Appendix

A. Details of 4D datasets

ACDC. The ACDC dataset features an average of 10.02±2.20 frames between the end-systolic and end-diastolic phases in the training set, with the test set presenting an average of 8.80±2.48 frames. All cardiac MRI scans have been uniformly resized. Following this resizing process, min-max scaling is applied to ensure consistent scaling across all scans.

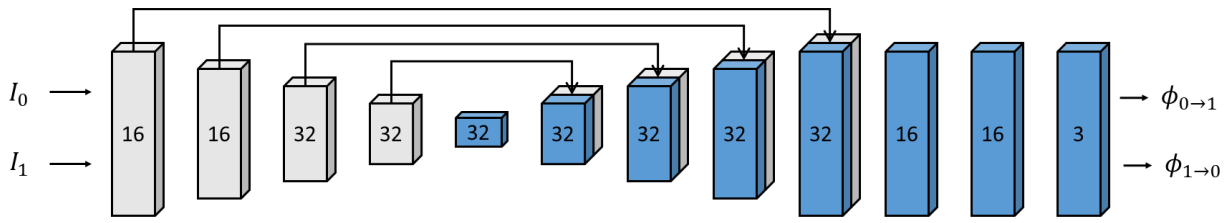
4D-Lung. In the case of the 4D-lung dataset, the models are trained to predict the four intermediate frames (10%, 20%, 30%, 40%) between the end-inspiratory (0%) and end-expiratory (50%) phases. Only CT images captured using kilovoltage energy are included in the study due to their superior image quality. Each lung CT scan is adjusted to the lung window range (-1400 to +200 Hounsfield unit) [15] and subjected to centering and min-max scaling. Subsequently, bed removal is performed using the following method: pixels exceeding a certain threshold (-500 HU in this study) are assigned a value of 1, while all other pixels are set to 0, creating a binarized map. The binarized map undergoes erosion/dilation [62] to identify the most prominent body contour mask. By getting the resulting body contour mask to the corresponding voxel region of the given images, a bed-removed CT image is obtained. All the lung CT images are resized to $128 \times 128 \times 128$.

B. Details of baseline models

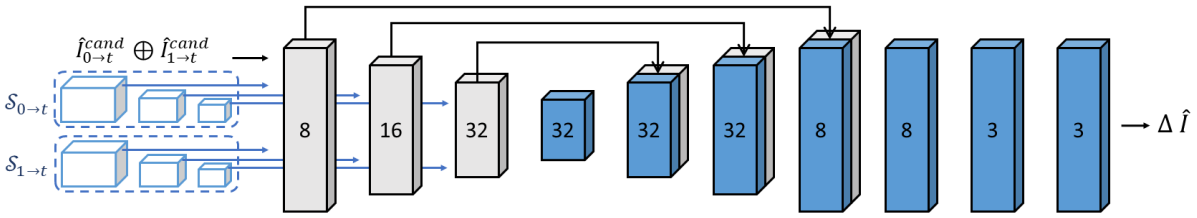
The following three unsupervised models and two supervised models are used as the baseline models for our main result: VoxelMorph [3], TransMorph [10], Fourier-Net+ [24], R2Net [27], IDIR [69], DDM [30] for unsupervised models, and SVIN [16], MPVF [68] for supervised models. To the best of our knowledge, this selection covers the most pertinent and all current baseline models in the field, providing a comprehensive benchmark for our study.

Unsupervised models. The VoxelMorph employs the exact same model architecture as our flow calculation model, as discussed in Appendix C.1. For TransMorph, we follow the TransMorph-Large framework from the original paper. In the case of Fourier-Net+, R2Net, IDIR, and DDM, we utilize the default architecture outlined in the original paper.

