Incremental Cross-view Mutual Distillation for Self-supervised Medical CT Synthesis

Chaowei Fang[1*] Liang Wang[1*] Dingwen Zhang[2,3†] Jun Xu[4] Yixuan Yuan[5] Junwei Han[2,3]
[1]Xidian University [2]Northwestern Polytechnical University
[3]Hefei Comprehensive National Science Center [4]Nankai University [5]City University of Hong Kong

Abstract

Due to the constraints of the imaging device and high cost in operation time, computer tomography (CT) scans are usually acquired with low within-slice resolution. Improving the inter-slice resolution is beneficial to the disease diagnosis for both human experts and computer-aided systems. To this end, this paper builds a novel medical slice synthesis to increase the inter-slice resolution. Considering that the ground-truth intermediate medical slices are always absent in clinical practice, we introduce the incremental cross-view mutual distillation strategy to accomplish this task in the self-supervised learning manner. Specifically, we model this problem from three different views: slice-wise interpolation from axial view and pixel-wise interpolation from coronal and sagittal views. Under this circumstance, the models learned from different views can distill valuable knowledge to guide the learning processes of each other. We can repeat this process to make the models synthesize intermediate slice data with increasing between-slice resolution. To demonstrate the effectiveness of the proposed approach, we conduct comprehensive experiments on a large-scale CT dataset. Quantitative and qualitative comparison results show that our method outperforms state-of-the-art algorithms by clear margins.

1. Introduction

High-resolution CT volume data can provide high-quality detail for organs and tissues, thus are valuable for computer-aided diagnosis. However, due to the constraints of the imaging device, the between-slice resolution of the acquired CT volume is not sufficiently high in practical clinical scenarios, which makes these volume data hard to provide the desired imaging detail for the disease diagnosis.

To solve this problem, a novel task, called medical slice synthesis, has been arising recently. The goal is to synthesize intermediate imagery content between original adjacent slices. Peng *et al.*, [24] made the earliest attempt by implementing pixel-wise interpolation processes on the coronal-view and sagittal-view images and then fusing the results interpolated from two views. However, this method requires large-scaled ground-truth training data, which we cannot conveniently acquire in practice.

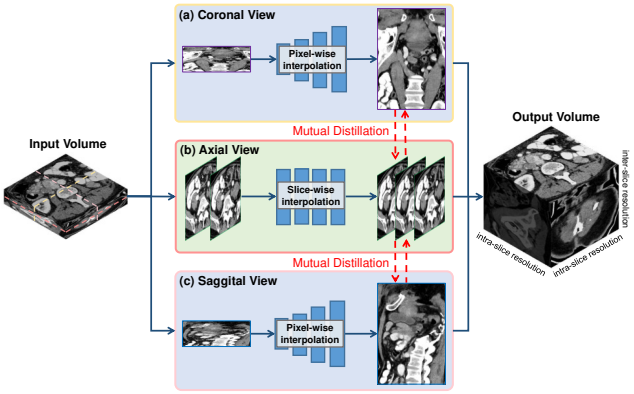


Figure 1. Pixel-wise interpolation in coronal and sagittal views, and slice-wise interpolation in axial view can increase the inter-slice resolution of the input volume individually. We propose a cross-view knowledge distillation framework to settle the self-supervised CT slice synthesis task.

This paper explores a self-supervised learning framework to train the slice synthesizer without the ground-truth data. Specifically, we find that another under-explored way is to formulate it as a slice-wise interpolation problem for the axial-view images (See Fig. 1). Namely, intermediate slices can be inferred from the context information of two adjacent slices in the axial view. Since pixel-wise and slice-wise interpolation modeling tries to synthesize the missing detail by exploring different kinds of spatial context, the two modeling processes tend to capture helpful yet distinct patterns towards the same ultimate goal. Thus, we can jointly use the two modeling processes to address the medical slice synthesis problem and collaborate them to provide complementary knowledge for each other. Each interpolation model can be learned under the guidance of the other ones, thus avoiding the requirement of ground-truth training data.

*Equal contribution.
†Corresponding author.

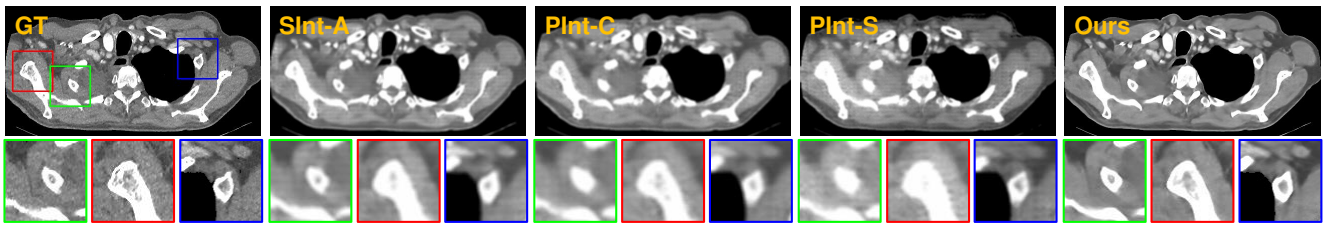


Figure 2. Slice-wise interpolation in axial view (SInt-A), and pixel-wise interpoloatin in coronal (PInt-C) and sagittal (PInt-S) views, have their own superiority in synthesizing inter-slice images. Our proposed cross-view mutual distillation can combine the learned knowledge from three types of interpolation algorithms.

We propose an incremental cross-view mutual distillation pipeline for training medical CT slice synthesis models to take advantage of slice synthesis algorithms from multiple views. Considering that structural information appears to have different characteristics across views and models learned from different views have their superiority (see Fig. 2), we involve three modeling components in the learning process: 1) slice-wise interpolation in axial view; 2) pixel-wise interpolation in coronal view; 3) pixel-wise interpolation in sagittal view. We set up a U-shape network with memorization capacity to implement the slice-wise interpolation and adopt an existing image super-resolution network [20] to achieve pixel-wise interpolation.

To lean such deep models, we propose a two-stage learning framework. In the first learning stage, we downsample the resolution of original volumes and then use the downsampled and original volume data to learn single-view slice synthesis models. To enable the model to upscale the resolution of the original volume data without any external supervision, we further design a cross-view mutual distillation process in the second learning stage. We constrain the pairs of predictions on the original volume data produced by axial-view slice-wise interpolation and coronal/sagittal-view pixel-wise interpolation models. An illustration of our proposed method is presented in Fig. 1. The knowledge distillation mechanism enables the slice-wise and pixel-wise interpolation models to learn from each other and fuse the advantages of different image recovery models learned from different perspectives. Finally, we incrementally increase the between-slice resolution from the three perspectives and apply the cross-view mutual distillation on predictions with very high resolution, enhancing the knowledge exchange across views in self-supervised slice synthesis.

