In this supplementary mateirial, we start by introducing implement details on the proposed K-Space Transformer and baseline methods included in our experiments. Then, we demonstrate ablation study on the influence of a deeper K-Space Transformer and how the hybrid learning works in the HR Decoder. Finally, we supplement more qualitative comparison results on settings that are not included in Section 4.2 .

# 6 Implement Details

**K-Space Transformer** is implemented with 4 Encoder layers, 4 Low-Resolution Decoder layers and 6 High-Resolution Decoder layers. Each encoder layer is equipped with a four-head self-attention layer and a feed forward network; each LR decoder layer is equipped with a four-head cross-attention layer, a four-head self-attention layer and a feed forward network; and each HR decoder layer is equipped with a four-head cross-attention, a feed forward network and a refinement module. The backone of refinement module consists of 5 convolution layers with leaky ReLU activations between. The first four layers has 64 filters of 3×3 kernel; while the last layer has 2 filters of 3×3 kernel. Necessary prediction, embedding and fft/ifft layers are inserted before and after the convolutions for transformation between k-space and image domain. By default, the embedding layers, prediction layers and feed forward networks mentioned in this paper are two-level MLPs with ReLu activations between. Following previous works [4, 7, 8, 21, 25], we treat the input and output complex MR signals as double channels, *i.e.*, real and imaginary. Features vectores and 2D positional encodings mentioned in Section 3 are set with the same dimensionality $d = 256$.

We adopt pixel-wise loss between reconstruction $\mathcal{I}$ and groundtruth $\mathcal{I}_{gt}$ in image domain, and apply deep supervision [13] on each layer of LR and HR decoders : $\mathcal{L} = \sum_i \|\mathcal{I}_i - \mathcal{I}_{gt}\|_2$, where $i$ indicates the layer depth. We train K-Space Transformer with AdamW optimizer and cosine annealing schedule, setting the initial learning rate as $5 \times 10^{-4}$. Inspired by [29], we adopt Pre-LN architecture to save warm-up stage. As the HR reconstruction depends on the LR decoder output (refer to Section 3.3), we find it helpful to stop the gradients from HR decoder outputs in early epochs, *i.e.*, only update the LR decoder. Training is conducted on 4 RTX 3090.

**Other Baseline Approaches.** We implement a standard U-Net [20] with data consistency layer before the final output for U-Net and K U-Net, which follows the same architecture as [8]. We train the network with the same objective function and learning rate as our methods but adopt an Adam optimizer. We implement D5C5 [21] according to its official configuration for 2D images reconstruction: a cascade network of 5 CNNs, each equipped with 5 layers and a data consistency layer. We train it with the same hyper-parameters as U-Net. For Deep ADMM [31], SwinMR [10] and OUCR [7], we adopt their official source codes and default hyper-parameters that are publicly available or provided by the authors.

# 7 Influence of Network Depth

We investigate the effect of layer depth in K-Space Transformer by increasing the depth of Encoder, LR Decoder and HR Decoder respectively and evaluate under Gaussian sampling, 5x acceleration. As demonstrated in Figure 6, increasing depth consistently improve the network performance in all three modules. Though we conjecture our model may benefit from

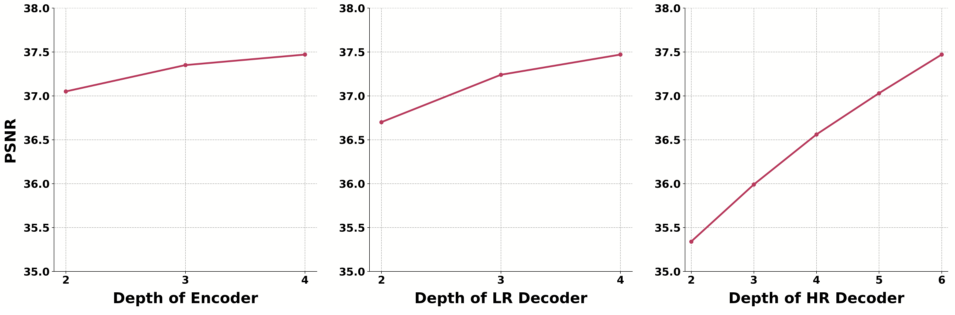


Figure 6: The performance influence of increasing module depth.

even deeper architecture, we take a default setting as 4 Encoder layers, 4 Low-Resolution Decoder layers and 6 High-Resolution Decoder layers to balance the computational cost.

