Radiology: Artificial Intelligence

SNRAware:

Improved Deep Learning MRI Denoising with Signal-to-noise Ratio Unit Training and G-factor Map Augmentation

Hui Xue[1]

Sarah M. Hooper[2]

Iain Pierce[4]

Rhodri H. Davies[3,4]

John Stairs[1]

Joseph Naegele[1]

Adrienne E. Campbell-Washburn[2]

Charlotte Manisty[4]

James C. Moon[4]

Thomas A. Treibel[4]

Michael S. Hansen*[1]

Peter Kellman*[1,2]

* M.S.H. and P.K. are co–senior authors.

Author affiliations, funding, and conflicts of interest are listed at the end of this article.

https://doi.org/10.1148/ryai.250227

Purpose: To develop and evaluate a novel deep learning-based MRI denoising method using quantitative noise distribution information obtained during image reconstruction to improve model performance and generalization.

Materials and Methods: This retrospective study included a training set of 2885236 images from 96605 cardiac cine series acquired on 3T MRI scanners from January 2018 to December 2020. 95% of these data were used for training and 5% for validation. The hold-out test set included 3000 cine series, acquired in the same period. Fourteen model architectures were evaluated by instantiating each of the two backbone types with seven transformer and convolution block types. The proposed SNRAware training scheme leveraged MRI reconstruction knowledge to enhance denoising by simulating diverse synthetic datasets and providing quantitative noise distribution information. Internal testing measured performance using peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM), whereas external tests conducted on 1.5T real-time cardiac cine, first-pass cardiac perfusion, brain, and spine MRIs assessed generalization across various sequences, contrasts, anatomies, and field strengths.

Results: SNRAware improved performance on internal tests conducted on a hold-out dataset of 3000 cine series. Models trained without reconstruction knowledge achieved the worst performance metrics. Improvement was architecture-agnostic for both convolution and transformer models; however, transformer models outperformed their convolutional counterparts. Additionally, 3D input tensors showed improved performance over 2D images. The best-performing model from the internal testing generalized well to external samples, delivering 6.5 × and 2.9 × contrast-to-noise ratio improvement for real-time cine and perfusion imaging, respectively. The model trained using only cardiac cine data generalized well to T1 MPRAGE (Magnetization-Prepared Rapid Gradient-Echo) brain 3D and T2 TSE (turbo spin-echo) spine MRIs.

Conclusion: The SNRAware training scheme leveraged data obtained during the image reconstruction process for deep learning-based MRI denoising training, resulting in improved performance and good generalization.

SNRAware, a model-agnostic approach for training MRI denoising models that leverages information from the image reconstruction process, improved performance and enhanced generalization to unseen imaging applications.

Abbreviations

SNR = signal-to-noise ratio, PSNR = peak signal-to-noise ratio, SSIM = structural similarity index measure

Key Points

In this retrospective study including 3000 cine series, the integration of quantitative noise distribution information from signal-to-noise ratio unit reconstruction and g-factor augmentation improved MRI denoising performance.

Two backbone types, HRnet and Unet, were instantiated with seven transformer and convolutional block types to evaluate 14 architectures.

Models were trained on a dataset of 96605 cine series and validated extensively on internal and external data, showing that the proposed method improved performance and generalization.

Deep neural networks outperform conventional filtering methods in restoring low signal-to-noise ratio (SNR) MRIs (1), benefiting applications such as low-field MRI (2,3), diffusion imaging (4), highly accelerated parallel imaging (5), and dynamic cardiac imaging (6). Deep learning-based MRI denoising further enhances diagnostic quality and clinical value.

Deep learning-based denoising can be performed using supervised or self-guided training. Supervised learning requires noisy-clean paired data, which can be difficult to acquire, especially for inherently low SNR applications. To overcome this limitation, self-guided training (7–10) methods that use only noisy images have been proposed. These methods include Noise2Noise (8), which learns from paired noisy images, and Noise2Void (9) and Noise2Fast (10), which require only one noisy image, and learn to predict blind-spot pixels from surrounding pixels or small patches from neighboring patches. These methods are slow at inference and underperform compared with supervised learning (11); therefore, recent advances have adopted diffusion-based generative training, such as denoising diffusion models (4) or score-based diffusion sampling (12), to further enhance image quality.

Existing training approaches often overlook potential performance gains from noise information and are typically trained on limited datasets that include only specific imaging protocols and contrasts (4,8,10,12). The resulting models, therefore, may not be robustly transferrable to other applications, particularly those with intrinsically low SNR, where the initial image quality is poor and denoising is more challenging. However, MRI noise distribution can be derived from the image reconstruction process, enhancing model performance and generalization.

To take advantage of noise distribution data, we proposed SNRAware, a novel MRI denoising training scheme that leverages noise prewhitening (13) with real g-factor maps to create spatially varying noise, noise correlation augmentation to integrate operations such as *k-space* filters, phase oversampling, and image resizing, and SNR unit reconstruction (14) to ensure a unity noise level and aid model learning. We also proposed a g-factor augmentation method to compute g-factor maps for acceleration factors R = 2 to 8, even when data for a given acceleration is unavailable. This method contrasts with prior work typically trained on normalized signal, directly integrating noise into training and eliminating the need for paired high-and low SNR images.

We trained models on an extensive dataset and conducted ablations to assess the impact of g-factor augmentation, realistic MRI noise, and SNR-based training, evaluating 14 architectures for model-agnostic improvements. We also aimed to investigate whether noise-centric training enhanced generalization to unseen applications across a range of imaging contrasts, sequences, field strengths, and anatomies.