The main contributions of this paper are as follows.

- A pioneering effort is made to implement the self-supervised CT slide synthesis, modeling slice-wise interpolation for the axial view and pixel-wise interpolation for the coronal and sagittal views.
- A novel self-supervised learning framework is established, based on single-view internal learning and incremental cross-view mutual distillation.
- Extensive experiments on a CT collection (composed of three existing CT datasets) demonstrate that our proposed method achieves state-of-the-art performance.

2. Related Work

Medical slice synthesis is targeted at hallucinating inter-slice detail which is critical to high-level disease diagnosis for both radiologists and computer-based intelligent systems. Recently, 3D neural networks [2, 6, 26, 28, 32] are extensively applied in processing and understanding medical volumetric data. The main drawback of using 3D neural networks is the huge amount of network parameters and memory consumption. SAINT [24] is a two-stage framework to solve the slice synthesis task. It first employs 2D convolutional neural networks (CNNs) to enlarge sagittal and coronal images individually, and then fuse the enlarged images of two views to produce the final result.

Learning slice synthesis CNNs requires a large number of paired LR and HR volumes. However, HR volumes are usually not available in practical medical scenarios. Thus, it is essential to develop unsupervised optimization algorithms for medical slice synthesis CNNs. As far as we know, few work is devoted to addressing this task. In this paper, we focus on the unsupervised slice synthesis task, and propose a cross-view mutual distillation pipeline , twisting slice-wise interpolation in axial view and pixel-wise interpolation in coronal and sagittal views.

Video Frame Interpolation. Slice-wise interpolation is highly related to video frame interpolation. In videos, the differences between consecutive frames are mainly caused by object or camera motions. Thus, video interpolation algorithms usually rely on optical flow fields [4, 14, 17, 22, 23], adaptive kernels [18, 19], or flow-based adaptive kernels [13] to interpolate intermediate transition frames from temporally neighboring frames. Aiming at tackling frame interpolation under complex motions and severe occlusions, [3, 10, 31] adopts an image reconstruction pipeline without using motion fields and adaptive kernels which are difficult to be estimated when there exist large motions and severe occlusions in the input video. The slice synthesis task is more challenging since the different slices contains totally different content and there exist no explicit correspondence relations between adjacent slices.

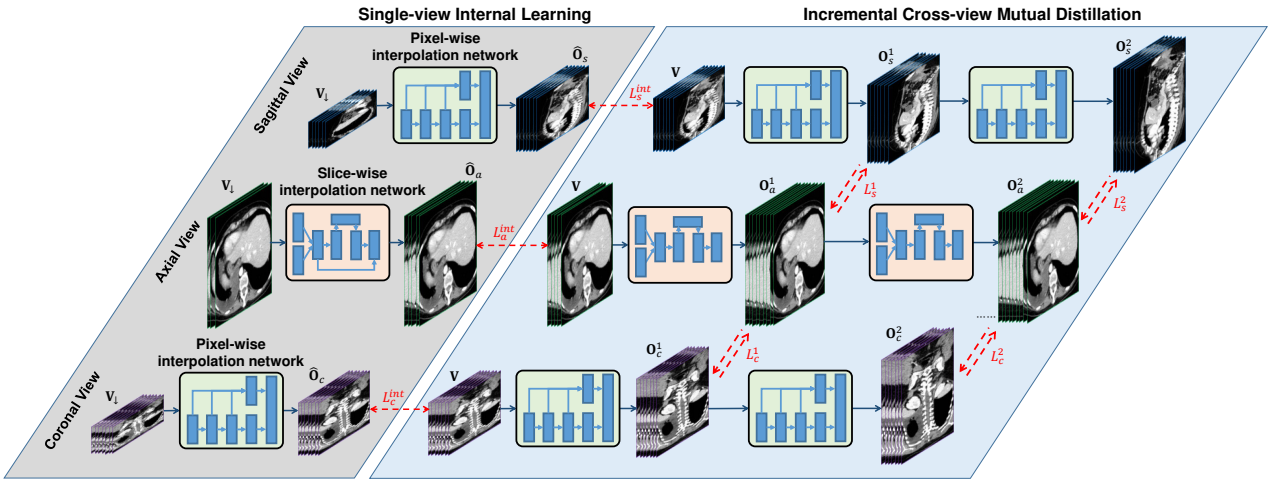


Figure 3. Overall pipeline of our method. First, internal learning is used to regularize single-view interpolation models via regarding down-sampled and original volumes as training samples. Then, an incremental cross-view mutual distillation pipeline is devised for knowledge exchange between the slice-wise interpolation in the axial view and the pixel-wise interpolation in the coronal and sagittal views.

Image Super-Resolution (SR). As a fundamental and long-lasting topic in image processing, super-resolution attracts lots of research attention. Dong *et al.* [5] apply convolutional neural networks in image super-resolution for the first time. Mainstream SR methods depend on various CNN backbones [5, 12, 29, 30, 35]. MetaSR [8] proposes to tackle the SR task of arbitrary scales through dynamic kernels learned from the pixel coordinates and upscaling factor. HAN [20] introduces the holistic attention to explore cross-position, cross-channel and cross-layer dependencies for promoting SR performance. The slice synthesis can be implemented via image SR in the coronal and sagittal views.

Knowledge Distillation. The concept of knowledge distillation is first proposed for model compression in [1]. Hinton *et al.* [7] define knowledge distillation as the task of transferring the knowledge of a teacher model which can be a very large model or an ensemble of multiple models to a student model. They also propose a distillation strategy through using the soft outputs of the teacher model to guide the training of the student model. Henceforth, a lot of literature focuses on devising more effective distillation algorithms [16, 25, 33]. Our proposed method is most related to the mutual learning [36], in which an ensemble of student models learn from each other. The major difference of our method to mutual learning is that, the student networks in our method are constructed from different views of the volumetric data and devised for addressing different tasks, namely slice-wise or pixel-wise interpolation.