Supervised models. In our study involving SVIN, we adhered to the official architecture as described in the foundational paper. For MPVF, we applied the architecture specified for the ACDC dataset, as outlined in the original publication. However, our experience with the 4D-lung dataset presented unique challenges. Despite the original study using a distinct lung preprocessing method, which resulted in larger data sizes, and reporting successful execution on a V100 GPU with 32GB of memory, our attempts to run their code on an A6000 GPU with 48GB of memory encountered memory issues. Upon contacting the authors, we learned that no official code was available for the 4D-lung dataset. Consequently, we were compelled to arbitrarily modify the model size to accommodate our 48GB memory constraint. This entailed reducing the encoder inplanes from [32, 64, 128] to [8, 16, 32], decreasing the number of ViT heads from 4 to 2, lowering the ViT num classes from 1000 to 300, and diminishing the hidden dimension from 256 to 64. Please note that although we reduced the model to fit a 48GB memory constraint, our measurements were conducted on a model size larger than the original model's 32GB specification.



(a) Architecture of flow calculation model



(b) Architecture of reconstruction model

Figure 6. The architecture of the flow calculation and reconstruction models consisting of encoder and decoder layers. The encoder layers are represented by gray boxes, while the decoder layers are represented by blue boxes. The numbers associated with each box indicate the number of features in the corresponding convolutional filter.

C. UVI-Net details

C.1. Flow calculation model

The flow calculation model follows the network architecture illustrated in Fig. 6a, which is based on VoxelMorph [3]. The model processes a single input by combining the images I_0 and I_1 into a 2-channel 3D image. Then, it outputs 3-channel 3D flows, where each channel represents the displacement along each dimension. The flow model incorporates 3D convolutions in both the encoder and decoder stages with a kernel size of 3. LeakyReLU layer with a negative slope of 0.2 follows each convolutional operation.

In the encoder, strided convolutions with stride size 2 are utilized to reduce the spatial dimensions by half at each layer. Conversely, the decoding involves a combination of upsampling, convolutions, and concatenation of skip connections. As a result, the model outputs the flows $\phi_{0\to1}$ and $\phi_{1\to0}$, each warping I_0 to resemble I_1 and I_1 to resemble I_0, respectively.

C.2. Reconstruction model

Fig. 6b describes the architecture of the reconstruction model, based on 3D-UNet [55]. We employ a single image $\hat{I}^{cand}_{0\to t} \oplus \hat{I}^{cand}_{1\to t}$, which is a weighted sum of two candidate images, in conjunction with three levels of multi-resolution features, each possessing channel dimensions of 4, 8, and 16, respectively. The model's first encoder layer receives an input composed of two channel-wise concatenated warped features and an image $\hat{I}^{cand}_{0\to t} \oplus \hat{I}^{cand}_{1\to t}$. Advancing to the subsequent layers, the model concatenates features of half and quarter resolutions at the second and third encoder layers. Thereafter, the model returns the image difference $\Delta\hat{I}$, which will be added to the input to acquire the final estimated image $\hat{I}_t$. The architecture of the reconstruction model follows details similar to those of the flow calculation model.

C.3. Additional training details

In our training process, we employ the Adam optimizer [34] with a learning rate 2×10^{-4} for 200 epochs, configuring the batch size as 1. For instance-specific optimization, models are fine-tuned for 100 epochs on the given test sample while maintaining the same experimental settings as in the previous training. The results are presented in a straightforward setup, with all loss coefficients uniformly set to 1.

