End-to-End 3D Liver CT Image Synthesis from Vasculature Using a Multi-Task Conditional Generative Adversarial Network

Qianmu Xiao 1 and Liang Zhao 1,2,*

1 School of Computing and Electronic Informatics, Guangxi University, Nanning 530004, China; 2013301052@st.gxu.edu.cn
2 Precision Medicine Research Center, Taihe Hospital, Hubei University of Medicine, Shiyan 442000, China
* Correspondence: S080011@e.ntu.edu.sg

Abstract: Acquiring relevant, high-quality, and heterogeneous medical images is essential in various types of automated analysis, used for a variety of downstream data augmentation tasks. However, a large number of real image samples are expensive to obtain, especially for 3D medical images. Therefore, there is an urgent need to synthesize realistic 3D medical images. However, the existing generator models have poor stability and lack the guidance of prior medical knowledge. To this end, we propose a multi-task (i.e., segmentation task and generation task) 3D generative adversarial network (GAN) for the synthesis of 3D liver CT images (3DMT-GAN). To the best of our knowledge, this is the first application for a 3D liver CT image synthesis task. Specifically, we utilize a mask of vascular segmentation as the input because it contains structural information about a variety of rich anatomical structures. We use the semantic mask of the liver as prior medical knowledge to guide the 3D CT image generation, reducing the calculation of a large number of backgrounds, thus making the model more focused on the generation of the region of the liver. In addition, we introduce a stable multiple gradient descent algorithm (MGDA) reconstruction method into our model to balance the weights of the multi-task framework. Experiments were conducted on a real dataset, and the experimental results show that the segmentation task achieves a Dice similarity coefficient (DSC) of 0.87, while the synthesis task outperforms existing state-of-the-art methods. This study demonstrates the feasibility of using vascular images to synthesize images of the liver.

Keywords: medical image synthesis; GAN; 3D liver reconstruction; multi-task learning

1. Introduction

With the advancement of medical imaging and computer graphics technology, 3D liver image reconstructions play an increasingly significant role in medical education, technician training, and simulated surgery [1,2]. However, acquiring complete and high-quality 3D liver CT images necessitates patients undergoing a full CT tomography procedure. Such data inherently involve patient privacy concerns and regional differences in medical equipment conditions. Additionally, the limited number of images for certain rare diseases makes it difficult for physicians to identify and learn from relevant cases.

Recently, deep learning (DL)-based liver image analysis has been a great success in the medical diagnosis field [3–5]. However, DL-based methods require a large number of image samples for learning and training. Currently, collecting and labelling medical image samples is a labor-intensive task that can consume a lot of manpower and material resources, further limited by medical ethics and patient privacy issues. Therefore, it is very difficult to obtain real medical image samples at this stage, especially 3D medical images [6–9]. Image generation techniques based on DL have made it possible to obtain more realistic segmented images, providing new methods. Nowadays, generative adversarial network (GAN)-derived image generation models have demonstrated remarkable
results in the field of synthetic images [10,11], effectively addressing the issue of obtaining substantial amounts of medical image data for medical image synthesis.

However, the previous GAN-based generation methods are unstable and have high uncertainty [12–14]. Furthermore, different from natural images, medical images often contain anatomical knowledge, i.e., prior clinical knowledge, and previous medical image synthesis methods fail to consider this knowledge in depth [6,15]. Furthermore, previous images have been mostly based on the generation of 2D medical images, with few methods focused on the generation of 3D medical images. In this paper, we propose a multi-task 3D generative adversarial network (GAN) with liver mask segmentation and liver generation tasks to generate 3D liver CT images using vascular segmentation labels (see Figure 1). The contributions of this paper can be summarized as the follows:

- We propose a multi-task conditional generation adversarial model that synthesizes 3D liver CT images by inputting liver vessel labels to provide a priori anatomical structure information and using real liver segmentation labels for guidance. To the best of our knowledge, this is the first application for 3D liver CT image generation.
- We introduce a robust multi-gradient method to optimize multiple tasks, specifically by balancing the weights of individual tasks in a multi-task framework.
- Extensive experiments were conducted on the collected real data, and the experimental results show that the performance of our method outperforms existing state-of-the-art methods. In addition, this study demonstrates the possibility of using vascular images to synthesize images of the liver.