3. Proposed Method

Given a 3D volume $\mathbf{V} \in \mathbb{R}^{h\times w\times l}$, we assume that $r-1$ ($r \geq 2$) slices should be interpolated between every two consecutive slices. This means that a volume defined by $\mathbf{O} \in \mathbb{R}^{h\times w\times(rl-r+1)}$ is expected to be produced. $\mathbf{V}$ can be decomposed into 2D images in the axial, coronal and saggital views, yielding $\{\mathbf{X}_a^i \in \mathbb{R}^{h\times w}\}_{i=1}^{l}$, $\{\mathbf{X}_c^j \in \mathbb{R}^{w\times l}\}_{j=1}^{h}$, and $\{\mathbf{X}_s^k \in \mathbb{R}^{h\times l}\}_{k=1}^{w}$, respectively. We can achieve the goal with three models that perform slice-wise interpolation in the axial view and pixel-wise interpolation in the coronal and sagittal views. The concrete model design can be referred to in Sec. 3.1.

Since actual training data is hard to obtain, we follow the degradation operation in [24] or [9] to approximate the real downsampling case. Under this circumstance, single-view internal learning is first used to constrain the three models with the help of down-sampled volumes. Then, the slice-wise and pixel-wise interpolation models are constrained via the consistency between volume data enlarged by them for knowledge distillation across views. The overall framework of our method is presented in Figure 3. Though 3D convolution can be alternatively used, we implement our framework with 2D convolution-based modules in consideration of computational efficiency.

3.1. Interpolation Models

3.1.1 Slice-wise Interpolation

The slice synthesis can be implemented via inserting intermediate slices between every two adjacent slices. Inspired from [3], we build up a CNN model for slice-wise interpolation in the axial view (see Fig. 4). Given two consecutive slices $\mathbf{X}_a^i \in \mathbb{R}^{h\times w}$ and $\mathbf{X}_a^{i+1} \in \mathbb{R}^{h\times w}$, a convolution layer with the kernel size of 3×3 and the dimension of 3 is used to extract two preliminary feature maps. They are rearranged into tensors $\mathbf{F}^i \in \mathbb{R}^{\frac{h}{8}\times\frac{w}{8}\times192}$ and $\mathbf{F}^{i+1} \in \mathbb{R}^{\frac{h}{8}\times\frac{w}{8}\times192}$ through the space-to-depth transformation operation. Then, a U-shape architecture is devised to fully explore multiple features of different layers to estimate the intermediate slices between $\mathbf{X}_a^i$ and $\mathbf{X}_a^{i+1}$, namely $\{\mathbf{Y}_a^{(i-1)r+t}\}_{t=2}^{r}$.

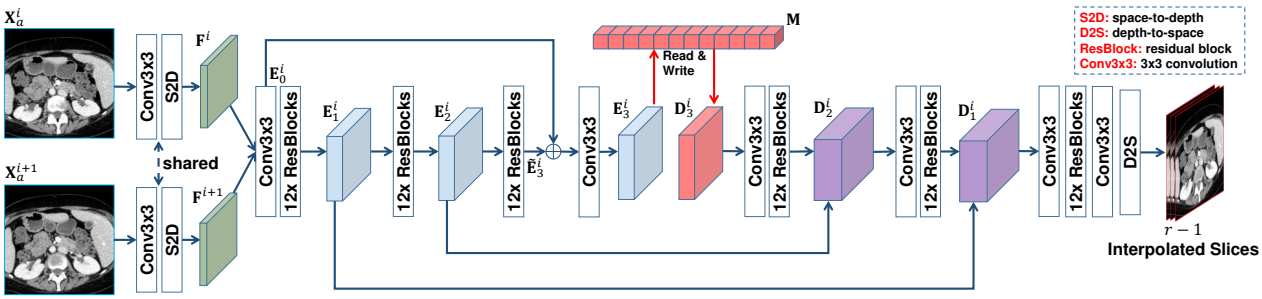


Figure 4. Network architecture of the slice-wise interpolation model. Given two adjacent slices, a U-shape network constituted by convolution layers, residual groups [3] and a memory bank [21], synthesizes $r-1$ intermediate slices.

$\mathbf{F}^i$ and $\mathbf{F}^{i+1}$ are concatenated and then compressed into a tensor $\mathbf{E}_0^i \in \mathbb{R}^{\frac{h}{8}\times\frac{w}{8}\times 192}$ via a 3×3 convolution layer. Then, three groups of residual blocks with channel attentions [3] are used to produce multiple feature maps $\mathbf{E}_1^i$, $\mathbf{E}_2^i$ and $\tilde{\mathbf{E}}_3^i$. Each group is composed of 12 residual blocks. $\tilde{\mathbf{E}}_3^i$ is added to $\mathbf{E}_0^i$ through a skip connection, and another 3×3 convolution is attached to produce the final feature map $\mathbf{E}_3^i$.

Considering CT images usually share high similarities (e.g., anatomical structures) across persons, we incorporate a memory bank [21] $\mathbf{M}\in\mathbb{R}^{m\times d}$ to store the common patterns. m is the number of items in the memory bank. All points in $\mathbf{E}_3^i$ are reconstructed with $\mathbf{M}$, deriving a new feature map $\mathbf{D}_3^i$. The linear combination of items in $\mathbf{M}$ is used to infer every point in $\mathbf{D}_3^i$,

$$\mathbf{D}_3^i[x,y]=\sum_{z=1}^{m} p_{x,y,z}^i \mathbf{M}[z], \tag{1}$$

where $\mathbf{D}_3^i[x,y]$ represents the feature vector at position (x,y) of $\mathbf{D}_3^i$, and $\mathbf{M}[z]$ indicates the z-th item of the memory bank $\mathbf{M}$. $p_{x,y,z}^i$ indicates the weight coefficient between $\mathbf{E}_3^i[x,y]$ and $\mathbf{M}[z]$,

$$p_{x,y,z}^i=\frac{\exp(\mathbf{E}_3^i[x,y]\circ\mathbf{M}[z])}{\sum_{z'=1}^{m}\exp(\mathbf{E}_3^i[x,y]\circ\mathbf{M}[z'])}. \tag{2}$$

Here, '$\circ$' indicates the inner product operation. During the training stage, the memory bank is continuously updated through accumulating the emerging patterns in $\mathbf{E}_3^i$.

$$q_{x,y,z}^i=\frac{\exp(\mathbf{M}[z]\circ\mathbf{E}_3^i[x,y])}{\sum_{(x',y')\in\mathcal{U}_k}\exp(\mathbf{M}[z]\circ\mathbf{E}_3^i[x',y'])}, \tag{3}$$

$$q_{x,y,z}^i\leftarrow q_{x,y,z}^i/\max_{(x',y')\in\mathcal{U}_k} q_{x',y',z}^i, \tag{4}$$

$$\mathbf{M}[z]\leftarrow\mathbf{M}[z]+\sum_{(x',y')\in\mathcal{U}_k} q_{x',y',z}^i\mathbf{E}_3^i[x',y'], \tag{5}$$

$$\mathbf{M}[z]\leftarrow\mathbf{M}[z]/||\mathbf{M}[z]||_2. \tag{6}$$

$\mathcal{U}_k$ represents the set of points whose nearest neighbor in the memory bank is $\mathbf{M}[k]$. Based on the above process, the memory bank is updated by accumulating patterns across all training CT slices. It can store representative visual patterns for CT slice reconstruction.

