2024.07.19

Physics in Medicine & Biology

PAPER • OPEN ACCESS

2D medical image synthesis using transformer-based denoising diffusion probabilistic model

Shaoyan Pan[1,2], Tonghe Wang[3], Richard L J Qiu[1], Marian Axente[1], Chih-Wei Chang[1], Junbo Peng[4], Ashish B Patel[1], Joseph Shelton[1], Sagar A Patel[1], Justin Roper[1] Show full author list

Published 5 May 2023 • © 2023 The Author(s). Published on behalf of Institute of Physics and Engineering in Medicine by IOP Publishing Ltd

Physics in Medicine & Biology, Volume 68, Number 10

Citation Shaoyan Pan *et al* 2023 *Phys. Med. Biol.* **68** 105004

DOI 10.1088/1361-6560/acca5c

Diffusion model ; Swin-transformer;
medical image synthesis

Yeon Su Park

Pukyong National University
Division of Computer Engineering And Artificial Intelligence
Medical AI Lab.

Introduction

Why synthesis methods?

- **Conventional image augmentation (scalomg, rotation, affine and deformed transformations …) provides limited intrinsic diversity of the dataset.**

- **Synthesis methods can supplement conventional methods. When Large-scale data is needed.**
  - **Imaging system design and development**
  - **Virtual clinical trials**
  - **Statistical model development**

- **Synthesis data should be similar in morphology and texture to real data and add greater diversity in visual appearances and data characteristics to enrich existing datasets.**

Introduction

MT-DDPM

- **GAN was SOTA in various tasks like …**
  - **Image creation**
  - **Image super-resolution**
  - **Image-to-image translation**
  - **Etc …**

- **But, GAN has been suffer from unstable training and mode collapse.**
- **Diffusion Model can address this problem. In this paper the wirter proposes Transformer-based Denosing Diffusion Probabilistic Model (MT-DDPM) incorporate Swin-vision transformer.**

- **U-Net is good method to image denosing. but it has limited ability to model global-level dependency.**
- **U-Swin-transformer can capture both global and local information to perform high-quality diffustion process.**