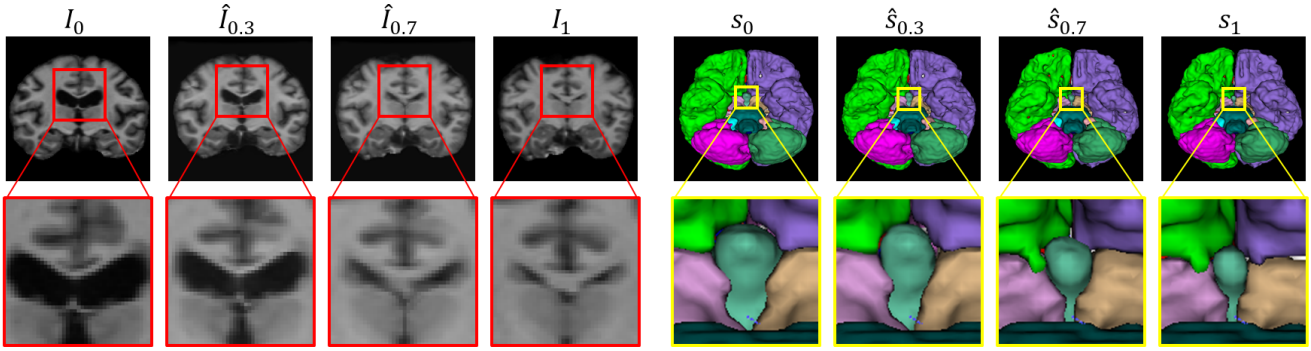


Figure 7. Visualization of data augmentation using our approach. Given I_0, I_1 and s_0, s_1, we report the generated image and label when $t = 0.3, 0.7$. This is a visualization based on data from the OASIS dataset.

D. Downstream Task

D.1. Method

We propose an effective 3D data augmentation technique based on our interpolation framework. To extend the interpolation task to 3D data augmentation, we generate new data by inputting randomly selected pairs of 3D images from the training dataset that share common types of segmentation labels. Here, we utilize time t as an interpolation degree for augmentation. Furthermore, inspired by previous works [3, 10], we incorporate the segmentation labels as supplementary information to enrich the augmented dataset.

Let s_0 and s_1 represent the organ segmentation of I_0 and I_1. When calculating flow fields, we only use I_0 and I_1, excluding segmentation labels. Using the calculated flows, we calculate $\hat{s}^{cand}_{t_1\to 0}$, $\hat{s}^{cand}_{t_2\to 0}$, $\hat{s}^{cand}_{t_2\to 1}$ and $\hat{s}^{cand}_{t_3\to 1}$ similar to the procedure of image. Finally, we ensure that $\hat{s}^{cand}_{t_1\to 0}$ and $s^{cand}_{t_2\to 0}$ have cycle consistency between s_0, while $s^{cand}_{t_2\to 1}$ and $s^{cand}_{t_2\to 1}$ have cycle consistency with s_1.

When labels are used during training, we expand the segmentation map into K binary masks to enable backpropagation, where K represents the total number of labels in the segmentation maps. Since Dice score [13] is commonly used to quantify optical flow performance [3, 10], we directly minimized the Dice loss [44].

D.2. Experimental setting

Datasets. For the segmentation dataset for augmentation, three 3D medical datasets are used. OASIS [20] is a brain dataset comprising 414 T1-weighted MRI scans and the corresponding segmentation labels for 36 organs, including the background label released from VoxelMorph [3]. IXI [1] is another brain MRI dataset with segmentation labels for 31 organs, including the background [10] released from TransMorph [10]. All the brain MRI scans are skull-stripped and resized to $128\times128\times128$. In both datasets, the first 20 samples are used for training, while the rest are included in the test set. Lastly, MSD-Heart [60] is an MRI dataset with one label (excluding background) and resized to $128\times128\times64$. Since MSD-Heart has only 20 data, we use 10 data for training and 10 for testing with background loss.

[1] https://brain-development.org/ixi-dataset/

Method	OASIS	IXI	MSD-Heart
Vanilla	0.821	0.801	0.755
VM [3]	0.825	0.813	0.803
TM [10]	0.831	0.810	0.773
Fourier-Net+ [24]	0.822	0.802	0.809
R2Net [27]	0.621	0.688	0.789
DDM [30]	0.826	0.806	0.818
Ours (w/o inst opt.)	**0.843**	**0.818**	**0.831**

Table 3. Segmentation results on three datasets. Experiments are conducted by adding augmentation data at a scale of 10x to the real data. Dice score is used as the averaged performance metric for three segmentation models.

