A Self-Pretraining Paradigm For CBCT-CT Translation

Runqi Wang[1], Zheng Zhang[1], Ruizhi Hou[1], Lei Xiang[1], and Tao Song[1]

Subtle Medical, Shanghai, China
elliotqii@gmail.com , {zheng,ruizhi,lei,taosong}@subtlemedical.cn

Abstract. The impact of medical imaging on oncological patients' diagnosis and therapy has grown significantly over the last decades. Especially in radiotherapy (RT), imaging plays a crucial role in the entire workflow, from treatment simulation to patient positioning and monitoring. Thus, we proposed a self-pretraining paradigm for handling some unpaired problems. Then, we fintune the pretrained model on the paired dataset and achieve good performance. A gradual training schedule is also adopted in this method. Specifically, our method could be achieved by following steps: 1) Self-supervised pertaining with masked images. 2) Gradually fintuning with different losses on paired CBCT-CT dataset. 3) Weighted ensemble for different types of models. Experiments on SynthRAD Challenge dataset show that our method is effective.

Keywords: CBCT-to-CT synthesis · Masked Autoencoder · Transformer.

1 Introduction

Cone-beam computed tomography (CBCT) images are widely used in image-guided radiotherapy (IGRT), however its clinical application is limited due to the low image quality. The synthesis of Computed Tomography (CT) images from Cone-Beam Computed Tomography (CBCT) scans has gained significant attention in medical imaging research. Recent advances in deep learning techniques have shown promising results in generating high-quality CT images from low-dose CBCT scans. In this study, we aim to explore the effectiveness of pretraining a Masked Auto-Encoder[1] (MAE) using the SwinIR[2] architecture for translating CBCT image to synthetic CT (sCT) image that preserves both CT image quality and CBCT anatomical.

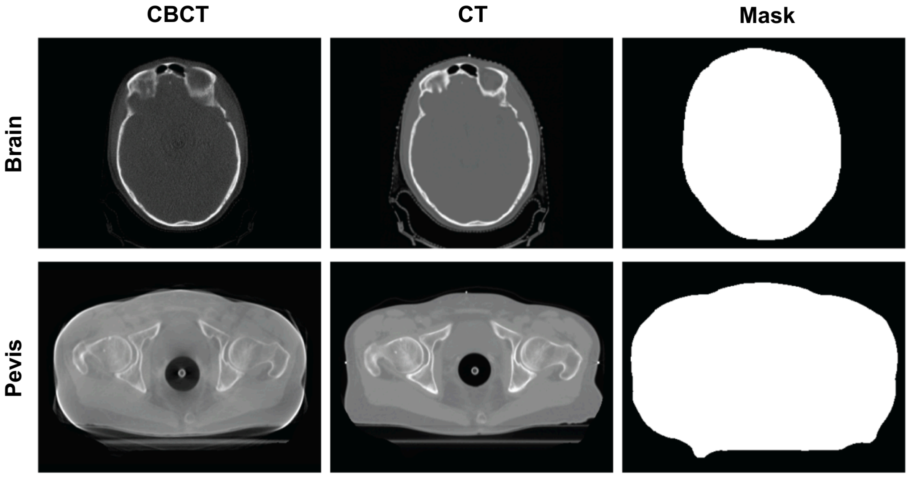

Fig. 1. Synthesis CT needs to preserve both CT image quality and CBCT anatomical

2 Method

We proposed a Swin Transformer synthesis network based on masked autoencoder pertaining paradigm. Also, a global feature branch is introduced to better capture the global features of the image, thereby guiding image synthesis. So, we divide this section into three parts: masked autoencoder, swin transformer and group propagation block.

2.1 Masked Autoencoder

Recently, MAE has gained recognition as a highly effective strategy for self-supervised learning in diverse computer vision applications. It is specifically designed to learn meaningful representations from input data and generate high-quality output images. The masked autoencoder consists of an encoder and a decoder, which work together to reconstruct the input images while learning a compressed representation in the middle. The masked autoencoder architecture is flexible and can be adapted to different medical image synthesis tasks, including CBCT to CT synthesis. By training the model on a large dataset of paired medical images, the masked autoencoder can learn to generate high-quality synthetic images that closely resemble the ground truth images. This technique has shown promising results in various medical imaging applications, including image denoising, super-resolution, and image synthesis.