![Figure 1. An example of image synthesis using our proposed method.](image)

### 2. Related Work
#### 2.1. GAN-Based Model

Generative adversarial networks (GANs) [12] first introduced the adversarial relationship between the generator and discriminator for synthesizing pseudo-data that emulate the composite target dataset distribution. Subsequently, Radford et al. [13] incorporated convolutional neural networks (CNNs) to replace the deep neural network-based generator model, which significantly improved the quality and stability of the synthesized images. Due to the instability of the adversarial learning mechanism, prone to gradient vanishing or adversarial collapse, to constrain the randomness of the synthesis results, Mirza et al. proposed conditional GANs (CGANs) [14] to synthesize images that meet specific requirements by adding conditions to the random vector. Isola et al. [2] introduced the concept of image translation by removing the input of the random vector and directly using the complete image as the input condition. However, the translation mechanism of Pix2Pix requires paired image data inputs, which places high demands on the dataset. Therefore, Zhu et al. proposed unsupervised learning-based image style transfer models, such as CycleGAN [16] and StyleGAN [17]. Another interesting piece of research [18] was the proposed improved 3D-Unet, which was then embedded into the GAN framework.
to segment the 3D liver. This suggests that a GAN-based approach may enhance the performance of medical segmentation tasks. Recently, image generation tasks have become an extremely popular research topic, with some efficient models already more effective than GAN-based mechanisms. For instance, the latest variational autoencoder (VAE)-based diffusion models [19,20] and the language-to-image models developed by OpenAI [21] have yielded impressive results. However, such models are not fully open source and require huge amounts of data for training.

2.2. Medical Image Generation

The high cost of medical image sampling and the need for professional manual annotation pose a series of challenges for building large-scale medical image datasets with annotations. This problem can be effectively solved by augmenting a dataset with authentic-style pseudo-data synthesized by GANs and their derived models [12,22]. This includes the synthesis of new data from random noise, as reported in [23]. Additionally, for specific rare disease images with low data volumes, images can be synthesized with target lesion characteristics based on specific tags on the input [24,25]. These synthesized images can be used to supplement rare cases in the dataset, as well as for teaching and training purposes. The study of 3D liver image synthesis in this paper also belongs to this field of application. Similar related studies can be found in [6,8]. The former first proposed synthesizing fundus retinal images using the Pix2Pix model based on 2D retinal vessel segmentation labels, while the latter suggested adding classification feature loss and improved retinal byte loss to jointly constrain the synthesis results, thereby providing more detail. In another piece of research, Mende et al. [7] synthesized lung images with specific lesions based on both lung and lesion segmentation labels. However, similar work was primarily oriented towards the synthesis of 2D images, while the synthesis of lesion results required a separate input of the specific lesion labels for supervision. Ying et al. [26] used GAN to reconstruct 3D lung images using 2D lung X-ray images, the first application of using 2D data to synthesize 3D images. Previous 3D generated image tasks did not introduce the medical priori knowledge to guide image synthesis.

2.3. Multi-Task Learning

In this paper, we propose a novel synthetic image multi-task method for 3D liver CT image generation. One of the advantages of multi-task learning is that different tasks can share parameters and assist each other, eventually achieving joint improvement of multiple tasks [27,28]. For example, Pu et al. [29] obtained better performance by splitting the standard section recognition into a section type recognition task and a whether-or-not standard task. The task of section classification can assist the whether-standard task. Similarly, Zhao et al. [30] proposed that the performance of different tasks can be improved by using multiple tasks in the quality control of ultrasound images. To improve the neuroimage quality, Wang et al. [31] proposed a multi-task deep learning method to jointly synthesize multi-contrast neuroimaging using signal relaxation relationships and spatial information. Huang et al. [32] designed a multi-task decoherent modality transferable GAN, i.e., MCMT-GAN, to tackle the issue of brain MRI synthesis using an unsupervised method. The weights of multiple tasks in previous multi-task methods were set manually, potentially resulting in one task being overfitted and another not converged. More details of multi-task learning are reported in [33].

3. Materials and Methods

3.1. Dataset Description and Processing

The image generation model requires a large amount of real data for training. In this paper, we require complete 3D liver segmentation images and corresponding liver vessel segmentation labels to conduct the experiment. The commonly used public dataset with liver–vessel segmentation labels 3D-Iracadb [34] has only 20 patient scan sequences, and the complete sequence length is only about 30 sheets. Concerning the MSD [35] dataset,
although the number of patients is high, it lacks accurate liver contour segmentation labels, while the sequence length is also generally in the upper 50 sheets. Therefore, we used the LiVS [36] dataset to evaluate the performance of our model. This dataset includes 515 complete patient CT image sequences containing 82,428 contrast-enhanced CT liver segmentation sections, of which 15,449 slices have been manually annotated for liver vessel segmentation. Due to the high workload and difficulty of manually annotating vessels, a large number of efficient liver vascular segmentation models [37–39] can be used to predict the segmentation label. We used part of the expert-labelled dataset, then trained the efficient liver vessel segmentation model [39] using the annotated dataset, predicting the remaining segmentation labels using the trained model.