Part 2,

methods

Network architecture

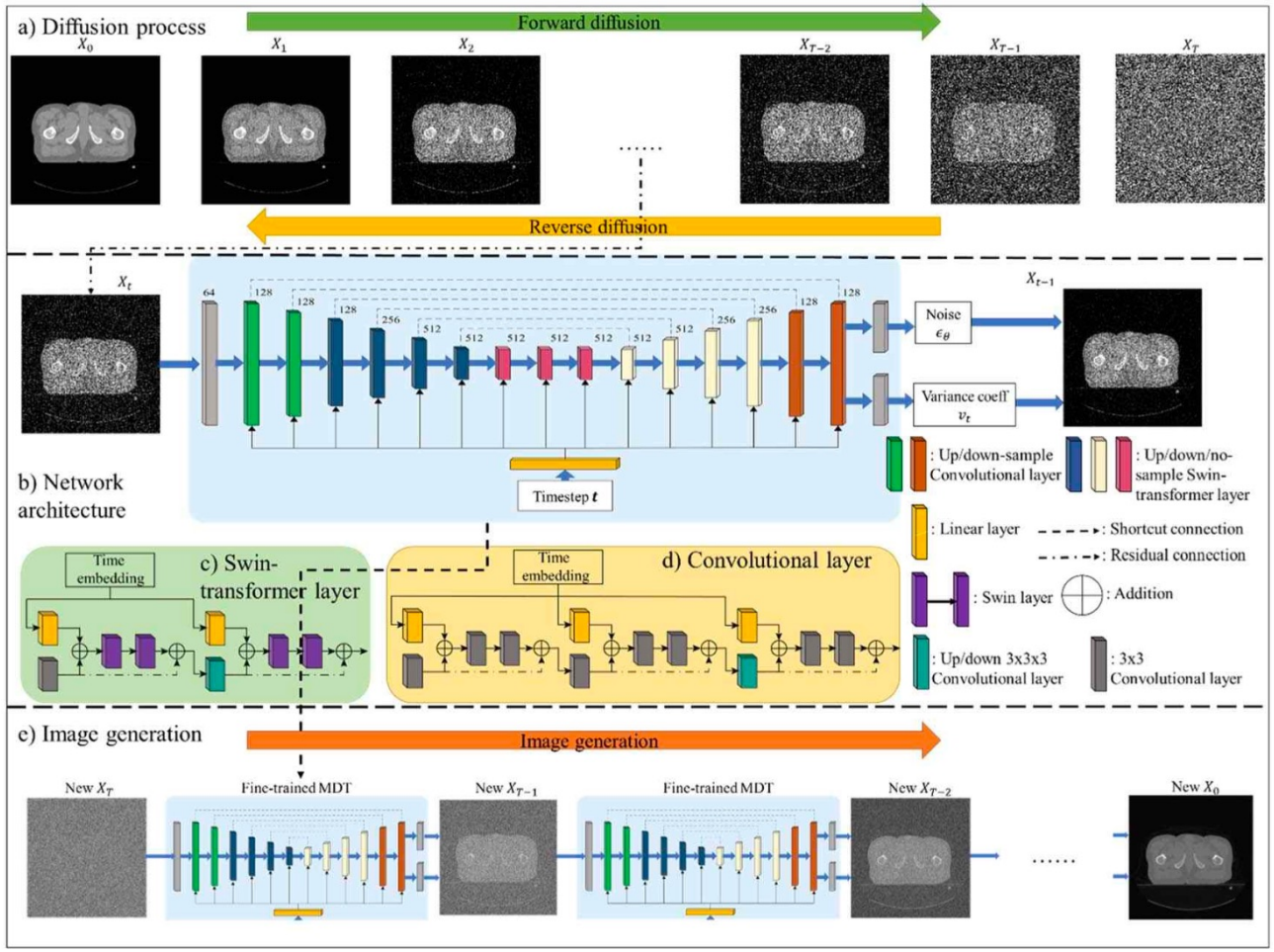
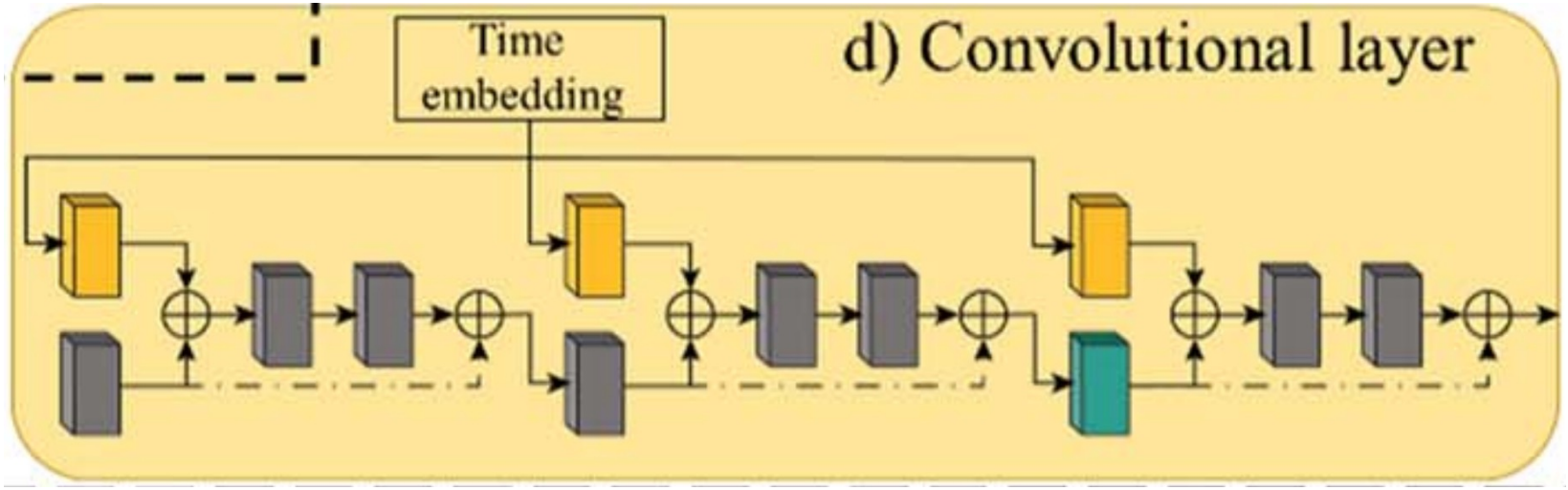


Figure 1. (a) Diffusion process of the MT-DDPM framework: The forward diffusion transforms medical images into pure Gaussian noise by adding a small amount of noise iteratively. The reverse process requires a network to repeatedly denoise the Gaussian noise to the noise-free image. (b) Network architectures of the MT-DDPM network: a symmetrical encoder–decoder architecture to learn the reverse process. We can calculate the denoised image by predicting the noise and variance coefficient by equation (23). (c) Swin-transformer block: the purple Swin layer is built by a window self-attention and a shifted window self-attention module. (d) Convolutional block: Three convolutional residual structures learn local features. (e) Image generation: the fine-training MT-DDPM network can transform a new pure Gaussian noise image into a synthetic image new to the dataset.