Segmentation models. To perform 3D segmentation, we utilize three publicly available models from MONAI package[2]: 3D-UNet [55], VNet [1], and UNETR [18]. The segmentation models are trained for 15,000 iteration steps the final Dice score at the last iteration is recorded. Adam optimizer [34] with an initial learning rate 1×10^{-4} is used, and batch size is set to 1. For loss function, the weighted sum of Dice [44] and Cross Entropy [57] losses is used. For augmented data generation, which expands the original dataset size by a factor of ten, we employed alpha sampling ratios of $t = 0.1, 0.2, \ldots, 1.0$.

D.3. Result

We have successfully generated pairs of images and labels, as illustrated in Fig. 7. Detailed results presented in Tab. 3 reveal that our approach consistently outperforms competing methods, delivering superior performance across a diverse range of conditions. This includes variations in dataset types and the use of different segmentation models, underscoring the robustness and versatility of our methodology.

[2] https://monai.io/

Dataset	Loss function			PSNR ↑	NCC ↑	SSIM ↑	NMSE ↓	LPIPS ↓
	$\mathcal{L}_{warp}$	$\mathcal{L}_{image}$	$\mathcal{L}_{reg}$					
Cardiac	✓	✓		33.01	0.563	0.975	2.679	1.076
	✓		✓	33.16	0.562	0.975	2.691	1.194
	✓	✓	✓	**33.57**	**0.565**	**0.977**	**2.409**	**1.134**

Table 4. Ablation results of loss terms. $\mathcal{L}_{image}$ and $\mathcal{L}_{reg}$ are components of $\mathcal{L}_{cyc}$. NMSE and LPIPS are written in units of 10^{-2}.

Dataset	$\mathcal{L}_{image}$		PSNR ↑	NCC ↑	SSIM ↑	NMSE ↓	LPIPS ↓
	NCC	ρ					
Cardiac	✓		33.55	**0.565**	**0.977**	**2.406**	1.189
		✓	33.50	**0.565**	**0.977**	2.437	1.316
	✓	✓	**33.57**	**0.565**	**0.977**	2.409	**1.134**

Table 5. Ablation results of loss terms. NMSE and LPIPS are written in units of 10^{-2}. $\mathcal{L}_{image}$ is used for warping loss and cyclic loss, and ρ stands for Charbonnier loss.

Dataset	Feature extractor	PSNR ↑	NCC ↑	SSIM ↑	NMSE ↓	LPIPS ↓
Cardiac	None	33.53	**0.565**	**0.977**	2.410	1.163
	Edge detection	33.49	**0.565**	**0.977**	2.434	**1.101**
	U-Net	33.50	**0.565**	**0.977**	2.445	1.151
	Single-scale CNN	33.49	0.564	**0.977**	2.448	1.116
	Multi-scale CNN	**33.57**	**0.565**	**0.977**	**2.409**	1.134

Table 6. The ablation results for the feature extraction module. Extract type "None" indicates not using feature extraction.

E. Additional experimental results

We further substantiate our methodology through a series of ablation studies designed to broaden the empirical results. All reported outcomes represent the values derived from three distinct experimental runs.

E.1. Ablation studies of loss term

The ablation results of loss terms conducted on the ACDC dataset are summarized in Tab. 4 and Tab. 5. As indicated in Tab. 4, integrating each component of cyclic loss, which are $\mathcal{L}_{image}$ and $\mathcal{L}_{reg}$, significantly improves the performance of intermediate image synthesis. Furthermore, Tab. 5 demonstrates that the combined application of NCC and Charbonnier losses leads to a performance improvement compared to the application of each loss term independently.

