Supplementary Materials

1 Implementation Details

All the experiments are implemented in Python 3.8.12 with Pytorch 2.0.1 on an NVIDIA GeForce RTX 4090 GPU card with 24GiB of memory. In addition, the DiffuCE framework is based on the HuggingFace libraries with HuggingFace diffusers 0.19.3[1](Apache-2.0 license). We leverage the Adam optimizer with a learning rate of 1×10^{-5} for the training of DBE, CDD, and CRD, while the learning rate of the discriminator in the CRD training is set to 5×10^{-6}. The training epochs of the DBE, CDD, and CRD are 600, 100, and 160, respectively.

1.1 Pretrained Encoder & Decoder

The DiffuCE framework is derived from the Stable Diffusion framework, and its encoder and decoder are from a pretrained variational autoencoder(VAE). The ratio of the height and width of the input image size and the latent embedding is $8:1$, which further accelerates the inference speed of the denoising UNet since the smaller image can be processed faster.

In the training phase of the Domain Bridging Encoder(DBE) and Conditional Refinement Decoder(CRD), we manually integrate a LoRA layer into the only attention kernel of the pretrained VAE. With this technique, the GPU usage in the training is significantly reduced compared with full parameter fine-tuning.

1.2 Denoising UNet

The Conditional Diffusion Denoiser(CDD) is derived from the UNet of the Stable Diffusion framework. With LoRA fine-tuning, the GPU usage is significantly reduced like the training of the DBE and CRD.

With cross-attention design, conditions such as bone, lucent area, and low-frequency components of wavelet transform can be integrated into the CDD, leading to a more controllable and accurate prediction.

2 Algorithms

In this section, we display the details of the training algorithms for every part of the DiffuCE pipeline. The training of the entire DiffuCE framework is in three stages. First, the Conditional Diffusion Denoiser(CDD) is trained with real CT images, learning to generate high-quality medical images with given conditions, and the training algorithm is described in **Algorithm 1**.

Second, the DBE is trained with pseudo-CBCT and CT pairwise dataset, learning to bridge the distribution gap between the CBCT and CT images, and the training algorithm is described in **Algorithm 2**.

Third, the CRD is also trained with pseudo-CBCT, CT, and corresponding conditions, learning to decode the denoised latent to the pixel space with guidance from the condition. The training algorithm is described in **Algorithm 3**.

Lastly, the inference of DiffuCE can also be divided into three steps: (1) The DBE adds noise to the given CBCT images and bridges it to the CT distribution. (2) The CDD gradually removes the added noise, while preserving the details in every iteration with given constraints obtained from f_{cp}. (3) The CRD decodes the sample from latent space back to pixel space with given constraints to guide again. The inference algorithm is described in **Algorithm 4**.

Algorithm 1 Training algorithm of the CDD

Input: pre-trained diffusion models θ_{diff}, encoder θ_{enc}, CT dataset D, constraints preprocess f_{cp}, stop gradient operator sg, learning rate η

Output: CDD θ_{CDD}.

Let $\theta_{CDD} \leftarrow \theta_{diff}$.
while not converged **do**
  Sample from CT dataset $x_{ct} \sim D$
  Produce constraints $c \leftarrow f_{cp}(x_{ct})$
  Sample noise $z \leftarrow \mathcal{N}(0, \mathrm{I})$
  Sample timestep $t \leftarrow \mathcal{U}(0, \mathrm{T})$
  Encode latent embeddings $\epsilon_{ct} \leftarrow sg(\theta_{enc}(x_{ct}, z, t))$
  Compute the noise level $\hat{\epsilon} \leftarrow \theta_{CDD}(\epsilon_{ct}, c)$
  $\theta_{CDD} \leftarrow \theta_{CDD} - \eta\nabla(\|\hat{\epsilon} - z\|_2^2)$
end while
return θ_{CDD}

Algorithm 2 Training algorithm of the DBE

Input: down-sample function f_{down}, pre-trained encoder θ_{enc}, CT dataset D. alignment loss l_a, stop gradient operator sg, learning rate η.

Output: DBE θ_{DBE}.

Let $\theta_{DBE} \leftarrow \theta_{enc}$.
while not converged **do**
  Sample from CT dataset $x_{ct} \sim D$
  Produce pseudo-CBCT $x_{cbct} \leftarrow f_{down}(x_{ct})$
  Encode CT latent embedding $\epsilon_{ct} \leftarrow \theta_{enc}(x_{ct})$
  Encode CBCT latent embedding $\epsilon_{cbct} \leftarrow \theta_{DBE}(x_{cbct})$
  Compute the alignment loss.
  $\mathcal{L}_a \leftarrow l_a(sg(\epsilon_{ct}), \epsilon_{cbct})$
  $\theta_{DBE} \leftarrow \theta_{DBE} - \eta\nabla\mathcal{L}_a$
end while
return θ_{DBE}

Algorithm 3 Training algorithm of the CRD

Input:down-sample function f_{down}, pre-trained decoder θ_{dec}, fixed DBE and CDD $\theta_{DBE}, \theta_{CDD}$, CT dataset D, stop gradient operator sg, learning rate η, constraint preprocessing f_{cp}, adversarial loss l_{adv}, perceptual loss $l_{percept}$, condition loss l_{cons}

Output: CRD θ_{CRD}.