In the selection of the sequence length for the input 3D image, i.e., the size of the 3D image in the axial direction, we acquired image sequences with lengths ranging from 70 to 300. In order to unify the size of the input 3D image and reduce the computing pressure, we divided a complete liver CT sequence into fixed-length sequence segments, with half the sequences overlapping each other between the two segments to ensure continuity between the sequence segments and to achieve data enhancement (as shown in Figure 2).

![Figure 2. Example slices of the 3D image sequence. The whole vasculature is divided into fixed vascular pieces at the input. The third and fourth figures represent the reconstructed liver pieces and the whole liver, respectively.](image)

3.2. 3DMT-GAN

3.2.1. Overview

Our model (Figure 3) includes two main components, the generator and the discriminator. The vascular mask slice is the input as a fixed 2D sequence, fed through the generator to produce a 2D liver image sequence, i.e., a 3D liver image. The discriminator then judges whether the generated liver is a real 3D liver image or not.

![Figure 3. Overview of the proposed multi-task image translation model.](image)
3.2.2. Generation Task Based on Segmentation Task Guide

In this paper, we introduce a real liver segmentation mask to guide the synthetic image based on the existing basic pixel loss function. The general method to calculate $L_1$ loss requires counting all pixel losses of the whole image. However, in the medical generation task, the image always contains a large amount of background, e.g., regions with pixel values of 0. These backgrounds not only consume a lot of computational resources when computing the loss but also reduce the sensitivity of the model to focus on the generation of the liver region. To solve this issue, we added a liver supervision segmentation task to the image synthesis model and calculated the average pixel loss of the liver region. The loss function of the segmentation task is as follows:

$$L_s^G = \frac{(L_{bce}(m_p, m_{gt}) + L_{dsc}(m_p, m_{gt}))}{2}, \tag{1}$$

where $L_{bce}$ denotes the cross-entropy loss function and $L_{dsc}$ indicates the Dice loss function, $m_p$ represents the predicted liver segmentation mask and $m_{gt}$ denotes the ground truth. The definition of these two loss functions is as follows:

$$L_{bce} = -\frac{1}{N} \sum_{i=1}^{N} [m_{gt} \log(m_p) + (1 - m_{gt}) \log(1 - m_p)], \tag{2}$$

$$L_{dsc} = 1 - \frac{2|m_{gt} \cap m_p|}{|m_{gt}| + |m_p|}, \tag{3}$$

where $m_{gt}$ represents the $i$-th real image, $m_p$ represents the $i$-th synthesized image, and $N$ represents the number of samples. In addition to generating the segmentation mask, we also need to synthesize the liver image. The loss of the image generation liver task is defined as follows:

$$L_t^G = (|l_g - l_{gt}| \times m_{gt}) / \text{sum}(m_{gt}), \tag{4}$$

$$|l_g - l_{gt}| = \frac{1}{HWZ} \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{k=1}^{Z} |\hat{l}_{ijk} - \tilde{l}_{ijk}|, \tag{5}$$

where $l_g$ represents the generated pseudo-liver image, $l_{gt}$ represents the real liver image, $m_{gt}$ represents the real liver segmentation mask, and $\text{sum}(m_{gt})$ represents the number of pixels in the liver region in the liver segmentation mask. In the synthetic liver image task, the pixel difference between the whole synthetic image and the true image was calculated and the absolute value was found, then the true liver segmentation label was multiplied by the loss value pixel by pixel. At this time, the pixel loss of the liver region was preserved and the pixel loss of the non-liver region was set to 0, and finally the average pixel loss of the liver region was obtained by dividing the number of pixels in the liver segmentation region. This reduces the computational effort and allowed our model to focus more on the generation of liver regions rather than a large number of background regions.

Our multi-task generator adopted a U-net-like \[40,41\] encoder–decoder architecture with skip connections, except in the decoder we added a new task branch to form a Y-shaped generator structure, and the two decoder branches were responsible for two tasks, liver image generation and liver region segmentation, with the former activated by the $\tanh$ function and supervised by the real liver segmentation image using 3D-$L_1$ loss (Equation (5)). Furthermore, the latter task was activated by the $\text{sigmoid}$ function and supervised by the liver segmentation label using Equations (4) and (5). The image synthesis task focused on synthesizing liver images and liver masks. Therefore, the total loss function contained two parts defined as follows (Equation (6)):
where \( \mathcal{L}^G \) represents the total generator loss, and \( \mathcal{L}^G_s \) denotes the loss of the liver image synthesis, i.e., texture loss. \( \mathcal{L}^G_t \) denotes the loss of the synthetic liver segmentation mask.