Part 2,

methods

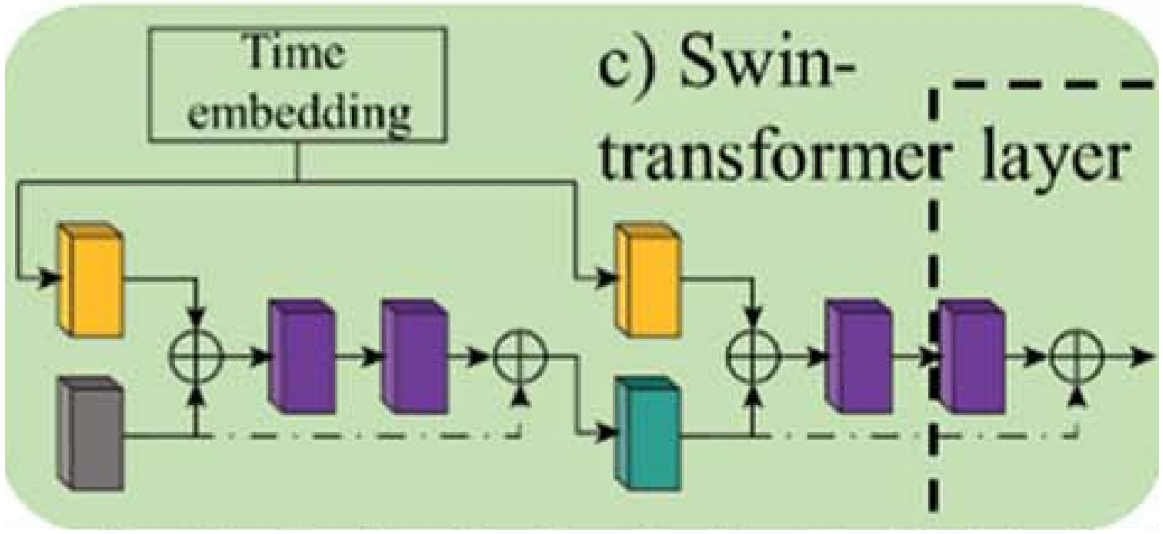
Convolutional Block



- Each up/down-sampling convolutional block combines three sequential residual convolutional sections.
- Each section is built by a timestep expanding layer, an early convolutional layer, and two subsequent residual layers.
- Each convolutional layer has 3*3 spatial filters with stride 1 in all axes. In the last section, a trilinear up/ down-sampling interpolation is deployed before the early convolutional layer. the early convolutional layer accepts the input features to learn local semantic information.

methods

Swin-transfomer Block



- The Swin-transformer block has a similar architecture with the convolutional block: two sequential Swintransformer sections.
- Each section also consists of a timestep expanding layer, which re-embeds the timestep embedding, and an early 3 * 3 convolutional layer, which learns local semantic information from input features. Finally, their outputs are summed up and fed to the following Swin-transformer block, which consists of window-attention (WA) and shifted window-attention (S-WA) modules to capture global information.

What is window-attention (WA) and shifted window-attention (S-WA)?

As shown in figure 1(c)), WA and S-WA modules consist of a window-partition layer to divide the input features F into non-overlapping windows. Then a multi-head self-attention (MHSA) layer, where each head indicates a set of independent weight matrices Q, K, V, is applied to calculate the spatial information within each window. Practically, we split the input feature $F \in \mathbb{R}^{H\times W\times C}$ into $F_{split} \in \mathbb{R}^{\frac{H}{l}\times\frac{W}{l}\times(l^2\times C)}$, to obtain $\frac{H}{l} \times \frac{W}{l}$ 2D windows with size of $l^2 \times C$. For each window W, the MHSA layer performs:

$$head_m = softmax\left(\frac{WQ_m(WK_m)^T}{\sqrt{d}}\right)(WV_m) \tag{1}$$

$$T = Concat\,(head_1,\ldots,head_M)\,O \tag{2}$$

where Q_m, K_m, V_m are the weights matrices of the $m'th$ head, O is another weight matrices, $T \in \mathbb{R}^{l^2\times C}$ is the attention output from the window. By gathering the attention from all the windows, we can finally acquire an attention map $T_{all} \in \mathbb{R}^{\frac{H}{l}\times\frac{W}{l}\times(l^2\times C)}$ and reshaped back to the original size of the non-split feature F. A fully-connected layer is then applied to further refine the attention map.

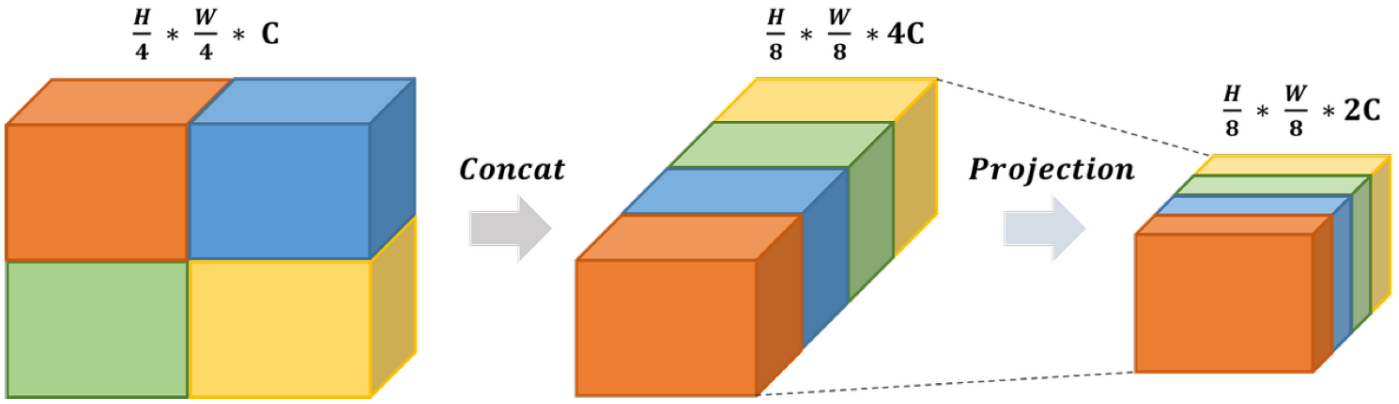
The attention map is then input to the S-WA layer. The W-SA layer is identical to the SA layer, which shifts the window by $\left[\frac{l}{2}, \frac{l}{2}\right]$ in the x and y-axis. Accordingly, the Swin-transformer section can also learn attention across different windows, better capturing global-level information and improving network performance. In summary, the Swin-transformer layer performs the following:

