, the discriminator aims to defeat the generator by recognizing real from the fake (generated) images. When the field of view (FOV) is the same, a higher resolution is equivalent to more voxels. In this way, we use "high-resolution image" and "large-size image" interchangeably in the paper. In clinical application, radiologists rely on high-resolution CT to make accurate diagnose decisions [8]. While there are previous works that propose to use 3D GAN for diverse medical applications [9], [10], the generated images are limited to the small size of $128 \times 128 \times 128$ or below, due to insufficient memory during training.

In this paper, we introduce a Hierarchical Amortized GAN (HA-GAN) to bridge the gap. Our model adopts different configurations between training and inference phases. In the training phase, we simultaneously generate a low-resolution image and a randomly selected sub-volume of the high-resolution image. Generating sub-volumes amortizes the memory cost of the high-resolution image and keeps local details of the 3D image. Furthermore, the low-resolution image ensures anatomical consistency and the global structure of the generated images. We train the model in an end-to-end fashion while retaining memory efficiency. The gradients of the parameters, which are the memory bottleneck, are needed only during training. Hence, sub-volume selection is no longer needed and the entire high-resolution volume can be generated during inference. In addition, we implement an encoder in a similar fashion. The encoder enables us to extract features from a given image and prevents the model from mode collapse. We test HA-GAN on thorax CT and brain MRI datasets. Experiments demonstrate that our approach outperforms baselines in image generation. We also present two clinical applications with proposed HA-GAN, including data augmentation for supervised learning and clinical-relevant feature extraction. Our code is publicly available at https://github.com/batmanlab/HA-GAN

In summary, we make the following contributions:

1) We introduce a novel end-to-end HA-GAN architecture that can generate high-resolution volumetric images while being memory efficient.
2) We incorporate a memory-efficient encoder with a similar structure, enabling clinical-relevant feature extraction from high-resolution 3D images. We show that the encoder improves generation quality.
3) We discover that moving along specific directions in latent space results in explainable anatomical variations in generated images.
4) We evaluate our method by extensive experiments on different image modalities as well as different anatomy. The HA-GAN offers significant quantitative and qualitative improvements over the state-of-the-art.

II. Related Work

In the following, we review the works related to GANs for medical images, memory-efficient 3D GAN and representation learning in generative models.

A. GANs for Medical Imaging

In recent years, researchers have developed GAN-based models for medical images. These models are applied to solve various problems, including image synthesis [11], data augmentation [12], modality/style transformation [13], and model explanation [14]. However, most of these methods concentrate on generating 2D medical images. In this paper, we focus on solving a more challenging problem, i.e., generating 3D images.

With the prevalence of 3D imaging in medical applications, 3D GAN models have become a popular research topic. Shan *et al.* [15] proposed a 3D conditional GAN model for low-dose CT denoising. Kudo *et al.* [16] proposed a 3D GAN model for CT image super-resolution. Jin *et al.* [17] propose an auto-encoding GAN for generating 3D brain MRI images. Cirillo *et al.* [9] proposed to use a 3D model conditioned on multi-channel 3D Brain MR images to generate tumor masks for segmentation. While these methods can generate realistic-looking 3D MRI or CT images, the generated images are limited to the small size of $128 \times 128 \times 128$ or below, due to insufficient memory during training. In contrast, our HA-GAN is a memory-efficient model and can generate 3D images with a size of $256 \times 256 \times 256$.

B. Memory-Efficient GANs

Some works are proposed to reduce the memory demand of high-resolution 3D image generation. In order to address the memory challenge, some works adopt slice-wise [7] or patch-wise [10] generation approach. Unfortunately, these methods may introduce artifacts at the intersection between patches/slices because they are generated independently. To remedy this problem, Uzunova *et al.* [18] propose a multi-scale approach that uses a GAN model to generate a low-resolution version of the image first. An additional GAN model is used to generate higher resolution patches of images conditioned on the previously generated patches of lower resolution images. However, this method is still patch-based; the generation of local patches is unaware of the global structure, potentially leading to spatial inconsistency. In addition, the model is not trained in an end-to-end manner, which makes it challenging to incorporate an encoder that learns the latent representations for the entire images. In comparison, our proposed HA-GAN is global structure-aware and can be trained end-to-end. This allows HA-GAN to be associated with an encoder.

C. Representation Learning in Generative Models

Several existing generative models are fused with an encoder [2], [19], [20], which learns meaningful representations for images. These methods are based on the belief that a good generative model that reconstructs realistic data will automatically learn a meaningful representation of it [21]. A generative model with an encoder can be regarded as a compression algorithm [22]. Hence, the model is less likely to suffer from mode collapse because the decoder is required to reconstruct all samples in the dataset, which is impossible if mode collapse happens such that only limited varieties of samples are generated [2]. Variational autoencoder (VAE) [19] uses an encoder to compress data into a latent space, and a decoder is used to reconstruct the data using the encoded representation. BiGAN [20] learns a bidirectional mapping between data space and latent space. α-GAN [2] introduces not only an encoder to the GAN model, but also learns a disentangled representation by implementing a code discriminator, which forces the distribution of the code to be indistinguishable from that of random noise. Variational auto-encoder GAN (VAE-GAN) [23] adds an adversarial loss to the variational evidence lower bound objective. Despite their success, the methods mentioned above can analyze 2D images or low-resolution 3D images, which are less memory intensive for training an encoder. In contrast, our proposed HA-GAN is memory efficient and can be used to encode and generate high-resolution 3D images during inference.

D. Our Previous Work

Sun *et al.* [24] first proposed to utilize hierarchical amortized GAN for high resolution 3D medical image generation. The current work presents several extensions compared to the preliminary version: 1) We incorporate a memory-efficient encoder into our model, enabling clinical-relevant feature extraction from high-resolution 3D images. We also show that the encoder improves generation quality. 2) We perform two new clinical applications, including characterizing the severity of COPD, and data augmentation for supervised learning. 3) We discover that moving along specific directions in latent space results in explainable anatomical variations in generated images. 4) We perform cross-validation evaluation and statistical tests for comparison of generated image quality with baseline methods to improve the adequacy of performance evaluation. We also conduct ablation studies to validate the contribution of proposed components.

III. Method

We first review Generative Adversarial Networks (GANs) in Section III-A. Then, we introduce our method in Section III-B, followed by the introduction of the encoder in Section III-C.