### Table 1. Configuration of the generator encoder.

<table>
<thead>
<tr>
<th>Input Size</th>
<th>320 × 320 × 64</th>
<th>320 × 320 × 48</th>
<th>320 × 320 × 32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generator Encoder</td>
<td>160 × 160 × 32</td>
<td>160 × 160 × 24</td>
<td>160 × 160 × 32</td>
</tr>
<tr>
<td></td>
<td>80 × 80 × 16</td>
<td>80 × 80 × 12</td>
<td>80 × 80 × 16</td>
</tr>
<tr>
<td></td>
<td>40 × 40 × 8</td>
<td>40 × 40 × 6</td>
<td>40 × 40 × 8</td>
</tr>
<tr>
<td></td>
<td>20 × 20 × 4</td>
<td>20 × 20 × 3</td>
<td>20 × 20 × 4</td>
</tr>
<tr>
<td></td>
<td>10 × 10 × 2</td>
<td>10 × 10 × 1</td>
<td>10 × 10 × 2</td>
</tr>
<tr>
<td>Bottleneck</td>
<td>5 × 5 × 1</td>
<td>5 × 5 × 1</td>
<td>5 × 5 × 1</td>
</tr>
</tbody>
</table>

#### 3.2.3. Multi-Task Generator Optimized by MGDA-UB

One key issue in the multi-task framework was the balance of weights for different tasks. For example, some tasks were more difficult and required a larger weight, while some tasks were simpler and required less weight. The core idea of this approach was to explore the shared parameters of multiple related tasks during the training process to improve the performance of the model on each task and to improve the generalization ability of the model. The structure of our multi-task generator model constitutes a typical multi-task learning model with hard shared parameters for two tasks focused on assigning the weights of the composite loss function composed between multiple tasks.

Nowadays, there are several mature optimization ideas for this method, such as the hard parameter sharing model with shared and unique parameters, using fixed or dynamic weights to construct compound loss functions to obtain better training efficiency. In this paper, we select the upper bound optimized multiple gradient descent algorithm (MGDA-UB) to dynamically update the weights in the composite loss function in each iteration, which has been demonstrated to be significantly better than the existing fixed weight method of each task, the gradient normalization method (Gradnorm). Meanwhile, it is computationally more efficient than the original multiple gradient descent algorithm (MGDA). The dynamic weight parameters for each task were updated according to the KKT (Karush–Kuhn–Tucker) condition to find the Pareto optimal point.

In Equation (6), the weights of the two tasks are \( \alpha \) and \( 1 - \alpha \), respectively, and based on the MGDA-UB algorithm, the optimization objective of our model is:

\[
\min_{\alpha \in [0,1]} \| \alpha \nabla_Z \hat{L}^1(\theta_{sh}, \theta^1) + (1 - \alpha) \nabla_Z \hat{L}^2(\theta_{sh}, \theta^2) \|_2^2,
\]

where \( \alpha \) ranges from 0 to 1, \( \theta_{sh} \) represents the parameters shared between the different tasks of the model, which in this paper corresponds to the encoder. \( \theta^1 \) and \( \theta^2 \) denote the parameters of the two specific task branches, which in this paper correspond to the parameters of the two decoders. \( \hat{L}^1 \) and \( \hat{L}^2 \) represent the loss functions of different tasks, \( +, f^+ \) denotes a clip function similar to Equation (9), and \( Z \) represents the bottleneck vector obtained by downsampling the input vessels in the model. Finally, in each iteration, \( \alpha \) can be updated as:
The main idea of MGDA is to find a gradient direction for multi-task learning that reduces the loss of all tasks at the same time. This optimization method finds a gradient direction such that this direction achieves the best possible performance on each task. MGDA can represent the gradients of different tasks as vectors in a gradient space, and then to find a new gradient direction that has a certain degree of projection on the gradients of each task. By this method, we can find a direction that combines the gradient information of all tasks.

