fastMRI: Region-Aware Image Reconstruction

Luis Tupac
Georgia Institute of Technology
ltupac3@gatech.edu

Igor Kamenetskiy
Georgia Institute of Technology
ikamenetskiy3@gatech.edu

Emma Resmini
Georgia Institute of Technology
eresmini3@gatech.edu

Naveen Vellaturi
Georgia Institute of Technology
nvellaturi3@gatech.edu

Abstract

Magnetic Resonance Imaging, a popular noninvasive diagnostic procedure, is not currently used in some applications due to the duration of the scan. In many of these applications the pathology is highly localized, yet while the data is collected in the frequency domain (k-space) that must be fully sampled, its under-sampling leads to distorted reconstructed images. Increasing the speed of MRI has been approached by training deep learning models (U-Net, DDPM, Cold Diffusion) to reconstruct under-sampled (and hence faster to collect) k-space data. We propose to further improve the quality of the reconstructed images by locating the Region-of-Interest (ROI) and weighting the learning rate in the ROI to focus training of the model on the pathology region. We proved the concept with U-Net model, using both manually chosen ROI and one detected by the Sobel edge-detecting operator, where improvement in the image quality characteristics (SSIM, NMSE, and PSNR) were demonstrated. The same approach was further applied to a Cold Diffusion model.

1. Introduction & Background

Throughout the past several decades, Magnetic Resonance Imaging (MRI) has gained a wide popularity among physicians and patients as a non-invasive diagnostic tool that, in many applications, is capable of providing more comprehensive and accurate diagnostic information than alternative methods, such as Computer Tomography or Ultrasound [1]. Despite multiple advantages of MRI, it has several disadvantages, with long scanning times and high monetary cost being arguably the main ones, that have prevented the use of MRI in some applications. To mitigate these limitations, scientists, engineers and physicians have been working to improve the technical characteristics of the scanners and imagining techniques, *e.g*. by increasing the magnetic field within the scanner from 0.2 to 7T or by introducing multichannel imaging [2]. Although improving the physical characteristics of imaging equipment remains an active area of academic and industrial research, there are still applications where the use of MRI is prohibitively long and expensive. For example, knee MRI may take more than an hour, during which time the patient should be completely still. However, in the case of the knee, the pathology is highly localized and it would suffice to concentrate only on the area of possible pathology. However, this is not possible using conventional methods because during the scan, the data is collected in the frequency domain (so-called "k-space", please refer to [3] for detailed discussion) that should be fully sampled. Its undersampling leads to the image distortion called Shannon-Nyquist aliasing [4], rather than to the image with lower resolution.

The latest advances in the area of Machine Learning in general and Deep Neural Networks in particular brought research aimed at reducing MRI scan times to a new level. For example, advanced progress was achieved in collaboration between Meta and the NYU Center for Biomedical Imaging [2] where the researchers used undersampled MRI data to train deep learning models to reconstruct full-resolution MRI scans. An extended version of this dataset, called fastMRI+ [5], was released by Stanford and includes expert annotations such as bounding boxes for pathology in knee and brain MRI scans.

Recently, Denoising Diffusion Probabilistic Models (DDPM) [6] gained popularity due to their ability to generate and reconstruct images. The idea behind DDPM is that the model is trained to de-noise images that were distorted by adding a random Gaussian noise. This technique was also successfully applied to fast MRI [7].

More recently, the Cold Diffusion technique was introduced [8]. The term "Cold" means that the added noise is not random but deterministic, *e.g*. Blur or Snow. G. Shen, *et al*. [9] trained the diffusion model on fully sam-

pled knee MRI images by applying progressively undersampling mask on the raw k-space data. This technique allowed the authors to achieve superior image quality (measured in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM)) compared to previously used methods (U-Net, W-Net, and E2E-VarNet).

Our contribution was motivated by the fact that, as previously mentioned, in many practical cases (*e.g.* for knee) the pathology is highly localized, therefore the performance of the reconstruction neural network may benefit from focusing on the Region-of-Interest (ROI). This can be achieved by weighting the loss function applied to ROI differently compared to the rest of the image. We first tested our hypothesis on a U-Net model training with "naively" chosen ROI in the center of the image and observed a modest but noticeable improvement in the image quality metrics (PSNR and SSIM). We further applied the loss weighting based on the Sobel edge-detecting operator [10] that allowed to generalize the choice of ROI.

Inspired by the improvement observed when the loss function is ROI-weighted in the U-net model, we applied the same idea to the Cold Diffusion model, where equally-weighted L1 loss was originally used. We believe that focusing on the ROI should improve the Cold Diffusion knee image reconstruction from the under-sampled k-space even further.

We trained our model on the publicly available fastMRI [2] single-coil knee MRI dataset. Each file includes multiple 2D slices of a knee scan, along with metadata. The dataset contains over 1,500 fully sampled knee MRI scans collected at NYU Langone Health on Siemens 3T and 1.5T scanners[11]. It was designed to support research in accelerated MRI reconstruction, aiming to improve scan speed without sacrificing image quality. The data did not require any preprocessing and can be used for model training directly, without additional cleaning or alignment. However, potential limitations include the lack of demographic information (e.g., age, sex, ethnicity), which may obscure biases in patient representation, potentially affecting model generalization across diverse populations.

