Poster Session

THU-AM-192

CycleINR: Cycle Implicit Neural Representation for Arbitrary-Scale Volumetric Super-Resolution of Medical Data

Wei Fang[1,2], Yuxing Tang[1], Heng Guo[1,2], Mingze Yuan[1], Tony C. W. Mok[1,2], Ke Yan[1,2], Jiawen Yao[1,2], Xin Chen[3], Zaiyi Liu[3], Le Lu[1], Ling Zhang[1], Minfeng Xu[1,2]

[1]DAMO Academy, Alibaba Group, [2]Hupan Lab, [3]Guangdong Province People's Hospital

CVPR SEATTLE, WA JUNE 17-21, 2024

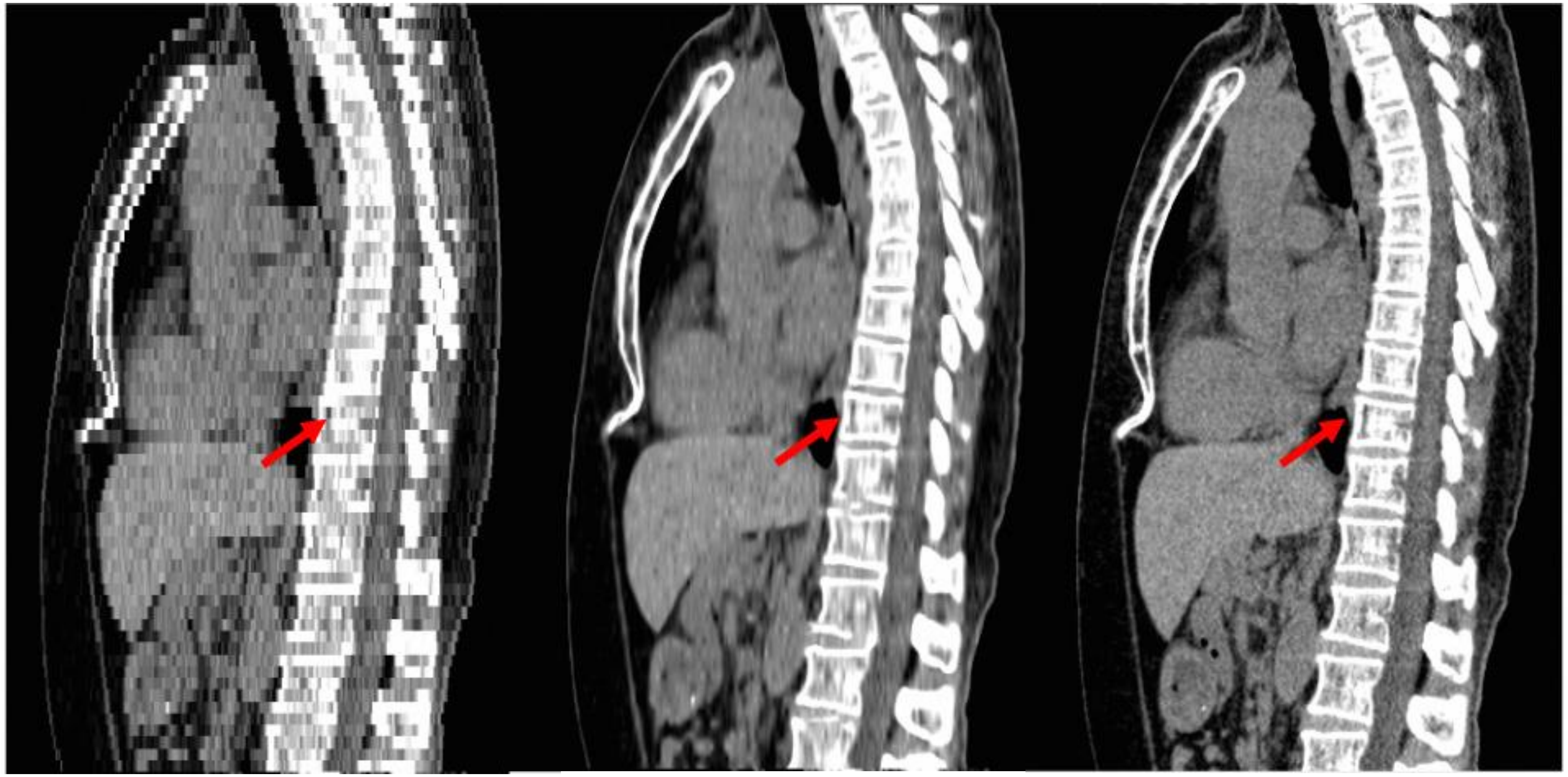
CycleINR Quick View

A super-resolution model for Medical data, such as CT and MRI.