TABLE I
IMPORTANT NOTATIONS IN THIS PAPER

Models	
$G^A(\cdot)$	The common block of the generator.
$G^L(\cdot)$	The low-resolution block of the generator.
$G^H(\cdot)$	The high-resolution block of the generator.
$D^H(\cdot)$	The discriminator for high-resolution images.
$D^L(\cdot)$	The discriminator for low-resolution images.
$E^H(\cdot)$	The high-resolution block of the encoder.
$E^G(\cdot)$	The ground block of the encoder.
Functions	
$S^H(\cdot,\cdot)$	The high-resolution sub-volume selector.
$S^L(\cdot,\cdot)$	The low-resolution sub-volume selector.
Variables	
Z	Latent representations.
$\widehat{Z}$	Reconstructed latent representations.
c	GOLD score.
r	The index of the tarting slice for sub-volume selection.
X^H	The real high-resolution image.
X^L	The real low-resolution image.
$\widehat{X}^H$	The generated high-resolution image.
$\widehat{X}^H_r$	The generated high-resolution sub-volume starting at slice r.
$\widehat{X}^L$	The generated low-resolution image.
A	Intermediate feature maps for the whole image
A_r	Intermediate feature maps for the sub-volume starting at slice r
$\widehat{A}$	Reconstructed intermediate feature maps for the whole image.
$\widehat{A}_v$	Reconstructed intermediate feature maps for the v-th sub-volume.
$\{T_v\}_{v=1}^V$	The indices of the starting slices for a partition for X^H.

We conclude this section with the optimization scheme in Section III-D and the implementation details in Section III-E. The notations used are summarized in Table I.

A. Background

Generative Adversarial Networks (GANs) [1] is widely used to generate realistic-looking images. The training procedure of GANs corresponds to a two-player game that involves a generator G and a discriminator D. In the game, while G aims to generate realistic-looking images, D tries to discriminate real images from the images synthesized by G. The D and G compete with each other. Let P_X denote the underlying data distribution, and P_Z denote the distribution of the random noise Z. Then the objective of GAN is formulated as below:

$$\min_G \max_D \mathop{\mathbb{E}}_{X\sim P_X}[\log D(X)] + \mathop{\mathbb{E}}_{Z\sim P_Z}[\log(1 - D(G(Z)))]. \quad (1)$$

B. The Hierarchical Structure

Generator: Our generator has two branches that generate the low-resolution image $\widehat{X}^L$ and a randomly selected sub-volume of the high-resolution image $\widehat{X}^H_r$, where r represents the index for the starting slice of the sub-volume. The two branches share initial layers G^A and after they branch off:

$$\widehat{X}^L = G^L(\underbrace{G^A(Z)}_{A}), \quad (2)$$

$$\widehat{X}^H_r = G^H(\underbrace{S^L(G^A(Z); r)}_{A_r}), \quad (3)$$

where $G^A(\cdot)$, $G^L(\cdot)$ and $G^H(\cdot)$ denote the common, low-resolution and high-resolution blocks of the generator, respectively. $S^L(\cdot, r)$ is a selector function that returns the sub-volume of input image starting at slice r, where the superscript L indicates that the selection is made at low resolution. The output of this function is fed into $G^H(\cdot)$, which lifts the input to the high resolution. We use A and A_r as short-hand notation for $G^A(Z)$ and $S^L(G^A(Z); r)$, respectively. We let $Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be the input random noise vector. We let r be the randomly selected index for the starting slice that is drawn from a uniform distribution, denoted as $r \sim \mathcal{U}$; i.e., each slice is selected with the same probability. Therefore, the randomly selected sub-volumes can be overlapping, which can better cover the junctions between sub-volumes than non-overlapping sub-volume selection. The schematic of the proposed method is shown in Fig. 1. Note that $\widehat{X}^H_r$ depends on a corresponding sub-volume of A, which is A_r. Therefore, we feed A_r rather than complete A into G^H during training, making the model memory-efficient.

Discriminator: Similarly, we define two discriminators D^H and D^L to distinguish a real high-resolution sub-volume X^H_r and a low-resolution image X^L from the fake ones, respectively. D^H makes sure that the local details in the high-resolution sub-volume look realistic. At the same time, D^L ensures the proper global structure is preserved. Since we feed a sub-volumes $S^H(X^H; r)$ rather than the entire image X^H into D^H, the memory cost of the model is reduced. The location of the sub-volume r is also fed into D^H to help it distinguish sub-volumes from different locations.

There are two GAN losses $\mathcal{L}^H_{GAN}$ and $\mathcal{L}^L_{GAN}$ for high and low resolutions respectively:

$$\mathcal{L}^H_{GAN}(G^A, G^H, D^H) = \min_{G^H, G^A} \max_{D^H} \mathop{\mathbb{E}}_{r\sim U} \Big[\mathop{\mathbb{E}}_{X\sim P_X}[\log D^H(S^H(X^H; r), r)] + \mathop{\mathbb{E}}_{Z\sim P_Z}[\log(1 - D^H(\widehat{X}^H_r, r)] \Big], \quad (4)$$

$$\mathcal{L}^L_{GAN}(G^L, G^A, D^L) = \min_{G^L, G^A} \max_{D^L} \mathop{\mathbb{E}}_{X\sim P_X}[\log D^L(X^L)] + \mathop{\mathbb{E}}_{z\sim P_Z}[\log(1 - D^L(\widehat{X}^L)]. \quad (5)$$

Note that the sampler $S^H(\cdot; r)$ in (3) and $S^L(\cdot; r)$ in (4) are synchronized, such that r corresponds to the indices for the same percentile of slices in the high- and low-resolution.

Inference: The memory space needed to store gradient is the main bottleneck for 3D GANs models; however, the gradient is not needed during inference. Therefore, we can directly generate the high-resolution image by feeding Z into G^A and G^H sequentially, *i.e.*, $\widehat{X}^H(Z) = G^H(G^A(Z)))$. Note that to generate the entire image during inference, we directly feed the complete feature maps $A = G^A(Z)$ rather than its sub-volume A_r into the convolutional network G^H.

C. Incorporating the Encoder

We also adopt a hierarchical structure for the encoder, by defining two encoders $E^H(\cdot)$ and $E^G(\cdot)$ encoding