Radiologists and patients would benefit directly from this. Radiologists will have access to faster and more accurate scan results, and patients will spend less time and money doing this procedure. If our improvements work, they could push this already promising technology even further, making MRIs more practical and accessible to people all over the world.

2. Approach

Our approach consisted of improving the Cold Diffusion framework (applied to the reconstruction of undersampled MRI scans by G. Shen, *et al.* [9]) by employing an ROI-weighted loss function. The proof of concept was first obtained on the fastMRI U-Net model [2] where the weighting of the loss function showed improvement in image quality metrics such as the Peak Signal-to-Noise Ratio (PSNR), Normalized Mean Squared Error (NMSE), and Structural Similarity Index Measure (SSIM).

2.1. ROI-Weighted Loss

In this approach we focused on modifying the U-Net model Meta provided [12], to prioritize the ROIs during image reconstruction instead of treating the entire image the same. This approach involved creating a binary mask over the ROI and used it to modify the loss function by increasing the contribution of pixels within the ROI using equation 1.

$$\text{roi\_loss} = (1 + (\alpha - 1) \cdot \text{roi\_mask}) \cdot \text{base\_loss} \quad (1)$$

Here, α is a new hyperparameter we introduced to control the weight given to the ROI during training. Its effect is defined as:

- $\alpha = 1.0$: No change, the ROI contributes equally to the loss as the rest of the image. **roi_loss** = **base_loss**.
- $\alpha < 1.0$: The ROI is underweighted. Its contribution to the total loss is reduced. **roi_loss** < **base_loss**.
- $\alpha > 1.0$: The ROI is upweighted. It contributes more to the total loss. **roi_loss** > **base_loss**.

We used L1 loss to compute the ***base_loss*** because it is less sensitive to outliers than L2 loss and tends to produce sharper images. This aligned well with our approach, as it helps preserve the overall structure without introducing unnecessary blurring from large error terms.

To test the success of this approach we made two implementations, a naive implementation and an edge-based implementation.

Naive ROI-Weighted Loss

For this, we manually defined the ***roi_mask*** as a fixed square at the center of each MRI image. Since knee scans are typically centered, we assumed this would reasonably approximate the location of the joint across most slices. In this implementation, we treated the knee joint as the only region of interest (ROI) and only upweighted the loss contribution within that region.

To validate the effectiveness of this approach, we trained the U-Net model on 1% of the single coil knee MRI dataset. We used Meta's default training configuration for consistency as shown in Table B.1.

We tested the following combinations of α and square ROI size to assess this naive strategy:

- **Alpha Values**: $\{0.25, 0.75, 1.0, 1.25, 2.0\}$

- **ROI Sizes**: {60, 80, 100, 120}

We anticipated that the square mask wouldn't always align with the joint, especially in the earlier slices, but we prioritized this simple implementation to determine whether the idea of region-weighted loss was even worth pursuing.

One challenge we encountered during this phase was reproducibility. Even after setting `pl.seed_everything()`, we weren't always able to get identical results across runs. We believe this was mainly due to randomness in the `mask_func` used during training, as well as the sampling of different 1% subsets of the dataset. We weren't able to fully resolve this, so as a workaround we modified the training script to first load the dataset and then share the same `DataLoader` across different training runs. This helped ensure that models were trained on the same data slices. Due to time and resource constraints on the PACE cluster, we re-trained the $\alpha = 1.0$ baseline model alongside each experimental run to maintain a consistent baseline for comparison.

Edge-Based ROI-Weighted Loss

If the modified loss function showed good results, then we will implement a more robust method using Sobel filtering to automatically detect edges in the MRI slices. The Sobel edge detection method applies two 3x3 convolution kernels to compute gradients in the horizontal and vertical directions. These gradients are combined to calculate the gradient magnitude, which highlights edges in the image. A threshold is applied to the normalized gradient magnitude to create a binary edge map, where pixels above the threshold are marked as edges. To form a broader ROI, the edge map is dilated using a square kernel, expanding the region around detected edges to include surrounding anatomical structures critical for reconstruction. The resulting ROI mask is a binary tensor with 1s in the ROI and 0s elsewhere, applied to the L1 loss using Equation 1 with the hyperparameter α controlling the weighting. This will generate more accurate ROI masks around the actual knee structures instead of relying on a fixed square.

2.2. Cold Diffusion

Conventionally, the diffusion models use a Markov chain to convert the noise distribution to the data distribution. It has the form of $p_\theta(\mathrm{x}_0) := \int p_\theta(\mathrm{x}_{0:T})\, d\mathrm{x}_{1:T}$ where $\mathrm{x}_1, \ldots, \mathrm{x}_T$ are latent variables of the same dimensionality as the data distribution $\mathrm{x}_0 \sim q(\mathrm{x}_0)$. The reverse process $p_\theta(\mathrm{x}_{0:T})$ is a joint distribution and is defined as a Markov chain with learned Gaussian transitions starting with $p_\theta(\mathrm{x}_T) = \mathcal{N}(\mathrm{x}_T; 0, \mathbf{I})$:

$$
\begin{aligned}
p_\theta(\mathrm{x}_{0:T}) &:= p_\theta(\mathrm{x}_T) \prod_{t=1}^{T} p_\theta(\mathrm{x}_{t-1} | p_\theta(\mathrm{x}_t)), \\
p_\theta(\mathrm{x}_{t-1} | \mathrm{x}_t) &:= N\left(\mathrm{x}_{t-1}; \mu_\theta(\mathrm{x}_t, t), \sigma_t^2 \mathbf{I}\right)
\end{aligned} \tag{2}
$$