E.2. Ablation studies of feature extractor model

The Tab. 6 presents the results of ablation studies on the feature extraction model, conducted on the ACDC dataset. In our comparative analysis, we demonstrate that our feature extraction methodology exhibits superior performance compared to scenarios where no feature extraction model is implemented. Additionally, we explored alternative methods of feature extraction, including: (1) using the Canny edge detector, (2) employing a simple U-Net architecture, and (3) utilizing a CNN module with single-scale warped images. Our approach outperformed other feature extraction modules in overall metric aspects. Moreover, some metrics in those modules showed performance worse than cases where no feature extraction was applied.

E.3. Additional qualitative results

We present a series of additional qualitative results in Fig. 8. Our approach demonstrate the superior results against various baseline methods. This not only underscores our method's enhanced alignment and coordination but also showcases its ability to generate outcomes that are more accurate and realistic. The visual evidence presented here plays a crucial role in substantiating the quantitative metrics we have reported, offering a holistic view of our model's capabilities in real-world scenarios.

E.4. Visualization for extrapolation

The Fig. 9 visualizes the extrapolation results, particularly for $\hat{I}_{-0.5}$ and $\hat{I}_{1.5}$, along with the corresponding optical flow and source images I_0 and I_1. These images represent the most extreme cases of extrapolation in our study. To ensure the credibility and real-world applicability of the results, they have been rigorously examined by a board-certified radiation oncologist. The evaluation focused on determining whether the extrapolated images exhibit any excessive or unnatural changes that could undermine their practical utility. This ensures that using extrapolation in our method does not present significant complications.

E.5. Visualization results for sequential 4D images

Fig. 10 visualizes the prediction results over time for the entire 4D sequence. As the baseline results, we introduce the interpolated images through the application of linear scaling to VoxelMorph, which serves as the backbone registration model within our framework. It can be observed that our approach more effectively captures fine-grained details and predicts the ground truth compared to the baseline.