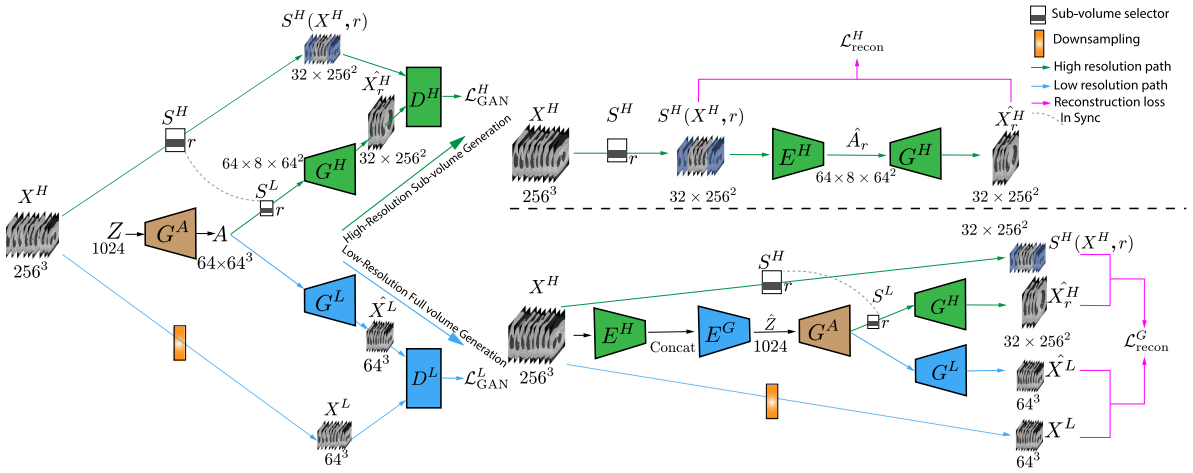


Fig. 1. **Left:** The architecture of HA-GAN (encoder is hidden here to improve clarity). At the training time, instead of directly generating high-resolution full volume, our generator contains two branches for high-resolution sub-volume and low-resolution full volume generation, respectively. The two branches share the common block G^A. A sub-volume selector is used to select a part of the intermediate feature for the sub-volume generation. **Right:** The schematic of the hierarchical encoder trained with two reconstruction losses, one on the high-resolution sub-volume level (upper right) and another one on the low-resolution full volume level (lower right). The meanings of the notations used can be found in Table I. The model adopts 3D architecture with details presented in Supplementary Material.

the high-resolution sub-volume and the entire image respectively. We partition the high-resolution image X^H into a set of V *non-overlapping* sub-volumes, *i.e.*, $X^H = \texttt{concat}(\{S^H(X^H, T_v)\}_{v=1}^V)$, where $\texttt{concat}$ represent concatenation, $S^H(\cdot)$ represents the selector function that returns a sub-volume of a high-resolution image, and T_v represents the corresponding starting indices for the non-overlapping partition.

We use $\widehat{A}_v$ to denote the sub-volume-level feature maps for the v-th sub-volume, i.e., $\widehat{A}_v = E^H(S^H(X^H; T_v))$. To generate the image-level representation $\widehat{Z}$, we first summarize all sub-volume representation for the image through concatenation, such that $\widehat{A} = \texttt{concat}(\{A_v\}_{v=1}^V)$. Then we feed $\widehat{A}$ into the encoder $E^G(\cdot)$ to generate the image-level representation $\widehat{Z}$, i.e., $\widehat{Z} = E^G(\widehat{A})$

In order to obtain optimal E^H and E^G, we introduce the following objective functions:

$$\mathcal{L}^H_{recon}(E^H) = \min_{E^H} \mathop{\mathbb{E}}_{X\sim P_X, r\in U} \left\| S^H(X^H; r) - G^H(\widehat{A}_r) \right\|_1, \tag{6}$$

$$\mathcal{L}^G_{recon}(E^G) = \min_{E^G} \mathop{\mathbb{E}}_{X\sim P_X} \left[\left\| X^L - G^L(G^A(\widehat{Z})) \right\|_1 + \mathop{\mathbb{E}}_{r\sim U} \left[\left\| S^H(X^H; r) - G^H(S^L(G^A(\widehat{Z}); r)) \right\|_1 \right] \right]. \tag{7}$$

Equation (6) ensures a randomly selected high-resolution sub-volume $S^H(X^H; r)$ can be reconstructed. (7) enforces both the low-resolution image X^L and a random selected $S^H(X^H; r)$ can be reconstructed given $\widehat{Z}$. Note that in (6), the sub-volume is reconstructed from the intermediate feature maps $\widehat{A}_v$; while in the second term in (7), the sub-volume is reconstructed from the latent representations $\widehat{Z}$. In these equations, we use ℓ_1 loss for reconstruction because it tends to generate sharper result compared to ℓ_2 loss [25]. The structure of the encoders are illustrated in Fig. 1.

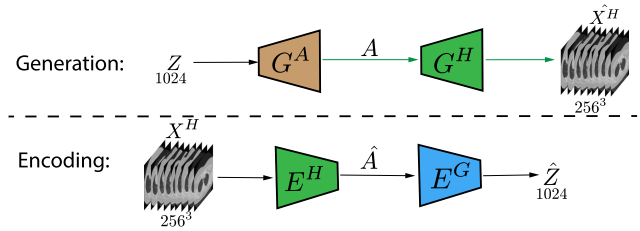


Fig. 2. Inference with the hierarchical generator and encoder. Since the memory demand is lower at inference time, we directly forward input through the high-resolution branch for full image generation and encoding.

When optimizing for (6), we only update E^H while keeping all other parameters fixed. Similarly, when optimizing for (7), we only update E^G. We empirically find that this optimization strategy is memory-efficient and leads to better performance.

Inference: In the inference phase, we can get the latent code $\widehat{Z}$ by feeding the sub-volumes of X^H into E^H, concatenating the output sub-volume feature maps into $\widehat{A}$ and then feeding the results into E^G, *i.e.*, $\widehat{Z} = E^G(\texttt{concat}(\{E^H(S^H(X^H; T_v))\}_{v=1}^V))$. The idea is illustrated at the bottom of Fig. 2.

D. Overall Model

The model is trained in an end-to-end fashion. The overall loss function is defined as:

$$\mathcal{L} = \mathcal{L}^H_{GAN}(G^H, G^A, D^H) + \mathcal{L}^L_{GAN}(G^L, G^A, D^L) + \lambda_1 \mathcal{L}^H_{recon}(E^H) + \lambda_2 \mathcal{L}^G_{recon}(E^G), \tag{8}$$

where λ_1 and λ_2 control the trade-off between the GANs losses and the reconstruction losses. The optimizations for generator

(G^H, G^L and G^A), discriminator (D^H, D^L), and encoder (E^H, E^G) are altered per iteration.

During training, we sample noise from Gaussian distribution and pass it through the generator to create randomly synthesized images for minimizing the adversarial loss. We also sample real images and pass it through the encoder, followed by the generator to create reconstructed images for minimizing the reconstruction loss. Our overall optimization balances between the losses to learn parameters for the encoder, generator, and discriminator in end-to-end training.

E. Implementation Details

We train the proposed HA-GAN for 80000 iterations, the training and validation curves can be found in Supplementary Material. We let the learning rate for generator, encoder, and discriminator be 1×10^{-4}, 1×10^{-4}, and 4×10^{-4}, respectively. We also set $\beta_1 = 0$ and $\beta_2 = 0.999$ for the Adam optimizer. The batch size is set as 4. We let the size of the X^L be 64^3. The size of the randomly selected sub-volume $S^H(X^H; r)$ is defined to be 32×256^2, where r is randomly selected on the batch level. We let feature maps A have 64 channels with a size of 64^3. The dimension of the latent variable Z is chosen to be 1,024. The trade-off hyper-parameters λ_1 and λ_2 are set to be 5. The experiments are performed on two NVIDIA Titan Xp GPUs, each with 12 GB GPU memory. The detailed architecture can be found in Supplementary Material.