The forward or diffusion process is the approximate posterior $q(\mathrm{x}_{1:T}|\mathrm{x}_0)$, which is fixed to a Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$:

$$
\begin{aligned}
q(\mathrm{x}_{1:T}|\mathrm{x}_0) &:= \prod_{t=1}^{T} q(\mathrm{x}_t — \mathrm{x}_{t-1}), \\
q(\mathrm{x}_t — \mathrm{x}_{t-1}) &:= N\left(\mathrm{x}_t; \sqrt{1-\beta_t}\mathrm{x}_{t-1}, \beta_t \mathbf{I}\right)
\end{aligned} \tag{3}
$$

In practice, the forward process is achieved by gradually adding Gaussian noise following the variance schedule. This process does not encounter learnable parameters. The reverse process, on the other hand, is implemented with a learnable deep neural network.

A. Bansal *et al*. [8] showed that it is possible to train the de-noising model in the case of the deterministic (or "called") noise. We followed the approach of G. Shen, *et al*. [9] who used Cold Diffusion for image degradation and restoration. Given an image x_0, consider a degradation operator D with severity t; then the degraded $\mathrm{x}_t = D(\mathrm{x}_0, t)$ should vary continuously in t and the degradation should satisfy $D(\mathrm{x}_0, 0) = \mathrm{x}_0$. To reverse this process and generate an image, the restoration operator R approximately inverts D and has the property of $R(\mathrm{x}_t, t) \approx \mathrm{x}_0$. The restoration operator R is implemented via a deep neural network in practice and parameterized by θ. This network can then be trained via the minimization problem:

$$
\min_\theta \mathbb{E} \left\| R_\theta(D(\mathrm{x}, t), t) - \mathrm{x} \right\| \tag{4}
$$

Once the degradation is chosen and the network is trained properly to perform the restoration, the network can be used to sample images from the degraded image.

In Cold Diffusion, x_t is sampled via intermediate variables:

$$
\mathrm{x}_{t-1} = \mathrm{x}_t - D(\widehat{\mathrm{x}}_0, t) + D(\widehat{\mathrm{x}}_0, t-1) \tag{5}
$$

This sampling strategy is beneficial especially when the higher-order terms in the Taylor expansion of the degradation $D(\mathrm{x}, t)$ are non-negligible. It enables more reliable reconstructions for Cold Diffusion models with a smaller total step number T and a variety of image restoration operations such as deblurring, inpainting, super-resolution, snowification, etc. [8]

Application of Cold Diffusion to MRI Image Reconstruction

MRI scanner acquires measurements in the frequency domain (k-space) that relates to the image via Fourier transformation:

$$k = \mathcal{F}(\mathrm{x}) + \epsilon \tag{6}$$

where $\mathcal{F}$ is the Fourier transformation operator and ϵ is the measurement noise. The MRI acquisition speed is limited by the amount of k-space data to obtain. This acquisition process can be accelerated by down-sampling only a portion of the k-space data. However, this may lead to aliasing artifacts after applying the inverse Fourier transformation to reconstruct the image [3].
In order to mitigate such image degradation, the Cold Diffusion model is trained to reverse such degradation:

$$\widehat{\mathrm{x}}_0 = R(\mathrm{x}_t, t) = R\left(\mathcal{F}^{-1}(M_t \circ \mathrm{k}), t\right) \tag{7}$$

where $\mathcal{F}^{-1}$ is inverse Fourier transform, M_t is the sampling mask, and $\circ$ indicates Hadamard product. Then, equation 5 is used to predict the fully sampled reconstruction $\widehat{\mathrm{x}}_0$ [9].

Our main challenge in implementing the Cold Diffusion model was computational resources. It took the authors of the original paper [9] eight days of continuous training with a powerful GPU - 1 second per step - to fully train the model, while training on the PACE cluster was taking at least twice as long per step, even with two GPUs. Therefore, we reduced training to 35,000 steps, in contrast to 700,000 steps in the original paper, and only used 10% of the training data. We also had a similar challenge of reproducibility, as outlined in Section 2.1, and similar steps were taken with the `DataLoader` to ensure uniform data slices during training, and testing metrics were averaged out over several runs.

2.3. Evaluation

To evaluate model performance, we compared the following metrics:

- **SSIM (Structural Similarity Index)** — We used this to evaluate structural consistency in reconstructed MRI images.
- **NMSE (Normalized Mean Squared Error)** — We used this to compare pixel errors between the original and reconstructed MRI images.
- **PSNR (Peak Signal-to-Noise Ratio)** — We used this to comparing the noise introduced during reconstruction versus the strength of the original signal.

We expected that upweighting the loss in the ROI would lead to improvements across all three metrics by reducing noise in the joint area, preserving key anatomical features, and balancing global accuracy with localized focus.

3. Frameworks and Codebases Used

We used PyTorch and PyTorch Lightning as our deep learning frameworks. Both Meta's fastMRI repository [12] and the Cold Diffusion implementation repository [13] are built on these frameworks, which made it easier to work with.