$$F_w = linear\,(WA\,(F)) \tag{3}$$

$$Y = linear\,(S - WA\,(F_w)) \tag{4}$$

where *linear* is fully-connected layer.

What is window-attention (WA) and shifted window-attention (S-WA)?

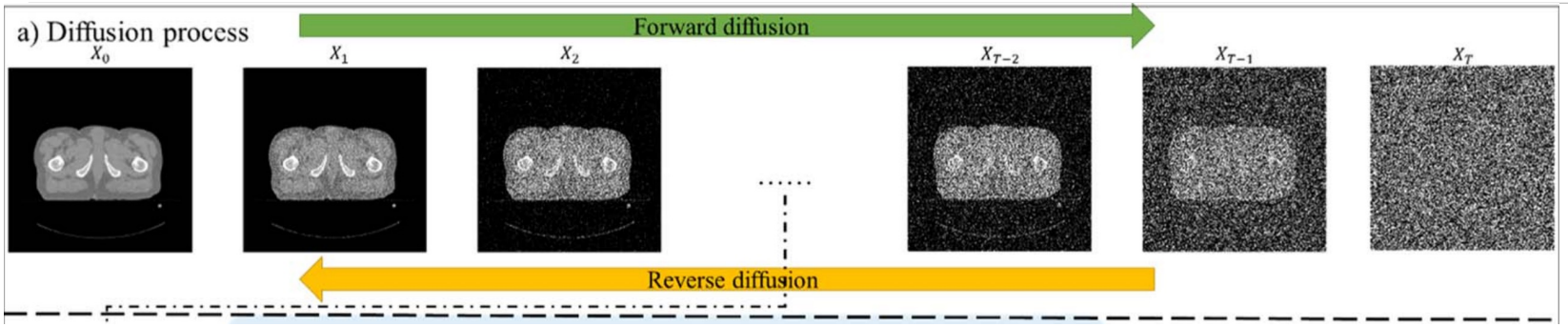


- Stage 1의 출력인 $\frac{H}{4} * \frac{W}{4} * C$의 차원을 **2 * 2 그룹들로 나눕**니다.
- 나눠진 하나의 그룹은 $\frac{H}{8} * \frac{W}{8} * C$의 **차원을 가지고, 4개의 그룹들을 채널을 기준으로 병합**합니다(Concat).
- 병합된 $\frac{H}{8} * \frac{W}{8} * 4C$의 **차원 축소를 위해 절반**인 $2C$의 차원으로 축소합니다.
- 위 과정들은 모든 Stage에서 동일하게 작용합니다.

Part 2,

methods

Diffusion process – forward diffusion



- The noisy image generation process q as a Markov process which that the noisy image at timestep t only depends on the noisy image at timestep t – 1

$$\alpha_t := 1 - \beta_t$$

$$q(X_T|X_0) = \mathcal{N}\left(X_t; \sqrt{\prod_{i=1}^{t} \alpha_i} X_0, \left(1 - \prod_{i=1}^{t} \alpha_i\right) I\right).$$

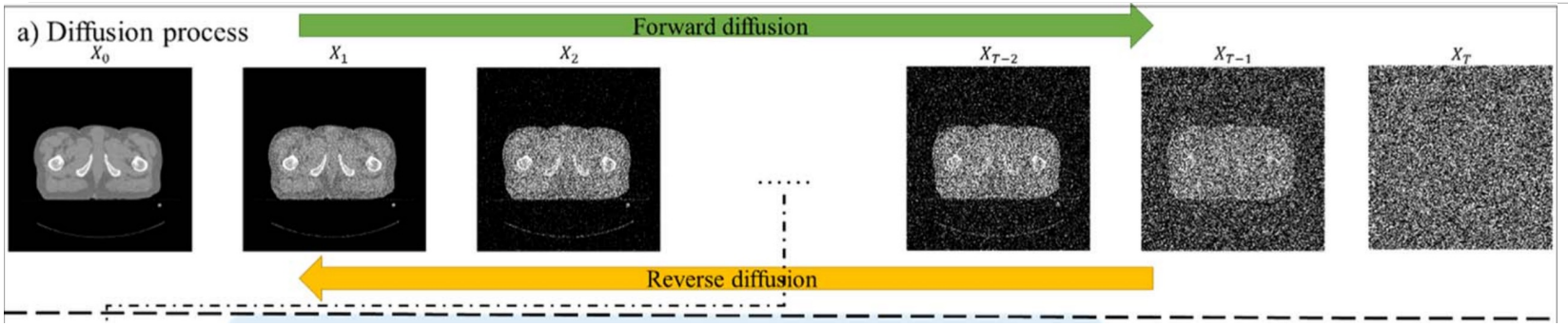
Practically, the noisy image at timestep t is generated as