Materials and Methods

Data Collection

This retrospective study utilized retrospective-gated cardiac cine MRI data from 3T clinical scanners (MAGNETOM Prisma, Siemens AG Health care) with a balanced steady-state free precession (B-SSFP) sequence. Consecutive data were acquired with R = 2 acceleration across standard cardiac views (two-chamber, three-chamber, four-chamber, and short-axis stack), with raw *k-space* signals saved for reconstruction between January 2018 and December 2020. This dataset has not been used in previous publications.

Data from the National Institutes of Health Cardiac MRI Raw Data Repository, hosted by the Intramural Research Program of the National Heart Lung and Blood Institute, were curated with the required ethical and/or secondary audit use approvals or guidelines permitting the retrospective analysis of anonymized data without requiring written informed consent for secondary usage for the purpose of technical development, protocol optimization, and/or quality control. The data were fully anonymized and used for training without exclusion. The training and test datasets are summarized in Table 1. The training set included 96605 cine series (2885236 images) from 7590 patients, with 95% of the scans used for training and 5% for validation, and the internal test set included 3000 cine series, with no overlap.

Four external tests were conducted to evaluate generalization: (i) 10 real-time cine slices with B-SSFP contrast but different sequence parameters, (ii) five free-breathing first-pass perfusion scans for dynamic contrast changes, (iii) a T1 MPRAGE 3D brain scan (R = 2 × 2), and (iv) a high-resolution (0.76 mm^2) T2 TSE multislice 2D spine scan (R = 2), all acquired at 1.5T (MAGNETOM Aera, Siemens Healthineers, Germany). For 2D + T cases such as the cine series, the third dimension is time, while for 3D scans like T1 MPRAGE of the brain it is the second encoding dimension or depth and for spine scans with multiple 2D slices, it is the slice dimensionPhantom scans were also acquired at 1.5T with R = 2 and 4 acceleration using standard FLASH (fast low angle shot) readouts.

Training Method

SNRAware was aimed toward improving MRI denoising by generating low SNR data and providing noise distribution data to the network. The data generation process is shown in Figure 1, while Figure 2 outlines the training scheme and model design.

Training data generation with g-factor augmentation.—

Training data were acquired with R = 2 undersampling to minimize g-factor-related noise amplification. To generalize higher accelerations (R = 3 to 8), we proposed a g-factor based data augmentation scheme to compute real g-factor maps at higher accelerations. In parallel imaging, noise scales with the g-factor due to the ill-posed inversion of the calibration matrix (15,16); however, this noise amplification varies from scan to scan. After reconstruction, the noise standard deviation for pixel location p is $g(p)$, where g increases dramatically at higher accelerations (Figs 2, 3).

MRI systems acquire raw signals and store them in *k-space*, which represents the Fourier transform of the image intensities. To accelerate data acquisition, only a portion of the *k-space* data are collected; for instance, acquiring half of the data yields a speedup factor of R = 2. Parallel imaging techniques are then used to estimate the missing *k-space* data to reconstruct the full image. This process, however, amplifies the noise present in the acquired signals, leading to reconstructed images with reduced SNR. G-factor maps computed during parallel imaging reconstruction quantify noise amplification, with each pixel value indicating exactly how much the noise has been amplified. Assuming SNR unit scaling has been applied, and the original images have unitary noise (noise SD of 1.0), the g-factor directly reflects the postreconstruction noise standard deviation. Of note, this noise amplification varies spatially among pixels due to the geometry and coupling of the receiver coils, the human body, and the specifics of the reconstruction algorithm. Therefore, the g-factor map provides crucial information about where noise is present and to what extent (Fig 1).

The SNRAware training theme (Fig 1A) computed g-factor maps from 2D GRAPPA (Generalized Autocalibrating Partial Parallel Acquisition) coefficients derived from auto-calibrated or fully sampled *k-space* lines. These coefficients were converted into image domain unmixing coefficients (17,18) and g-factor maps were obtained as the sum of their squares. Although the original scans used R = 2, g-factor maps for other accelerations were computed from corresponding unmixing coefficients. A randomly selected g-factor map was used to amplify white, complex noise via pointwise multiplication during training.

Reconstruction steps such as *k-space* filtering and zero-filling resizing introduce spatial noise correlation to influence spatial noise distribution. To mimic these effects, we developed a training data augmentation process that varies noise correlation based on common reconstruction techniques by sampling white noise with a randomly selected sigma (ranging from 0 to 32.0) from each training image. This noise was amplified by a g-factor map and then modified using a *k-space* filter (Gaussian filter, sigma of 0.8, 1.0, 1.5, 2.0, or 2.25), after which a partial Fourier filter was applied with a probability of 0.5 (tapered Hanning filter (19); partial Fourier sampling ratio of 1.0, 0.85, 0.7, 0.65, or 0.55). Reduced resolution was mimicked by masking out high

frequency samples (ratio: 1.0, 0.85, 0.7, 0.65, or 0.55). Each operation was independently applied to readout and phase encoding directions.

By randomly selecting starting noise sigma, acceleration, and *k-space* filters, the augmentation procedure produced a wide range of spatially varying noise that closely resembles what would be observed in clinical imaging. Figure 1B shows noisy samples generated with different SNR levels (Supplemental Movie 1).