The decoding stage is constituted by three consecutive modules. Each module contains one 3×3 convolution layer and twelve residual blocks. $\mathbf{D}_3^i$ is regarded as the input of the first stage. Skip connections are used to propagate $\mathbf{E}_2^i$ and $\mathbf{E}_1^i$ into the second and third stages of the decoder, respectively. Finally, a 3×3 convolution followed by a depth-to-space operation is employed to produce intermediate slices. By means of the above slice interpolation model, the input volume is interpolated into a new volume with more slices, $\{\mathbf{Y}_a^i\}_{i=1}^{rl-r+1}$, where $\mathbf{Y}_a^i=\mathbf{X}_a^{(i-1)\%r}$, if $i\%r=1$. We denote the interpolated volume as $\mathbf{O}_a=\mathrm{SInt}_a(\mathbf{V}|\Theta_s)$. Θ_s denotes the parameters of the interpolation model.

3.1.2 Pixel-wise Interpolation

The other perspective for slice synthesis is the pixel-wise interpolation, based on the super-resolving of the images in the coronal or sagittal view. We use the image super-resolution network proposed in [20] for pixel-wise interpolation. The coronal and sagittal views share the same model. The coronal images $\{\mathbf{X}_c^j\}_{j=1}^h$ and sagittal images $\{\mathbf{X}_s^k\}_{k=1}^w$ are super-resolved by a factor of r along the longitudinal axis, resulting in $\{\mathbf{Y}_c^j\}_{j=1}^h$ and $\{\mathbf{Y}_s^k\}_{k=1}^w$ respectively. The last $r-1$ columns of super-resolved images are abandoned to make the shape consistent with the volume produced by the slice-wise interpolation model. These super-resolved coronal and sagittal images can be stacked into new volumes $\mathbf{O}_c$ and $\mathbf{O}_s$ respectively. We denotes the pixel-wise interpolation processes in the coronal and sagittal view as, $\mathbf{O}_c=\mathrm{PInt}_c(\mathbf{V}|\Theta_p)$ and $\mathbf{O}_s=\mathrm{PInt}_s(\mathbf{V}|\Theta_p)$ respectively. Θ_p denotes parameters of the pixel-wise interpolation model.

During the inference phase, the final interpolation result $\mathbf{O}$ is obtained via fusing $\mathbf{O}_a$, $\mathbf{O}_c$ and $\mathbf{O}_s$,

$$\mathbf{O}[x,y,z]=\begin{cases}\frac{\mathbf{O}_c[x,y,z]+\mathbf{O}_s[x,y,z]}{2} & \text{if } z\%r=1\\ \frac{\mathbf{O}_a[x,y,z]+\mathbf{O}_c[x,y,z]+\mathbf{O}_s[x,y,z]}{3} & \text{else}\end{cases} \tag{7}$$

3.2. Learning Procedure

3.2.1 Single-view Internal Learning

An internal learning strategy is adopted to optimize individual single-view slice-wise or pixel-wise interpolation models. The original volume is down-sampled by the factor of r along the axial view, resulting in $\mathbf{V}_{\downarrow} \in \mathbb{R}^{h\times w\times \lfloor \frac{l}{r} \rfloor}$. Feeding $\mathbf{V}_{\downarrow}$ into the slice-wise and pixel-wise interpolation models, we can obtain upsampled volumes: $\hat{\mathbf{O}}_a = \text{SInt}_a(\mathbf{V}_{\downarrow})$, $\hat{\mathbf{O}}_c = \text{PInt}_c(\mathbf{V}_{\downarrow})$, and $\hat{\mathbf{O}}_s = \text{PInt}_s(\mathbf{V}_{\downarrow})$. Here, parameters are neglected for briefness.

Regarding the original volume as the ground-truth, we calculate the training loss with the mean square error (MSE) function. Besides, to strengthen the restoration on high-frequency details, we extract three scales of wavelet coefficients and use MSE to constrain the distances on the LH (horiz), HL (vertic), and HH (diag) coefficients of each wavelet decomposition scale. The overall loss functions used in the single-view internal learning are as follows.

$$L_a^{int} = \text{MSE}(\hat{\mathbf{O}}_a, \mathbf{V}) + \sum_{t=1}^{3} \text{MSE}(\text{WT}_a^{(t)}(\hat{\mathbf{O}}_a), \text{WT}_a^{(t)}(\mathbf{V})), \tag{8}$$

$$L_c^{int} = \text{MSE}(\hat{\mathbf{O}}_c, \mathbf{V}) + \sum_{t=1}^{3} \text{MSE}(\text{WT}_c^{(t)}(\hat{\mathbf{O}}_c), \text{WT}_c^{(t)}(\mathbf{V})), \tag{9}$$

$$L_s^{int} = \text{MSE}(\hat{\mathbf{O}}_s, \mathbf{V}) + \sum_{t=1}^{3} \text{MSE}(\text{WT}_s^{(t)}(\hat{\mathbf{O}}_s), \text{WT}_s^{(t)}(\mathbf{V})). \tag{10}$$

$\text{WT}_a^{(t)}(\cdot)$, $\text{WT}_c^{(t)}(\cdot)$, and $\text{WT}_s^{(t)}(\cdot)$ calculates the t-th scale of wavelet coeffficients from the axial, coronal, and sagittal images of the input volume respectively. The restored volumes may have a smaller size than $\mathbf{V}$ due to the quantization effect, and excess voxels of $\mathbf{V}$ are neglected when calculating the above loss functions.