3.2.4. Patch Discriminator

In [49], Demir et al., proposed to determine the authenticity of a composite image by chunking it instead of giving a comprehensive authenticity output for the whole image. Inspired by this work, we also incorporated this core idea into our discriminator. In contrast to [49], we employed 3D convolution operations and the adversarial loss function of our patch discriminator composition as follows:

$$L_D = E[\log(D(v, l_r))] + E[\log(1 - D(v, l_g))],$$

where $D(v, l_r)$ represents the input of the discriminator. $D$ represents the discriminator, $v$ represents the input real vascular structure, $l_r$ represents the corresponding real liver and $l_g$ represents the liver generated by the generator. $D(v, l_g)$ represents the output of the discriminator, which is a probability result of the different input patched 3D data, and a 3D matrix taking values between 0 and 1. The final loss function of our whole model is as follows:

$$\mathcal{L} = \beta \mathcal{L}_G + \mathcal{L}_D,$$

where the parameter $\beta$ is set to 100, consistent with the Pix2Pix [2] model. The adjustment of $\beta$ is considered to be a trade-off between the realism of the generated results and the restoration, when the larger the value of this parameter the more biased the synthesis results will be in the style of restoration, and the smaller the choice will be closer to the original image corresponding to it. The details of the patch discriminator configuration are listed in Table 2.

**Table 2. Configuration of the patch discriminator.**

<table>
<thead>
<tr>
<th>Input Size</th>
<th>320 × 320 × 64</th>
<th>320 × 320 × 48</th>
<th>320 × 320 × 32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discriminator</td>
<td>160 × 160 × 32</td>
<td>160 × 160 × 24</td>
<td>160 × 160 × 16</td>
</tr>
<tr>
<td></td>
<td>80 × 80 × 16</td>
<td>80 × 80 × 12</td>
<td>80 × 80 × 8</td>
</tr>
<tr>
<td></td>
<td>40 × 40 × 8</td>
<td>40 × 40 × 6</td>
<td>40 × 40 × 4</td>
</tr>
<tr>
<td></td>
<td>20 × 20 × 4</td>
<td>20 × 20 × 3</td>
<td>20 × 20 × 2</td>
</tr>
<tr>
<td>Output Size</td>
<td>10 × 10 × 4</td>
<td>10 × 10 × 3</td>
<td>10 × 10 × 2</td>
</tr>
</tbody>
</table>

4. Experimental Results

4.1. Implementation Settings

We designed experiments with different input sequences for the segmentation task, as listed in Table 3. For the training settings, we set the batch size to 2, optimizer as an
Adam optimizer with a learning rate of 0.00001. The epoch was set to 100. We implemented the source code using the Pytorch framework and trained the model on a Linux server with a single Nvidia A100-40 GB GPU. All liver segments were divided into training, validation, and testing sets in a ratio of 7:2:1. In this paper, we loaded the pre-training weights for the model \[39\] and then continued to fine-tune them on our data. The mask of the segmented vessels was then used as part of the input data for the 3D generation task. On the validation set (i.e., all the vascular segmentation masks in our experiment) the DSC was 0.803.

### Table 3. Number of input sequences for the segmentation task.

<table>
<thead>
<tr>
<th>Segmentation Layers</th>
<th>64 s</th>
<th>48 s</th>
<th>32 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of liver sequence segments</td>
<td>1621</td>
<td>2323</td>
<td>3579</td>
</tr>
</tbody>
</table>

#### 4.2. Evaluation Metrics

Three evaluation metrics, FID (Fréchet inception distance) \[50\], KID (kernel inception distance) \[51\], and LPIPS (learned perceptual image patch similarity) \[52\], were used to evaluate the performance of the synthesized task. These three metrics are defined as follows:

\[
\text{FID}(l_g, l_p) = \| \mu_g - \mu_p \|^2_2 + \text{Tr}(\Sigma_g + \Sigma_p - 2(\Sigma_g \Sigma_p)^{1/2})
\]

(12)

\[
\text{KID}(l_g, l_p) = \text{MMD}_k(l_g, l_p)^2
\]

(13)

\[
\text{LPIPS}(f_g, f_p) = \sum_{l=1}^{L} w_l \| \Phi_l(f_g) - \Phi_l(f_p) \|_2^2
\]

(14)

where \(l_g\) and \(l_p\) represent the high-dimensional feature vectors of the real liver image and the synthetic liver image extracted using the Inception V3 network, respectively. \(\mu_g\) and \(\mu_p\) represent the mean values of these two vectors, respectively. \(\Sigma_g\) and \(\Sigma_p\) represent the covariance of these two vectors, respectively, \(\text{Tr}\) represents the trace operation. In Equation (14), \(f_g\) and \(f_p\) represents the real liver image and the generated liver image, \(L\) represents the number of layers in the VGG network used for feature extraction, \(w_l\) represents the weight of the \(l\) layer, and \(\Phi_l(f_g)\) represents the feature mapping of the image \(f\) at the \(l\) layer. \(\| \cdot \|_2\) represents the L2 norm, denoting the Euclidean length of the vector.