The fastMRI repository provided a well-organized, production-ready codebase with clear abstractions for deep learning models, example scripts for the modules, custom data loaders, and a wide variety of parameters. This gave us a great starting point to build from once we were able to get through the tricky set up. The code was written in an older version of PyTorch and PyTorch Lightning, which created a lot of debugging and dependency issues. Resolving these version mismatches was one of the early challenges of the project, but once resolved, extending the codebase became significantly easier. We used and modified the U-Net model module heavily for our experiments with ROI-weighted loss.

The Cold Diffusion codebase was less structured than fastMRI but still useful. It used more recent versions of PyTorch and PyTorch Lightning making the setup simpler. While this conflicted with fastMRI's older dependency stack, the code helped us understand how Cold Diffusion works in practice. We followed their Cold Diffusion logic closely when adapting it into our own fastMRI-based training pipeline.

4. Experiments and Results

4.1. Naive ROI-Weighted Loss

We compared each alpha and ROI configuration against the baseline loss using SSIM, NMSE, and PSNR. Table 1 shows the top 5 alpha and ROI size combinations that maximized SSIM and PSNR while minimizing NMSE.

α	ROI Size	SSIM	NMSE	PSNR
1.25	100	**0.7135**	**0.0366**	**31.45**
1.00	-	0.7130	0.0366	31.44
0.75	100	0.7125	0.0366	31.45
0.75	80	0.7116	0.0371	31.34
0.25	100	0.7114	0.0369	31.39

Table 1. Top configurations sorted by SSIM and NMSE

These results suggest that $\alpha = 1.25$ with roi_size $= 100$ slightly outperforms the baseline ($\alpha = 1.0$) model. Since these results were promising, we retrained the baseline model with parameters more closely to the best parameters defined by Meta. Due to the resource constraints we experienced, we modified the parameters as shown in Table B.2.

While the differences are subtle, the upweighted loss with $\alpha = 1.25$ outperformed the base model in all metrics as shown in Table 2. Figure C.1 shows a comparison be-

α	ROI Size	SSIM	NMSE	PSNR
1.25	100	**0.723**	**0.034**	**31.870**
1.00	–	0.721	0.035	31.843

Table 2. Comparison of best-tuned ROI-weighted loss vs. baseline

tween reconstruction in the ground truth vs baseline model vs ROI-weighted output.

We ran both the tuned models on unseen data to test their generalization. The ROI-weighted model came out slightly ahead with a loss of 0.2918 compared to the baseline model with a loss of 0.2923. Focusing just on the ROI region, the gap widened a bit more with 0.3300 for the baseline model vs. 0.3292 for the ROI-weighted model. This further supports our hypothesis that giving more importance to the ROI during training helps the model generalize better, especially in localized regions.

Despite being a naive implementation, this approach produced measurable improvements in the key metrics SSIM, NMSE, and PSNR while generalizing better on unseen data than the baseline model. These results encouraged us to develop a more sophisticated variant using Sobel filtering for dynamic ROI detection.

4.2. Edge-Based ROI-Weighted Loss

To enhance the naive ROI approach, we implemented an edge-based ROI loss using Sobel filtering to dynamically detect knee structures. We visualized edge maps to select a Sobel threshold of 0.4, which effectively covered the knee region (see Figure C.3). We conducted a grid search over hyperparameters $\alpha \in \{0.25, 0.75, 1.0, 1.25, 1.5\}$ and Sobel threshold $\in \{0.3, 0.4, 0.5\}$. Dilation size was fixed at 9. Due to randomness in data loading, performance varied slightly across runs. Table 3 shows all configurations sorted by SSIM.

The best configuration ($\alpha = 1.5$, Sobel threshold=0.3) appears to outperform the naive ROI approach and baseline approach seen in Table 1, and has the best SSIM, NMSE, and PSNR metrics out of all the models in Table 3. Configurations with $\alpha < 1.0$ (e.g., $\alpha = 0.75$, Sobel=0.5, SSIM=0.7150) also performed well, likely because global SSIM also emphasizes non-ROI regions, which dominate the image. SSIM may undervalue knee-specific improvements. However, some configurations also underperformed - likely due to suboptimal edge detection or data variability. Figure C.2 compares the edge-based ROI mask against the naive approach, showing improved knee coverage.

To further evaluate the best-tuned Sobel-based model ($\alpha = 1.5$, Sobel threshold=0.3, dilation size=9), we compared it against the baseline ($\alpha = 1.0$) with parameters closer to the best defined by Meta. Table 4 summarizes the results. Figure C.4 is a visual comparison between the outputs of these two models. Figure C.5 is a visual comparison of the most improved slice with the Sobel model.