Let $\theta_{CRD} \leftarrow \theta_{dec}$.
while not converged **do**
  Sample from CT dataset $x_{ct} \sim D$
  Produce constraints $c \leftarrow f_{cp}(x_{ct})$
  Produce pseudo-CBCT $x_{cbct} \leftarrow f_{down}(x_{ct})$
  Get reconstructed sample.
  $x_{recon} \leftarrow \theta_{CRD}(sg(\theta_{CDD}(\theta_{DBE}(x_{cbct}), c)), c)$
  Compute the loss
  $\mathcal{L}_{adv} \leftarrow l_{adv}(x_{recon}, x_{ct})$
  $\mathcal{L}_{percept} \leftarrow l_{percept}(x_{recon}, x_{ct})$
  $\mathcal{L}_{cond} \leftarrow l_{cond}(x_{recon}, c)$
  $\mathcal{L} \leftarrow \mathcal{L}_{adv} + \mathcal{L}_{percept} + \mathcal{L}_{cond}$
  $\theta_{CRD} \leftarrow \theta_{CRD} - \eta\nabla\mathcal{L}$
end while
return θ_{CRD}

Algorithm 4 Inference algorithm of the DiffuCE framework

Input:fixed DBE, CDD, and CRD $\theta_{DBE}, \theta_{CDD}, \theta_{CRD}$, CBCT image x, constraint preprocessing f_{cp}, strength s

Output: reconstructed sample y

Produce constraints $c \leftarrow f_{cp}(x)$
Get bridged embedding $\hat{x} \leftarrow \theta_{DBE}(x)$
for $k = s$ to 1 **do**
    Denoise $\hat{x} \leftarrow \theta_{CDD}(\hat{x}, c, k)$
end for
Reconstruct to pixel space $y \leftarrow \theta_{CRD}(\hat{x}, c)$
return y

3 Pseudo-CBCT

The pseudo-CBCT is an ideal low-quality sample to a high-quality CT image. To train the model in a supervised manner, the pairwise dataset is essential, which is almost impossible to obtain in a practical clinical situation. The clinicians are either unable to obtain high-quality images or do not need to obtain low-quality images since high-quality images are always available. To produce pairwise training data, we employ a method inspired by "A streak artifact reduction algorithm in sparse-view CT using a self-supervised neural representation." In this section, we initially introduce the sinogram and its relationship with the CT image. Subsequently, we explain the practical algorithm used for pseudo-CBCT production. Finally, we showcase some paired data from our training dataset.

3.1 Sinogram

A typical medical image scanning involves observations from various angles, and the raw data representation from such scans is termed a sinogram. The sinogram can be reconstructed into the image domain using filter backprojection (FBP), a technique that combines signal processing, filtering, and linear transformation. FBP enables the reconstruction of a full-view sinogram, incorporating enriched scanning information from multiple angles with high quality. However, artifacts may emerge during the FBP process when dealing with down-sampled sinograms, which contain less information from limited angles. This sparsity in data is leveraged in our artifact generation algorithm.

3.2 Artifact Generation Algorithm

The idea beyond the artifact generation algorithm(AGA) is to manually produce the sparsity from the full-view sinogram. Initially, we have the original CT image x_{full} sampled from the CT image dataset D_{full}, and the sinogram of x_{full}, denoted as s_{full}, can be obtained by performing an inverse FBP process FBP^{-1}. The sparsity of s_{full} can be easily done by filling zeros to some columns, and a sparse version of data can be obtained, denoted as s_{sparse}. The selection of columns can be determined or random, depending on whether the actual scanning pattern is known or not. In our case, the scanning patterns from the CT imaging machines are unknown, thus we randomly drop 40% of data to create the sparsity. Practically, we use a mask $\mathcal{M}_{drop}$ to perform the dropping operation. Finally, we use FBP to transform the data from the sinogram domain back to the image domain, and the reconstructed image is denoted as x_{sparse}.

$$x_{full} \sim D_{full}, \tag{1}$$

$$s_{full} = FBP^{-1}(x_{full}), \tag{2}$$

$$s_{sparse} = \mathcal{M}_{drop} \times s_{full}, \tag{3}$$

$$x_{sparse} = FBP(s_{sparse}) \tag{4}$$

See **Figure 6** for the pipeline overview, and **Figure 7** for some visual demonstrations.