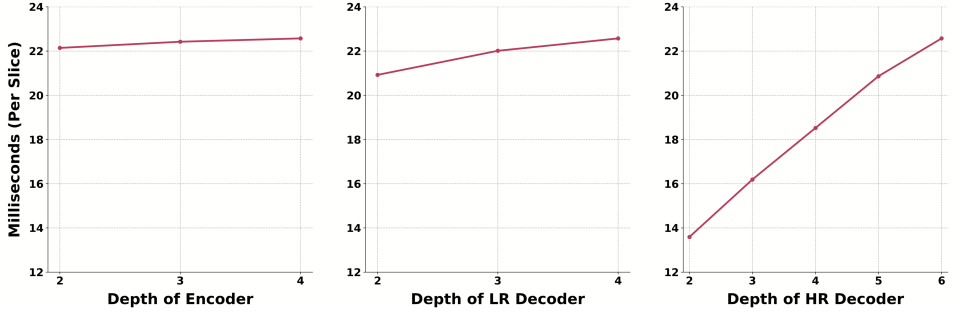


Figure 7: The speed influence of increasing module depth.

**On Reconstruction Speed.** Here, we evaluate the reconstruction time, as expected, deeper networks tend to incur longer processing time, shown in Figure 7. For example, depending on the depth of the HR decoder, the average inference time for each slice could range from 13.6ms to 22.6ms. By comparison, it's significantly faster than Deep ADMM (around 1s) and traditional CS-based methods (up to hundreds of seconds) and comparable to SwinMR (around 18ms), and only slightly slower than the existing CNN-based methods (a few milliseconds). However, we believe the processing speed of our model is sufficient for real-time imaging, thus its remarkable performance improvement on high acceleration settings should be more valuable to improve the patients' scanning experience and reduce medical cost in clinical application.

# 8 Effectiveness of Hybrid Learning

We have shown that image domain refinement does provide complementary information to k-space decoding in Section 4.3 . Here, we further investigate how the alternating process works in the hybrid learning. After evaluating the deep supervised reconstructed results of each HR decoder layer under Gaussian sampling, 5x acceleration (shown in Figure 8),

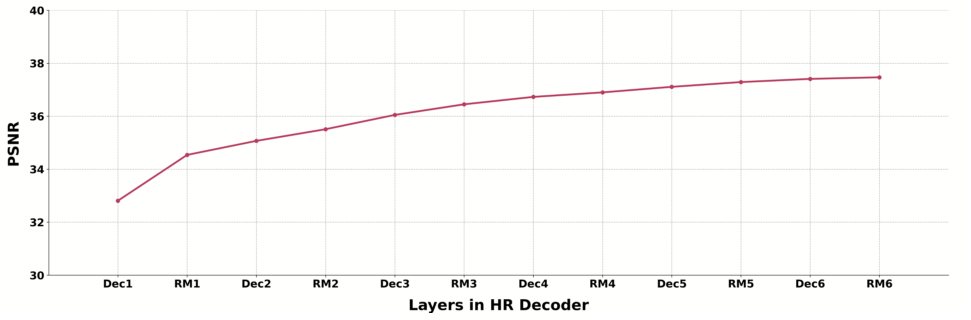


Figure 8: Evaluation on deep supervised reconstruction from each k-space decoding(denoted as Dec) and image refinement module(denoted as RM) in the HR Decoder.

we observe that PSNR increases consistently throughout the alternation between k-space decoding and image domain refinement, and this finding is consistent across all the settings.

# 9 Supplementary Qualitative Comparison with Baselines

We show more qualitative comparisons that are not included in Section 4.2 . As can be seen from Figure 9 and Figure 10, on settings like 2.5x uniform sampling or 10x Gaussian sampling, our method achieve remarkably higher reconstruction quality over other approaches, demonstrating the robustness to more significant aliasing artifacts. While on settings with less structure loss and aliasing artifacts, the top methods perform quite close.