IV. Experiments

We evaluate the proposed model's performance in image synthesis, and demonstrate two clinical applications with HA-GAN: data augmentation and clinical-relevant feature extraction. We also explore the semantic meaning of the latent variable. We perform 5-fold cross-validation for the image synthesis experiments. We compare our method with baseline methods, including WGAN [26], VAE-GAN [23], α-GAN [27], Progressive GAN [28], 3D StyleGAN 2 [29] and CCE-GAN [30].

A. Datasets

The experiments are conducted on two large-scale 3D datasets, including the COPDGene dataset [31] and the GSP dataset [32]. Both are publicly available and details about image acquisition are presented in Supplementary Material.

COPDGene Dataset: We use 3D thorax computerized tomography (CT) images of 9,276 subjects from COPDGene dataset in our study. Only full inspiration scans are used in our study. We trim blank axial slices with all-zero values and resize the images to 256^3. The Hounsfield units of the CT images have been calibrated and air density correction has been applied. The Hounsfield Units (HU) are mapped to the intensity window of $[-1024, 600]$ and normalized to $[-1, 1]$.

GSP Dataset: We use 3D Brain magnetic resonance images (MRIs) of 3,538 subjects from the Brain Genomics Superstruct Project (GSP) [32] in our experiments. The FreeSurfer package [33] is used to remove the non-brain region in the images, bias-field correction, intensity normalization, affine registration to Talairach space, and resampling to 1 mm^3 isotropic resolution. We trim the blank axial slices with all-zero values and rescale the images into 256^3. The intensity value is clipped at top 0.1% quantile to remove outliers, and then normalized into $[-1, 1]$.

B. Image Synthesis

We examine whether the synthetic images are realistic-looking quantitatively and qualitatively, where synthetic images are generated by feeding random noise into the generator.

1) Quantitative Evaluation: If the synthetic images are realistic-looking, then the synthetic images' distribution should be indistinguishable from that of the real images. Therefore, we can quantitatively evaluate the quality of the synthetic images by Fréchet Inception Distance (FID) [34], Maximum Mean Discrepancy (MMD) [35] and Inception Score (IS) [36]. Lower values of FID/MMD and higher values of IS indicate that the distributions of generated images are closer to real ones, implying more realistic-looking synthetic images. We evaluate the synthesis quality at two resolutions: 128^3 and 256^3. Due to memory limitations, the baseline models can only be trained with the size of 128^3 at most. To make a fair comparison with our model (HA-GAN), we apply trilinear interpolation to upsample the synthetic images of baseline models to 256^3. We adopt a 3D ResNet model pre-trained on 3D medical images [37] to extract features for computing FID and MMD. Note the scale of FID relies on the feature extraction model. Thus our FID values are not comparable to FID value calculated on 2D images, which is based on feature extracted using model pre-trained on ImageNet. For the IS scores, following the practice of [29], we measure the Inception Scores on the middle slices on axial, coronal, and sagittal planes of the generated 3D images and report averaged performance. As shown in Table II and Table III, HA-GAN achieves lower FID and MMD as well as higher IS than the baselines, which implies that HA-GAN generates more realistic images. We found that at the resolution of 128^3, HA-GAN still outperforms the baseline models, but the lead has been smaller compared with the result at the resolution of 256^3. In addition, we performed statistical tests on the evaluation results at 256^3 resolution between methods. More specifically, we performed two-sample t-tests (one-tailed) between HA-GAN and each of the baseline methods. At a significance level of 0.05, HA-GAN achieves significantly higher performance than baseline methods for both datasets.

2) Ablation Study: We perform three ablation studies to validate the contribution of each of the proposed components. The experiments are performed at 256^3 resolution. Shown in Table IV, we found that adding a low-resolution branch can help improve results, since it can help the model learn the global structure. Adding an encoder can also help improve performance, since it can help stabilize the training. For the deterministic r experiments, we make the sub-volume selector to use a set of deterministic values of r (equal interval between them) rather than the randomly sampled r currently used. From the results, we can see that randomly sampled r outperforms deterministic r.

3) Qualitative Evaluation: To qualitatively analyze the results, we show some samples of synthetic images in Fig. 3. The figure illustrates that HA-GAN generates sharper images than the baselines.

TABLE II
EVALUATION FOR IMAGE SYNTHESIS ON COPDGENE DATASET

Resolution	128^3			256^3		
	FID↓	MMD↓	IS↑	FID↓	MMD↓	IS↑
WGAN	$0.012_{\pm.011}$	$0.092_{\pm.059}$	$1.99_{\pm.07}$	$0.161_{\pm.044}$	$0.471_{\pm.110}$	$1.97_{\pm.05}$
VAE-GAN	$0.139_{\pm.002}$	$1.065_{\pm.008}$	$1.19_{\pm.03}$	$0.328_{\pm.007}$	$1.028_{\pm.008}$	$1.18_{\pm.03}$
α-GAN	$0.010_{\pm.004}$	$0.089_{\pm.056}$	$1.89_{\pm.04}$	$0.043_{\pm.094}$	$0.323_{\pm.080}$	$1.96_{\pm.03}$
Progressive GAN	$0.015_{\pm.007}$	$0.150_{\pm.072}$	$1.75_{\pm.11}$	$0.107_{\pm.037}$	$0.287_{\pm.123}$	$1.76_{\pm.11}$
StyleGAN 2	$0.011_{\pm.001}$	$0.071_{\pm.002}$	$2.03_{\pm.02}$	$0.081_{\pm.003}$	$0.225_{\pm.008}$	$2.06_{\pm.01}$
CCE-GAN	$0.010_{\pm.004}$	$0.087_{\pm.039}$	$1.97_{\pm.05}$	$0.074_{\pm.038}$	$0.252_{\pm.116}$	$1.95_{\pm.04}$
HA-GAN	$\mathbf{0.005_{\pm.003}}$	$\mathbf{0.038_{\pm.020}}$	$\mathbf{2.05_{\pm.05}}$	$\mathbf{0.008_{\pm.003}}$	$\mathbf{0.022_{\pm.010}}$	$\mathbf{2.09_{\pm.06}}$

TABLE III
EVALUATION FOR IMAGE SYNTHESIS ON GSP DATASET

Resolution	128^3			256^3		
	FID↓	MMD↓	IS↑	FID↓	MMD↓	IS↑
WGAN	$0.006_{\pm.002}$	$0.406_{\pm.143}$	$1.37_{\pm.02}$	$0.025_{\pm.013}$	$0.328_{\pm.139}$	$1.43_{\pm.03}$
VAE-GAN	$0.075_{\pm.004}$	$0.667_{\pm.026}$	$1.03_{\pm.01}$	$0.635_{\pm.040}$	$0.702_{\pm.028}$	$1.06_{\pm.06}$
α-GAN	$0.010_{\pm.007}$	$0.606_{\pm.204}$	$1.39_{\pm.03}$	$0.029_{\pm.016}$	$0.428_{\pm.141}$	$1.34_{\pm.08}$
Progressive GAN	$0.017_{\pm.008}$	$0.818_{\pm.217}$	$1.25_{\pm.10}$	$0.127_{\pm.055}$	$1.041_{\pm.239}$	$1.25_{\pm.10}$
StyleGAN 2	$0.014_{\pm.001}$	$0.369_{\pm.175}$	$1.26_{\pm.01}$	$0.048_{\pm.001}$	$0.370_{\pm.020}$	$1.32_{\pm.01}$
CCE-GAN	$0.005_{\pm.004}$	$0.301_{\pm.147}$	$1.38_{\pm.02}$	$0.030_{\pm.011}$	$0.411_{\pm.106}$	$1.41_{\pm.04}$
HA-GAN	$\mathbf{0.002_{\pm.001}}$	$\mathbf{0.129_{\pm.026}}$	$\mathbf{1.41_{\pm.02}}$	$\mathbf{0.004_{\pm.001}}$	$\mathbf{0.086_{\pm.029}}$	$\mathbf{1.50_{\pm.03}}$