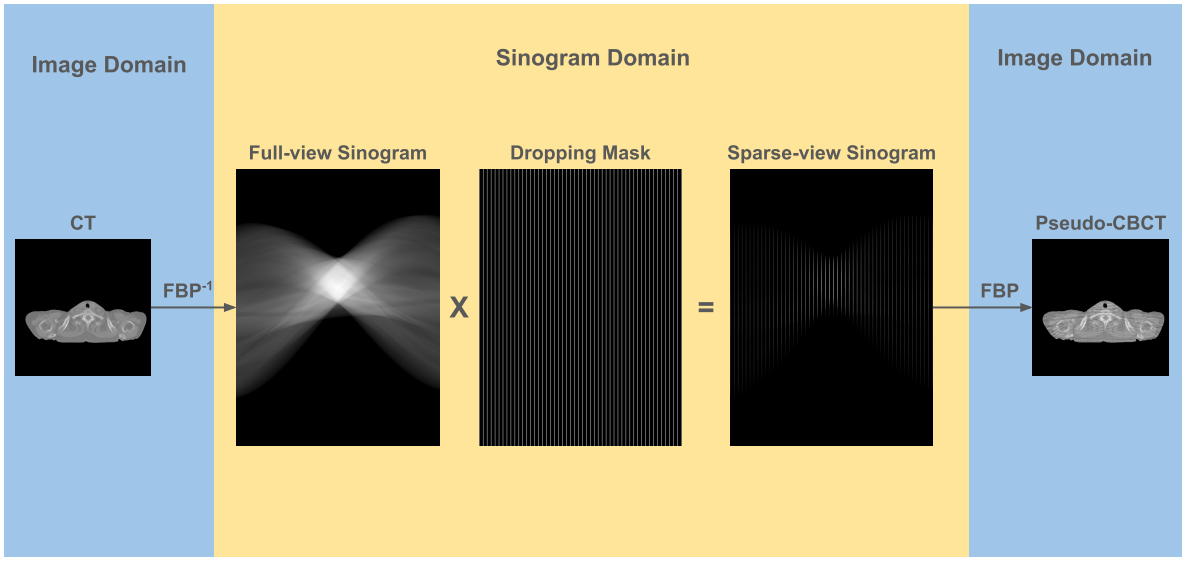

Figure 6: **AGA pipeline.** The fundamental concept behind the Artifact Generation Algorithm (AGA) is to intentionally introduce sparsity by selectively dropping data in the sinogram domain. The transformation between the image and sinogram domain is facilitated by Filter BackProjection (FBP). We employ a mask to govern the dropping ratio and location information, enabling the creation of a down-sampled sinogram. This down-sampled sinogram is obtained by multiplying the full-view sinogram with the mask. Details of the AGA can be found in **Supplementary Materials Section 1.2**.

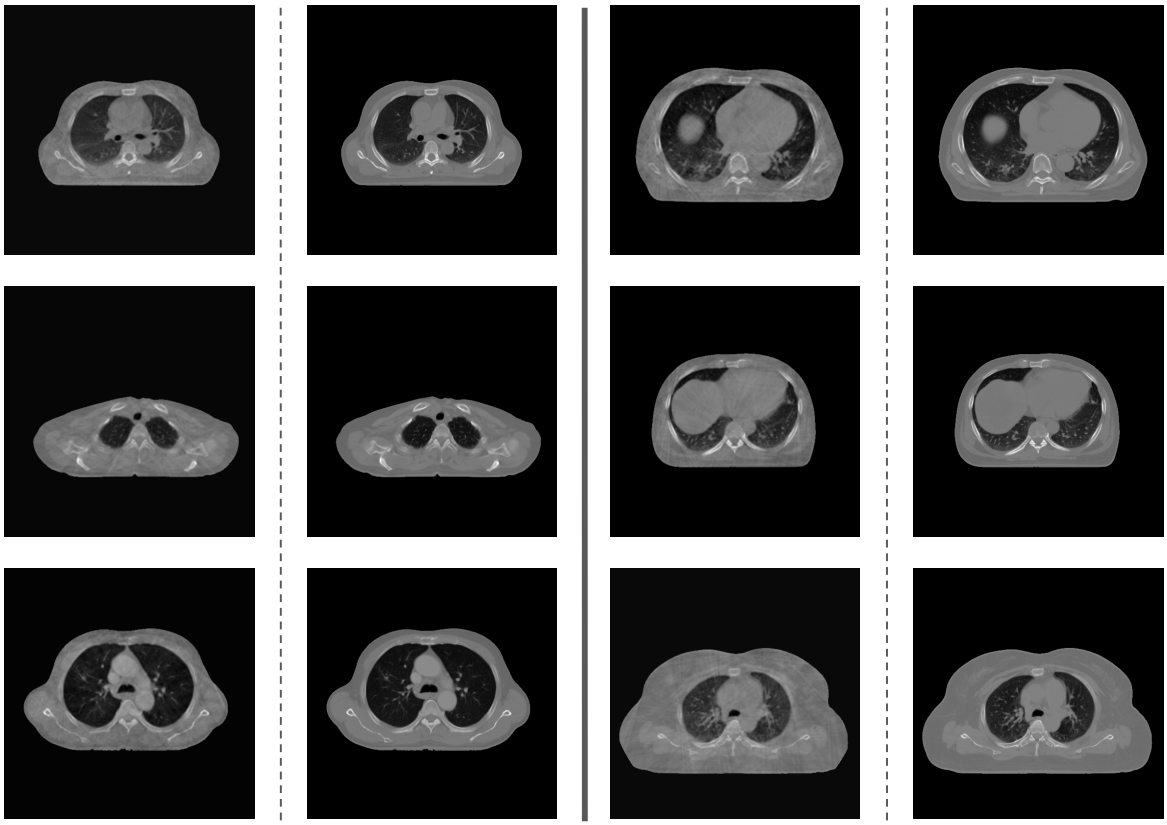

Figure 7: **Dataset Demostration.** We have chosen 6 pairs of training data from our dataset for demonstration purposes. In each pair, the down-sampled image (pseudo-CBCT) is presented on the left-hand side, while the original full-view CT image is displayed on the right-hand side. Notably, streak artifacts are observable in the pseudo-CBCT, while the fundamental structural features are consistent with their corresponding ground truth.

