## Appendix 1

### *Training details*

Our implementation is based on the Pytorch 1.8 library (27). Pytorch is an open-source deep-learning library that supports the use of GPUs. It dramatically accelerates the iteration of the training process.

We simulated fully sampled K-space data from the input image by trajectory. Afterward, we used 20% of the data as the test data set. We also used 10-fold cross-validation on the remaining data set. After implementing the deep-residual U-net, we applied the He Initializer (28) to initialize its weight. The weight decay rate was set at 0.0001. We also fine-tuned the hyperparameters to achieve better performance. In this study, we used the Adam optimizer, with an initial learning rate of 0.0003. The batch size and the momentum were set to 18 and 0.9, respectively. A dynamic early stopping method (19) was used in this work to address the issue of overfitting. The initial number of epochs was set to 10,000. The early stopping method automatically stopped the training process when the metrics of the validation data set did not decrease further (or the range of decreases did not reach the threshold). First, the early stopping method required an evaluation metric and an early stopping interval. The parameters that had the best performance compared to the parameters of the former epochs were saved. If the network did not improve after the number of early stopping intervals passed, it stopped automatically.

Training took about four minutes per epoch (in both the training and validation processes) for the deep-residual network on an NVIDIA RTX 8000 GPU. We also ran our code on the same GPU at the test time to exploit its computational speed. Predictions on the whole testing data took, on average, 30 minutes.

The actual training process is shown in *Figure S1*.

### *Visual analysis*

*Figures S2-S4* depict the comparison of the Dice coefficients and Hausdorff distances between the proposed and traditional architecture. As the figures show, for the different under-sampling rates, the proposed method usually achieved a higher Dice coefficient and lower Hausdorff distance than the traditional method. In relation to the view level, the WT dice measurement based on our architecture did not change greatly at a different rate for the remaining K-space data. However, the line chart of the traditional architecture appeared to significantly increase as the information in the images increased. Thus, our architecture was much more stable, as it was not greatly affected by the amount of K-space data, and it showed excellent adaption to the under-sampling situation, which is widely used at present.

Nevertheless, the line representing our architecture is generally positioned above those of the traditional architectures in the line charts. It also clearly demonstrates the advancements of the novel method. As for the Hausdorff distance at different statuses, our approach also had a stable horizontal line compared with the traditional methods, which demonstrates the superiority of our method in generating the detailed borders of the lesions.

## References


27. Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32 (NeurIPS 2019). 2019.
28. He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. 2015:1026-34.


 https://dx.doi.org/10.21037/qims-23-946

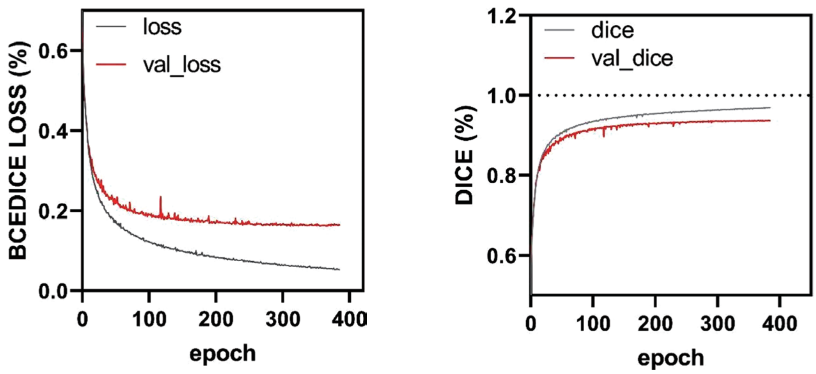



**Figure S1** The training process on fully sampled K-space data. The gray line shows how training loss or training dice changed over the epoch. The red line shows how validation loss or validation dice changed over the epoch.