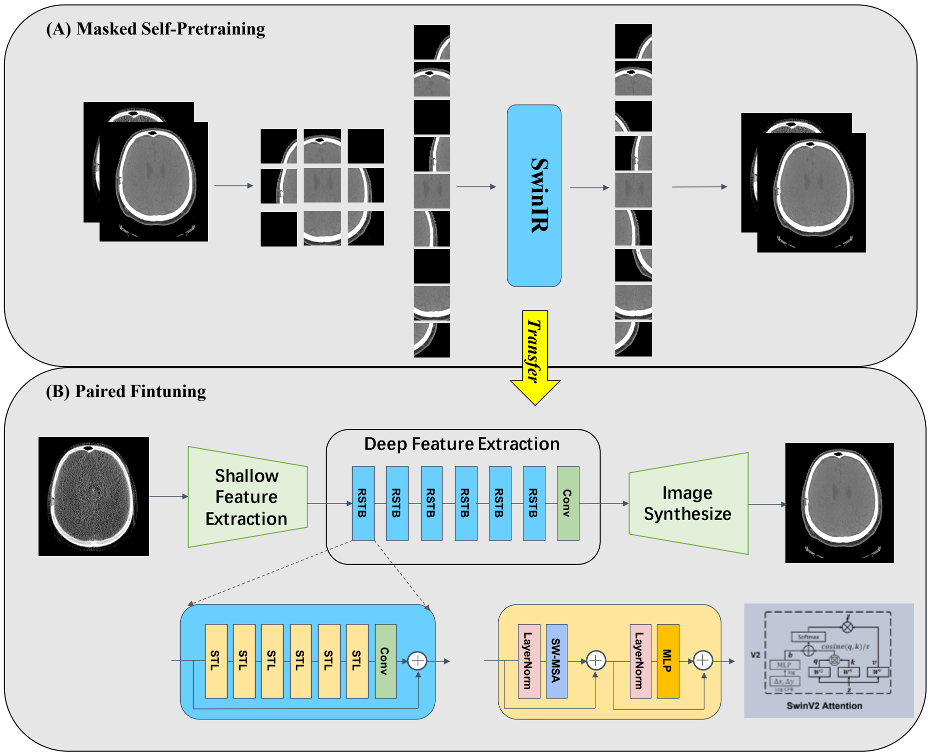

Fig. 2. Our initial pipeline for CT synthesis. Further, we also apply SwinV2 architecture and add a global branch for capturing the global features.

2.2 SwinIR

The Swin transformer has innovatively incorporated hierarchical attention with shifted windows, enabling the fusion of contextual information while mitigating the computational burden. Among its notable applications in the domain of low-level vision, SwinIR[2] stands out. Figure 2 visually illustrates the architecture, comprising multiple convolutional blocks and Swin transformer modules. Specifically, for a Swin transformer module, an input token sequence $T \in R^{b\times w\times n\times do}$ is first layer-normalized (LN). Here b is the batch size, w is the window number, n is the number of tokens, and do is the token embedding dimension.

2.3 Group Propagation Block

Apart from SwinV2 architecture[4], we also try to apply group propagation blocks [3] into our Swin Transformer.In each GP block, the features are first grouped by a fixed number of learnable group tokens; then the group is propagated to exchange global information between the grouped features; finally, the global information in the updated grouped features is returned to the image feature through the converter decoder. For the GP Block module, image features are grouped with a fixed number of learnable group tokens. Then use the MLP-Mixer module to exchange global information and update the characteristics of the grouping. Next, the grouped features are queried and connected in series with the image features to pass the global information to each image feature. Finally, the updated image features are converted through the feedforward network to generate output.

2.4 Loss Function

The loss functions of our methods are constructed by a gradual scheduler. Specifically, in the first stage, the $L1$ loss is used with the learning rate of 1×10^{-4} . Data augmentation includes the horizontal flip, the vertical flip, and the rotation with 90 degrees. In the second stage, the MSE loss with a learning rate of 2×10^{-5} is applied. Finally, the perceptual loss is adopted, and the learning rate is set to 1×10^{-5}.

3 Experiments

3.1 Datasets

We only use the synthesizing computed tomography for radiotherapy challenge (SynthRAD 2023) dataset to evaluate the efficiency of our method. The following pre-processing steps were performed on the data: 1. Conversion from dicom to compressed nifti. 2. Rigid registration between CT and CBCT. 3. Anonymization (face removal, only for brain patients). 4. Patient outline segmentation (provided as a binary mask). 5. Crop MR/CBCT, CT and mask to remove background and reduce file sizes.

Table 1. The number of subjects in each phase.

Subdataset	Center-A	Center-B	Center-C
Training set	60	60	60
Validation set*	10	10	10
Test set*	20	20	20

3.2 Experiment Settings

All experiments in this section are performed on Ubuntu 20.04.5 LTS, with Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz. All models are run with NVIDIA GeForce RTX 3090 using PyTorch. Here're some experiment settings:

During pretraining, the batchsize is 24, masked patch size is 8 with 75% masked rate. The model depths for the SwinIR are [6, 6, 6, 6, 6, 6] with 180 embedding dimensions. During finetuning, the batchsize is 16 with cropping. The crop size is set to 160. We use Adam optimizer with a 1e-4 learning rate. All the models are trained with 100 epochs. After obtaining the results of different models, we first calculate the mean value of each output. We use the difference between each output and the mean value to get the weight of the corresponding output. Then, we norm the weight to ensemble all outputs.