Model	sharpness↑	contour↑	preserv.↑	bone↑	muscle↑	heart↑	lung↑	recon.↑	satisfy↑
GAN	0.70	0.70	0.82	0.86	0.66	0.62	0.72	0.75	0.61
RegGAN	5.48	5.42	6.13	6.34	5.30	5.13	3.31	5.79	4.90
CycleGAN	<u>6.33</u>	<u>6.05</u>	**6.72**	<u>7.12</u>	<u>6.14</u>	5.54	4.01	<u>6.50</u>	<u>5.69</u>
CDGAN	6.27	5.85	**6.72**	6.63	5.87	<u>5.55</u>	3.98	6.22	5.63
SD.v2	5.62	5.21	3.81	3.03	4.22	4.34	**6.63**	3.88	4.49
DiffuCE	**6.54**	**7.13**	<u>6.45</u>	**7.36**	**6.16**	**7.00**	<u>5.17</u>	**6.57**	**7.07**

Table 4: **Questionnaire Results.** The scores have been averaged across samples from 10 patients. The first and the second place on each metric are marked with bold letters and underlined, respectively. Noted that the "preserv." and "recon." represent "tissue preservation" and "reconstruction", respectively.

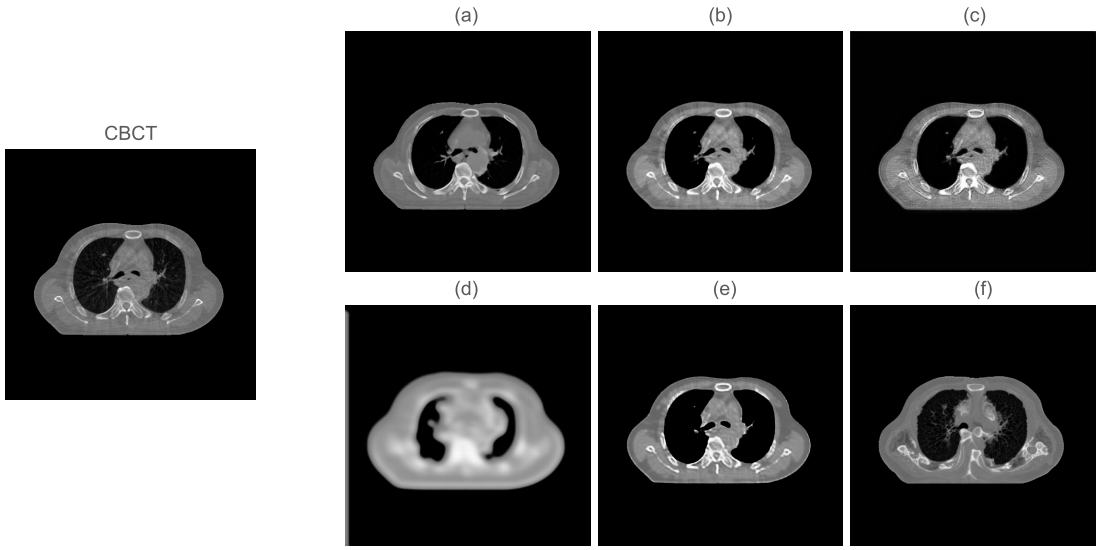

Figure 8: **Questionnaire Example.** This figure illustrates an example of the questionnaire utilized for experts' assessment. For each set of CT images, we present one real CBCT image alongside 6 reconstructed images from various deep learning-based approaches, arranged in random order. Experts are instructed to rank the reconstructed images using given metrics on a scale from 0 to 10. In this specific example, the order of reconstructed images is as follows: (a) DiffuCE, (b) CycleGAN, (c) CycleDeblurGAN, (d) GAN, (e) RegGAN, (f) Stable Diffusion v2.

4 Experts' Assessment

To assess the performance of DiffuCE from a clinical perspective, we engage five radiologists from the local medical center in our assessment. The assessment consists of CT images from 10 patients and their corresponding reconstructed images from different models. The assessment result is provided in **Table 4**, and a visual example is shown in **figure 8**.

5 Ablation Study

Our framework, DiffuCE, consists of three parts: Domain Bridging Encoder (DBE), Conditional Diffusion Denoiser (CDD), and Conditional Refinement Decoder (CRD). In the ablation study, we remove one component at a time and compare the outputs across every combination. In this section, we provide visual examples to further explain the findings across different combinations in the ablation study.

In the DiffuCE framework, the DBE plays a crucial role in bridging latent representations from CBCT into the CT domain. If the DBE is excluded, CBCT latents are directly fed into the CDD. However, since these latents have a distinct distribution in the latent space, this direct feed can result in inaccurate outputs. An example is provided in **Figure 9**.

The role of CDD is to systematically eliminate the noise introduced in the latent space by the DBE while retaining details through the assistance of conditional guidance modules. If the CDD is omitted, the noise