3.2.2 Incremental Cross-view Mutual Distillation

Given axial, coronal, and sagittal images originating from the same volume, the slice-wise and pixel-wise interpolation models have specific superiority in synthesizing details since different context is explored. We devise an MSE-based consistent constraint to make the two kinds of models teach each other so that the specific advantages of the three interpolation schemes are combined to promote the ultimate interpolation performance. Such a cross-view mutual distillation method can tackle the dilemma in which the ground-truth training data is absent. Practically, we repeat the slice-wise and pixel-wise interpolation for n times, deriving of $\mathbf{O}_a^n = \text{SInt}_a^{(n)}(\mathbf{V})$, $\mathbf{O}_c^n = \text{PInt}_c^{(n)}(\mathbf{V})$, and $\mathbf{O}_s^n = \text{PInt}_s^{(n)}(\mathbf{V})$. The consistency constraints between the slice-wise interpolation result in axial view and the pixel-wise interpolation result in coronal/sagittal view are formulated as follows,

$$L_c^n = \sum_{(x,y,z)\in \mathbb{T}_c^n(\gamma)} \frac{(\mathbf{O}_a^n[x,y,z] - \mathbf{O}_c^n[x,y,z])^2}{|\mathbb{T}_c^n(\gamma)|}, \tag{11}$$

$$L_s^n = \sum_{(x,y,z)\in \mathbb{T}_s^n(\gamma)} \frac{(\mathbf{O}_a^n[x,y,z] - \mathbf{O}_s^n[x,y,z])^2}{|\mathbb{T}_s^n(\gamma)|}, \tag{12}$$

where $\mathbb{T}_c^n(\gamma)$ ($\mathbb{T}_s^n(\gamma)$) denotes the set of γ percents of points with smallest loss values between $\mathbf{O}_a^n$ and $\mathbf{O}_c^n$ ($\mathbf{O}_s^n$). Assume he largest number of interpolation times be N. The overall objective functions for the cross-view mutual distillation are formulated as, $L_c^{cmd} = \frac{1}{N}\sum_{n=1}^{N} L_c^n$, and $L_s^{cmd} = \frac{1}{N}\sum_{n=1}^{N} L_s^n$.

3.2.3 Overall Objective Function

Apart from the single-view internal learning and cross-view mutual distillation loss functions, the compactness (L^{com}) and separateness (L^{sep}) constraints as in [21], are used to regularize the memory bank,

$$L^{com} = \sum_{i=1}^{l-1}\sum_{x=1}^{h/8}\sum_{y=1}^{w/8} \|\mathbf{E}_3^i[x,y] - \mathbf{M}[z_{\text{pos}}^i(x,y)]\|_2, \quad s.t.\ \ z_{\text{pos}}^i(x,y) = \underset{z'}{\text{argmax}}\, p_{x,y,z'}^i; \tag{13}$$

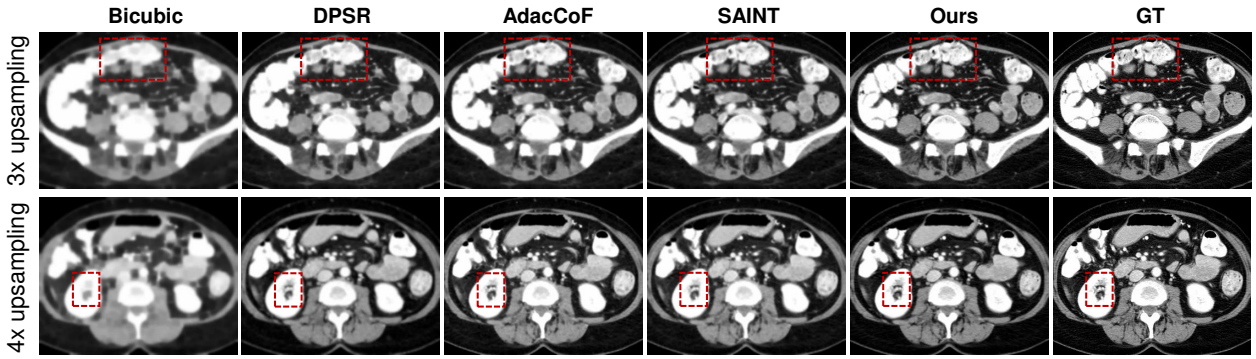
$$L^{sep} = \sum_{i=1}^{l-1}\sum_{x=1}^{h/8}\sum_{y=1}^{w/8} \max(\|\mathbf{E}_3^i[x,y] - \mathbf{M}[z_{\text{pos}}^i(x,y)]\|_2 - \|\mathbf{E}_3^i[x,y] - \mathbf{M}[z_{\text{neg}}^i(x,y)]\|_2 + \alpha, 0), \quad s.t.\ \ z_{\text{neg}}^i(x,y) = \underset{z'\neq z_{\text{pos}}^i(x,y)}{\text{argmax}}\, p_{x,y,z'}^i. \tag{14}$$

$\alpha(=1)$ is a constant. The complete objective function is formed through summing up the above losses, $L = L_a^{int} + L_c^{int} + L_s^{int} + 0.15*(L_c^{cmd} + L_s^{cmd}) + 0.1*(L^{com} + L^{sep})$. The weighting factors are chosen empirically.

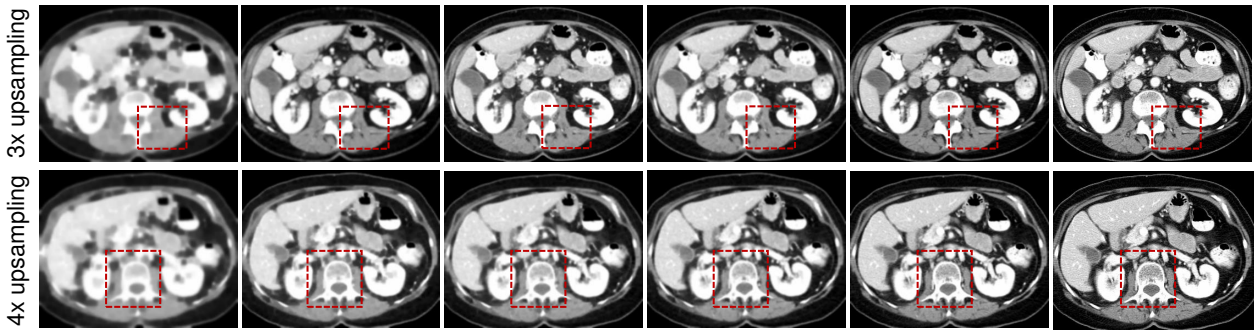
4. Experiments

4.1. Experimental Settings

Dataset. The CT Dataset consists of 560 volumes, which are collected from the Medical Segmentation Decathlon challenge [27], including 131, 126, and 303 volumes for liver, colon and hepatic vessel segmentation, respectively. The spatial size is 512×512 and the number of slices is in the range of 24 to 917. The within-slice resolution ranges from 0.5mm to 1.0mm, and the between-slice resolution ranges from 0.7mm to 8.0mm. Fifty volumes with the thinnest slices are used for testing, and the other 510 volumes are used for training. All volumes are down-sampled by the factor of r in the axial view, while high-resolution volumes are only used for validating algorithm



(a) LR volumes are generated via direct downsampling.