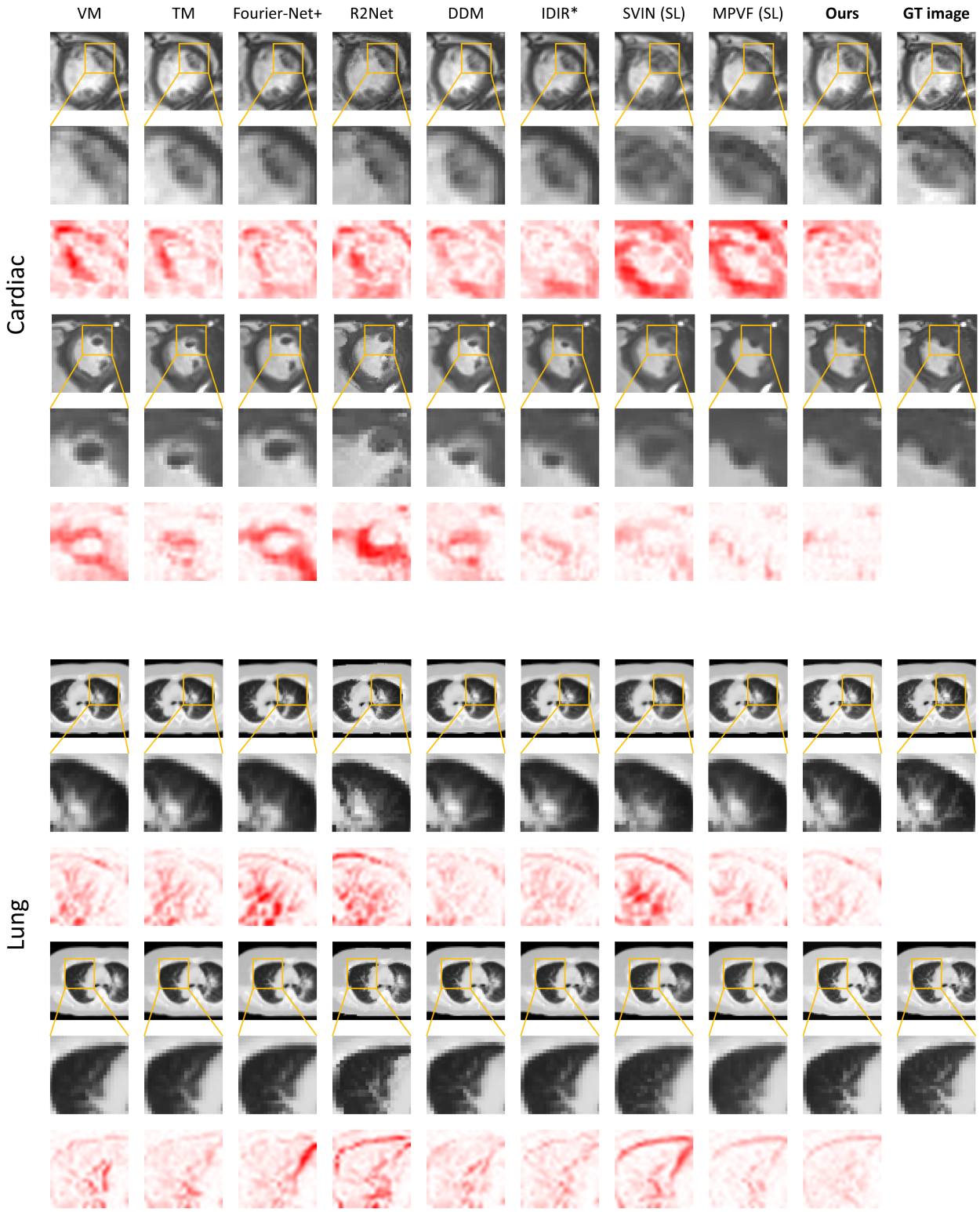


Figure 8. Additional visualization examples demonstrating our proposed method's effectiveness for 4D interpolation. The model marked with an '*' is trained exclusively on the test set, while models marked with '(SL)' are trained using supervised learning. Every third row shows the difference between each model and the ground truth image, where greater pixel value indicates a larger divergence from the ground truth.