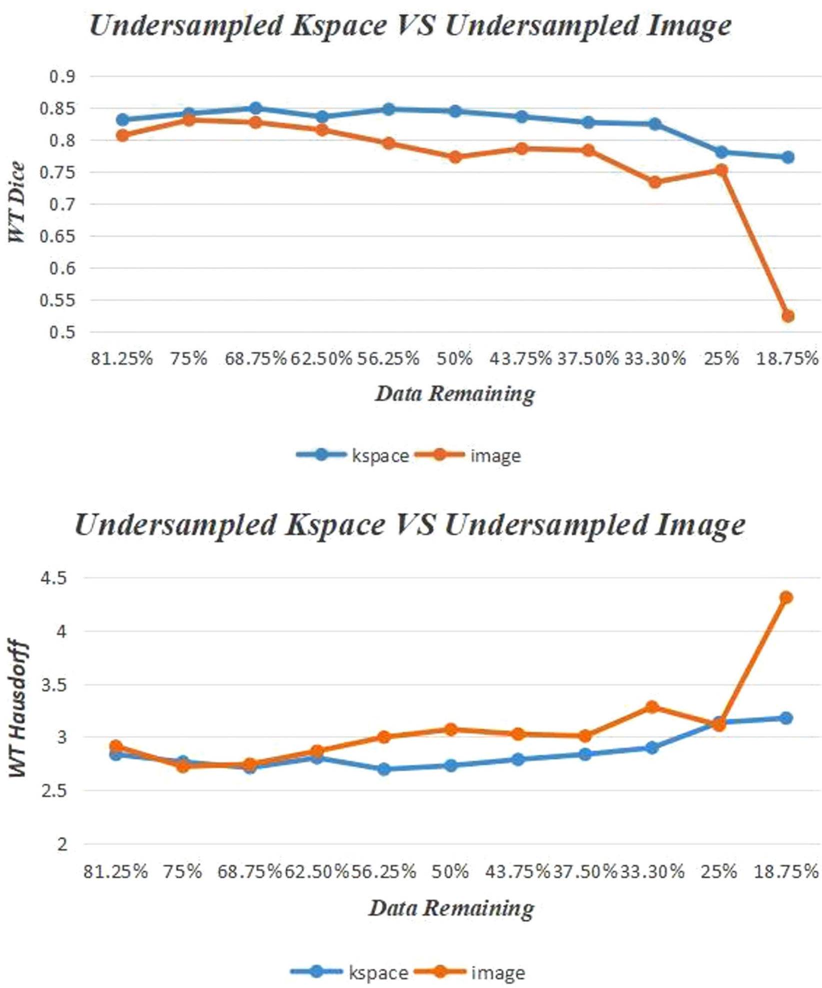
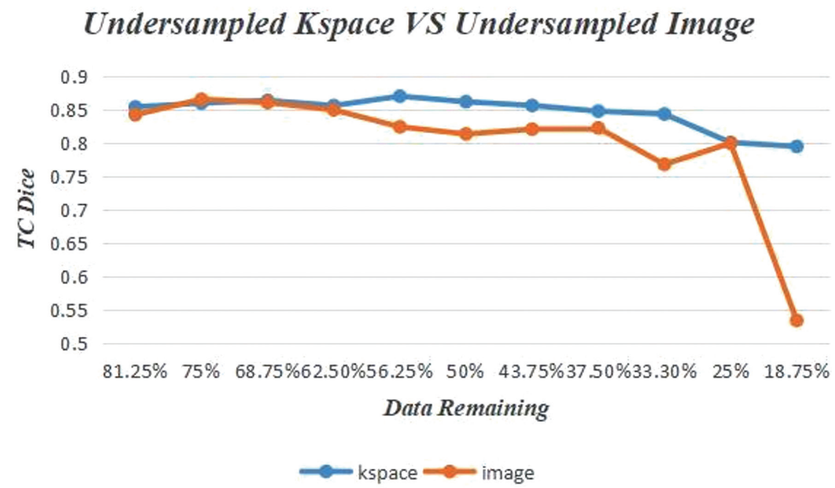



**Figure S2** Comparison of the segmentation of the WT based on under-sampled K-space data with different under-sampling rates and the corresponding image data, respectively. The orange line represents the quantitative results based on the K-space data segmentation. The blue line represents the quantitative results based on image data segmentation. WT, whole tumor.