$$X_t = \sqrt{\prod_{i=1}^{t} \alpha_i} X_0 + \sqrt{1 - \prod_{i=1}^{t} \alpha_i}\, \epsilon_t\Bigg),$$

Part 2,

methods

Diffusion process – reverse diffusion



- The reverse diffusion aims to calculate the conditional probability $q(X_{t-1} | X_t)$ Then, we can denoise a random Gaussian noise X_T to X_{T-1} recursively in reverse direction until we are able to generate a corresponding medical image X_0. The conditional probability can be calculated in a closed form only when conditioned on X_0.

- m_{param} is a mean matrix and S_{param} is a variance matrix of the approximated Gaussian distribution learned by the network param. Once the MT-DDPM learns $p_{param}(X_{t-1} | X_t)$ for all the X_0 existing in the training dataset, by using the reparameterization trick again, we can recursively generate denoised image X^{new}_{t-1} until obtaining the new X^{new}_0 without the needs of conditioning on X^{new}_0. where $epslion_t$ is a random Gaussian noise sampled from N(0,I).

$$X_{t-1}^{new} = \mu_\theta(X_t^{new}, t) + \sigma_\theta(X_t^{new}, t) * \epsilon_t$$

How to estimate mean?

2.2.2.1. Estimating the mean

Directly estimate the mean μ by the MT-DDPM could be the most obvious option. However, learning μ can be a difficult task for the MT-DDPM, since the value of the mean is not constrained, which causes the training to be unstable and therefore detrimental the synthetic image's quality. Alternatively, to improve the network's stability and synthetic image's quality, we optimize the MT-DDPM as a noise predictor, as shown in figure 1(a), instead of a direct mean predictor.

Practically, as demonstrated in (Ho *et al* 2020), the first output of MT-DDPM , ϵ_θ, can be optimized by a mean square error (MSE) loss between the ground truth noise ϵ_t and predicted noise $\epsilon_\theta(X_t,\ t)$:

$$\underset{\epsilon_\theta}{\operatorname{argmin}}\ \mathrm{L}_{\mathrm{mean}} = MSE(\epsilon_t,\ \epsilon_\theta(X_t,\ t)) = E(E_t\ ||\epsilon_t - \epsilon_\theta(X_t,\ t)||_2^2), \tag{14}$$

where E indicates the mean of the square error over all pixels, E_t indicates the mean over all timesteps. Once an accurate ϵ_θ is obtained, we can calculate the mean μ_θ:

$$\mu_\theta(X_t,\ t) = \frac{1}{\sqrt{\alpha_t}}\left(X_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(X_t,\ t)\right) \tag{15}$$

Since then, we can obtain the $\mu_\theta(X_t,\ t)$ which can be used for the new image synthesis.

How to estimate variance?

2.2.2.2. Estimating the variance

Regarding to the variance, following Nicol *et al*'s demonstration (Dhariwal and Nichol 2021), the Σ_θ can be calculated by the pre-scheduled variance β_t and an interpolation coefficient v predicted by the MT-DDPM :

$$\Sigma_\theta(X_t,\ t,\ v_t) = \exp\left(v_t * \log\beta_t + (1 - v_t) * \log\left(\frac{1 - \prod_{i=1}^{t-1}\alpha_i}{1 - \prod_{i=1}^{t}\alpha_i}\beta_t\right)\right) \tag{16}$$

The predicted variance Σ_θ should minimize a variational lower bound (VLB) between the $p_\theta(X_{t-1}|X_t)$ and $q(X_{t-1}|X_t,\ X_0)$.

Practically, we summarize the VLB loss (Nichol and Dhariwal 2021) at timestep t as:

$$\text{argmin}_v\ \mathrm{L}_{\text{var}} = E(E_t(\mathrm{L}_{\text{VLB}}(\mathrm{t}))) \tag{17}$$