α	Sobel Thr.	Dil. Size	SSIM	NMSE	PSNR
1.50	0.3	9	**0.7164**	**0.0357**	**31.58**
0.75	0.5	9	0.7150	0.0362	31.51
0.75	0.3	9	0.7143	0.0365	31.46
0.25	0.3	9	0.7141	0.0367	31.44
0.25	0.5	9	0.7137	0.0364	31.48
1.50	0.5	9	0.7125	0.0364	31.46
1.00	0.3	9	0.7119	0.0362	31.50
0.25	0.4	9	0.7112	0.0369	31.39
1.25	0.4	9	0.7111	0.0368	31.38
1.25	0.5	9	0.7100	0.0366	31.40
1.50	0.4	9	0.7099	0.0369	31.41
1.25	0.3	9	0.7042	0.0383	31.23
1.00	0.4	9	0.7042	0.0372	31.29
1.00	0.5	9	0.6999	0.0396	30.85
0.75	0.4	9	0.6680	0.0397	30.83

Table 3. Edge-based ROI configurations sorted by SSIM. Randomness in data loading may contribute to performance variations.

The metrics suggest that we have not been able to make an improvement with this method. The visualizations indicate that the Sobel model's outputs better match the ground truth in the knee portion but not the background noise, possibly explaining the suboptimal metrics.

α	Sobel Thr.	Dil. Size	SSIM	NMSE	PSNR
1.0	–	–	**0.7217**	**0.0344**	**31.8516**
1.5	0.3	9	0.7213	0.0345	31.8437

Table 4. Comparison of best-tuned ROI-weighted loss vs. baseline

4.3. Cold Diffusion

The Cold Diffusion model was trained with the parameters listed in Table B.3 by applying a progressive undersampling mask to the fully sampled k-space. Since training the Cold Diffusion model is very computationally expensive, we were unable to assess the effect of the ROI-weighted loss with a fully-trained model with our given resources, and were also unable to optimize the weights of the loss function and the parameters of the Sobel algorithm. Instead, we trained using the optimized hyperparameters for the U-Net model (see Table B.4 and Section 4.2).

The image reconstructions produced by the undertrained model are not ideal and do not allow to draw a conclusion on the advantage of using ROI-wighted loss, as can be seen in Figure C.6 (for uniform L1 loss) and Figure C.7 (for ROI-weighted L1 loss). However, comparison of the evaluation metrics (see Section 2.3) obtained with the Cold Diffusion model trained with the equally weighted L1 loss function to those obtained with the model trained with the ROI-weighted loss function presented in Table 5 demonstrate modest benefit from using the ROI-weighted loss. The metrics shows that the ROI-weighted loss helps the

model deliver more refined reconstructions in ROIs, which leads to better image quality as compared to L1 loss or the U-Net model.

Loss Function	SSIM	NMSE	PSNR
ROI-weighted	**0.7970**	**0.0075**	**32.1300**
Equally-weighted L1	0.7653	0.0078	31.8660

Table 5. Comparison of metrics for Cold Diffusion models with equally-weighted L1 loss and ROI-weighted loss functions. The metrics are averaged over 20 measurements to smooth the randomness in data loading.

5. Conclusion

We showed that training the U-Net model with the ROI-weighted loss function produces more accurate image reconstruction measured in terms of PSNR, SIMM and NMSE. We first tried manually selected ROI (at the center of the image) and then generalized the approach using the Sobel edge detection framework, which suggested further improvement in image quality. We ran multiple experiments to find the best loss function weight and the Sobel algorithm parameters.

We then applied the ROI weighting with the parameters optimized on the U-Net model to the Cold Diffusion model. Due to the computational intensity of the Cold Diffusion model training, we had to reduce the training steps and use a subset of the training data. Moreover, even the image reconstruction was time consuming. Consequently, we could not prove that the ROI-focused loss function produces superior under-sampled image reconstruction with the fully trained Cold Diffusion model. However, the results observed with the "under-trained" model look promising, as discussed in Section 4.3. We hope to be able to continue working towards a fully-trained Cold Diffusion model. However, with our computational resources it may take several months. For the ideal end product, the loss function weights and the Sobel edge-detecting parameters would be calibrated using the Cold Diffusion model itself, rather than using the optimal parameters calibrated on the U-Net model. However, this will likely be prohibitively computationally expensive.

For future work, the Stanford fastMRI+ dataset [5] could be explored, as it provides expert annotated regions of interest (ROIs) such as bounding boxes and labels for 22 different pathology categories. This could enable more precise supervision compared to our current approach, which either uses a centrally defined manual ROI or treats the entire knee as the region of interest.

6. Work Division

Summary of contributions are provided in Table A.1 in Appendix A.

References

[1] National Institute of Biomedical Imaging and Bioengineering. Magnetic resonance imaging (mri). https://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri, Accessed: 2025-04-27. 1

[2] Florian Knoll, Jure Zbontar, Anuroop Sriram, Matthew J. Muckley, Mary Bruno, Aaron Defazio, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael Rabbat, Pascal Vincent, James Pinkerton, Duo Wang, Nafissa Yakubova, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastmri: A publicly available raw k-space and dicom dataset of knee images for accelerated mr image reconstruction using machine learning. *Radiology: Artificial Intelligence*, 2(1):e190007, 2020. PMID: 32076662. 1, 2

[3] David Moratal, Ana Vallés-Luch, Luis Martí-Bonmatí, and Michael Brummer. k-space tutorial: an mri educational tool for a better understanding of k-space. *Biomed Imaging Interv J*, 4(1):e15, Jan 2008. PMID: 21614308. 1, 4

[4] E. T. Whittaker. On the functions which are represented by the expansions of the interpolation theory. *Proceedings of the Royal Society of Edinburgh*, 35:181–194, 1915. 1