 https://dx.doi.org/10.21037/qims-23-946



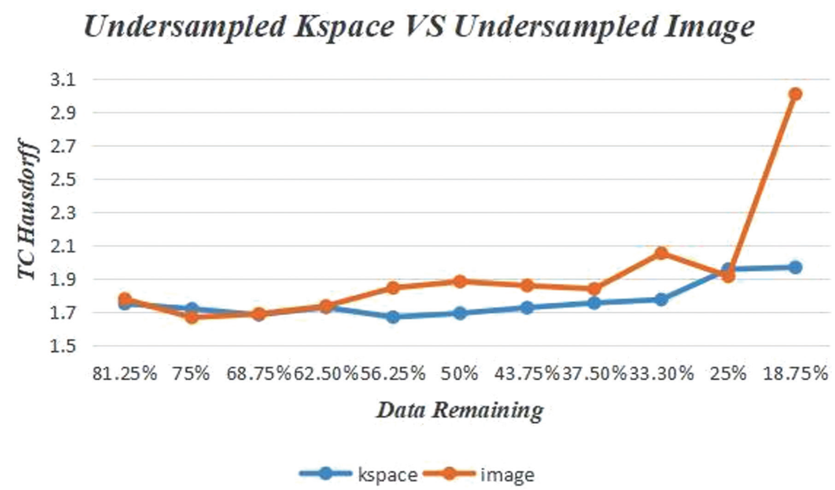




**Figure S3** Comparison of the segmentation of the TC based on under-sampled K-space data with different under-sampling rates and the corresponding image data, respectively. TC, tumor core.

 https://dx.doi.org/10.21037/qims-23-946

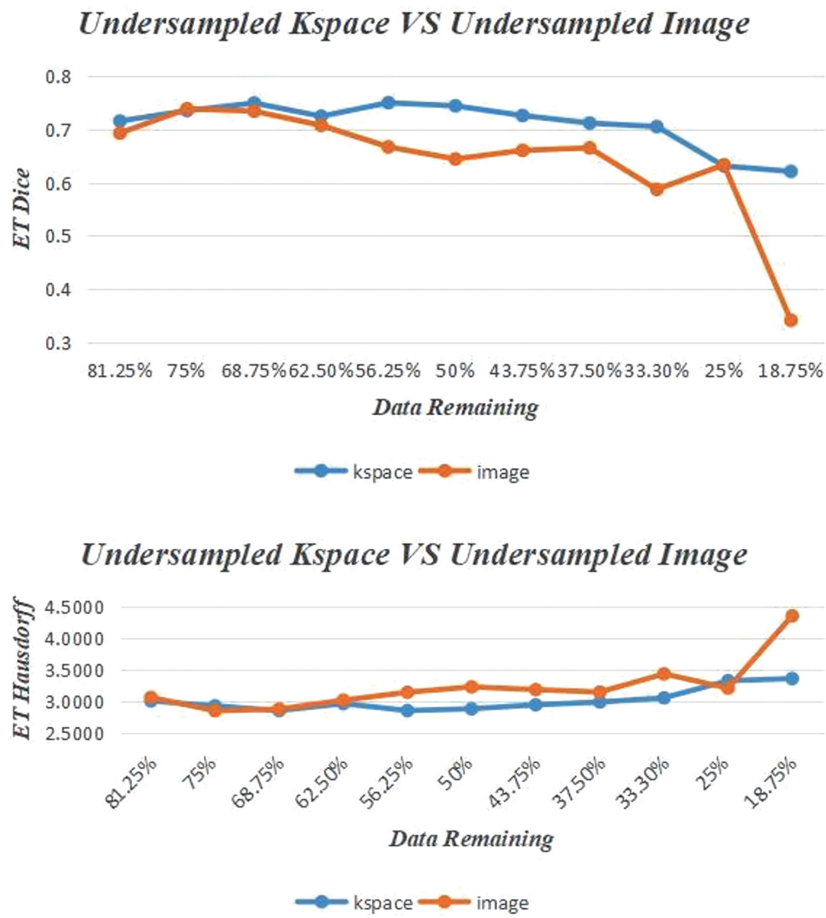



**Figure S4** Comparison of the segmentation of the ET part based on under-sampled K-space data with different under-sampling rates and the corresponding image data, respectively. ET, enhanced tumor.


https://dx.doi.org/10.21037/qims-23-946