Providing noise distribution information to the network.—

We provided information about the noise distribution to the network to aid in the denoising task and used SNR unit (14) reconstruction for all training and test data to carefully scale the noise SD to unity and maintain this scaling through the reconstruction process. We aimed to determine if this method would aid the denoising model by reducing the variation in noise distributions which the model must learn. To perform SNR unit reconstruction, noise-only data were acquired before every scan (14). The noise readouts were used to compute the covariance matrix and perform noise prewhitening on the imaging readouts (17,20) and the noise SD was scaled to 1.0 by compensating for the equivalent noise bandwidth for every receiver coil or channel. The imaging data with unity noise went through FFT (Fast Fourier Transform), parallel imaging GRAPPA reconstruction (16), and coil combination to produce the final complex images, with noise scaling was kept constant throughout (14). The complex images were finally resized to the target matrix size with zero-filling.

We utilized similar techniques to keep the noise SD constant when generating synthetic noise for our training data. Given a high SNR image $\boldsymbol{I}$, which was reconstructed while maintaining unit noise variance scaling through all steps except parallel imaging unmixing, and a corresponding native g-factor $\boldsymbol{g}$, the corresponding SNR unit image is $\boldsymbol{S} = \boldsymbol{I} / \boldsymbol{g}$. Generated correlated noise, $\boldsymbol{n}$, with a selected variance of σ^2 and augmented as described above, was added to the SNR unit image to create a noisy sample: $\boldsymbol{S}_n = (\boldsymbol{S} + \boldsymbol{n} * \boldsymbol{g}_{aug}) / \sqrt{\sigma^2 + 1}$, where $\boldsymbol{g}_{aug}$ is a g-factor map computed by the aforementioned g-map augmentation. The ratio $1 / \sqrt{\sigma^2 + 1}$ accounts for original unity plus added noise and returns the image scaling to unit noise variance excluding the noise amplification introduced by parallel imaging unmixing. Every training pair consisted of a clean sample, $\boldsymbol{S}$, and noise augmented sample, $\boldsymbol{S}_n$. The clean image had unity noise, whereas the noisy image had spatial varying noise multiplied by $\boldsymbol{g}_{aug}$.

The process aforementioned is illustrated in Figure 2A. In addition to providing the network images with unity noise, we also provided the g-factor map to the network stacked along the channel dimension. This directly provided the network with information about the spatial amplification of noise in the image.

Sample code and detailed explanations can be found in previous publications and tutorials (13,17,18,20) for the noise prewhitening, SNR unit scaling, and computation of pixel-wise g-factor maps from parallel imaging calibration. Gadgetron framework

(https://github.com/gadgetron/gadgetron) provided an open-source, high-performance implementation.

Model and Training

The inputs for all models were 5D tensors—batch, channel, time or slice or depth, height, and width (B, C, T/S/D, H, W)—providing the flexibility to support different imaging formats.

We evaluated 14 model architectures based on two adapted backbone types: HRnet (21) and Unet (22) (Fig 2B). The transformer layers were inspired by the Swin (23), ViT (24) and more recent CNNT (Convolutional neural network transformer) (11) models, where input tensors are split into patches across T/S/D, H, W and attention was computed over patches. Detailed backbone configurations can be found in Supplemental Appendix E1.

The loss was the sum of Charbonnier loss (25), MRI perpendicular loss (26) designed to match complex values, VGG (Visual Geometry Group) perceptual loss on magnitude (27), and gradient loss, computed as the L1 difference of intensity gradient between reference standard and predicted tensors.

The dataset was split into 95% for training and 5% for validation. Model was trained with the Sophia (28) optimizer with the one-cycle learning rate scheduler (29) and cosine annealing. All models were implemented using PyTorch (30).

More information for model training is available in Supplemental Appendix E2.

Statistical Analysis

Internal test.—

Low SNR images were generated for the test dataset of 3000 series using the aforementioned process. The resulting data were fed into the trained model, and peak SNR (PSNR; $10\times log_{10}(\frac{MAX^2}{MSE})$, MSE: mean square difference, MAX: maximal value of image pixels) and structural similarity index measure (SSIM) (31) were computed for model outputs against the reference standard data. Since the image signals were floating values and noise level was unity, MAX was set to 2048.0, as SNR above this threshold was highly unlikely.

PSNR and SSIM were reported for 14 tested models (two backbone types; layer types: CNNT, CNNT-large, Swin3D, ViT3D, ViT2D, Conv3D, Conv2D). Next, the following ablations were performed:

Without g-factor map: models were trained without the g-factor map as an extra input channel and inference did not consider the g-factor map

Without MRI noise: training included the g-factor map, but the generated noise was not transformed by filters; instead, uncorrelated white noise was added to the high SNR images

Magnitude training without imaging knowledge: as simulation of reconstruction information was not available, training was conducted on magnitude images (as found in DICOM (Digital

Imaging and Communications in Medicine) images). G-factor maps were excluded, while MRIs with realistic noise and magnitude images were used with a channel dimension size of one.

External tests.—

High SNR reference standard images were not available for the imaging data acquired with higher acceleration; therefore, SNR gain was estimated using the Monte-Carlo simulation method (14) by repeatedly adding a fixed amount of noise to the input data for $n = 64$ times. The noise level in the model outputs was measured by computing the SD across repetitions and the SNR increase was measured as the reduction of noise standard deviation. Regions-of-interests were drawn in the myocardium and blood pool, from which SNR and the CNR (contrast-to-noise ratio, as $\frac{2\times(Signal_{blood}-Signal_{myo})}{(noise_{blood}+noise_{myo})}$) were measured.