TABLE IV
RESULTS OF ABLATION STUDY

Dataset	COPDGene (Lung)		GSP (Brain)	
	FID↓	MMD↓	FID↓	MMD↓
HA-GAN w/o Low-resolution branch	$0.030_{\pm.018}$	$0.071_{\pm.039}$	$0.118_{\pm.078}$	$0.876_{\pm.182}$
HA-GAN w/o Encoder	$0.010_{\pm.003}$	$0.034_{\pm.006}$	$0.006_{\pm.003}$	$0.099_{\pm.028}$
HA-GAN w/ Deterministic r	$0.014_{\pm.003}$	$0.035_{\pm.007}$	$0.061_{\pm.016}$	$0.612_{\pm.157}$
HA-GAN	$\mathbf{0.008_{\pm.003}}$	$\mathbf{0.022_{\pm.010}}$	$\mathbf{0.004_{\pm.001}}$	$\mathbf{0.086_{\pm.029}}$

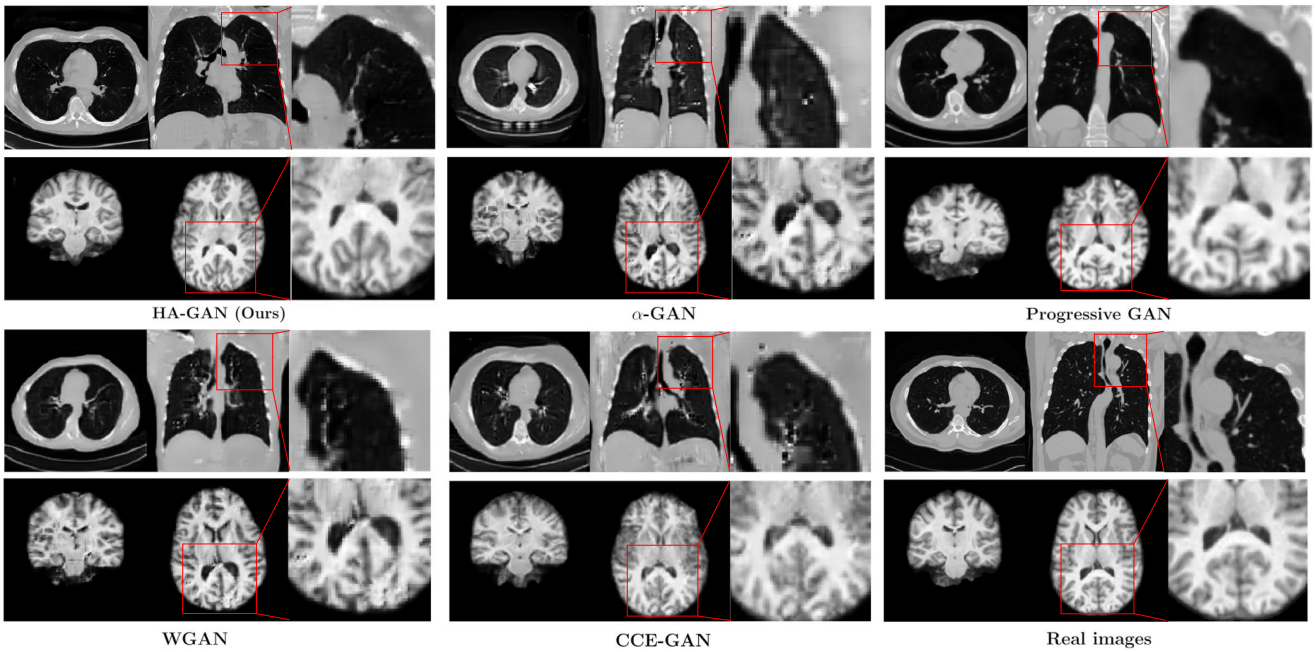
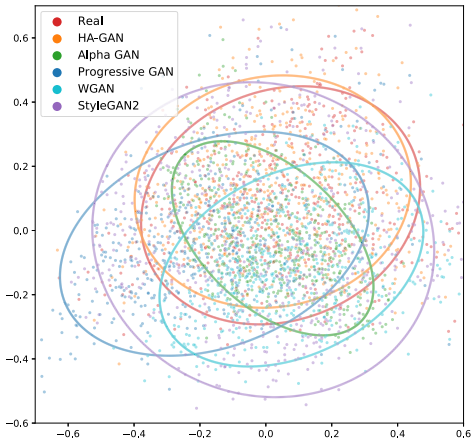
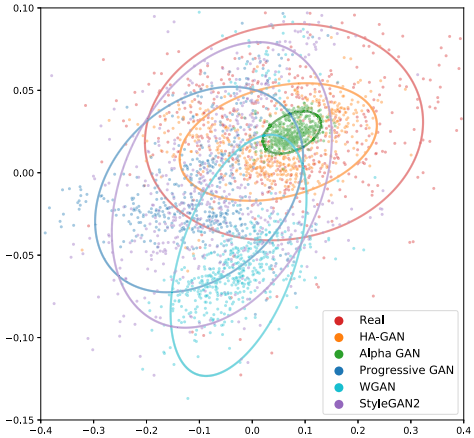


Fig. 3. *Randomly generated* images by different models and the real images. The figure illustrates that HA-GAN generates sharper images than the baselines.



(a) MDS visualization on COPDGene dataset.



(b) MDS visualization on GSP dataset.

Fig. 4. Comparison of the embedding of different models. We embed the features extracted from synthesized images into 2-dimensional space with MDS. The ellipses are fitted to scatters of each model for better visualization. The figures show that the embedding region of HA-GAN has the most overlapping with real images, compared to the baselines.

To examine the diversity and authenticity of generated images, we embed the synthetic and real images into the latent space. If the synthetic images are indistinguishable from the real images, then we expect that the synthetic and real images occupy the same region in the embedding space. Following the practice of [27], we first use a pretrained 3D medical ResNet model [37] to extract features for 512 synthetic images by each method. As a reference, we also extract features for the real image samples using the same ResNet model. Then we conduct MDS to embed the exacted features into 2-dimensional space for both COPDGene and GSP datasets. The results are visualized in Fig. 4(a) and 4(b), respectively. To avoid cluttering dots, we only visualize four representative baseline methods. In both figures, we fit an ellipse for the embedding of each model with the least square. In the figures, we observe that synthetic images by HA-GAN better overlap with real images, compared with the baselines. This implies that HA-GAN generates more realistic-looking images than the baselines.