Flexible super-resolution ratio.

Mitigate over-smoothing problem by introducing cycle-consistent loss.

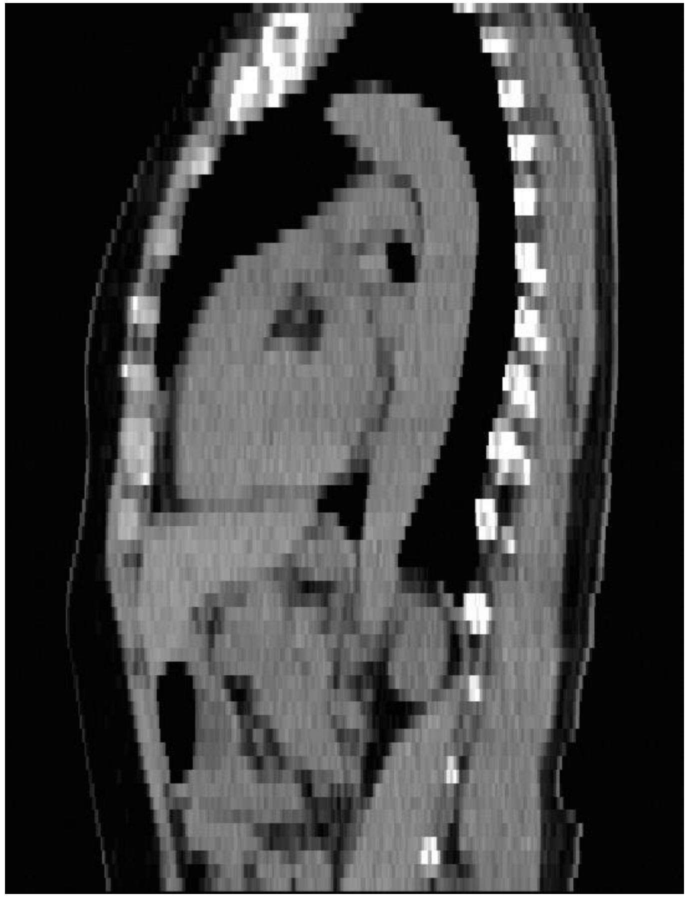
Extensive experiments on image generation and downstream task.



Original 5mm-spacing image | CycleINR Predicted, 1mm-spacing image | Ground Truth, 1mm-spacing image

Background and Motivation

- Anisotropic resolution in volumetric medical data
- Z spacing is worse than X and Y axis
- Hindering optimal viewing experiences
- Impeding the use of downstream analysis algorithms



One 5 mm spacing image sample

Two main challenges

- **Non-adaptive super-resolution ratios**
  - Most previous methods can only handle one specific super-resolution ratio for one model.
  - Fractional super-resolution ratio like 5/3 will make this problem more complex.

- The **over-smoothing** problem
  - The newly generated slices often exhibit over-smoothing for previous deep learning methods.
  - This will create a noticeable slice-wise inconsistency issue in volumetric scenarios, which is especially obvious when scrolling through the slices.

Our CycleINR Solution

SEATTLE, WA CVPR JUNE 17-21, 2024

- INR Model for Flexible Arbitrary-scale Super-resolution
  - INR model uses a neural network to represent a image. The network's input is the coordinate and the output is the corresponding pixel value.