FID and KID were used to calculate whether the ground truth and the synthesized image have the same distribution. The FID was calculated using the Inception V3 network \[53\], by removing the final fully connected and pooling layers, a high-dimensional feature can be obtained, in this paper we used the highest 2048 dimensions. The vectors extracted by the Inception V3 network from our real liver CT images obey a specific distribution. If the corresponding extracted feature vectors of the synthesized liver image also obey the same distribution, then the synthesized image has a high degree of realism. In other words, by calculating the distance between the real image and the feature vector of the synthesizing image, the smaller the distance obtained, the stronger the performance of the model. The calculation of KID is similar to the FID, which also requires the extraction of high-dimensional feature vectors from the Inception V3 networks, but the maximum mean discrepancy (MMD) is calculated for the two feature vectors instead of the Fréchet distance as in the case of FID. The KID and FID statistics can be quickly obtained by installing and using the Python package torch-fidelity \[54\].

LPIPS forces the generator to learn to reconstruct the reverse mapping of the real image from fake images, by learning the reverse mapping of the generated image to real image and prioritizing the perceived similarity between them. In this paper, we used the Python package lpips to count LPIPS. Although the focus of this task was not to accurately segment the liver, but to demonstrate the authenticity of the synthetic 3D liver image, we still used the DSC (Equation (6)) coefficient to evaluate the error of the synthetic 3D liver
with the real liver. FID, KID and LPIPS were used to evaluate the detail texture of the synthetic image. DSC was used to evaluate the contour of the synthetic images.

4.3. Baseline Methods

Three classic advanced GAN-based methods used in medical image synthesis are: CGAN [14], Pix2Pix [2], and CycleGAN [16]. The two former methods are supervised training, while the latter is unsupervised training. The training strategy of the Pix2Pix and CGAN are very similar, both using supervised training. The main difference lies in the parameter settings for the generator’s input and upsampling. The latter uses random noise fused with the input conditions to create random synthetic results, while the former only uses images as the input conditions and only uses dropout operations in the generator’s upsampling stage to achieve diversity in the synthetic results. CycleGAN, on the other hand, uses unsupervised training, consisting of two pairs of generators and discriminators. Compared with the first two models, CycleGAN does not require paired datasets for training, but demands a higher training data volume.

4.4. Quantitative Evaluation

As shown in Table 4, compared with CGAN, Pix2Pix and CycleGAN for three different input sequence images based on the four evaluation metrics, our method obtained the best performance in 10 cases, indicating the effectiveness of the proposed method. From Table 4, the texture of the synthesized image was the best (i.e., the value of FID, KID, and LPIPS are highest) when the number of the sequence is 48 (i.e., 48 s). The contour of the synthesized image was optimal (i.e., DSC is the best) when the number of the sequence was 64 (i.e., 64 s). From analysis of the experimental results, in terms of the accuracy of the contour of the final synthesized image, when the sequence length was longer, the more accurate was the contour of the generated image. In terms of the realism of the generated image texture, using a sequence length that was too long or too short made the texture details of the generated image poor, and an appropriate sequence length (i.e., 48) should be chosen. For a sequence length of 32, less information was contained about the overall structure of the vessel, and the synthesized image was poorer in terms of contour and detailed texture. For a more direct display of the results, we give a line plot of the comparison results, as shown in Figure 4.

Table 4. Comparisons of all methods. * indicates the best evaluated metric in the current round of training, bold font indicates that the best flat cabinet value was obtained in all experiments.