A paired *t* test was performed and a *P* value < .05 was considered statistically significant. Matlab R2022a (Mathworks, USA) was used for statistical computing.

Results

Phantom

Figure 3 presents phantom test results with g-factor maps shown for R = 2 and 4 scans. For R = 4, noise was elevated in the center of the water phantom. Denoising with HRnet-CNNT and Unet-Conv3D improved performance, with the highest SNR achieved when reconstruction information was incorporated into model training, and the poorest results observed when reconstruction knowledge was excluded. The transformer-based CNNT model outperformed Conv3D.

Internal Tests

Models trained with SNRAware performed the best, removing g-factor or realistic noise knowledge degraded performance. Table 2 summarizes the internal tests and ablation studies. Detailed results are given in Supplemental Appendix E3.

External Tests

Table 3 presents the SNR and CNR of external tests.

Figure 4 presents real-time cine results for R = 5, with B-SSFP readout acquired using a protocol different from training data. The proposed training approach produced the highest SNR, with improvements ($P < .001$, Table 3). Differences between the model outputs and raw images showed that removed noise exhibited a pattern like the g-factor map. When the g-factor map was not used in training, amplified noise was removed to a lesser extent (Supplemental Movie 2). Across 10 cases, the mean SNR increased by 5.2 × for the blood pool and 3.5 × for the myocardium, while CNR improved 6.5×. Additional cine examples appear in Supplemental Movie 3.

Figure 5 shows the results of perfusion imaging, in which a contrast bolus was passed through the heart. Perfusion dynamic contrast changes are different than the cine training data . The model trained with the addition of reconstruction knowledge successfully enhanced SNR across the field of view, whereas models lacking full imaging information performed worse. The proposed method increased the mean SNR by 3.0 × for the blood pool and 3.7 × for the myocardium, with a 2.9 × increase in CNR (Table 3). Supplemental Movie 4 shows the corresponding videos.

Figure 6 demonstrates CNNT-large model generalization to unseen anatomies, including head and spine imaging. Despite differences in sequence, resolution, and contrast, the model improved SNR while preserving details. The white and gray matter contrast remained unchanged (Fig 6A), whereas a high-resolution T2 TSE spine scan (0.76 mm^2) showed enhanced SNR in the vertebrae, discs, spinal cord, and cerebrospinal fluid (Fig 6B). Supplemental Movies 5 and 6 show brain and spine results, respectively.

Discussion

We aimed to develop and evaluate a novel deep learning-based MRI denoising method using quantitative noise distribution information obtained during image reconstruction to improve model performance and generalization. To this end, this study introduced SNRAware, a new training scheme for MRI denoising that integrated knowledge from the image reconstruction process into deep learning. Signal-to-noise ratio (SNR) unit reconstruction produced unit noise level images, simplifying the task of augmenting training data with realistic noise distributions. G-factor maps were added as model input data, providing quantitative information about spatially varying noise amplification. The proposed augmentation method subsequently computed real g-factor maps for R = 2 to 8, requiring only a single acceleration factor for training data. MRI realistic noise was generated on-the-fly to lower the SNR of reference standard images, closely resembling the noise distribution in reconstructed images with higher acceleration factors and varying *k-space* filtering effects. This method eliminated the need for paired high and low SNR images for every acceleration factor and filter configuration, which is impractical for imaging with every acceleration or resolution. The training scheme was tested on 14 model architectures with two backbone types instantiated with transformer-based layers and convolution. Results of testing performed using phantoms, in vivo internal test data of 3000 cine series, and four external datasets showed that integrating reconstruction information into the training process consistently improved model performance. At higher acceleration factors, such as the R = 5 real-time cine test, the proposed method effectively corrected g-factor noise amplification, whereas models trained without g-factor maps did not.

The clinical benefits of denoising models include improved SNR and imaging efficiency. The improved SNR enables faster acquisitions with higher acceleration, equating to cost/time savings which can shorten scan time and reduce sensitivity to motion, and can also be traded for increased spatiotemporal resolution. For example, thinner slice thickness becomes more achievable, which reduces through-slice dephasing and suppresses artifacts from implants. In a clinical study with inherently poor native SNR, such as low-field MRI, denoising models can play a key role in improving image quality for more accurate diagnoses (32).

The training method was model architecture-agnostic, as shown in the experiment where different models were trained with SNRAware. Leveraging quantitative noise distribution requires the availability of g-factor map information from the reconstruction process, a limitation of this method. The model inference in this method is fully automated and thus can be integrated in the clinical setting into the imaging workflow after reconstruction to process complex images with g-factor maps. The reproducibility of this method can be studied through the scan-rescan studies and expert image reading, enhanced with the quantitative measurement of SNR gain. Further analysis of noise characteristics can be found in Supplemental Appendix E4. Furthermore, previous studies have proposed using g-factor maps in MRI denoising for CNN (Convolutional Neural Network) models (33,34). Supplemental Appendix E5 gives an extended review on this topic.