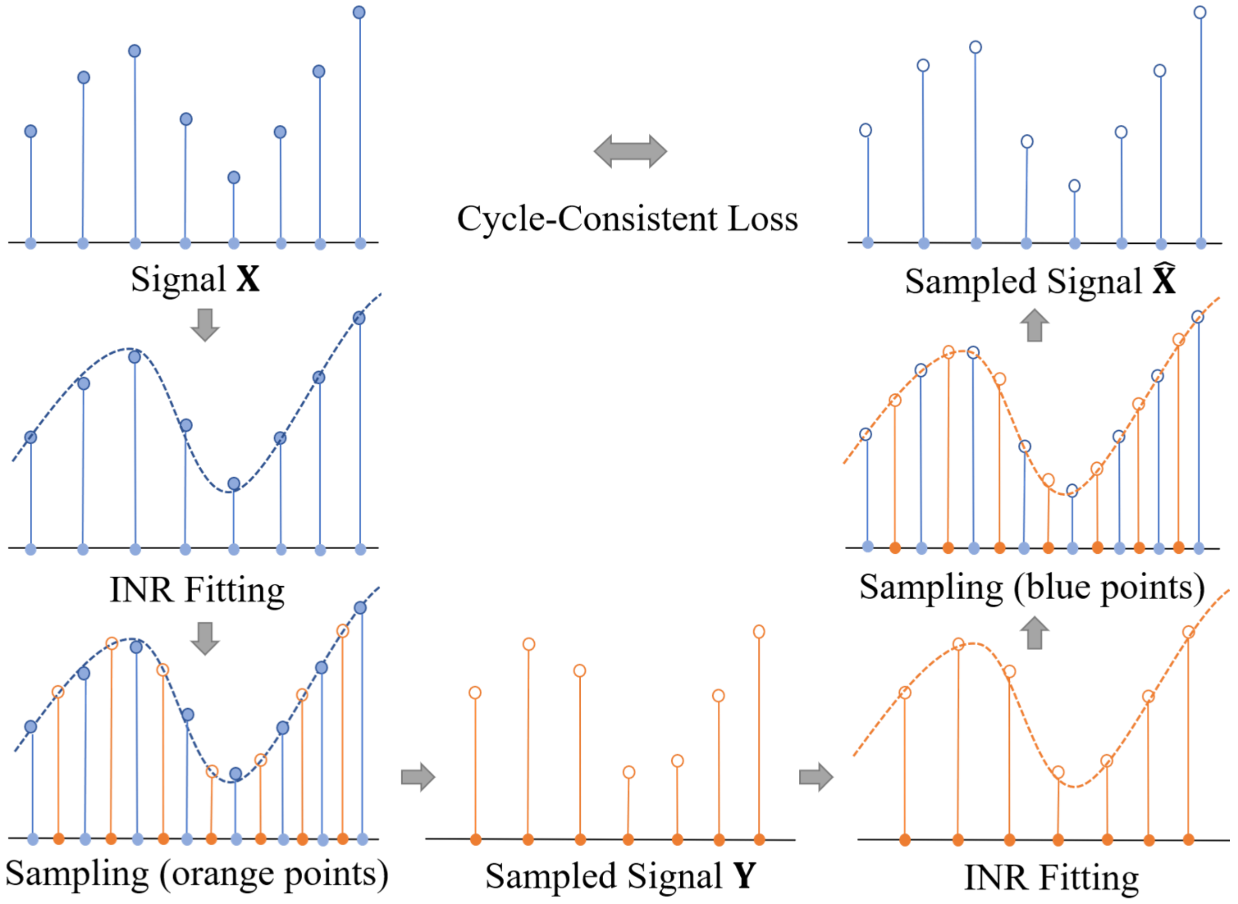
$$I = f_\theta(\mathbf{x})$$

  - Thus LR and HR images are sampled data of one continual signal at different sampling rates.
  - Once we get the trained INR network, we achieve an arbitrary-scale super-resolution model.
- Cycle-consistent Loss for Overcoming Over-smoothing
  - Utilizing the cycle-consistency between generated slices and the original slices under the INR setting.

Cycle-consistent loss

Steps for constructing cycle-consistent loss:

- Use signal $\mathbf{X}$ to fit a continuous Implicit Neural Representation (INR) function
- New points ($\mathbf{Y}$) are sampled from this function to create a new INR function.
- The signal $\hat{\mathbf{X}}$ is then sampled from the new function at the same positions as $\mathbf{X}$.
- Constructing cycle-consistent loss by assessing the similarity between $\hat{\mathbf{X}}$ and $\mathbf{X}$.