And the details of the VLB loss is summarized as:

$$L_t := \frac{1}{2}(\log\Sigma_\theta(X_t,\ t,\ v_t) - \log\Sigma(X_t,\ t) + \frac{\log\Sigma(X_t,\ t)}{\log\Sigma_\theta(X_t,\ t,\ v_t)} + \frac{(\mu_\theta(X_t,\ t) - \mu(X_t,\ t))^2}{\Sigma_\theta(X_t,\ t,\ v_t)}), \tag{18}$$

$$L_{high} := (\Sigma_\theta(X_t,\ t,\ v_t) * GELU(X_t - \mu_\theta + \delta), \tag{19}$$

$$L_{low} := (\Sigma_\theta(X_t,\ t,\ v_t) * GELU(X_t - \mu_\theta - \delta), \tag{20}$$

$$\mathrm{L}_{\text{VLB}}(\mathrm{t}) = \begin{cases} L_t,\ t > 0 \\ \log(L_{high} - L_{low}),\ t = 0\ and\ 1 - \delta < X_t < \delta \\ \log(1 - L_{low}),\ t = 0\ and\ X_t > \delta \\ \log(L_{high}),\ t = 0\ and\ X_t < 1 - \delta \end{cases} \tag{21}$$

where δ is the pixel intensity threshold set to $\frac{1}{256}$. The *GELU* (Gaussian Error Linear Units function) is a fast approximation of Cumulative Distribution Function for Gaussian Distribution (Hendrycks and Gimpel 2016). Notice that there is only one learnable parameter v_t in the equation and the other parameters are known. Once an optimal v_t is obtained, we can obtain the Σ_θ for the image synthesis. The overall optimization function is presented as

$$L = L_{mean} + \gamma L_{var}, \tag{22}$$

where γ is a weighting parameter which is empirically selected as 0.01.

Part 2,

methods

Synthetic image generation

- This paper evenly spaced numbers between 1 and 4000 (the training timestep T) by 500 timesteps and denote the new number set as S. By denoting the new Gaussian noisy sample as X_S the MT-DDPM can generate a less noisy image X_{S-1}. all we need is to provide epsilon$_s$ ~ N(0, I).
- By recursively generate the X_{S-1} until we obtain the noise-free image X_0.

$$X_{s-1} = \frac{1}{\sqrt{1-\beta_s}}\left(X_s - \frac{\beta_s}{\sqrt{1-\overline{\alpha_s}}}\epsilon_\theta(X_s, s)\right)$$

$$+\exp\left(v_s * \log\beta_s + (1-v_s) * \log\left(\frac{1-\prod_{i=1}^{s-1}\alpha_i}{1-\prod_{i=1}^{s}\alpha_i}\beta_s\right)\right) * \epsilon_s$$

Datasets

List of dataset and preprocessing

3.1. Public dataset: chest x-rays dataset

The chest x-rays dataset for our synthesis was built from the NIH Chest x-rays dataset (Wang *et al* 2017), which consists of 112120 chest x-rays 2D images saved in PNG format. The first 5500 x-rays images were selected to train the MT-DDPM network. Each training image was then resampled to resolution of 256×256. For training, the whole dataset was normalized to intensity range from $[0, 255]$ to $[-1, 1]$. For inference, the intensities of the synthetic images were clipped to $[-1, 1]$.

3.2. Public dataset: heart MRI dataset

We collected 1902 2D heart MRIs from 3D MR scans of the Automated Cardiac Diagnosis Challenge (ACDC) dataset (Bernard *et al* 2018). The MR scans were collected as a series of axial slices cover the heart base to the left ventricle with an axial spacing of 5 to 8 millimeters (mm). The 2D axial slices were then separated to form the training dataset. The axial slices of all scans were resampled to 2×2 mm, and boundary-padded to have a consistent resolution of 256×256. In training, the intensities of each scan were normalized to $[-1, 1]$.

3.3. Institutional dataset: pelvic CT dataset

The dataset contains 4157 2D CT slices collected from 94 patients with prostate cancer treated by external beam radiation therapy in our institution. Institutional review board approval was obtained, and informed consent was not required for this Health Insurance Portability and Accountability Act compliant retrospective analysis. All patient scans were acquired by CT simulation using a Siemens SOMATOM Definition AS CT scanner with a resolution of $512 \times 512 \times 128$ and voxel size of $0.977 \times 0.977 \times 2$ mm. The scans were then resampled to voxel size of $2 \times 2 \times 2$ mm. To reduce irrelevant body volumes, we only collect the central axial slices, covering the bladder, prostate, rectum, left femoral head, and right femoral head. The slice collection was guided by the patient organ segmentation map identified by two expert physicians. The slices were boundary-padded to has a consistent resolution of 256×256. For training, the pixel intensities of all scans were jointly normalized from attenuation coefficient values $[-1024, 3012]$ to $[0, 1]$. For inference, the intensities of synthetic images were clipped to $[0, 1]$ and converted back to attenuation coefficient values.

3.4. Public dataset: abdomen CT dataset

The dataset contains 2178 abdomen CT slices collected from the Beyond the Cranial Vault (BTCV) dataset (Landman *et al* 2015), published in the proceedings of the 2015 Medical Image Computing and Computer-Assisted Intervention conference in 2015. The dataset contains 30 patient scans with a resolution of $512 \times 512 \times [80\tilde{}225]$ and voxel size of $0.76 \times 0.76 \times 3$ mm. The patient scans were firstly resampled to a voxel size of $1.52 \times 1.52 \times 3$ mm and hence each axial slice has resolution of 256×256. With the guidance of the provided organ annotations, we only collected the slices with at least one organ appearing. The same normalization and attenuation coefficient value recovering strategies adopted in the institutional pelvic CT dataset were implemented in training and inference.