3.3 Metrics

The metrics measuring the accuracy of the algorithm are masked Peak Signal-to-Noise Ratio (PSNR), Mean absolute error (MAE), and Structural similarity index (SSIM) between sCT and CT.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|CT_i - sCT_i| \tag{1}$$

where n is the number of voxels in the mask.

$$PSNR = 10\log_{10}\left(\frac{Q^2}{\frac{1}{n}\sum_{i=1}^{n}(CT_i - sCT_i)^2}\right) \tag{2}$$

where n is the number of voxels in the mask, and Q is the typical range of voxel intensities in the CTs (3000 HU).

$$SSIM = \frac{(2\mu_{CT}\mu_{sCT} + C_1)(2\delta + C_2)}{(\mu_{CT}^2 + \mu_{sCT}^2 + C_1)(\delta_{CT}^2 + \delta_{sCT}^2 + C_2)} \tag{3}$$

where μ is the mean pixel value, δ is the variance . $C1 = (0.01Q)^2$ and $C2 = (0.03Q)^2$ are two variables to stabilize the division with weak denominators, where Q is the typical range of voxel intensities in the CTs (3000 HU).

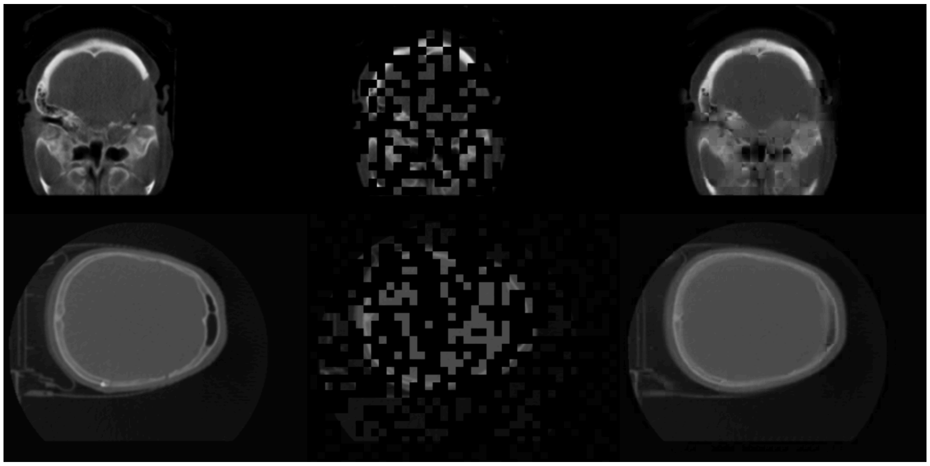

Fig. 3. The reconstruction visualization of pretrained model.

Table 2. PSNR, SSIM, MAE scores with different experiments

	PSNR ↑	SSIM ↑	MAE ↓
A) CNN-baseline	26.27	0.8402	60.34
B) SwinIR-baseline	27.37	0.8607	55.88
C) Adjust some parameters	27.40	0.8642	55.76
D) SwinV2 and global branch	27.43	0.8649	55.03
E) Ensemble	27.89	0.8650	54.84
F) Weighted Ensemble	27.89	0.8670	54.54

3.4 Results

In this part, we'll present the performance on synthrad challenge and visualize our results of synthesized CT (sCT). In Fig.3 we can see that the model has strong reconstruction ability after self-supervised pertaining. Thus, it may benefit solving unpaired problems using the encoded features. Table 2. shows the PSNR, SSIM and MAE scores of our methods during this challenge.

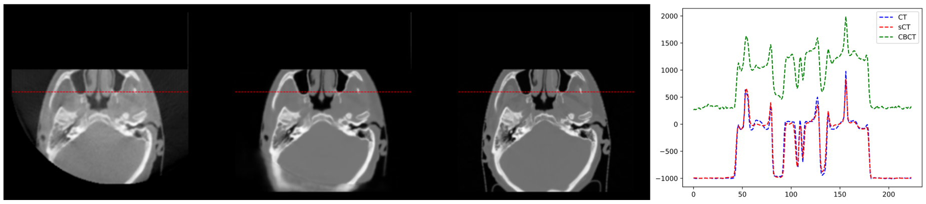

Fig. 4. The side-by-side comparison of CBCT, sCT, CT for a validation patient; The right sub-graph shows the line profile of the red line.

4 Conclusion

In this paper, we investigated the application of a masked auto-encoder pretrained SwinIR model for CBCT to CT synthesis. The results demonstrate the potential of the proposed approach in generating high-quality CT images from CBCT scans. The proposed method shows promise in capturing relevant visual features and producing accurate CT images. Further research and refinement of the methodology could lead to improved CBCT-to-CT synthesis techniques with significant clinical implications.

References

1. He, Kaiming, et al. Masked Autoencoders are Scalable Vision Learners. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

2. Liang, Jingyun, et al. SwinIR: Image Restoration Using Swin Transformer. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
3. Yang, Chenhongyi, et al. GPViT: A High Resolution Non-Hierarchical Vision Transformer with Group Propagation. Proceedings of the International Conference on Learning Representations. 2023.
4. Liu, Ze, et al. Swin Transformer v2: Scaling up Capacity and Resolution. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.