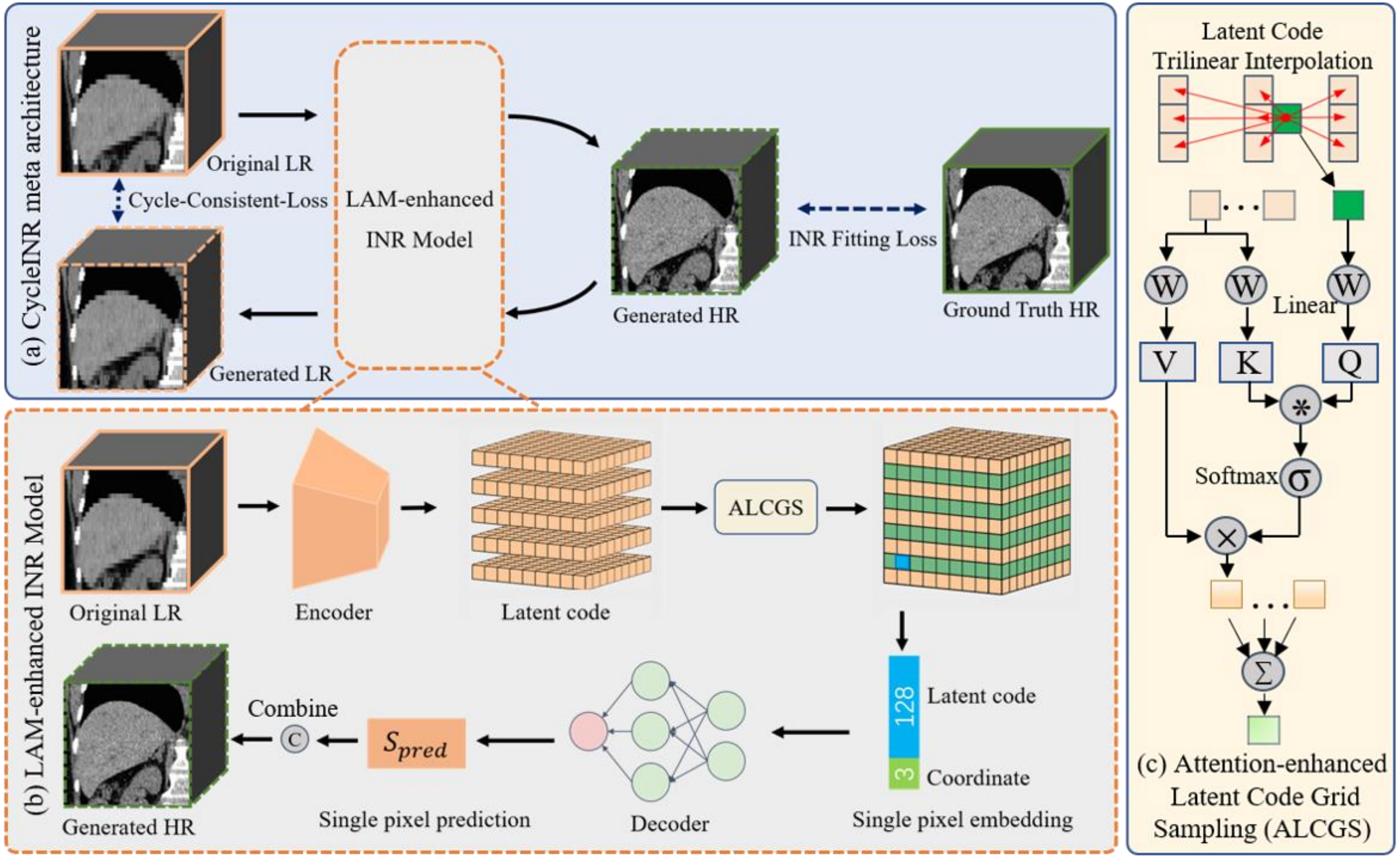


The concept of cycle-consistent loss

Framework Overview

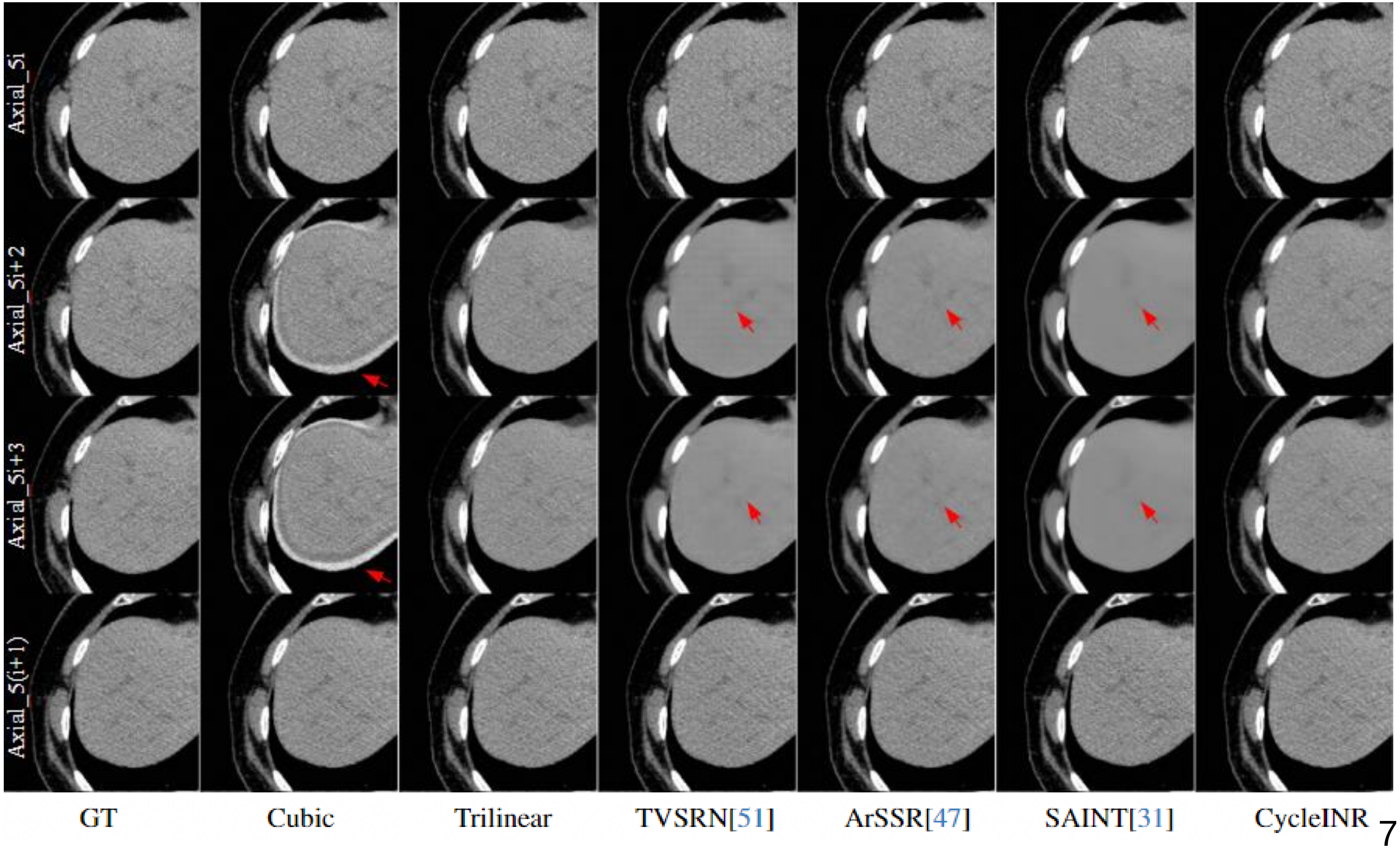
The INR model includes:

- CNN encoder
- Attention-enhanced Latent Code Grid Sampling (ALCGS)
- Fully connected decoder