(b) LR volumes are obtained via blurring, downsampling, and noise distortion.

Figure 5. Qualitative comparisons against existing slice synthesis algorithms on CT-s and MRI-s. The slices synthesized by our method are better than the results of DPSR [34], AdaCoF [13], and SAINT [24]. (Best viewed in close-up)

performance. Two degradation strategies are used for validating interpolation algorithms: 1) Low-resolution volumes are synthesized via directly sampling one slice every r slices in the axial view; 2) Low-resolution volumes are generated by blurring and down-sampling. Then, Gaussian noises are used to distort the down-sampled volumes.

Evaluation Metrics. We use two metrics, including PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index). SSIM is calculated independently on axial, coronal, and sagittal images, denoted by SSIM_a, SSIM_c, and SSIM_s, respectively.

Implementation Detail. During training, we only use central 256*256 regions of CT slices, which are further decomposed into 128×128 patches. Each training volume is composed of 15 slices. Adam [11] is chosen for network optimization. The model is trained for 50 epochs with a batch size of 4. The learning rate is initially set to 10^{-4} and decayed by 0.1 after ten epochs. By default, m, γ, and N is set to 10, 40%, and 2, respectively. We test three cases for the upsampling factor r (2, 3, and 4).

4.2. Comparisons against Existing Methods

In this section, we compare our method against pixel-wise interpolation algorithms (including RDN [35], DPSR [34] and MetaSR [8] which are originally devised for tackling image super-resolution), slice-wise interpolation methods (including RRIN [14] and AdaCoF [13] which are originally proposed for settling video frame interpolation), and the slice interpolation method SAINT [24].

Quantitative Comparisons. Experimental results on the CT dataset are reported in Table 1 and 2. Our proposed method outperforms all algorithms by clear margins on both degradation settings. For example, under the $4\times$ interpolation setting, our method achieves 41.11dB and 37.87dB PSNR, which are 2.69dB and 1.17dB higher than the scores of SAINT, on the two degradation strategies, respectively.

Qualitative comparisons of our method against existing methods are presented in Fig. 5. We also visualize the super-resolution performance of SAINT our method in coronal and sagittal views under the $4\times$ upsampling setting. Our method has more detailed structures and apparent organ boundaries than other methods.

Model Size & Time Cost. For $4\times$ slice synthesis, the number of parameters of SAINT and our method is 44.2M and 46.9M, respectively. The training processes of SAINT and our method cost 18 and 31 hours, respectively. When processing a $512 \times 512 \times 36$ volume, SAINT and our method consume 35.96 and 13.25 seconds, respectively.

4.3. Ablation Study

This subsection conducts extensive inner comparisons on the CT dataset under the $4\times$ interpolation setting. Here, LR volumes are synthesized via direct downsampling. Core

Table 1. Comparison with existing slice synthesis, pixel-wise interpolation, and slice-wise interpolation algorithms on the CT dataset, under $2\times$, $3\times$, and $4\times$ upsampling settings. LR volumes are generated via direct downsampling.

Method	$2\times$				$3\times$				$4\times$			
	PSNR	$SSIM_a$	$SSIM_c$	$SSIM_s$	PSNR	$SSIM_a$	$SSIM_c$	$SSIM_s$	PSNR	$SSIM_a$	$SSIM_c$	$SSIM_s$
RDN [35]	43.51	0.9539	0.9519	0.9512	39.52	0.9402	0.9398	0.9376	37.89	0.9199	0.9210	0.9212
DPSR [34]	43.83	0.9690	0.9691	0.9682	38.82	0.9434	0.9423	0.9424	38.13	0.9166	0.9135	0.9154
MetaSR [8]	43.68	0.9547	0.9549	0.9548	39.90	0.9419	0.9425	0.9414	38.00	0.9211	0.9198	0.9214
RRIN [14]	43.45	0.9688	0.9691	0.9682	38.82	0.9434	0.9423	0.9424	38.10	0.9255	0.9232	0.9252
SRGAN [12]	43.22	0.9524	0.9521	0.9522	38.54	0.9433	0.9429	0.9425	37.91	0.9213	0.9209	0.9207
3D-MDSR [15]	44.31	0.9692	0.9698	0.9689	40.22	0.9489	0.9489	0.9490	38.20	0.9307	0.9302	0.9310
AdaCoF [13]	44.88	0.9749	0.9746	0.9747	40.92	0.9513	0.9498	0.9451	38.23	0.9311	0.9148	0.9150
SAINT [24]	44.43	0.9694	0.9641	0.9632	40.81	0.9448	0.9388	0.9416	38.42	0.9259	0.9175	0.9203
Ours	**46.81**	**0.9792**	**0.9784**	**0.9786**	**42.94**	**0.9631**	**0.9589**	**0.9604**	**41.11**	**0.9404**	**0.9385**	**0.9382**

Table 2. Comparison with existing slice synthesis, pixel-wise interpolation, and slice-wise interpolation algorithms on the CT dataset, under $2\times$, $3\times$, and $4\times$ upsampling settings. LR volumes are generated via blurring, downsampling and noise distortion.