This study does have some limitations worth mentioning. First, noise prewhitening is required for SNR unit scaling, necessitating noise calibration data, and although the Siemens scanners used in this study automatically acquire noise readouts, some curated raw datasets (such fastMRI data (36)) do not come with noise scans. Therefore, the application of this method to raw datasets that lack noise scans is limited. Second, the introduction of new processing steps that alter noise distribution augmentation would require an extension of the framework. With some generalization tests, therefore, more evaluation is needed for other contrasts, resolutions, and anatomies. Third, global image quality metrics were used to measure improvement added by introducing imaging and reconstruction knowledge into the model training. Allow this allows the comparison of different models, it is not a substitute for clinical evaluation, meaning that radiologists' subjective assessments and diagnostic performance validation are still required to integrate models into the imaging workflow. Fourth, a single data source was used in our study—all data were acquired using Siemens MRI scanners at one site; therefore, more evaluations are needed for a multicenter setup. Fifth, training data diversity was limited, as cardiac cine data were the only type of data used for training the network; therefore, we assess how the addition of more diverse data for static image volumes improves model generalization over other anatomies. Finally, we conducted ablation tests with well-controlled training setups to demonstrate the added value of using quantitative noise information in denoising training. Although the improvements were agnostic to specific model architecture, we did not attempt end-to-end comparisons with other methods and systems, which would be crucial to the validation of models deployed into the clinical imaging workflow. In general, limitations such as single data source, limited training data diversity and clinical validation, and noise prewhitening were acknowledged, and are potential scopes for future studies.

This study used SSIM and PSNR to compare different training setups and evaluate whether integrating noise information from the MRI reconstruction process would boost model performance. However, global image quality metrics were used to compare two models, rather than validating diagnostic performance. Previous studies (35) showed that SSIM correlated positively with the radiologists' scores, but tended to overlook local degradation, given that it is a global quality metric. This study focused on showing that the proposed training scheme would improve all model architectures. In future studies, therefore, dedicated radiologist evaluations are needed to validate model performance on a per-imaging-task basis. Models trained to recognize

MRI noise distribution may generalize to unseen imaging applications, supported by experiments on perfusion imaging with dynamically contrast and brain and spine scans.

Author affiliations:

[1] Health Futures, Microsoft Research, 14820 NE 36th St, Bldg 99, Rm 4941, Redmond, WA 98052

[2] National Heart, Lung and Blood Institute, National Institutes of Health, Bethesda, Md

[3] Institute of Cardiovascular Science, University College London, London, UK

[4] Barts Heart Centre, Barts Health NHS Trust, London, UK

Received XXX; revision requested XXX; revision received XXX; accepted XXX.

Address correspondence to: H.X. (email: xueh@microsoft.com).

Funding: Authors declared no funding for work.

Author contributions: Guarantors of integrity of entire study, **H.X., M.S.H.**; study concepts/study design or data acquisition or data analysis/interpretation, all authors; manuscript drafting or manuscript revision for important intellectual content, all authors; approval of final version of submitted manuscript, all authors; agrees to ensure any questions related to the work are appropriately resolved, all authors; literature research, **H.X., S.M.H., J.N., M.S.H.**; clinical studies, **I.P.**; experimental studies, **H.X., S.M.H., I.P., R.H.D., A.E.C.W., P.K.**; statistical analysis, **H.X., J.N.**; and manuscript editing, **H.X., S.M.H., I.P., R.H.D., J.N., A.E.C.W., C.M., J.C.M., T.A.T., M.S.H., P.K.**

Disclosures of conflicts of interest: H.X. Full-time employee and stockholder, Microsoft Research. **S.M.H.** No relevant relationships. **I.P.** No relevant relationships. **R.H.D.** Consulting fees and shareholder, Mycardium AI. **J.S.** Full-time employee and stockholder, Microsoft Research. **J.N.** Full-time employee and stockholder, Microsoft Research. **A.E.C.W.** Principal Investigator on a Cooperative Research and Development Agreement with Siemens Healthineers. **C.M.** Board member, MycardiumAI. **J.C.M.** No relevant relationships. **T.A.T.** Institutional grant, JenaValve; consulting, AstraZeneca; speakers bureau, Siemens Healthineers; stocks, Mycardium AI. **M.S.H.** Salary and funding for compute infrastructure, Microsoft; stock grants as part of compensation, Microsoft; full-time employee; Microsoft; stockholder, Microsoft. **P.K.** No relevant relationships.

References

1. Tian C, Fei L, Zheng W, Xu Y, Zuo W, Lin CW. Deep Learning on Image Denoising: An overview. Neural Netw 2020;131:251–275.

2. Campbell-Washburn AE, Keenan KE, Hu P, et al. Low-field MRI: A report on the 2022 ISMRM workshop. Magn Reson Med 2023;90(4):1682–1694.

3. Campbell-Washburn AE, Ramasawmy R, Restivo MC, et al. Opportunities in interventional and diagnostic imaging by using high-performance low-field-strength MRI. Radiology 2019;293(2):384–393.

4. Xiang T, Yurt M, Syed AB, Setsompop K, Chaudhari A. DDM2: Self-Supervised Diffusion MRI Denoising with Generative Diffusion Models. ICLR. 2023. https://github.com/StanfordMIMI/DDM2.

5. Wang X, Uecker M, Feng L. Fast Real-Time Cardiac MRI: a Review of Current Techniques and Future Directions. Investig Magn Reson Imaging 2021;25(4):252–265.

6. Kellman P, Hansen MS, Nielles-Vallespin S, et al. Myocardial perfusion cardiovascular magnetic resonance: optimized dual sequence and reconstruction for quantification. J Cardiovasc Magn Reson 2017;19(1):43.

7. Huang T, Li S, Jia X, Lu H, Liu J. Neighbor2Neighbor: Self-Supervised Denoising from Single Noisy Images. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021. doi:10.1109/CVPR46437.2021.01454.

8. Lehtinen J, Munkberg J, Hasselgren J, et al. Noise2Noise: Learning Image Restoration without Clean Data. arXiv 2018. Preprint posted online March 12, 2018; https://arxiv.org/abs/1803.04189.