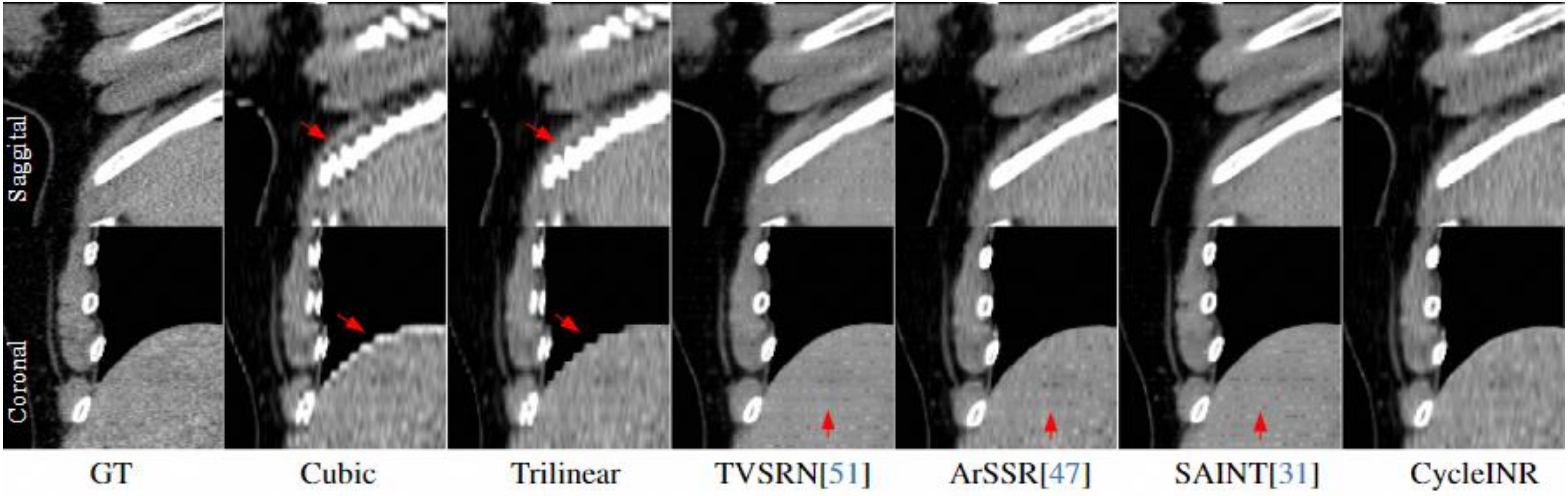


Axial View Results

- Better details in newly generated slices
- Better consistency between slices



Coronal and Sagittal Views



- Better bone structure reconstruction quality
- Less Jagged artefacts
- Mitigated horizontal lines due to improved slice-wise noise level consistency

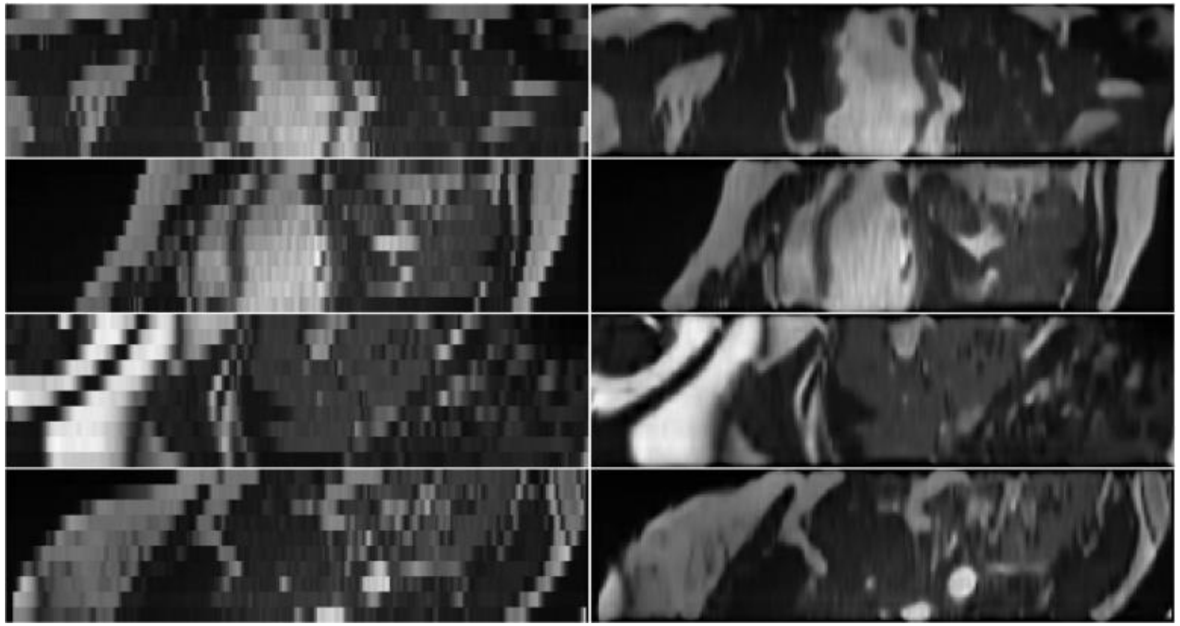
Quantitative Results on CT

CVPR
SEATTLE, WA JUNE 17-21, 2024

- Significant improvements on LPIPS
- Suboptimal PSNR and SSIM does not necessarily signify a negative outcome since these two metric favor smoothness