[5] Ruiyang Zhao, Burhaneddin Yaman, Yuxin Zhang, Russell Stewart, Austin Dixon, Florian Knoll, Zhengnan Huang, Yvonne W. Lui, Michael S. Hansen, and Matthew P. Lungren. fastmri+: Clinical pathology annotations for knee and brain fully sampled magnetic resonance imaging data. *Scientific Data*, 9(1):152, 2022. 1, 6

[6] P. Abbeel J. Ho, A. Jain. Denoising diffusion probabilistic models, 2020. 1

[7] Y. Wang S. Liu T. Chen H. Zheng D. Liang Y. Zhu C. Cao, Z.-X. Cui. High-frequency space diffusion models for accelerated mri, 2024. 1

[8] H.-M. Chu J. S. Li H. Kazemi F. Huang M. Goldblum J. Geiping T. Goldstein A. Bansal, E. Borgnia. Cold diffusion: Inverting arbitrary image transforms without noise. *NeurIPS 2023 poster*, 2020. 1, 3

[9] Guoyao Shen, Mengyu Li, Chad W. Farris, Stephan Anderson, and Xin Zhang. Learning to reconstruct

accelerated mri through k-space cold diffusion without noise. *Scientific Reports*, 14(1):21877, 2024. 1, 2, 3, 4

[10] Sobel edge detector. https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm,. 2

[11] NYU Langone Health and Meta AI. fastmri dataset, 2018. Accessed: 2025-04-28. 2

[12] Facebook AI Research and NYU. fastMRI GitHub Repository. https://github.com/facebookresearch/fastMRI, 2020. Accessed: 2025-04-19. 2, 4

[13] Guoyao Shen. K-space cold diffusion for accelerated mri. https://github.com/GuoyaoShen/K-SapceColdDIffusion, 2024. Accessed: 2025-04-19. 4

A. Appendix: Work Division

Student Name	Contributed Aspects	Details
Luis Tupac	Project Management, Custom Loss Implementation, Evaluation, Report Writing	Organized team meetings, set project milestones, and facilitated idea development. Designed, wrote validation scripts and tests, defined metrics, and implemented the ROI-weighted loss function across U-Net as shown in sections 2.3 and 4.1. Ran experiments validating the loss function with naive ROI detection. Led contributions to the first draft by structuring the paper, drafting answers to key guiding questions, and formatting the document for clarity.
Naveen Vellaturi	Sobel Edge Detection, Evaluation, Report Writing	Implemented Sobel edge detection. Tested and visualized various Sobel thresholds to optimize ROI mask generation. Trained all edge-based ROI models in Section 4.2 and evaluated their performance using SSIM, NMSE, and PSNR. Contributed to the paper, including sections on experiments, results, and general edits.
Igor Kamenetskiy	Cold Diffusion model optimization and training, Evaluation, Report Writing	Modified publicly available Cold Diffusion code to reduce training time, save/load the checkpoints and output the results metrics. Trained the model, produced and analyzed the results using SSIM, NMSE, and PSNR metrics (Section 4.3). Contributed to the paper, including sections on the approach, experiments, results and conclusion.
Emma Resmini	Cold Diffusion model with ROI-weighted loss training, Evaluation, Report Writing	Trained Cold Diffusion model with ROI-weighted loss function. Additionally, optimized training code to use multiple GPUs when available, which accelerated training time, and adjusted testing and visualization scripts to ensure reproducibility. Contributed to the paper, including sections on the approach, experiments, and results, as well as general edits.

Table A.1. Contributions of team members.

B. Appendix: Training Configurations

Parameter	Value
Loss Function	L1
Optimizer	RMSProp
Epochs	5
Learning Rate	0.001
Channels	32
Weight Decay	0.0

Table B.1: Baseline training configuration used across UNet experiments

Parameter	Value
α	1.25
Channels	128
Dropout Probability	0.0
Input Channels	1
Learning Rate	0.001
LR Gamma	0.1
LR Step Size	40
Number of Pool Layers	4
Output Channels	1
ROI Size	100
Weight Decay	0.0

Table B.2: Parameter set for best-performing ROI configuration

Parameter	Value
Loss Function	L1
Optimizer	Adam
Learning Rate	0.00002
Layers	4
Initial Channels	64
Weight Decay	0.0

Table B.3: Training and reconstruction parameters for the Cold Diffusion model with L1 loss

Parameter	Value
Loss Function	ROI
α	1.5
Sobel thresh.	0.3
Dilation size	9
Optimizer	Adam
Learning Rate	0.00002
Layers	4
Initial Channels	64
Weight Decay	0.0

Parameter	Value

Table B.4: Training and reconstruction parameters for the Cold Diffusion model with ROI-weighted loss