9. Krull A, Buchholz TO, Jug F. Noise2Void-Learning Denoising from Single Noisy Images. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/CVPR.2019.00223.

10. Lequyer J, Philip R, Sharma A, Hsu WH, Pelletier L. A fast blind zero-shot denoiser. Nat Mach Intell 2022;4(11):953–963.

11. Rehman A, Zhovmer A, Sato R, et al. Convolutional neural network transformer (CNNT) for fluorescence microscopy image denoising with improved generalization and fast adaptation. Sci Rep 2024;14(1):18184.

12. Chung H, Lee ES, Ye JC. MR Image Denoising and Super-Resolution Using Regularized Reverse Diffusion. IEEE Trans Med Imaging 2023;42(4):922–934.

13. Kellman P; ISMRM. Parallel Imaging: The Basics https://kellmanp.github.io/webpages/publications.htm. Published 2002. Accessed DATE.

14. Kellman P, McVeigh ER. Image reconstruction in SNR units: A general method for SNR measurement. Magn Reson Med 2005;54(6):1439–1447.

15. Pruessmann KP, Weiger M, Scheidegger MB, Boesiger P. SENSE: Sensitivity encoding for fast MRI. Magn Reson Med 1999;42(5):952–962.

16. Griswold MA, Jakob PM, Heidemann RM, et al. Generalized Autocalibrating Partially Parallel Acquisitions (GRAPPA). Magn Reson Med 2002;47(6):1202–1210.

17. Deshmane A, Gulani V, Griswold MA, Seiberlich N. Parallel MR imaging. J Magn Reson Imaging 2012;36(1):55–72.

18. Breuer FA, Kellman P, Griswold MA, Jakob PM. Dynamic autocalibrated parallel imaging using temporal GRAPPA (TGRAPPA). Magn Reson Med 2005;53(4):981–985.

19. Prabhu KMM. Window Functions and Their Applications in Signal Processing. CRC, 2014.

20. Hansen MS. Nuts & Bolts of Advanced Imaging Image Reconstruction-Parallel Imaging. ISMRM 2013. https://www.ismrm.org/13/S10.htm.

21. Wang J, Sun K, Cheng T, et al. Deep High-Resolution Representation Learning for Visual Recognition. arXiv 2019. Preprint posted online August 20, 2019; https://arxiv.org/abs/1908.07919.

22. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv 2015. Preprint posted online May 18, 2015; https://arxiv.org/abs/1505.04597.

23. Liu Z, Lin Y, Cao Y, et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows 2021. https://doi.org/10.1109/ICCV48922.2021.00986.

24. Dosovitskiy A, Beyer L, Kolesnikov A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv 2020. Preprint posted online October 22, 2020; https://arxiv.org/abs/2010.11929.

25. Barron JT. A General and Adaptive Robust Loss Function. arXiv 2017. Preprint posted online January 11, 2017; https://arxiv.org/abs/1701.03077.

26. Terpstra ML, Maspero M, Sbrizzi A, van den Berg CAT. ⊥-loss: A symmetric loss function for magnetic resonance imaging reconstruction and image registration with deep learning. Med Image Anal 2022;80:102509.

27. Johnson J, Alahi A, Li FF. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. arXiv 2016. Preprint posted online March 27, 2016; https://arxiv.org/abs/1603.08155.

28. Liu H, Li Z, Hall D, Liang P, MT. Sophia: A Scalable Stochastic Second-Order Optimizer For Language Model Pre-Training. arXiv 2023. Preprint posted online May 23, 2023; https://arxiv.org/abs/2305.14342.

29. Smith LN. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. arXiv 2018. Preprint posted online March 26, 2018; https://arxiv.org/abs/1803.09820.

30. Paszke A, Gross S, Massa F, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv 2019. Preprint posted online December 3, 2019; doi:10.48550/arxiv.1912.01703.

31. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: From error visibility to structural similarity. IEEE Trans Image Process 2004;13(4):600–612.

32. Xue H, Hooper S, Rehman A, et al. Routine CMR at 0.55T with Standard Spatial Resolution Using an Imaging Transformer. J Cardiovasc Magn Reson 2024;26(Supplement 1):100121.

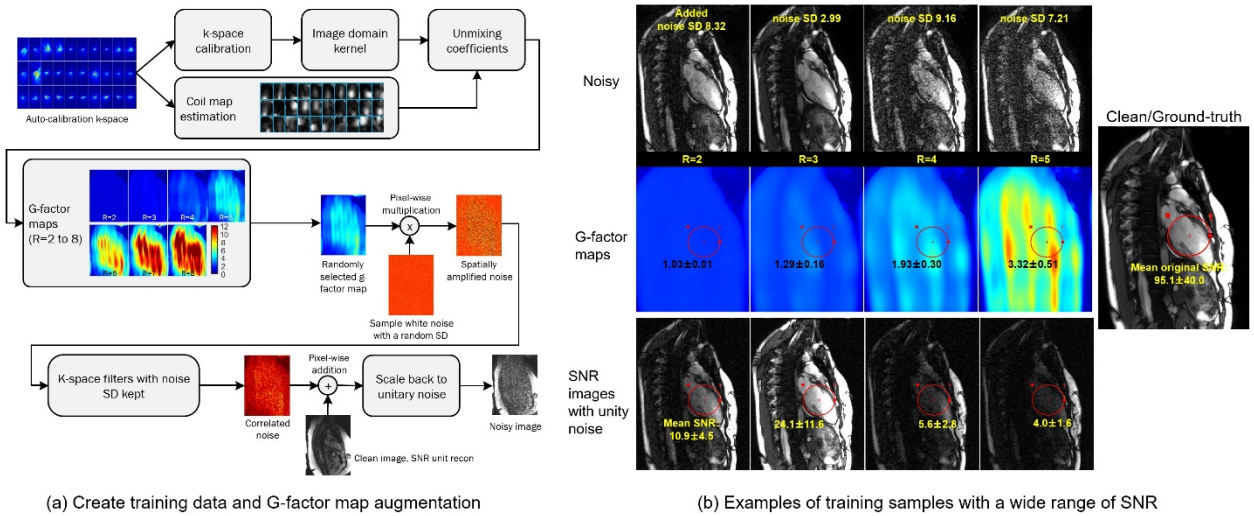
33. Pfaff L, Hossbach J, Preuhs E, et al. Self-supervised MRI denoising: leveraging Stein's unbiased risk estimator and spatially resolved noise maps. Sci Rep 2023;13(1):22629.