Performance evaluation

Visual turing assessment

- 3 medical physicists, were performed between the real and synthetic images to evaluate the quality of synthetic images.
- All physicists independently provided their results.
- 25 synthetic images + 25 real images

- Accuracy values of 50% represent random guess from the observers, which indicates a more realistic visual appearance of the synthetic images among the originals.

Performance evaluation

Quantitative comparison – intensity distribution evaluation

- Inception Score (IS): The higher IS indicates better quality and diversity.
- Fréchet Inception Distance Score (FID): the ower FID indicates a better quality.
- Feature Distribution Similarity (FDS): T-SNE algorithm are applied to embed each image into a two-point feature space to generated compressed features. Then Gaussian models are fitted to the features from real and synthetic data, respectively. The FDS is measured by KL divergence between the real and synthetic Gaussian distributions, where smaller KL indicates more similar distributions, therefore higher feature similarity and better quality.

$$IS(p_{gen}, p_{dis}) := \exp\left(\mathbb{E}_{x\sim p_{gen}}\left[D_{KL}\left(p_{dis}(\cdot|x)\| \int p_{dis}(\cdot|x)p_{gen}(x)dx\right)\right]\right)$$

$$FID = |\mu_T - \mu_G|^2 + Tr(\Sigma_T + \Sigma_G - 2(\Sigma_T\Sigma_G)^{1/2})$$

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x\in\mathcal{X}} P(x)\ \log\left(\frac{P(x)}{Q(x)}\right)$$

Performance evaluation

Quantitative comparison – diversity evaluation

- Diversity Score (DS): For each synthetic image, we calculated a SSIMs between it and every other synthetic image to find the most similar pair. Then the SSIMs among all the synthetic images with their corresponding most similar pair are reported nearest SSIMs. We assume that if the real images have most of their nearest SSIMs in a specific range, the synthetic images should also have their nearest SSIMs concentrated in that same range.
- nearest SSIM difference, is calculated using the KL divergence between the distribution of nearest SSIMs in real and synthetic images. a low DS suggests that the synthetic images have a similar diversity to the real images.

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Results

Quantitative results

(A) visual turing study - The average accuracy values for chest x-rays and heart MRIs are close to 0.5, the average sensitivity value is 0.65, and the average specificity value is 0.45. The result indicates that physicists had difficulty identifying real or synthetic images. The average accuracies for pelvic and abdomen CT images are around 0.75 and 0.65. With relatively high sensitivities, the specificities are still close to 0.5.

(B) MT-DDPM and competing networks Regarding two comprehensive evaluations – the highest IS and the lowest FID among all competing algorithms from all datasets. In addition, the MT-DDPM's synthetic images have the lowest KL divergence and the lowest DS in all modalities

	Chest x-rays	Heart MRI	Pelvic CT	Abdomen CT
Accuracy	0.57	0.56	0.77	0.64
Sensitivity	0.70	0.64	0.97	0.83
specificity	0.48	0.45	0.59	0.46