Scale	Method	PSNR(↑)	SSIM(↑)	LPIPS_alex(↓)	LPIPS_vgg(↓)	LPIPS_squeeze(↓)	SNLI(↓)
x2	Cubic	39.5039	0.9705	**0.0148**	0.0639	**0.0161**	0.5318
	Trilinear	40.6750	0.9757	0.0201	**0.0618**	0.0189	0.8450
	TVSRN [52]	43.6167	0.9820	0.0311	0.0824	0.0310	1.4307
	ArSSR [48]	42.6713	0.9799	0.0370	0.0913	0.0357	1.5738
	SAINT [32]	**44.3977**	**0.9833**	0.0361	0.0867	0.0354	1.6724
	CycleINR (Ours)	43.0137	0.9805	0.0201	0.0625	0.0206	**0.3527**
x3	Cubic	35.0140	0.9393	0.0306	0.1084	0.0292	**0.5322**
	Trilinear	36.5867	0.9515	0.0328	0.0973	0.0293	0.7422
	TVSRN [52]	40.4857	0.9696	0.0607	0.1419	0.0583	1.4389
	ArSSR [48]	39.3398	0.9659	0.0443	0.1135	0.0413	1.3026
	SAINT [32]	**40.8705**	**0.9711**	0.0625	0.1404	0.0599	1.7535
	CycleINR (Ours)	39.2748	0.9644	**0.0293**	**0.0902**	**0.0280**	0.6674
x5	Cubic	31.0470	0.8896	0.0562	0.1627	0.0480	**0.5336**
	Trilinear	32.6606	0.9106	0.0525	0.1413	0.0433	0.6682
	TVSRN [52]	36.8459	0.9503	0.0927	0.1989	0.0862	1.3593
	ArSSR [48]	35.1960	0.9394	0.0611	0.1485	0.0528	1.1714
	SAINT [32]	**36.9940**	**0.9519**	0.0996	0.2044	0.0951	1.6996
	CycleINR (Ours)	35.0022	0.9354	**0.0464**	**0.1289**	**0.0399**	0.6050

Quantitative Results on MR



Visualization results before and after super-resolution.

Method	PSNR(↑)	SSIM(↑)	LPIPS_a(↓)	LPIPS_v(↓)	LPIPS_s(↓)	SNLI(↓)	FID(↓)
Cubic	29.097	0.854	0.0202	0.152	0.0461	0.141	25.884
Trilinear	30.670	0.879	0.0196	0.133	0.0435	0.126	26.712
TVSRN [51]	30.710	0.882	0.0468	0.253	0.0969	0.140	21.006
ArSSR [47]	30.564	0.898	0.0323	0.148	0.0608	0.134	15.539
SAINT [31]	**32.120**	**0.911**	0.0536	0.177	0.0902	0.263	17.652
CycleINR	31.017	0.902	**0.0184**	**0.123**	**0.0424**	**0.120**	**15.449**