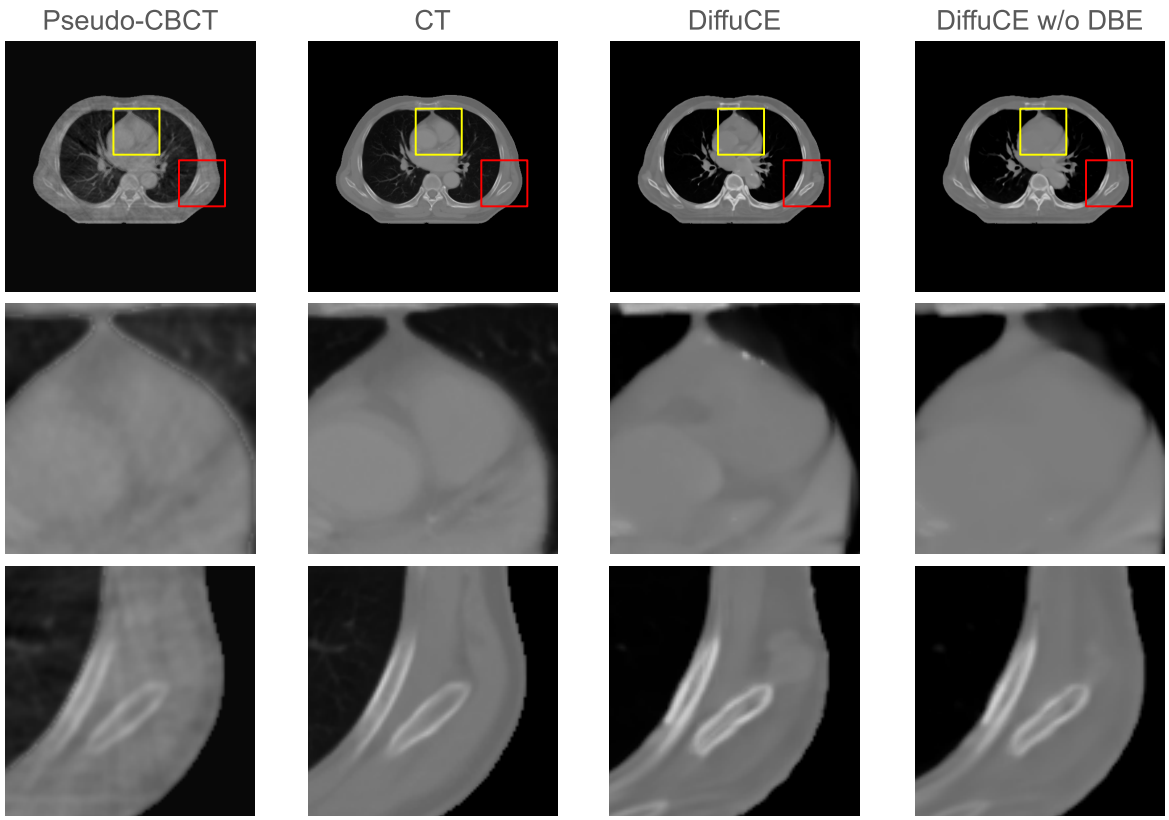

Figure 9: **Ablation: w/o DBE.** In this figure, we demonstrate that DiffuCE can reconstruct samples with enhanced contrast between different tissues, aided by the DBE. The areas framed in yellow and red are magnified in the second and third rows, respectively. In the second row, the DiffuCE output preserves finer details, whereas the output without the DBE tends to oversmooth the details. A similar outcome is observed in the third row.

within the latent space could severely distort the boundaries of soft tissues. Without the guidance information, reconstruction becomes challenging, leading to highly distorted results. An example is provided in **Figure 10**.

In the DiffuCE framework, the CRD plays a crucial role in guiding the transformation from latent space to pixel space. If the CRD is omitted, the structural details distorted by the CDD during the denoising process might propagate into pixel space, resulting in performance drawbacks. Case studies highlight the vital role of CRD in controlling the output at the pixel level. An example is provided in **Figure 11**.

6 Discussion

Why does DiffuCE need more conditions to optimize the performance?

Different from conventional denoising networks, the DiffuCE is more like generating high-quality images that very similar to the input low quality images.

In the natural image domain, the variation of the generative model can be seen as creativity, bringing more diversity to the generated content. With more control, such as more conditions and less timesteps, the diversity of diffusion models is suppressed, and the model tend to generate samples that are almost the same.

In medical image enhancement, the ideal solution is the one that only removes the noise pattern and leaves the rest of the contents unchanged. In other words, it's a generation task aiming to generate a high-quality version of the original input image, and the diversity during the generation should be as low as possible. Adopting the idea of the natural image generation, this ideal solution should be achieved with enough conditions that makes the generation process become almost deterministic.

If the generation process becomes deterministic, the generative model can be seen as a translator between two different image distributions, which is an ideal denoiser in a medical image enhancement task.

Why does the bridging module in the DBE work with the noisy latent?