C. Data Augmentation for Supervised Learning

In this experiment, we used the synthesized samples from HA-GAN to augment the training dataset for a supervised learning task. Previous work [38] has shown that GAN-generated samples improve the diversity of the training dataset, resulting in a better discriminative performance of the classifier. Motivated by their results, we designed our experiment with the following three steps: First, we extended our HA-GAN architecture to enable conditional image generation and trained a class-conditional variant of HA-GAN. Next, we used trained HA-GAN to generate new images with class labels. Finally, we combined the original training dataset and GAN-generated images to train a multi-class classifier, and evaluate the performance on the test set. We demonstrate our experiment on the COPDGene dataset using the GOLD score as a multi-class label. The GOLD score is a 5-class categorical variable ranging from 0–4.

We made two modifications to the original HA-GAN architecture to enable class-conditional image generation: 1) We updated the generator module $G^A(X;c)$ to take a one-hot code $c \sim p_c$ as input, along with latent variable $Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. c represents the target class for the conditional image generation. 2) We updated the discriminator to output two probability distributions, one over the binary real/fake classification (same as original HA-GAN), and another over the multi-class classification of class labels $P(C|X)$. Thus, the discriminator also acts as an auxiliary classifier for the class labels [39]. A schematic of the modified model can be found in Supplementary Material. In addition, two new terms are added to the original HA-GAN loss function for conditional generation:

$$\begin{aligned}\mathcal{L}^H_{class}(G^H, G^A, D^H) &= \mathbb{E}[\log P(C = c|X^H_r)] \\ &\quad + \mathbb{E}[\log P(C = c|\widehat{X}^H_r)] \\ \mathcal{L}^L_{class}(G^L, G^A, D^L) &= \mathbb{E}[\log P(C = c|X^L)] \\ &\quad + \mathbb{E}[\log P(C = c|\widehat{X}^L)] \end{aligned} \tag{9}$$

For comparison, we trained a class-conditional variant of α-GAN on COPDGene dataset. The same two modifications discussed above are incorporated into the original α-GAN model for conditional generation. We use a 3D CNN (implementation details are included in Supplementary Material Table VIII) as the classification model. We randomly sampled 80% of subjects as training set and the rest are used as test set. We use an image size of 128^3 for this experiment. We divided 80% of the subjects into training set, while the rest are included in a test set. For creating the augmented training set, we combine randomly generated images from class-conditioned GAN (20%) with the real images in the training set (80%). The proportion of different GOLD classes for generated images is the same as the original dataset. We train two classifiers on the original training set and the GAN-augmented training set for 20 epochs respectively, and evaluated their performance on a held-out test set of real images.

Table V shows the results on COPDGene dataset. Classifier trained with GAN augmented data performed better than the baseline model which trains on training set only consisted of real images. Augmentation with HA-GAN can further improve performance compared to α-GAN.

TABLE V
EVALUATION RESULT FOR GAN-BASED DATA AUGMENTATION

Method	Accuracy(%)
Baseline	59.7
Augmented with α-GAN	61.7
Augmented with HA-GAN	**62.9**

TABLE VI
R^2 FOR PREDICTING CLINICAL-RELEVANT MEASUREMENTS

Method	log FEV1pp	log FEV_1/FVC	log %Emphysema
VAE-GAN	0.215	0.315	0.375
α-GAN	0.512	0.622	0.738
HA-GAN	**0.555**	**0.657**	**0.746**

We do not include the results of WGAN and Progressive GAN, because they do not incorporate an encoder.

D. Clinical-Relevant Feature Extraction

In this section, we evaluate the encoded latent variables from real images to predict clinical-relevant measurements. This task evaluates how much information about the disease severity is preserved in the encoded latent features.

We select two respiratory measurements and one CT-based measurement of emphysema to measure disease severity. For respiratory measurements, we use percent predicted values of Forced Expiratory Volume in one second (FEV1pp) and its ratio with Forced vital capacity (FVC) (FEV_1/FVC). Given extracted features, we train a Ridge regression model with $\lambda = 1 \times 10^{-4}$ to predict the *logarithm* of each of the measurements. We report the R^2 scores on held-out test data. Table VI shows that HA-GAN achieves higher R^2 than the baselines. The results imply that HA-GAN preserves more information about the disease severity than baselines.

E. Exploring the Latent Space

This section investigates whether change along a certain direction in the latent space corresponds to semantic meanings. We segment the lung regions in the thorax CT images using Chest Image Platform (CIP) [40], and segment the bone tissues via thresholding. The detailed thresholding criteria can be found in Supplementary Material. Next, we train linear regression models that predict the total volume of the different tissues/regions with the encoded latent representations Z for each image, optimizing with least square. The learned parameter vector for each class represents the latent direction. Then, we manipulate the latent variable along the direction corresponding to the learned parameters of linear models and generate the images by feeding the resulted latent representations into the generator. More specifically, first a reference latent variable is randomly sampled, then the latent variable is moved along the latent direction learned until the target volume is reached, which is predicted by the linear regression model. As shown in Fig. 5, for thorax CT images, we identify directions in latent space corresponding to the volume of lung and bone respectively. When we go along these directions in latent space, we can observe the change of volumes for these tissues.

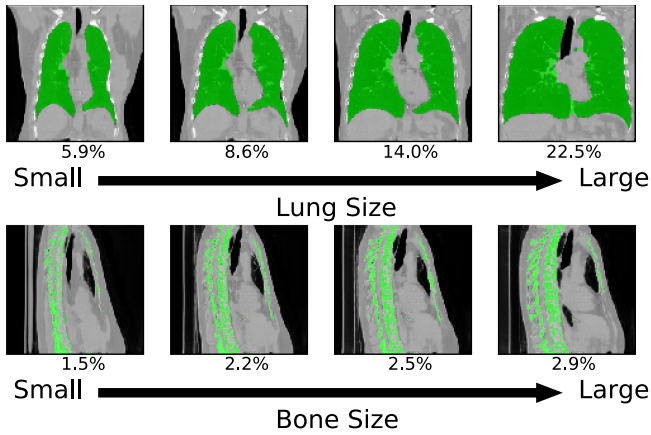


Fig. 5. Latent space exploration on thorax CT images. The figure reports synthetic images generated by changing the latent code in two different directions, corresponding to the lung and bone volume respectively. The number shown below each slice indicates the percentage of the volume of interest that occupies the volume of lung region of the synthetic image. The segmentation masks are plotted in green.