34. Dou Q, Wang Z, Feng X, Campbell-Washburn AE, Mugler JP, Meyer CH. MRI denoising with a non-blind deep complex-valued convolutional neural network. NMR Biomed 2025;38(1):e5291.

35. Mason A, Rioux J, Clarke SE, et al. Comparison of Objective Image Quality Metrics to Expert Radiologists' Scoring of Diagnostic Quality of MR Images. IEEE Trans Med Imaging 2020;39(4):1064–1072.

36. Zbontar J, Knoll F, Sriram A, et al. fastMRI: An Open Dataset and Benchmarks for Accelerated MRI. arXiv 2018. Preprint posted online November 21, 2018; https://arxiv.org/abs/1811.08839.

37. Zhang K, Zuo W, Chen Y, Meng D, Zhang L. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE Trans Image Process 2017;26(7):3142–3155.



(a) Create training data and G-factor map augmentation

(b) Examples of training samples with a wide range of SNR

Figure 1: Training sample creation. **(A)** The training samples are paired clean and noisy image series. The noisy images are created by computing the g-factor maps and using them to generate noise images. Geometry (G)-factor maps were computed from the auto-calibration *k-space*. GRAPPA (Generalized Autocalibrating Partial Parallel Acquisition) calibration was computed for accelerations R = 2 to 8. The GRAPPA k-space kernel was converted to image domain kernel and unmixing coefficients were computed by combining image domain kernels and coil maps. For data augmentation, a g-factor map is randomly selected and pixel-wise multiplied to the white noise. The resulting spatially varying noise further goes through k-space filtering steps to introduce correlation. The final noise is added to the clean image and scaled to be unitary, as the

noisy sample for training. **(B)** Four noisy samples are created from a clean cine series. The original SNR (signal-to-noise ratio) is 95.1 in the region of interest. By randomly selecting a g-factor map and changing the starting noise level, a wide range of SNRs can be produced. The SNR images are computed by dividing the images by the g-factor map.

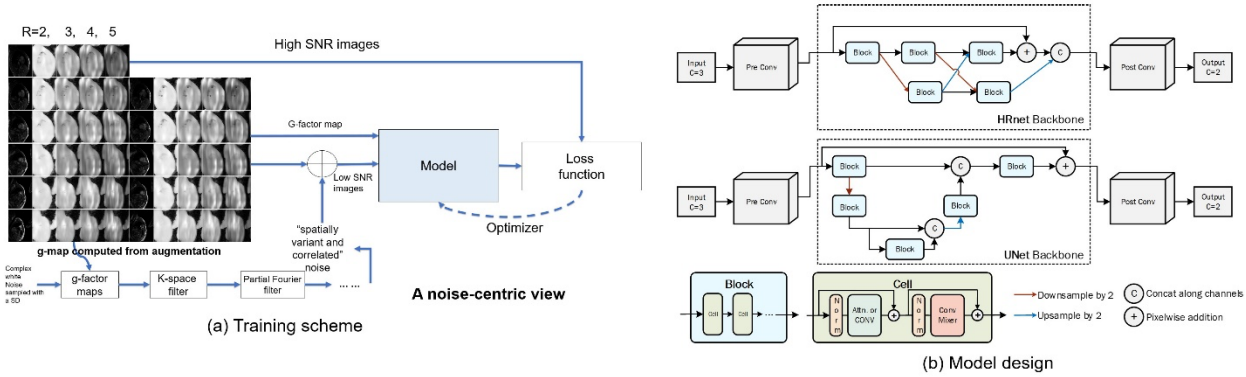


Figure 2: Overview of training scheme and model design. **(A)** Reconstructed cine images are augmented with spatially varying and correlated noise to create noisy samples. The corresponding g-factor maps are concatenated to the images and used as input into the model. The model predicts high SNR (signal-to-noise ratio) images. **(B)** All models consist of a preconv layer as the shallow feature extractor, a backbone and the output convolution. To simplify the evaluation on different models, a cell-block-backbone design is proposed for the backbone. Two backbone architectures tested here are HRnet and Unet. Both backbones process tensors through blocks which are connected by the downsample/upsample operation. Every block consists of 3 to 6 cells. Every cell follows the standard design, including normalization, attention or convolution layers and skip connections. By changing the module in the cell, different transformers and convolution models are instantiated and tested.