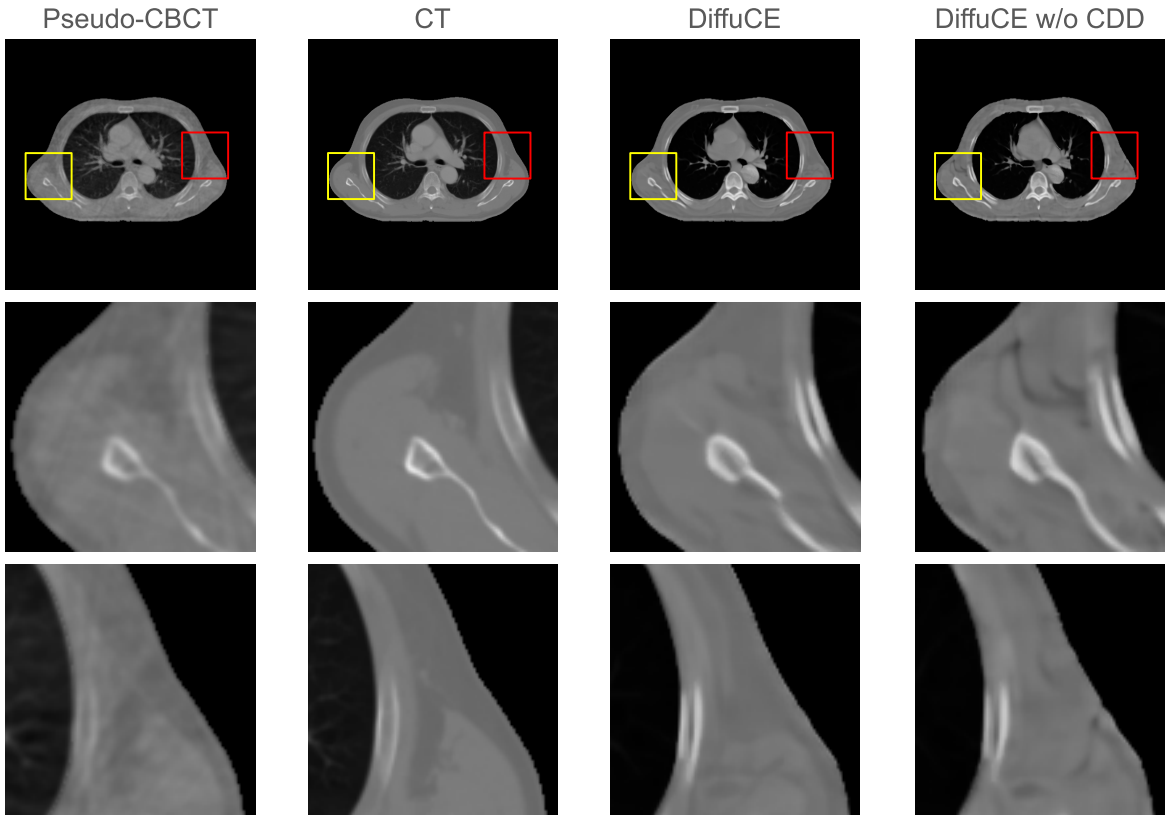

Figure 10: **Ablation: w/o CDD.** In this figure, we illustrate how DiffuCE can mitigate artifacts while preserving structural features with the assistance of CDD. The areas framed in yellow and red are magnified in the second and third rows, respectively. In the second row, the boundaries of soft tissue in the DiffuCE output are relatively well-defined, while the soft tissue becomes blurred and less realistic in the output without CDD. A similar outcome is observed in the third row.

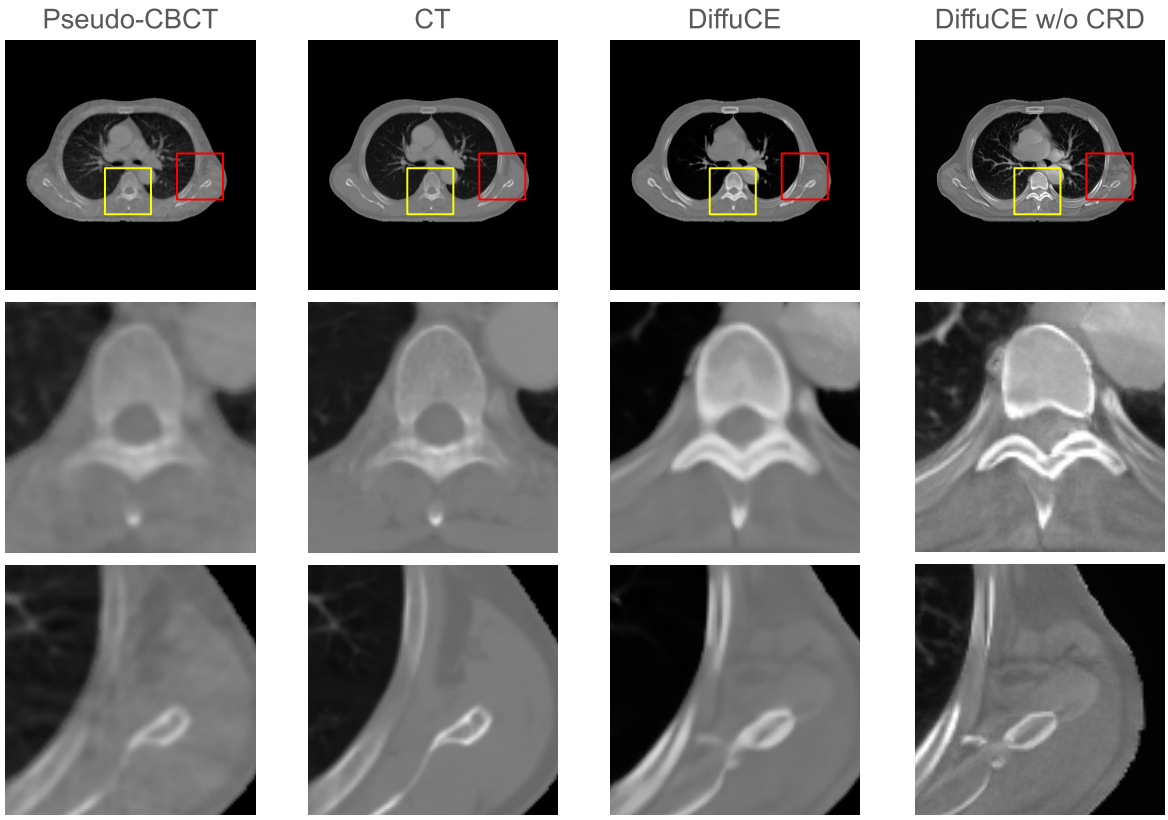

Figure 11: **Ablation: w/o CRD.** In this figure, we illustrate how CRD refines the output from CDD based on the provided conditions. The areas framed in yellow and red are magnified in the second and third rows, respectively. In the second row, the spine in the DiffuCE output is refined to a more realistic shape with the inclusion of CRD, whereas the spine in the output without CRD appears distorted. In the third row, the body contour is significantly distorted in the output without CRD, while the body contour in the output with CRD closely resembles the ground truth.