Downstream Task

Performed on MSD liver tumor dataset

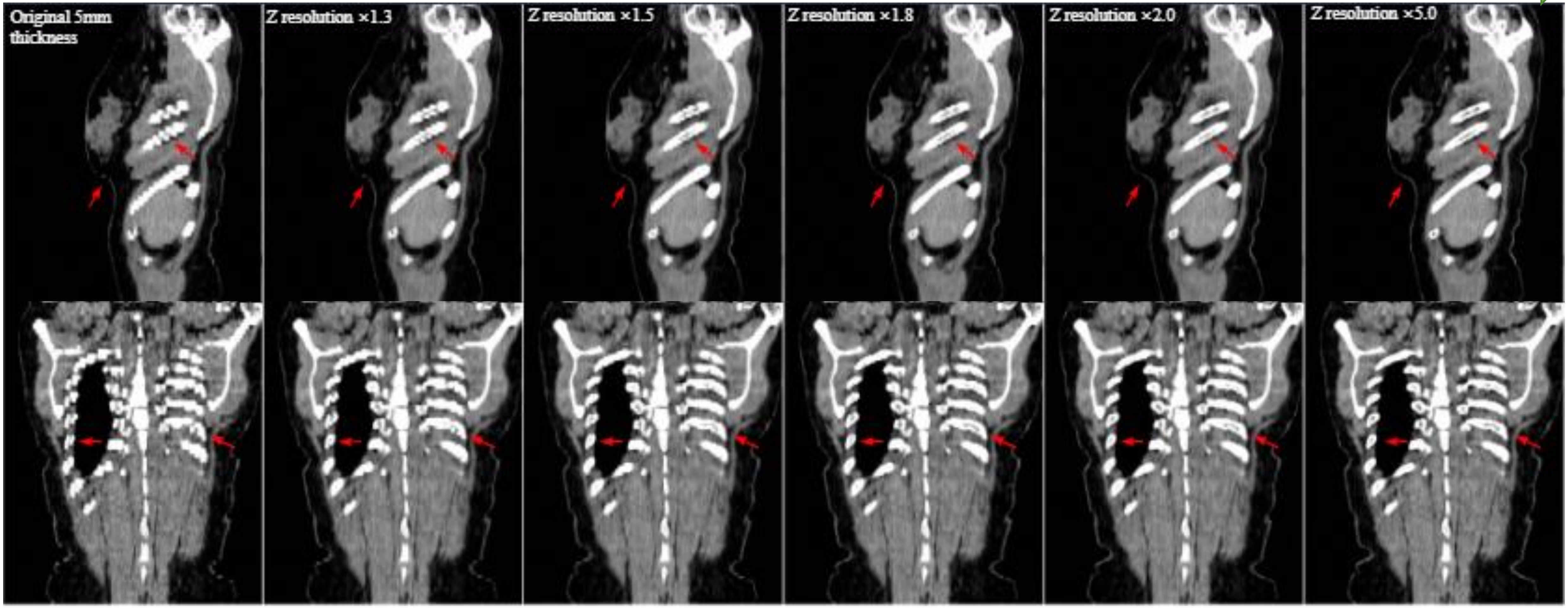
Comparison of segmentation across different methods with regard to the segmentation on the original HR data

'_L' and '_T' represent the liver and tumor respectively

Scale	Method	DSC_L↑	NSD_L↑	DSC_T↑	NSD_T↑
x2	Cubic	0.9868	0.9568	**0.9069**	**0.8632**
	Trilinear	0.9892	0.9622	**0.9069**	0.8575
	TVSRN [51]	0.9886	0.9514	0.8552	0.8065
	ArSSR [47]	0.9856	0.9459	0.8516	0.7952
	SAINT [31]	0.9884	0.9540	0.8476	0.7904
	CycleINR	**0.9911**	**0.9648**	0.8730	0.8226
x3	Cubic	0.9625	0.7862	0.7621	0.6891
	Trilinear	0.9783	0.9037	0.7858	0.7083
	TVSRN [51]	0.9783	0.9105	0.7672	0.6892
	ArSSR [47]	0.9777	0.9056	0.7811	0.6988
	SAINT [31]	0.9820	0.9250	0.8042	0.7338
	CycleINR	**0.9832**	**0.9294**	**0.8087**	**0.7381**
x5	Cubic	0.8678	0.4475	0.5644	0.4386
	Trilinear	0.9487	0.6873	0.6755	0.5385
	TVSRN [51]	0.9481	0.8228	0.6440	0.5431
	ArSSR [47]	0.9589	0.7907	0.6582	0.5297
	SAINT [31]	0.9619	0.8561	0.6006	0.5050
	CycleINR	**0.9620**	**0.8566**	**0.6881**	**0.5640**

Arbitrary-scale SR Results

Super-resolution ratio getting larger



Conclusion

- Introduced cycle-consistent loss mitigates over-smoothing
- Enables efficient high-resolution image generation without specific training for super-resolution ratios
- Offer enhanced visual quality, and robust downstream analysis