Method	$2\times$				$3\times$				$4\times$			
	PSNR	$SSIM_a$	$SSIM_c$	$SSIM_s$	PSNR	$SSIM_a$	$SSIM_c$	$SSIM_s$	PSNR	$SSIM_a$	$SSIM_c$	$SSIM_s$
RDN [35]	41.67	0.9366	0.9369	0.9373	37.24	0.9210	0.9214	0.9211	35.23	0.9004	0.9010	0.9011
DPSR [34]	41.92	0.9389	0.9391	0.9387	37.87	0.9221	0.9223	0.9225	35.98	0.9022	0.9025	0.9021
MetaSR [8]	41.99	0.9392	0.9398	0.9390	37.95	0.9262	0.9259	0.9264	36.20	0.9078	0.9081	0.9084
RRIN [14]	41.43	0.9344	0.9341	0.9336	37.35	0.9234	0.9226	0.9233	35.58	0.9045	0.9045	0.9054
SRGAN [12]	41.10	0.9319	0.9313	0.9321	37.04	0.9204	0.9201	0.9207	35.09	0.8992	0.9004	0.9001
3D-MDSR [15]	42.03	0.9411	0.9406	0.9412	38.25	0.9310	0.9303	0.9306	36.21	0.9112	0.9114	0.9115
AdaCoF [13]	42.36	0.9439	0.9436	0.9427	38.72	0.9313	0.9311	0.9320	36.63	0.9131	0.9124	0.9142
SAINT [24]	42.43	0.9434	0.9431	0.9432	38.88	0.9352	0.9358	0.9348	36.70	0.9139	0.9134	0.9133
Ours	**43.98**	**0.9570**	**0.9568**	**0.9569**	**40.91**	**0.9505**	**0.9499**	**0.9499**	**37.87**	**0.9244**	**0.9239**	**0.9248**

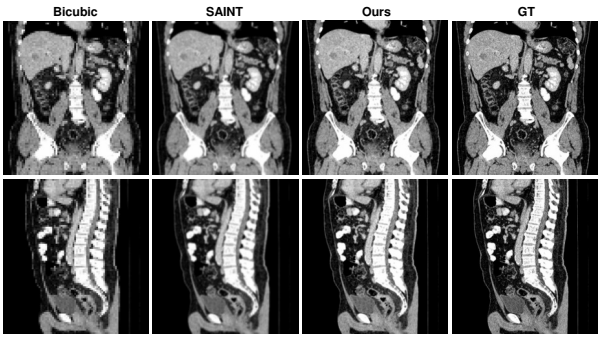


Figure 6. Visualization comparison. From left to right: bicubic interpolation; SAINT; our method, and ground-truth.

components of our method are teased apart to validate their effectiveness. The results are reported in Table 3.

Efficacy of cross-view mutual distillation is validated by removing consistency constraints L_c^{cmd} or L_s^{cmd}. In the baseline method, both L_c^{cmd} and L_s^{cmd} are not used, which means the cross-view mutual distillation is not applied. Compared to the baseline method, the full version of our approach brings PSNR and $SSIM_a$ gain of 2.53dB and 0.0118, respectively. Since pixel-wise interpolation in the coronal and sagittal views explore different context information for increasing the between-slice resolution, the knowledge learned from the two views is complementary to each other. Without distillation between axial view and coronal/sagittal view, the PSNR is decreased by 0.85dB/0.87dB in contrast to the PSNR of the full version. The distillation from two views performs better than the distillation with the single coronal or sagittal view. This can also be observed from an example of qualitative comparison in Fig. 7.

Table 3. Ablation study on critical components in our method. 'w/o L_c^{cmd} or L_s^{cmd}' means both L_c^{cmd} and L_s^{cmd} are not used for training. 'w/o L_c^{cmd}' ('w/o L_s^{cmd}') means L_c^{cmd} (L_s^{cmd}) is not used. 'w/o WT' means the loss on wavelet coefficients is not adopted. 'w/o memory' means the memory bank is not applied. For every variant, other parameters are set as in Section 4.1.

Variant	PSNR	$SSIM_a$
w/o L_c^{cmd} or L_s^{cmd}	38.58	0.9286
w/o L_c^{cmd}	40.26	0.9322
w/o L_s^{cmd}	40.24	0.9321
N=1	40.47	0.9325
w/o WT	40.56	0.9334
w/o memory	40.28	0.9324
final variant	41.11	0.9404

Efficacy of Incremental Interpolation. As shown in Table 3, using interpolation only once ($N = 1$) increases PSNR and $SSIM_a$ by 1.98dB and 0.0183, respectively. Applying two interpolation times ($N = 2$) can further improve the result with 0.64dB higher PSNR, compared to the variant with $N = 1$. This validates the effectiveness of the incremental interpolation scheme in our method. A qualitative compari-

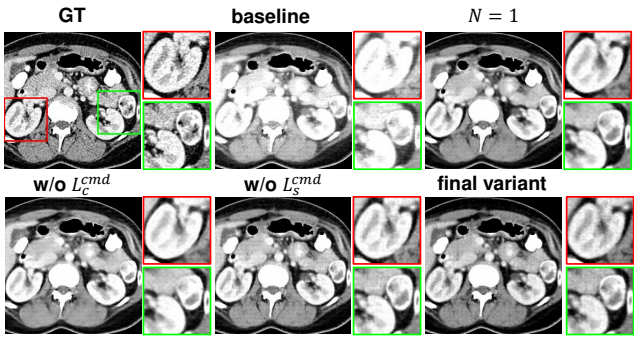


Figure 7. Examples of different variants of our method.

Table 4. Performance of different ensemble strategies for merging interpolation models.

Strategies	PSNR	$SSIM_a$
SInt-A	38.16	0.9140
SInt-A+PInt-C	38.47	0.9254
SInt-A+PInt-S	38.49	0.9261
SInt-A+PInt-C+PInt-S	38.58	0.9286
Ours SInt-A	38.49	0.9327
Ours SInt-A+PInt-C	40.24	0.9355
Ours SInt-A+PInt-S	40.26	0.9347
Ours SInt-A+PInt-C+PInt-S	**41.11**	**0.9404**

son is provided in Fig. 7. As can be observed, setting $N = 2$ induces an interpolation model capable of producing more accurate structures and textures.

Efficacy of Memory Bank. The adoption of the memory bank, which is used for storing common patterns. If the memory mechanism is not applied in the final variant of our method, the reduction on the PSNR metric reaches 0.63dB.

Performance of Using Different Ensemble Strategies. We report the results of single-view models and their simple combinations in Table 4. 'SInt-A', 'PInt-C', and 'PInt-S' stands for slice-wise interpolation in axial view, pixel-wise interpolation in coronal view, and pixel-wise interpolation in sagittal view, respectively. 'PInt-C/PInt-S+SInt-A' indicates 'PInt-C' or 'PInt-S' is integrated with 'SInt-A' through averaging their predictions. 'PInt-C+PInt-S+SInt-A' average the predictions of the three models. The simple combinations of pixel-wise and slice-wise interpolation models can improve the results of single models, which demonstrates that the two kinds of models are complementary to each other. Meanwhile, our proposed cross-view mutual distillation can help the combination strategies achieve much better performance.