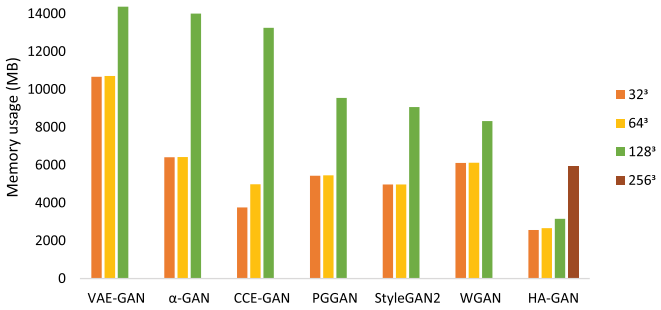


Fig. 6. Results of memory usage test. Note that HA-GAN is the only model that can generate images sized 256^3 without memory overflow on high-end GPU with 16 GB VRAM.

F. Memory Efficiency

In this section, we compare the memory efficiency of HA-GAN with baselines. We measure the GPU memory usage at the training time for all models under different resolutions, including 32^3, 64^3, 128^3, and 256^3. The results are shown in Fig. 6. Note that the experiments are performed on the same GPU (Tesla V100 with 16 GB memory), and we set the batch size to 2. The HA-GAN consumes much less memory than baseline models under different resolutions. In addition, HA-GAN is the only model that can generate images of sizes 256^3. All other models exhaust the entire memory of GPU; thus, the memory demand cannot be measured. In order to investigate where the memory efficiency comes from, we report the number of parameters for HA-GAN at different resolutions in Table VII. We found that as the resolution increases, the number of parameters only increases marginally, which is expected as the model only requires a few more layers as resolution increases.

In addition, we compare the computational efficiency of our HA-GAN model with baseline models. More specifically, we measure the number of iterations per second during training. One NVIDIA Tesla V100 GPU is used for each model and we set the batch size as 2. The comparison is performed under the

TABLE VII
NUMBER OF MODEL PARAMETERS AND MEMORY USAGE UNDER DIFFERENT RESOLUTIONS

Output Resolution	Memory Usage (MB)	#Parameters
32^3	2573	74.7M
64^3	2665	78.7M
128^3	3167	79.6M
256^3	5961	79.7M

TABLE VIII
TRAINING SPEED (ITER/S) FOR DIFFERENT MODELS (HIGHER IS BETTER)

WGAN	VAE-GAN	PGGAN	α-GAN	CCE-GAN	StyleGAN	HA-GAN
2.0	1.0	1.3	1.6	0.35	0.23	**3.8**

128^3 resolution where all models can fit in memory. The result is shown in Table VIII. Our HA-GAN is more computationally efficient than the baselines.

V. DISCUSSION

As shown quantitatively in Table II and Table III, HA-GAN achieves lower FID and MMD, as well as higher IS. This implies that our model generates more realistic images. This is further confirmed by the synthetic images shown in Fig. 3, where HA-GAN generates sharper images compared to other methods. We found that our method outperforms baseline methods at both the resolution of 128^3 and 256^3, but the lead is larger at 256^3 resolution than 128^3. Based on the results, we believe that the sharp generation results come from both the model itself and its ability to directly generate images at 256^3 without interpolation upsampling. For the baseline models, we found that α-GAN and WGAN have similar performance, and VAE-GAN tends to generate blurry images. WGAN is essentially the α-GAN without the encoder. Based on qualitative examples shown in Fig. 3, it can generate sharper images compared to α-GAN and Progressive GAN. However, it also generates more artifacts. According to the quantitative analysis shown in Table II, overall the generation quality of α-GAN is comparable with WGAN. Although our proposed HA-GAN achieves the highest quality comparing to the baseline models, we admit that there is still a gap between HA-GAN generated images and real images. We also note that in order to achieve optimal performance for HA-GAN, most of blank axial slices of training images need to be removed, because empty sub-volume may confuse the model. There are several directions that may further improve the performance, including using a pretrained segmentation network to regularize the generated images, etc. We hope that our method establishes a strong baseline that can be pushed further by future work.

For the ablation studies, first we found that adding a low-resolution branch can help improve results, we think it's because the low-resolution branch can help the model learn the global structure. Second, we observe in Table IV that HA-GAN with encoder outperforms the version without encoder in terms of image synthesis quality. The reconstruction loss in the objective function ensures that the reconstructed images are voxel-wise consistent with the original images. This term can encourage the generator to represent all data and not collapse, improving the performance of the generator in terms of image synthesis. Finally, using randomly selected r leads to randomly selected locations of sub-volumes. In this way, the junctions between sub-volumes can be better covered.

The embedding shown in Fig. 4(a) and Fig. 4(b) reveals that the distribution of the synthetic images by HA-GAN is more consistent with the real images, compared to all baselines. The scatters of WGAN/α-GAN show compressed support of real data distribution, which suggests that samples of WGAN (cyan) and α-GAN (green) have lower diversity than the real images. We think one reason is that the models only learn few attributes of samples in the dataset. To be more specific, the models learn an overly simplified distribution, so the generated images are of lower diversity. The HA-GAN model we proposed has an encoder module, which encourages different latent codes to map to different outputs, improving the diversity of generated samples. A portion of scatters of Progressive GAN (blue) and StyleGAN2 (purple) lay outside of real data distribution (red), which suggests that some generated images may contain artifacts.

In clinical applications, high-resolution CT can help radiologists make reliable diagnose decisions, including pulmonary eosinophilic granuloma, lymphangiomyomatosis, and emphysema [8]. High-resolution CT is especially beneficial in imaging tasks in which small anatomy and pathologic structure is the target, such as in-stent stenosis, lung nodules, coronary calcification, and temporal bones [41]. There are previous works that propose to use 3D GAN for diverse clinical applications [9], [10]. For instance, synthesized images can be used for data anonymization which enables privacy-preserving data sharing between institutions [42]. However, the generated images are limited to the small size of $128 \times 128 \times 128$ or below, due to insufficient memory during training. In most clinical CT applications, image matrix size of 512×512 or larger is used for in-plane direction [41]. Our proposed HA-GAN bridges the gap between them and serve as a plug-and-play module to improve performance for many GAN-based medical imaging applications.

We demonstrate two clinical applications in our paper: data augmentation and clinical-relevant feature extraction. For data augmentation, the results in Table V show that samples generated by HA-GAN can help the training of classification model. While samples generated by α-GAN can also help the training, the performance gain is smaller. We think one reason is that samples generated by HA-GAN are more realistic, also shown in Table II and Table III. GAN can learn a rich prior from existing medical imaging datasets, and the generated samples can help classifiers to achieve better performance.

For the experiment of feature extraction, we encode the full image into a flat variable to extract meaningful and compact feature representation for downstream clinical feature prediction. Table VI shows that HA-GAN can better extract clinical-relevant features from the images, comparing to VAE-GAN and α-GAN. Some clinical-relevant information might be hidden in specific details in the medical images, and can only be observed under high resolution. VAE-GAN and α-GAN can only process lower-resolution images of 128^3. We speculate that the high-resolution information leveraged by HA-GAN helps it learn better representations.