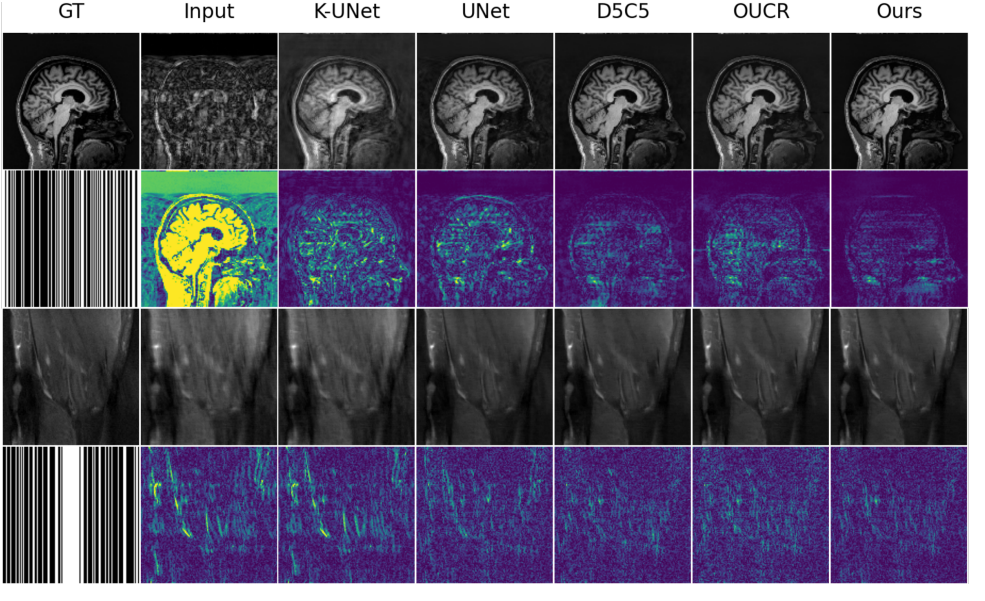


Figure 9: Supplementary qualitative comparison on uniform sampling settings.

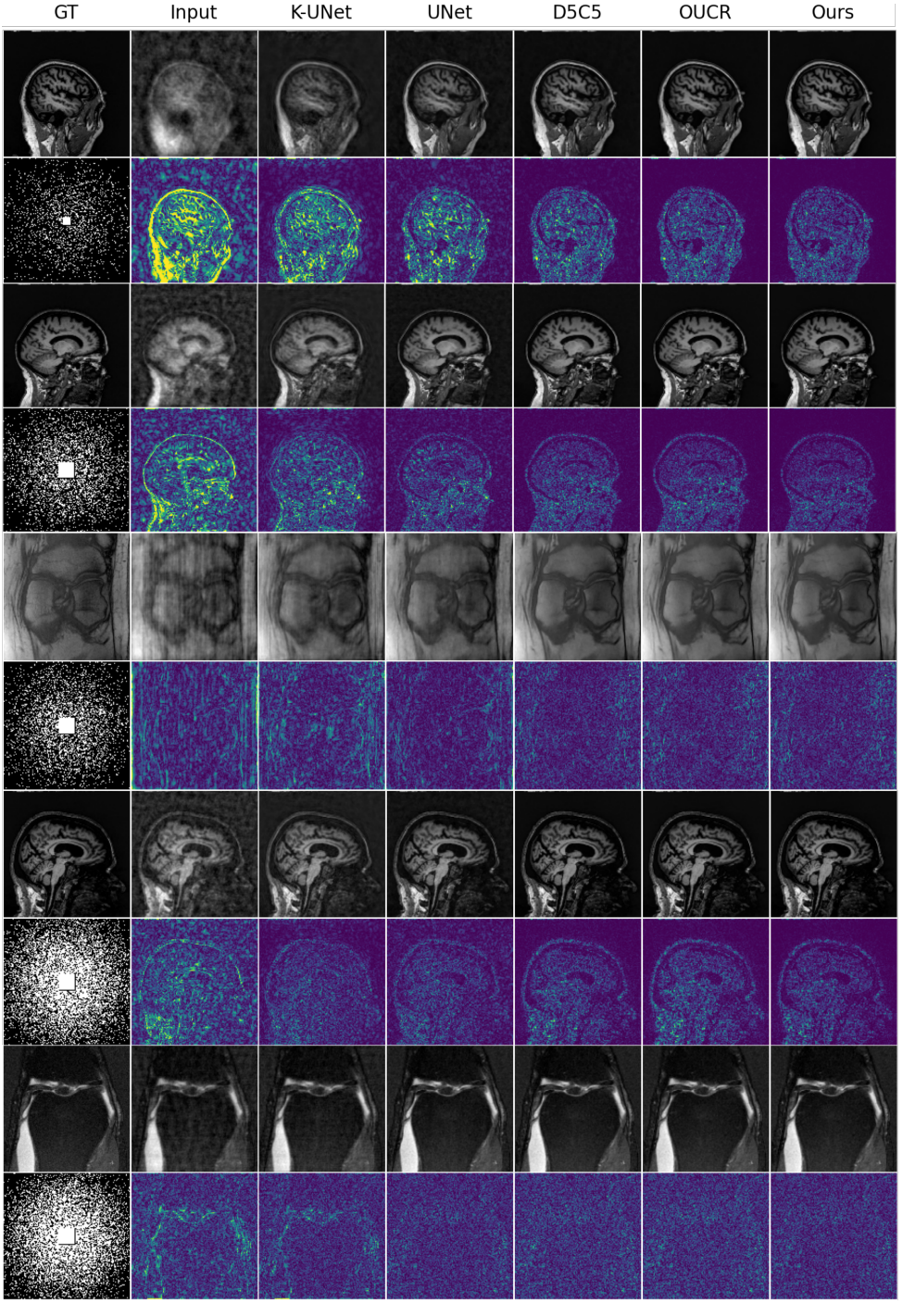


Figure 10: Supplementary qualitative comparison on Gaussian sampling settings.