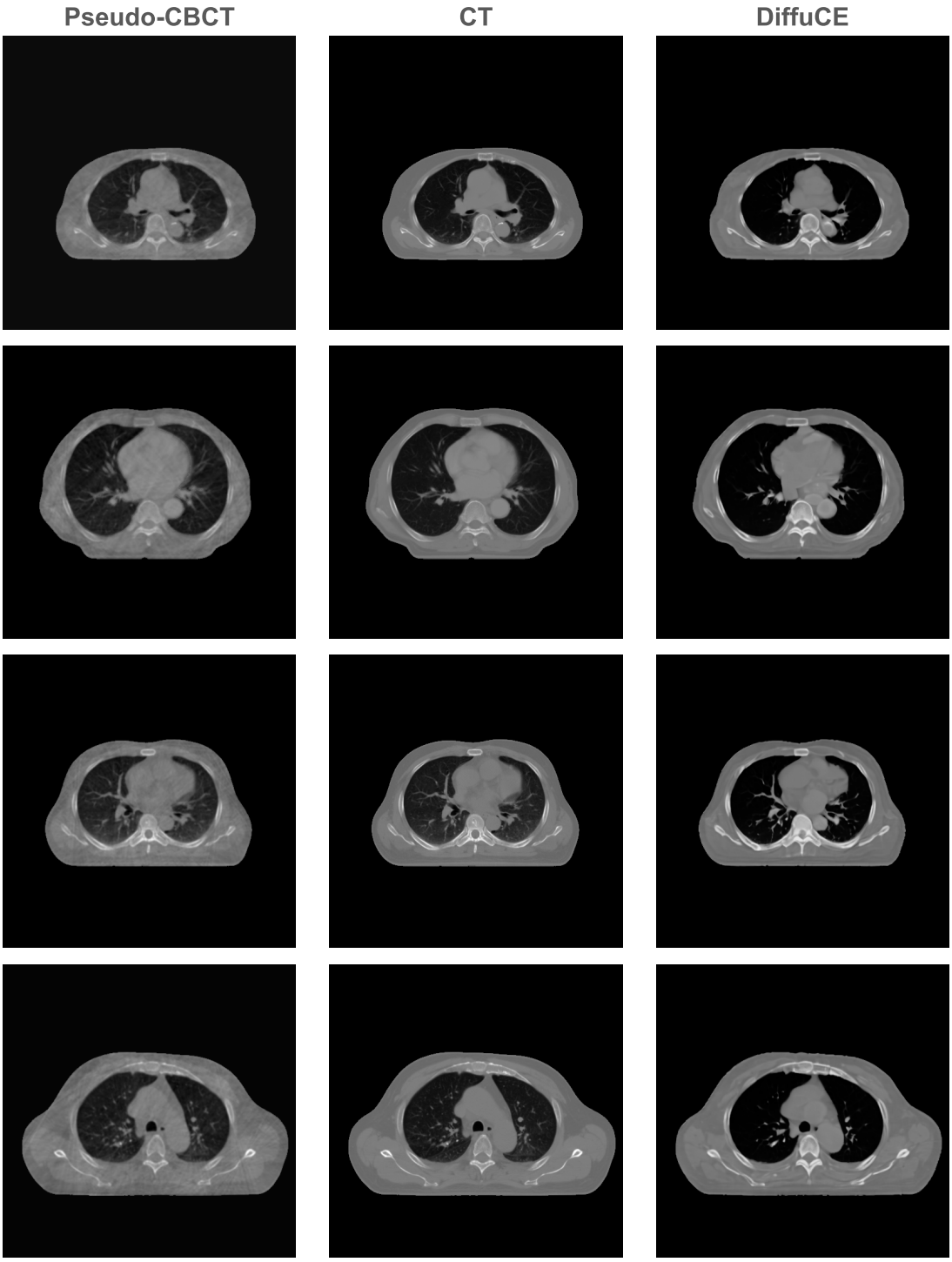

Figure 12: **Pseudo-CBCT case study.** Some reconstructed samples from DiffuCE.