From Table VII, we found that as the output resolution increases, the total number of model parameters does not increase much, but as the multiplier factor increases, the memory usage increases drastically. Therefore, we believe that the memory efficiency mainly comes from the sub-volume scheme rather than model parameters.

VI. Conclusion

In this work, we develop a hierarchical GAN model that can generate 3D high-resolution images. Experiments on 3D thorax CT and brain MRI show that HA-GAN achieves state-of-the-art performance in image synthesis and clinical applications. Our method enables various real-world medical imaging applications that rely on high-resolution image generation and analysis.

References

[1] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.

[2] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed, "Variational approaches for auto-encoding generative adversarial networks," 2017, *arXiv:1706.04987*.

[3] C. Han, K. Murao, and S. Satoh, "Learning more with less: GAN-based medical image augmentation," *Med. Imag. Technol.*, vol. 37, no. 3, pp. 137–142, 2019.

[4] H.-C. Shin *et al.*, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," in *Proc. Int. Workshop Simul. Synth. Med. Imag.*, 2018, 1–11.

[5] T. M. Quan, T. Nguyen-Duc, and W.-K. Jeong, "Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1488–1497, Jun. 2018.

[6] K. Armanious *et al.*, "MedGAN: Medical image translation using GANs," *Computerized Med. Imag. Graph.*, vol. 79, 2020, Art. no. 101684.

[7] Y. Lei *et al.*, "MRI-based synthetic CT generation using deep convolutional neural network," in *Proc. SPIE Med. Imag. Image Process.*, 2019, vol. 10949, Art. no. 109492T.

[8] F. Bonelli, T. Hartman, S. Swensen, and A. Sherrick, "Accuracy of high-resolution CT in diagnosing lung diseases," *Amer. J. Roentgenol.*, vol. 170, no. 6, pp. 1507–1512, 1998.

[9] M. D. Cirillo, D. Abramian, and A. Eklund, "Vox2vox: 3D-GAN for brain tumour segmentation," in *Proc. Int. MICCAI Brainlesion Workshop*, Springer, 2020, pp. 274–284.

[10] B. Yu, L. Zhou, L. Wang, J. Fripp, and P. Bourgeat, "3D CGAN based cross-modality MR image synthesis for brain tumor segmentation," in *Proc. IEEE 15th Int. Symp. Biomed. Imag.*, 2018, pp. 626–630.

[11] M. J. Chuquicusma, S. Hussein, J. Burt, and U. Bagci, "How to fool radiologists with generative adversarial networks? A visual turing test for lung cancer diagnosis," in *Proc. IEEE 15th Int. Symp. Biomed. Imag.*, 2018, pp. 240–244.

[12] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using GAN for improved liver lesion classification," in *Proc. IEEE 15th Int. Symp. Biomed. Imag.*, 2018, pp. 289–293.

[13] H. Zhao, H. Li, and L. Cheng, "Synthesizing filamentary structured images with GANs," 2017, *arXiv:1706.02185*.

[14] S. Singla, B. Pollack, J. Chen, and K. Batmanghelich, "Explanation by progressive exaggeration," in *Proc. 8th Int. Conf. Learn. Representations*, 2020.

[15] H. Shan *et al.*, "3-D convolutional encoder-decoder network for low-dose CT via transfer learning from a 2-D trained network," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1522–1534, Jun. 2018.

[16] E. Simo-Serra, "Virtual thin slice: 3D conditional GAN-based super-resolution for CT slice interval," in *Proc. Mach. Learn. Med. Image Reconstruction: 2nd Int. Workshop*, 2019, vol. 11905, Art. no. 91.

[17] W. Jin, M. Fatehi, K. Abhishek, M. Mallya, B. Toyota, and G. Hamarneh, "Applying artificial intelligence to Glioma imaging: Advances and challenges," *J. Neural Eng.*, vol. 17, no. 2, 2020, Art. no. 021002.

[18] H. Uzunova, J. Ehrhardt, F. Jacob, A. Frydrychowicz, and H. Handels, "Multi-scale GANs for memory-efficient generation of high resolution medical images," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2019, pp. 112–120.

[19] P. K. Diederik *et al.*, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Representations*, 2014.

[20] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *Proc. Int. Conf. Learn. Representations*, 2017.

[21] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAn: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[22] J. Townsend, T. Bird, J. Kunze, and D. Barber, "Hilloc: Lossless image compression with hierarchical latent variable models," in *Proc. Int. Conf. Learn. Representations*, 2020.

[23] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1558–1566.

[24] L. Sun, J. Chen, Y. Xu, M. Gong, K. Yu, and K. Batmanghelich, "Hierarchical amortized training for memory-efficient high resolution 3D GAN," *Proc. Med. Imag. Meets NeurIPS Workshop*, 2020.

[25] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[26] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[27] G. Kwon, C. Han, and D.-s. Kim, "Generation of 3D brain MRI using auto-encoding generative adversarial networks," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2019, pp. 118–126.

[28] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. Int. Conf. Learn. Representations*, 2018.

[29] S. Hong *et al.*, "3D-StyleGAN: A style-based generative adversarial network for generative modeling of three-dimensional medical images," in *Deep Generative Models, and Data Augmentation, Labelling, and Imperfections*. Berlin, Germany: Springer, 2021.

[30] S. Xing, H. Sinha, and S. J. Hwang, "Cycle consistent embedding of 3D brains with auto-encoding generative adversarial networks," in *Proc. Med. Imag. Deep Learn.*, 2021.

[31] E. A. Regan *et al.*, "Genetic epidemiology of COPD (COPDGene) study design," *COPD: J. Chronic Obstructive Pulmonary Dis.*, vol. 7, no. 1, pp. 32–43, 2011.

[32] A. J. Holmes *et al.*, "Brain genomics superstruct project initial data release with structural, functional, and behavioral measures," *Sci. Data*, vol. 2, 2015, Art. no. 150031.

[33] B. Fischl, "Freesurfer," *Neuroimage*, vol. 62, no. 2, pp. 774–781, 2012.

[34] 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.*, 2017.

[35] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A Kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, 2012.

[36] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[37] S. Chen, K. Ma, and Y. Zheng, "Med3D: Transfer learning for 3D medical image analysis," 2019, *arXiv:1904.00625*.

[38] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[39] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[40] R. San Jose Estepar, J. C. Ross, R. Harmouche, J. Onieva, A. A. Diaz, and G. R. Washko, "Chest imaging platform: An open-source library and workstation for quantitative chest imaging," *Amer. Thoracic Soc.*, 2015, pp. A4975–A4975.

[41] J. Wang and D. Fleischmann, "Improving spatial resolution at CT: Development, benefits, and pitfalls," *Radiology*, vol. 289, no. 1, pp. 261–262, 2018.

[42] P. Subramaniam *et al.*, "Generating 3D TOF-MRA volumes and segmentation labels using generative adversarial networks," *Med. Image Anal.*, vol. 78, 2022, Art. no. 102396.