<table>
<thead>
<tr>
<th>Number</th>
<th>Model/Evaluation Metrics</th>
<th>FID (dim = 2048)</th>
<th>KID</th>
<th>LPIPS</th>
<th>DSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CGAN_64s</td>
<td>101.459</td>
<td>0.1172</td>
<td>0.180</td>
<td>0.865</td>
</tr>
<tr>
<td>2</td>
<td>Pix2Pix_64s</td>
<td>85.045</td>
<td>0.097</td>
<td><strong>0.174</strong>  *</td>
<td><strong>0.867</strong></td>
</tr>
<tr>
<td>3</td>
<td>CycleGAN_64s</td>
<td>245.226</td>
<td>0.307</td>
<td>0.449</td>
<td>0.599</td>
</tr>
<tr>
<td>4</td>
<td>MTv2l_64s (Ours)</td>
<td><strong>79.295</strong>  *</td>
<td>0.086  *</td>
<td><strong>0.177</strong>  *</td>
<td><strong>0.872</strong>  *</td>
</tr>
<tr>
<td>5</td>
<td>CGAN_48s</td>
<td>98.485</td>
<td>0.1168</td>
<td>0.186</td>
<td>0.863</td>
</tr>
<tr>
<td>6</td>
<td>Pix2Pix_48s</td>
<td>111.586</td>
<td>0.136</td>
<td>0.192</td>
<td>0.856</td>
</tr>
<tr>
<td>7</td>
<td>CycleGAN_48s</td>
<td>195.270</td>
<td>0.234</td>
<td>0.422</td>
<td>0.614</td>
</tr>
<tr>
<td>8</td>
<td>MTv2l_48s (Ours)</td>
<td><strong>61.900</strong>  *</td>
<td><strong>0.062</strong>  *</td>
<td>0.181  *</td>
<td><strong>0.865</strong>  *</td>
</tr>
<tr>
<td>9</td>
<td>CGAN_32s</td>
<td>109.561</td>
<td>0.129</td>
<td>0.194</td>
<td>0.855</td>
</tr>
<tr>
<td>10</td>
<td>Pix2Pix_32s</td>
<td>107.912</td>
<td>0.124</td>
<td>0.186  *</td>
<td>0.833</td>
</tr>
<tr>
<td>11</td>
<td>CycleGAN_32s</td>
<td>297.366</td>
<td>0.344</td>
<td>0.515</td>
<td>0.464</td>
</tr>
<tr>
<td>12</td>
<td>MTv2l_32s (Ours)</td>
<td><strong>85.992</strong>  *</td>
<td>0.094  *</td>
<td>0.187</td>
<td><strong>0.858</strong>  *</td>
</tr>
</tbody>
</table>
4.5. Visualization of the Results

Figures 5–7 show a visualization of the results of all methods, including synthetic results of 3D shell, 3D CT images and 2D tomography, respectively. The visualization of the synthetic image from our model and the three comparison models is shown in Figure 5. The comparison of the 2D slices with the corresponding real image show that the synthesis results using the supervised learning Pix2Pix, CGAN, and the proposed method basically fit the real liver CT segmentation image style. In contrast, the correct synthesis cannot be performed when using the unsupervised learning of CycleGAN. We found that the texture style results of Pix2Pix and CGAN synthesis still have some errors with the real liver slice images, while our method achieved the best synthesis performance.

Figure 6 shows the 3D visualization of the internal structure of the synthetic liver segmentation images from three different viewpoints (i.e., 2D slices are stacked to synthesize 3D images). It can be seen from Figure 6 that the CycleGAN method can only retain the input vascular information and cannot synthesize more accurate internal liver structures, while the three other models can all synthesize some internal liver anatomical structures.

Figure 7 shows a visualization of the geometry of the synthetic liver. Our method extracted the contour information of the synthetic 3D liver image data and performed a 3D reconstruction to show how the synthetic liver image differs from the real liver in terms of geometry. It can be seen from the figure that the 3D liver image synthesized with our method was closer to the real liver in terms of contours, while the external contours of the comparison models all showed large errors.
Figure 5. The visualization of the 2D CT synthesis image.

Figure 6. The visualization of the 3D CT synthesis image.

Figure 7. The visualization of the 3D liver reconstruction.
5. Discussion
5.1. Discussion of the Discriminator Inputs

In this paper, there is an issue with determining what is fed into the discriminator. As shown in Figure 8, there are two types of inputs that can be used by the discriminator to make a judgment. The first lets the synthetic liver and the synthetic liver segmentation mask be multiplied together to filter out the background, and then input into the discriminator with the vessel segmentation mask. The second is the liver vascular segmentation mask, the synthetic liver, and the synthetic liver mask. The second type inputs three parts as three channels of the image into the discriminator, and the first type inputs two channels of the image, as illustrated in Figure 8. As shown in Figure 8, when used as the second input, the texture features generated by the generator was more realistic and detailed, and closer to the real image. As shown in Figure 9, for different inputs, i.e., two or three channels, we show more direct synthetic image results. In Figure 9, we can still observe that the detail of the synthetic image with three channels was closer to the real image.

Figure 8. Two inputs (i.e., two channels or three channels) for the discriminator.

Figure 9. The input of the discriminator. The first one treats the synthetic liver mask, vessel mask and synthetic liver as the three channels of the image. The second input treats the result of the synthetic liver and synthetic liver mask after multiplying them (for filtering the background) as one channel, and the vessel mask as another channel.
5.2. Advantages of the Multi-Task Generator

5.2.1. Discussion of the Differences in Texture Details

The single task does not consider the correlation between tasks and lacks a mechanism to share information among multiple-related tasks, making the learning ability of the model decrease during the training process. Our multi-task generator can share the information of the liver segmentation and liver synthesis tasks. Furthermore, the label of the liver segmentation can also supervise the liver synthesis task. In our comparison method, Pix2Pix and CGAN are single task generators. As shown in Figure 10, the region marked by the red circle is the portal vein in the real liver image, and this structure was usually contained within the liver region. In the comparison models Pix2Pix and CGAN, this structure was generated outside the liver region. Our model effectively avoids this problem, an improvement brought on by using multi-task generator based on a segmentation task to guide our generation task.