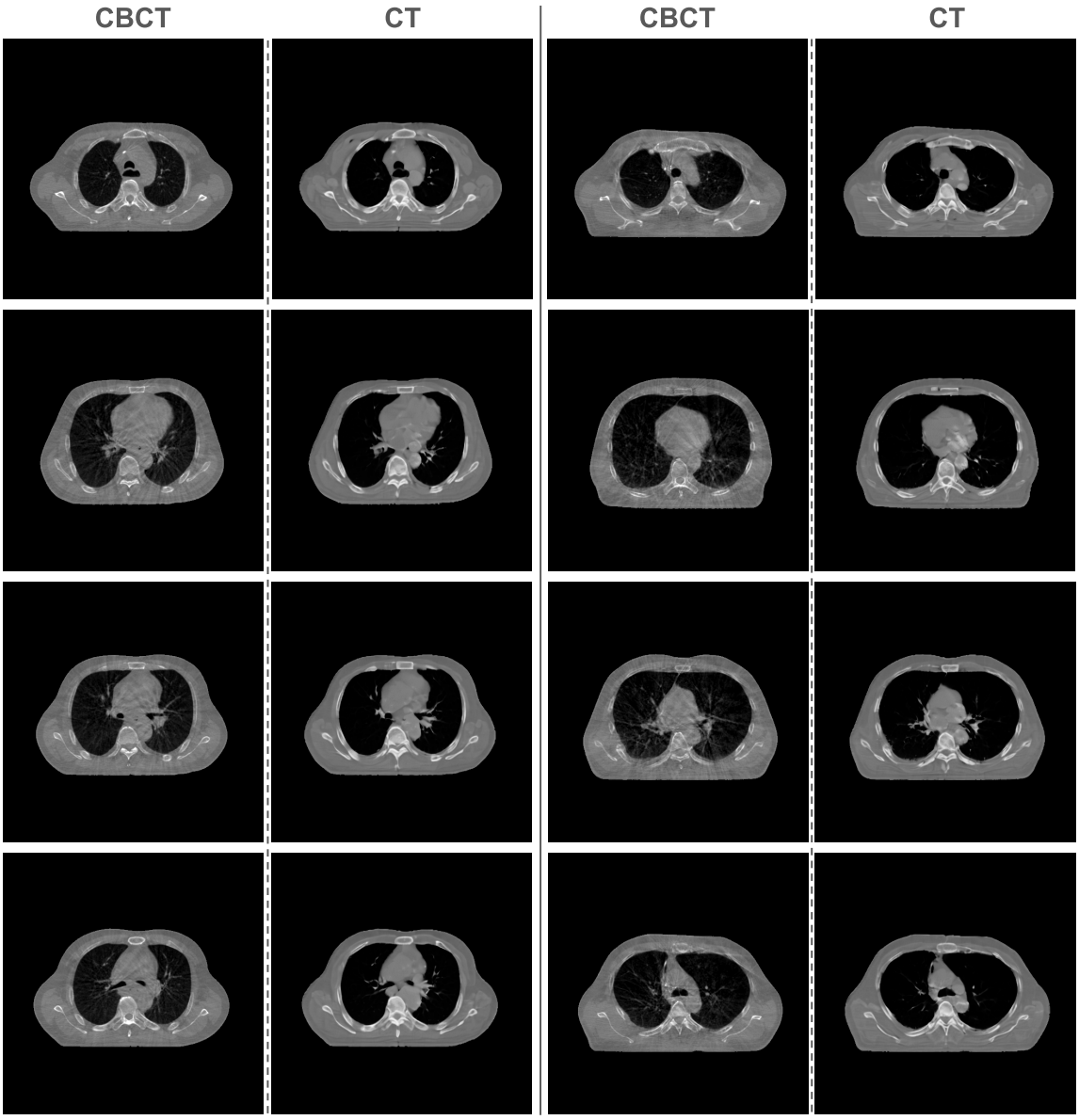

Figure 13: **Real CBCT case study.** In this figure, we showcase the capability of DiffuCE on real CBCT images. Each CBCT image exhibits noticeable artifacts, and DiffuCE effectively reduces these artifacts while preserving the structural features. However, it is noteworthy that some crucial features in vital organs, such as the heart, appear distorted or are missing. Addressing these specific challenges constitutes our primary focus for future work.

The input CBCT image is projected to CT distribution in the latent space by the DBE with added noise, and the CDD gradually removes the noise. The reason why the projection is performed on the noisy latent is because of the difference between the CBCT and CT images being partially compensated by the noise. Consider the following equation:

$$I_{noisyCT} = \alpha \times I_{CT} + \beta \times \epsilon, \tag{5}$$

$$I_{noisyCBCT} = \alpha \times I_{CBCT} + \beta \times \epsilon, \tag{6}$$

$$\lim_{\beta \to \infty} I_{noisyCT} = \lim_{\beta \to \infty} I_{noisyCBCT} \tag{7}$$

$$\tag{8}$$

, where $I_{noisyCT}, I_{noisyCBCT}, I_{CT}, I_{CBCT}, \epsilon$ refer to noisy CT, noisy CBCT, clean CT, clean CBCT image, and the noise. As the β being larger, the magnitude of added noise becomes stronger. Eventually, both $I_{noisyCT}$ and $I_{noisyCBCT}$ will become noise sampled from the same distribution if the β becomes infinitely large.

However, it's a trade-off between bridging distribution differences and preserving information. Thus we decide to train a bridging unit to perform the projection with the help of adding noise that shortens the distance between distributions.

Why does the performance of DiffuCE with different CRD weights are so close?

Since the output of the CRD should be high-quality medical images with an ideal appearance of clinical experts, the CRD is supposed to be optimized by the objective of the corresponding task. However, results in **Table 2** show that the weights optimized by different objectives result in similar performance. It might indicate that the optimal distribution of different CBCT enhancement tasks could be very close which is a reasonable assumption since high-quality CT images share features such as high contrast, clear border, and no noise pattern. Results in **Table 1** show the diffusion-based fine-tuned methods, SD.v2 and Ours, have the best FID performance, indicating the competitive ability to capture the CT image distribution. It might reveal the opportunity to build a more unified CBCT image enhancement network based on the pre-trained foundation diffusion model framework. We will leave this as the future work.

References

[1] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.