Efficacy of Constraint on Wavelet Coefficients. The constraint on the wavelet coefficients emphasizes the reconstruction of high-frequency information. Without using the constraint on the wavelet coefficients, the PSNR metric is reduced by 0.55dB.

Using Different Values for γ. In Fig. 8, we discuss the impact of using different values for the parameter γ, namely the percents of points used for calculating consistency losses (11) and (12). When the deviation between the inferences of slice-wise and pixel-wise interpolation models is too large, one of the two models must predict an incorrect output. However, it is unable to identify which model is more reliable. Hence, we neglect those points at which the loss values are too large. From Fig. 8, we can see that our method achieves the best performance when $\gamma = 25\%$.

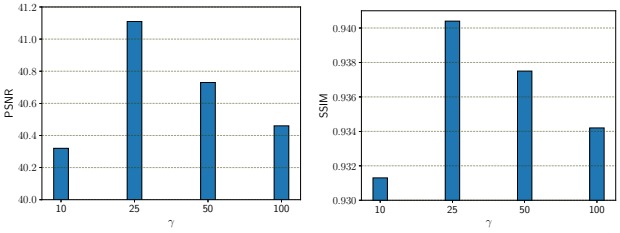


Figure 8. Performance of using different values for γ (%).

5. Conclusions

This paper proposes an incremental cross-view mutual distillation pipeline to tackle the self-supervised slice synthesis task. The mutual distillation between the slice-wise interpolation in the axial view and pixel-wise interpolation in the coronal and sagittal views contributes to a slice synthesizer with appealing performance. The learning process can be further enhanced via incrementally interpolating intermediate slices and then imposing cross-view distillation on these finer and finer intermediate slices. Extensive experiments on the CT dataset demonstrate the superiority of our method against existing slice synthesis methods.

Broader Impacts. Slices synthesized by our method still have apparent difference to real slices. In clinical applications, there exist risks for misleading the disease diagnosis process. It requires further research to improve the practicality of our method.

Limitations. In practical clinical scene, there exist many complicated artifacts during the acquisition of LR volumes, such as partial volume effect, motion blur, and streaks. In the current internal learning of our method, we use a simple way to approximate these artifacts. In the future, it deserves in-depth research on modeling the generation of these imaging artifacts for improving the generalization capacity in interpolating real-world LR CTs.

Acknowledgement. This work was supported in part by Key-Area Research and Development Program of Guangdong Province (No. 2021B0101200001), in part by the National Natural Science Foundation of China (No. 62003256, 61876140, 62027813, U1801265, and U21B2048), in part by Open Research Projects of Zhejiang Lab (No. 2019kD0AD01/010), and in part by MindSpore which is a new deep learning computing framework*.

*https://www.mindspore.cn/

References

[1] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 3

[2] Yuhua Chen, Feng Shi, Anthony G Christodoulou, Yibin Xie, Zhengwei Zhou, and Debiao Li. Efficient and accurate mri super-resolution using a generative adversarial network and 3d multi-level densely connected network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 91–99. Springer, 2018. 2

[3] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10663–10671, 2020. 2, 3, 4

[4] Myungsub Choi, Suyoung Lee, Heewon Kim, and Kyoung Mu Lee. Motion-aware dynamic architecture for efficient frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13839–13848, October 2021. 2

[5] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 3

[6] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 574–584, 2022. 2

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[8] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1575–1584, 2019. 3, 6, 7

[9] Changhui Jiang, Qiyang Zhang, Rui Fan, and Zhanli Hu. Super-resolution ct image reconstruction based on dictionary learning and sparse representation. *Scientific reports*, 8(1):1–10, 2018. 3

[10] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. *arXiv preprint arXiv:2012.08512*, 2020. 2

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[12] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 3, 7

[13] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5316–5325, 2020. 2, 6, 7

[14] Haopeng Li, Yuan Yuan, and Qi Wang. Video frame interpolation via residue refinement. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2613–2617. IEEE, 2020. 2, 6, 7

[15] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 7

[16] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020. 3

[17] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018. 2

[18] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017. 2

[19] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017. 2

[20] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020. 2, 3, 4

[21] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14372–14381, 2020. 4, 5

[22] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14539–14548, October 2021. 2

[23] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2398–2407, 2019. 2

[24] Cheng Peng, Wei-An Lin, Haofu Liao, Rama Chellappa, and S Kevin Zhou. Saint: spatially aware interpolation network for medical slice synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7750–7759, 2020. 1, 2, 3, 6, 7

[25] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets:

Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 3

[26] Irina Sánchez and Verónica Vilaplana. Brain mri super-resolution using 3d generative adversarial networks. *arXiv preprint arXiv:1812.11440*, 2018. 2

[27] Amber L Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram Van Ginneken, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*, 2019. 5

[28] Yucheng Tang, Dong Yang, Wenqi Li, Holger Roth, Bennett Landman, Daguang Xu, Vishwesh Nath, and Ali Hatamizadeh. Self-supervised pre-training of swin transformers for 3d medical image analysis. *arXiv preprint arXiv:2111.14791*, 2021. 2

[29] Longguang Wang, Xiaoyu Dong, Yingqian Wang, Xinyi Ying, Zaiping Lin, Wei An, and Yulan Guo. Exploring sparsity in image super-resolution for efficient inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4917–4926, June 2021. 3

[30] Wenbin Xie, Dehua Song, Chang Xu, Chunjing Xu, Hui Zhang, and Yunhe Wang. Learning frequency-aware dynamic network for efficient super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4308–4317, October 2021. 3

[31] Gang Xu, Jun Xu, Zhen Li, Liang Wang, Xing Sun, and Mingming Cheng. Temporal modulation network for controllable space-time video super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2

[32] Chenyu You, Guang Li, Yi Zhang, Xiaoliu Zhang, Hongming Shan, Mengzhou Li, Shenghong Ju, Zhen Zhao, Zhuiyang Zhang, Wenxiang Cong, Michael W. Vannier, Punam Saha, Eric Hoffman, and Ge Wang. Ct super-resolution gan constrained by the identical, residual, and cycle learning ensemble (gan-circle). *IEEE Transactions on Medical Imaging*, PP:1–1, 06 2019. 2

[33] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 3

[34] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Deep plug-and-play super-resolution for arbitrary blur kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1671–1681, 2019. 6, 7

[35] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018. 3, 6, 7

[36] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018. 3