![Figure 10](image-url)

*Figure 10.* Different methods for visualization of synthetic images. We extracted the synthesis results of different models at the same stage of early training using the same section scanning position. It can be seen that our model can quickly capture the key areas of the image in the early stage, and start synthesizing basic textures earlier. Moreover, in the later stages, the model can still maintain relatively correct edge shapes.

5.2.2. Discussion of the Effect of Lesion Synthesis

As shown in Figure 11, we give some synthetic visualizations of images containing lesions. Specifically, Figure 11 shows six tomographic scans of the liver region and the corresponding synthetic results for three patients, two with occupying lesions and one with terminal cancer. The red arrows indicate the areas of significant differences in the liver images produced with different generators. It can be seen that in the single-task model, only the L1-loss method was used for the whole image signal intensity constraint. As a result, the synthesized whole image only has the basic contour structure and the internal texture features were ignored. This disparity was reflected in the fact that regions with weaker signals in the real image were directly treated as signal-free regions in the synthesis.
results. In contrast, our method performed a better synthesis even though some images contain lesioned regions.

![Figure 11](image-url) Visualization of synthetic images containing areas with lesions based on the single- and our multi-task generator. We can see that after the masking operation using the synthetic mask, the discriminator’s attention is focused on the liver region, and this gap is particularly prominent in the tomography results of patients with advanced cancer (columns 2 and 3 in the figure) because the images of such vascular image with lesions are difficult to acquire accurate vascular structures.

### 5.3. Limitations of Our Method

Although our method achieved the best synthetic image performance, especially when synthesizing normal liver images, it still has some limitations. For example, our model is sensitive to the prediction and reconstruction of certain liver images with large lesions, such as advanced cancer and large occupying lesions. Because our method uses liver vessels to synthesize liver images, these images with particularly large lesion areas often have a large number of structures in the vessels that may have been lesioned, and the synthesis results were relatively poor compared to normal images. Meanwhile, the number of these samples was also very small, which limits the model’s performance. For liver samples with only minor lesions, our model was insensitive and performed better, because the vascular structures corresponding to such diseases are often not clearly distinguishable from normal liver images.

In addition, manually labelled vascular labels were still visually disconnected after 3D reconstruction, indicating that it was still difficult to obtain accurate vascular structure labels. This shows that clear vascular structural information is indispensable to synthesize 3D liver images. To enable the proposed method to be extended to clinical applications, we need to collect imaging data from different patients in the future. Furthermore, we plan to build a database of liver vascular images. In addition to this, we need to explore whether other anatomical structures can be generated in the liver.

### 6. Conclusions

In this paper, we present a multi-task generative adversarial network for synthesizing 3D liver images. To the best of our knowledge, this is the first application for the 3D liver synthesis task. Specifically, we have used a vascular segmentation mask as a input because it contains structural information about a variety of rich anatomical structures. We have
proposed using the masks labelled with liver segmentation for supervised and guided liver synthesis. In addition, we have introduced a stable gradient descent algorithm to balance the weight distribution among multiple tasks. In addition, we have utilized a masking mechanism in the discriminator and generator to filter out a large number of background regions, making our model more focused on the generation of liver regions rather than a large number of background regions. We have performed experiments on real data, and the experimental results show that our method improves both quantitative and visual analyses. This indicates that it is feasible to synthesize the liver using a vessel segmentation mask and that our multi-task generation method is effective. However, our model also has some limitations, such as being too sensitive to liver images containing a large number of lesions. In the future, we will focus on using multimodal information to synthesize 3D livers.

Author Contributions: Conceptualization, Q.X.; Methodology, Q.X.; Software, Q.X.; Validation, Q.X.; Resources, L.Z.; Data curation, L.Z.; Writing—original draft, Q.X.; Writing—review and editing, L.Z.; Visualization, Q.X.; Project administration, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was collectively supported by the National Natural Science Foundation of China (32060150) and the Advantages Discipline Group (Medicine) Project in Higher Education of Hubei Province (2021–2025) (2022XKQT5).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in reference number.

Conflicts of Interest: The authors declare no conflict of interest.

References


46. Karush, W. Minima of Functions of Several Variables with Inequalities as Side Constraints. Master’s Thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939.


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.