C. Appendix: Visual Results

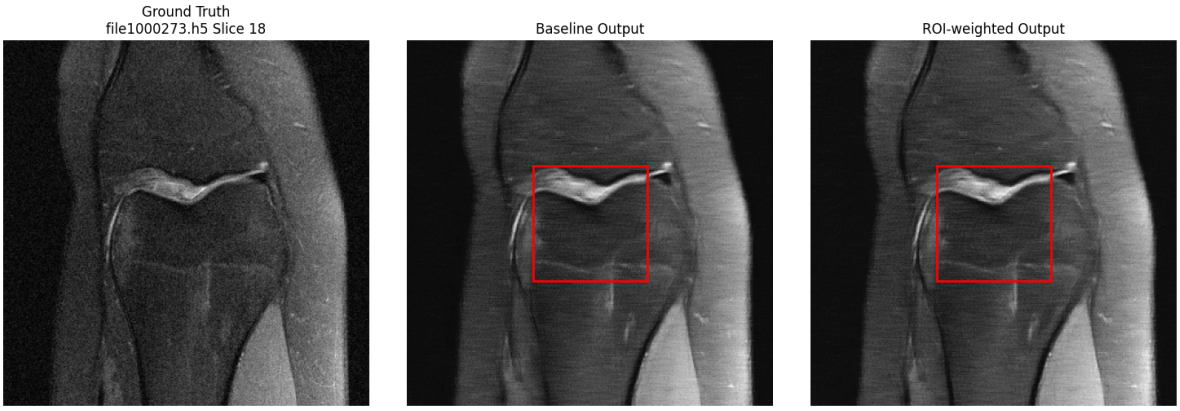

Figure C.1. Visual comparison of ground truth, baseline output, and ROI-weighted output ($\alpha = 1.25$, ROI size=100). The red box shows the region that was upweighted during training. While the reconstructions aren't noticeably different, the ROI-weighted output outperformed the baseline output in SSIM, PNSR, and NMSE.

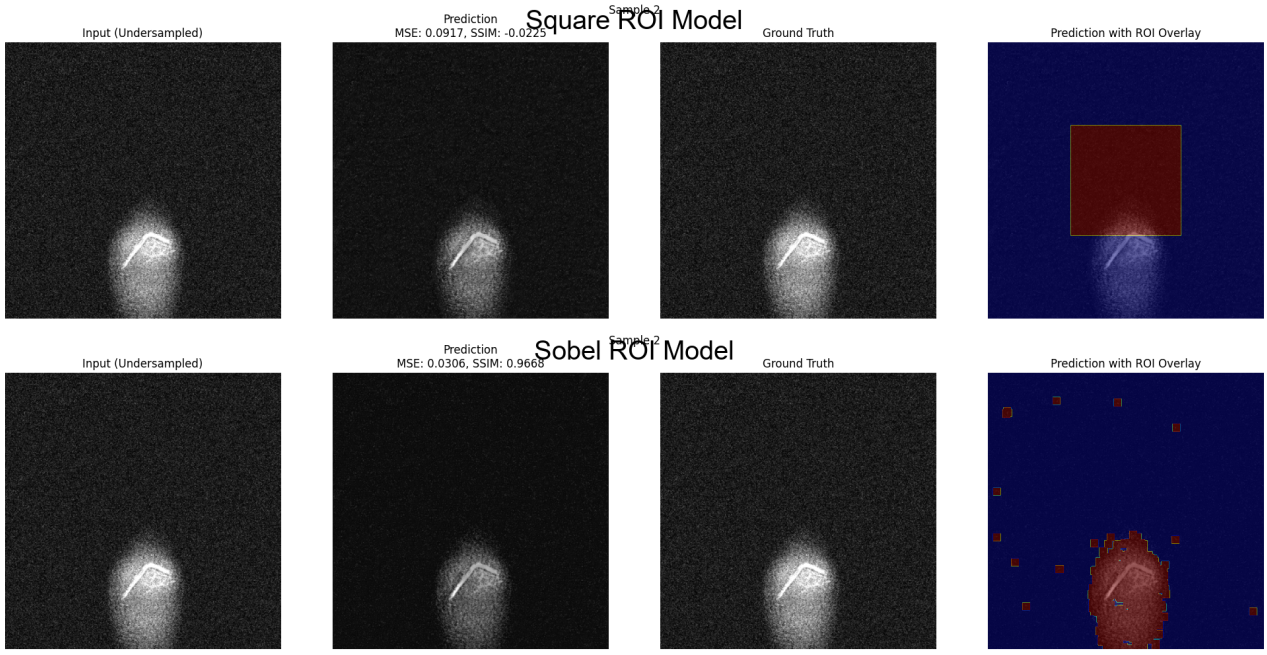

Figure C.2. Visual comparison of the undersampled input, the predictions along with the MSE and SSIM metrics, the ground truth, and the prediction with the ROI mask overlay for the model that used the Naive ROI mask vs the model with the Edge-based ROI Loss.