Part 5,

Results

Chest X-ray

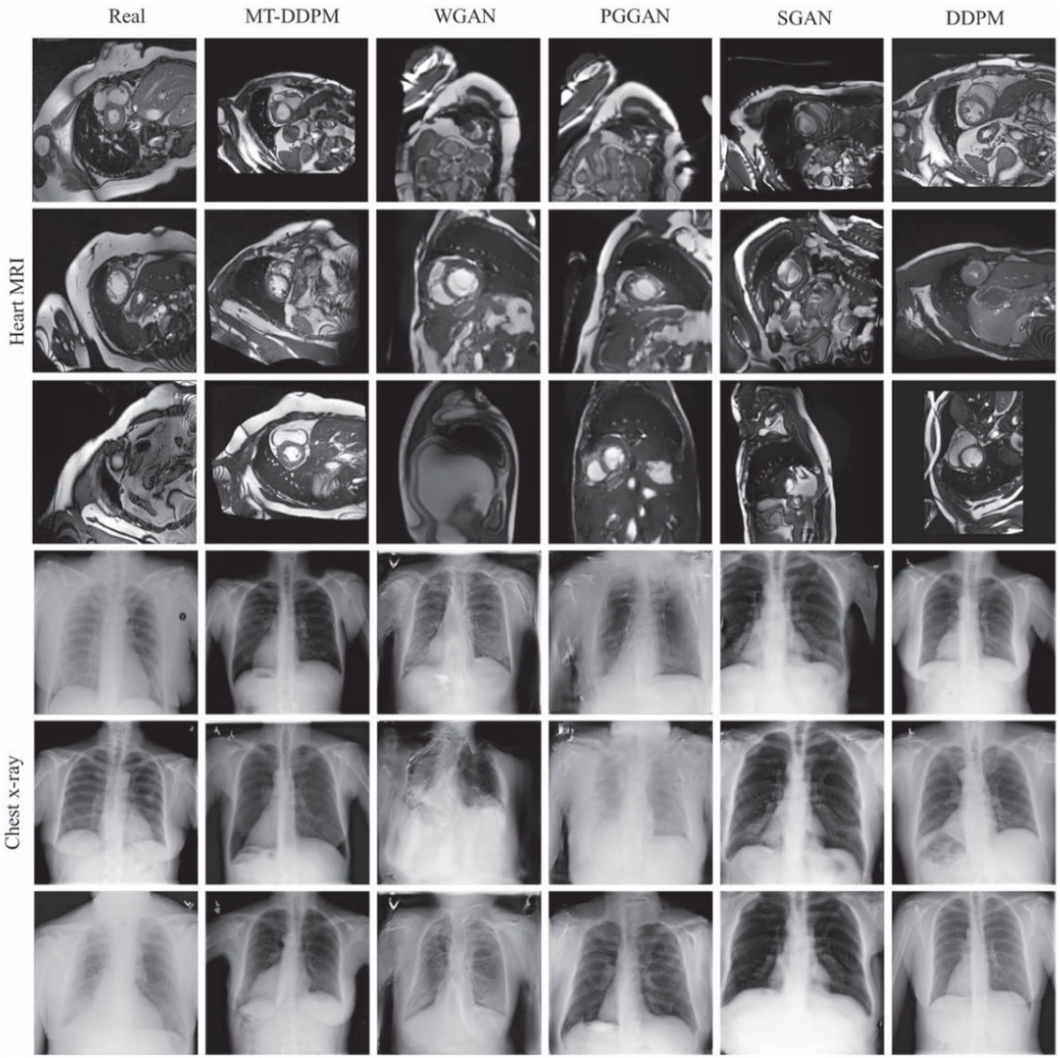


Figure 2. Synthetic and real images from chest x-rays, heart MRI. In each dataset, the real images, the synthetic images of the proposed MT-DDPM, WGAN, PGGAN, SGAN, and DDPM are presented in column-wise. Three slices of chest x-rays (row #1 to #3), heart MRIs (row #4 to #6) are displayed. Notice that for visualization purposes, the examples from different modalities were resized to the same resolution but could with different pixel spacings.

Part 5,

Results

Quantitative analysis

IS (↑)	Chest x-rays	Heart MRI	Pelvic CT	Abdomen CT
MT-DDPM	**2.07 ± 0.13**	**2.87 ± 0.16**	**2.07 ± 0.07**	**2.10 ± 0.12**
WGAN	2.05 ± 0.09	2.31 ± 0.11	1.77 ± 0.08	1.91 ± 0.06
PGGAN	2.06 ± 0.05	2.32 ± 0.10	1.97 ± 0.06	1.99 ± 0.06
SGAN	1.86 ± 0.09	2.43 ± 0.17	1.82 ± 0.05	2.01 ± 0.07
DDPM	2.04 ± 0.10	2.64 ± 0.14	2.00 ± 0.10	2.02 ± 0.14
FID (↓)	Chest x-rays	Heart MRI	Pelvic CT	Abdomen CT
MT-DDPM	**22.33**	**58.81**	**30.40**	**37.54**
WGAN	140.71	200.95	93.17	198.57
PGGAN	130.26	202.62	135.27	100.81
SGAN	182.80	240.86	100.62	170.05
DDPM	26.71	65.09	87.09	51.66
FDS (Feature KL) (↓)	Chest x-rays	Heart MRI	Pelvic CT	Abdomen CT
MT-DDPM	0.56	**0.15**	**0.03**	0.05
WGAN	2.82	2.96	0.77	1.50
PGGAN	**0.05**	3.06	2.29	**0.03**
SGAN	5.30	3.18	2.46	1.95
DDPM	0.73	0.31	2.77	0.10
DS (nearest SSIM diff) (↓)	Chest x-rays	Heart MRI	Pelvic CT	Abdomen CT
MT-DDPM	**0.29**	**0.94**	**1.84**	**0.36**
WGAN	0.95	6.32	2.44	2.94
PGGAN	1.26	1.19	17.26	1.19
SGAN	2.09	6.35	14.95	2.84
DDPM	1.11	2.07	5.24	0.70

Discussion

Conclusion

- The proposed MT-DDPM framework has demonstrated the promising quality of its synthetic images, which can potentially be used as training dataset in DL model for specific tasks.

- In the application evaluation, the MT-DDPM synthetic data achieved lower but statistically **comparable quantitative performance compared to the real images**. It was also able to be used as new way of data augmentation to enrich the existing real dataset and improve the training of the application network quantitatively. It outperforms commonly used existing generative models such as WGAN, PGGAN, SGAN, and DDPM, achieving state-of-the-art image quality and diversity.

- But MT-DDPM has problems. In this experiment it takes time **220 second** even WGAN and PGGAN takes **0.18 second** It has **$1.6*10^9$** parameters while the WGAN and PGGAN contain a total of parameters **$8.6*10^7$** for both the discriminator and generator. These limitations prevented us from extending the existing idea into 3D volume synthesis. In the future we have to address these problem and makes model generates 3D synthesis.