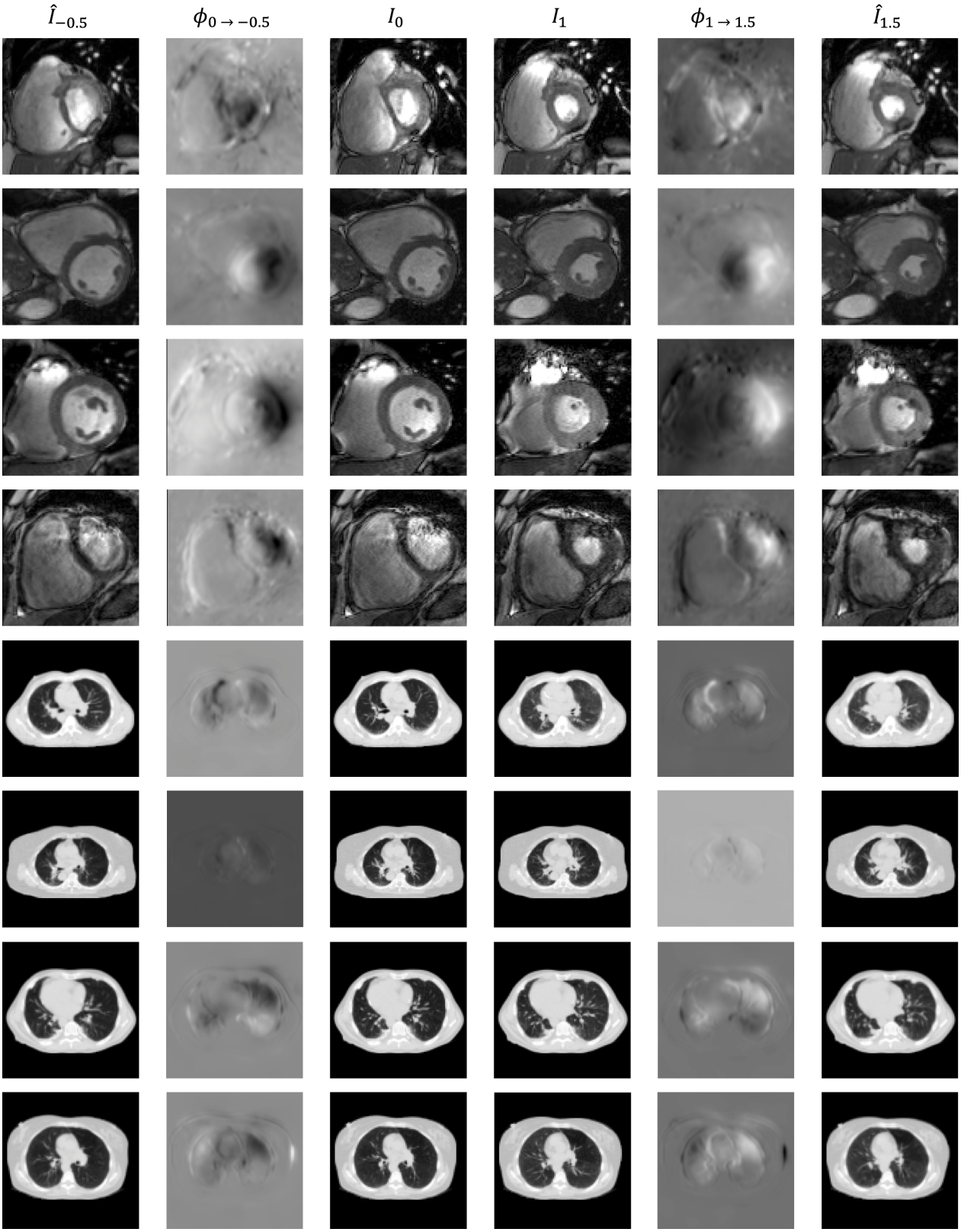
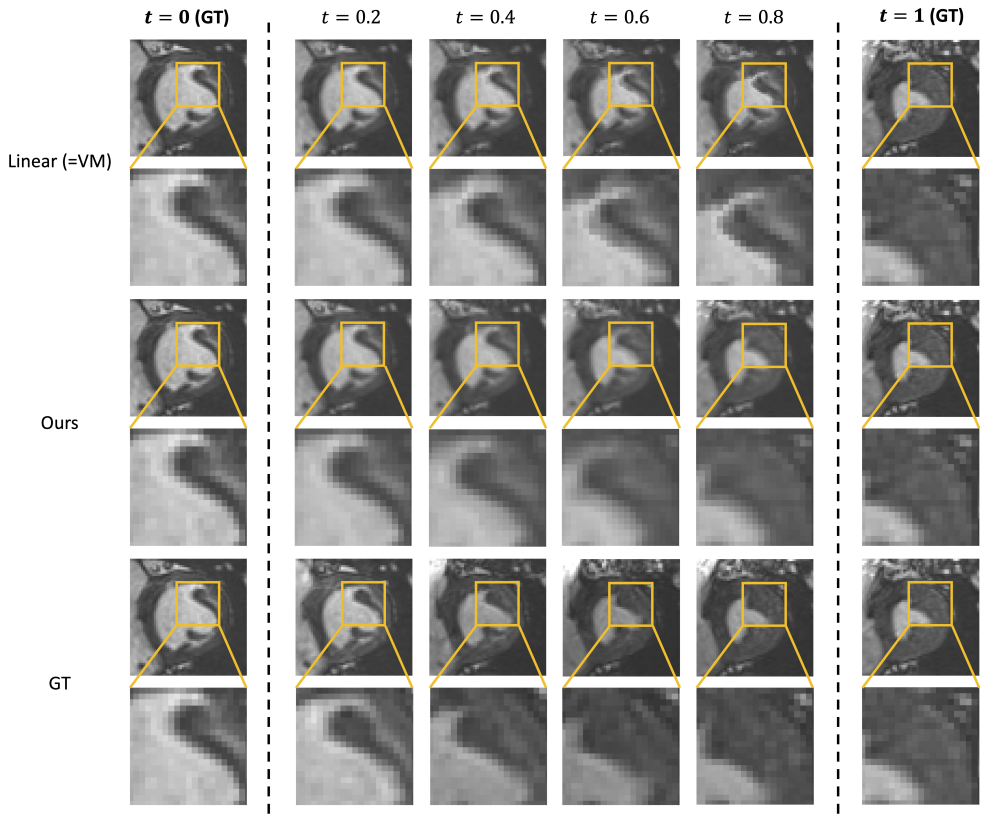
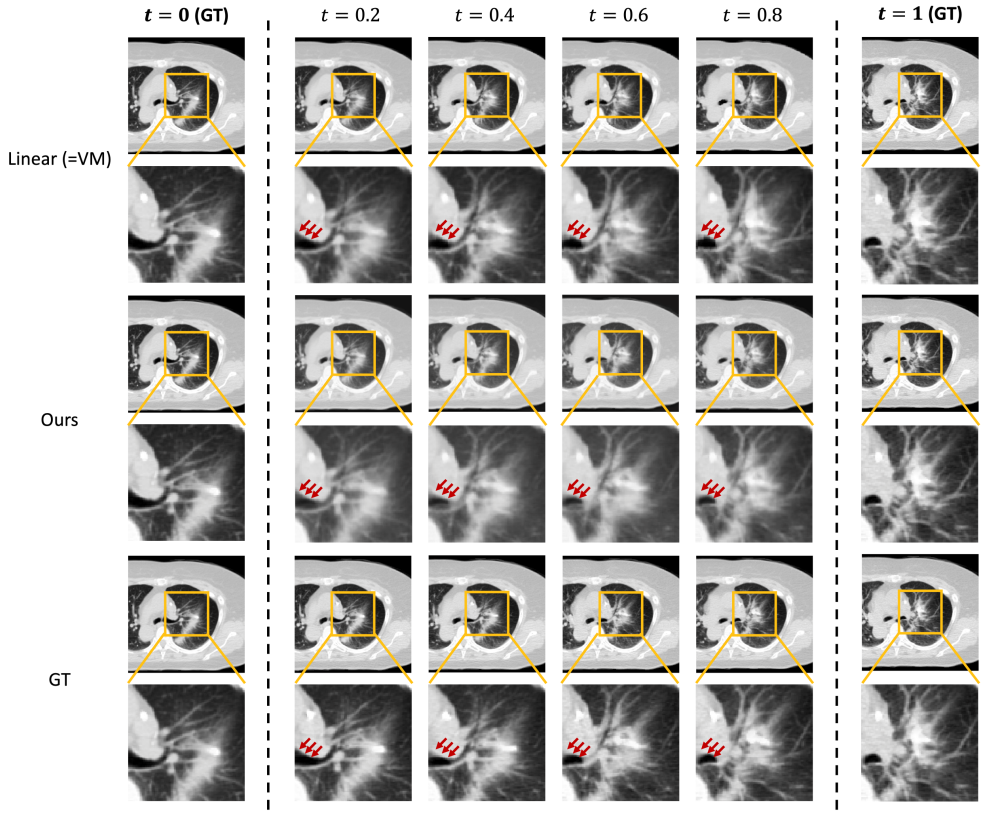


Figure 9. Extrapolation examples for the cardiac and lung datasets. The optical flows presented below pertain to the x-axis direction.



(a) Example of the cardiac dataset.



(b) Example of the lung dataset.

Figure 10. Qualitative results on the prediction of 4D image series over time. For lung images, we present the results in high resolution by upsampling the size of the registration field by a factor of four. In the provided figure, the first and last columns represent the ground truth images. Our model demonstrates a superior ability to capture the fine-grained structures like left main bronchus (indicated by red arrows) compared to the baseline.