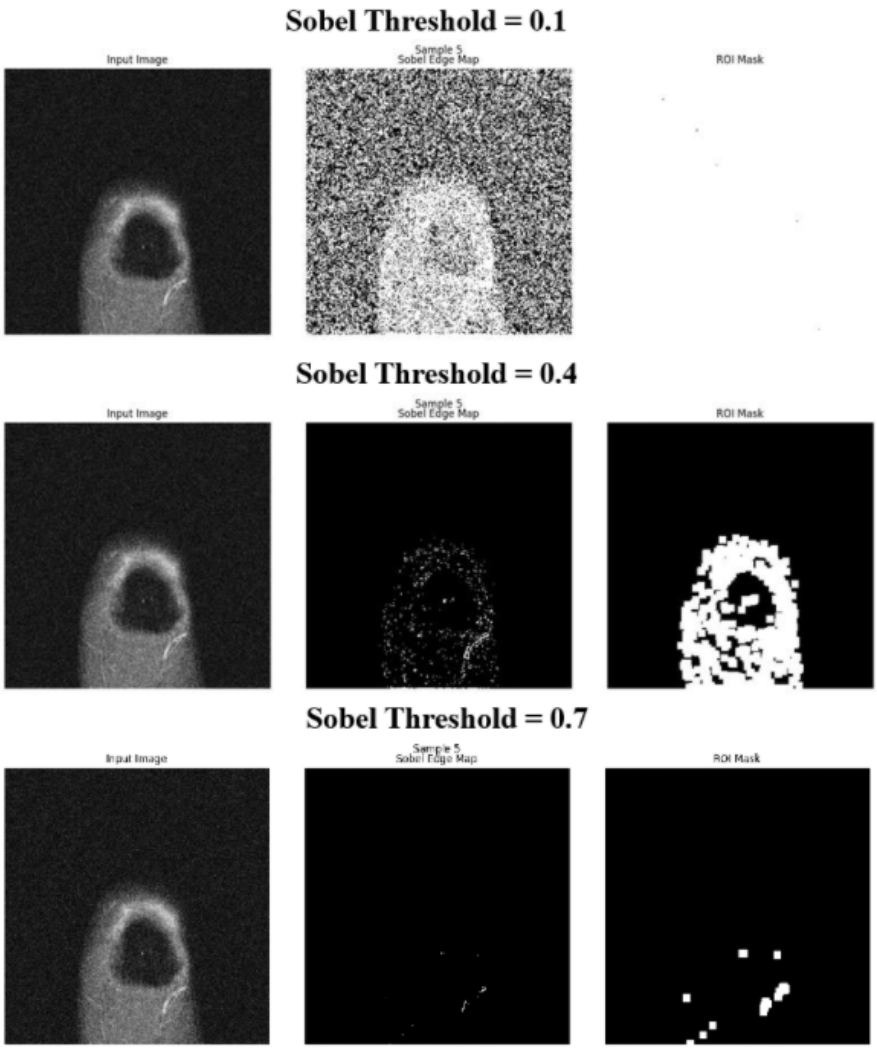

Figure C.3. Visual comparisons of some of the produced ROI masks with different values for the Sobel Threshold. 0.4 produced the mask that covered the greatest portion of the knee.

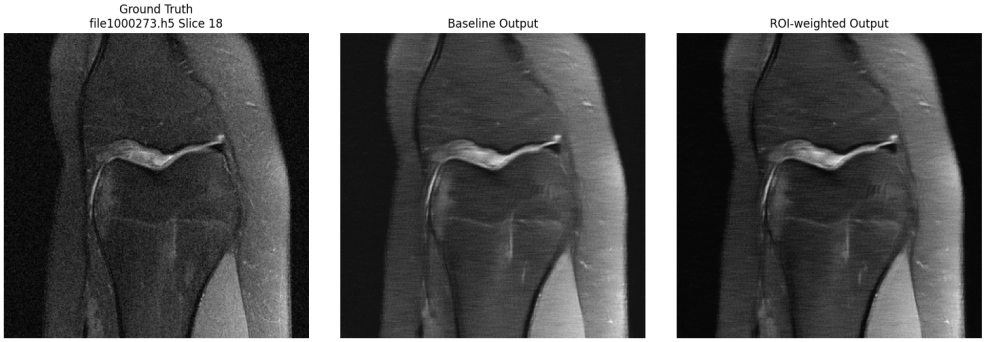

Figure C.4. Visual comparison with the outputs of the baseline and Sobel ROI models mentioned in Table 4

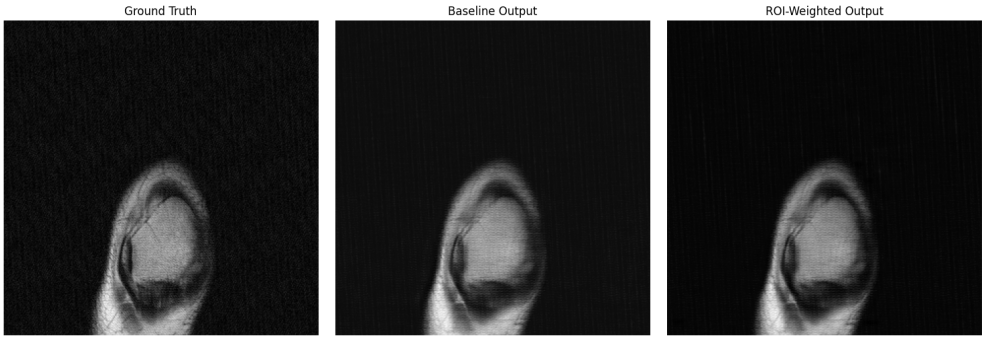

Figure C.5. A visual comparison of the outputs of the baseline and Sobel ROI models mentioned in Table 4 with the most improved slice based on ROI loss.

Figure C.6. Sample result of the image produced with the Cold Diffusion model with uniform L1 loss trained to 35,000 iterations.

Figure C.7. Visual comparison of ground truth, baseline output, and ROI-weighted loss output. The ROI-weighted output produces more granular details than the baseline.