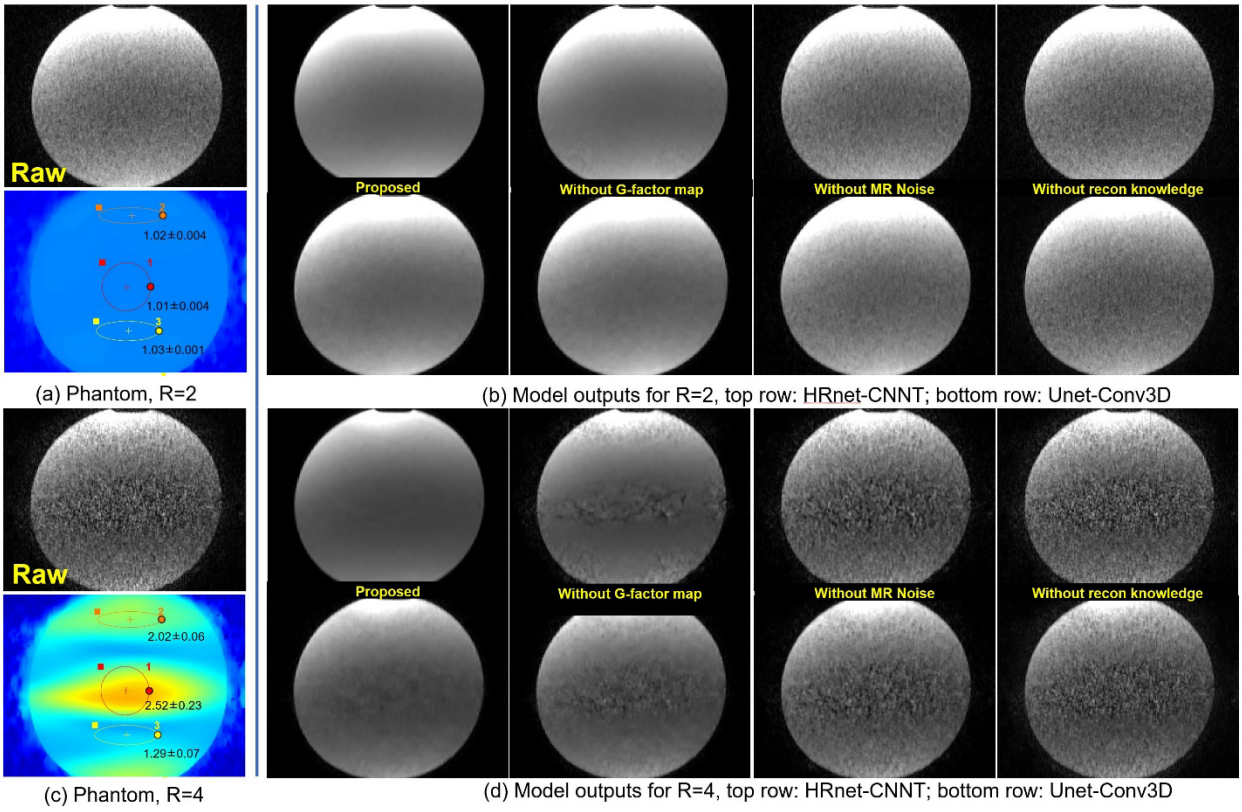


Figure 3: Phantom test results. **(A)** The raw images of R = 2 acceleration and g-factor map show a very minor noise amplification. **(B)** Model outputs with one transformer and one convolution architecture for proposed training, compared with ablation tests. **(C)** The images and g-factor map for R = 4 acquisition shows lower SNR (signal-to-noise ratio) and spatial noise amplification. The g-factor is higher at 2.52 at the center of field of view. The proposed method removes noise amplification. Training without g-factor map results in less efficient noise removal. For both accelerations, transformer models outperform convolution. Training without MR noise distribution further degrades the performance.

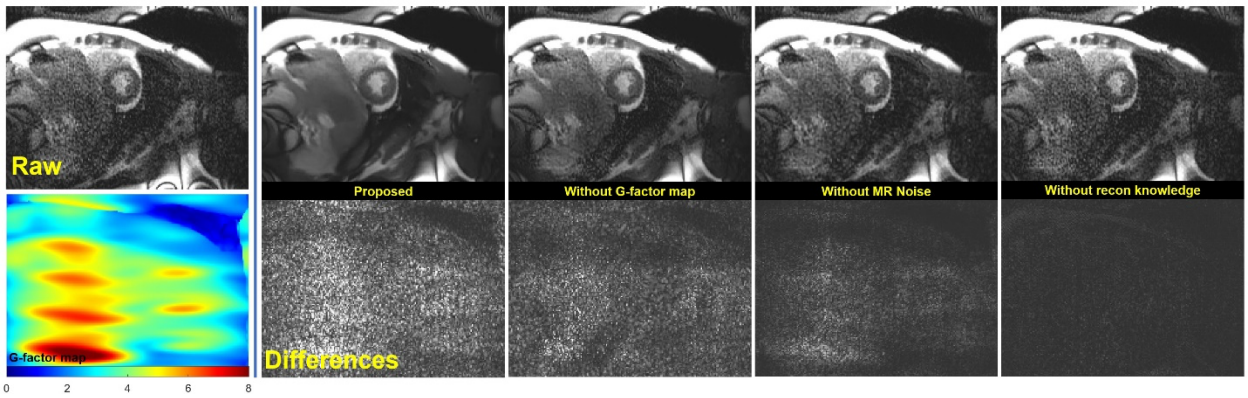


Figure 4: Real-time cine (acceleration R = 5) results produced with the HRnet-CNNT-large model. The raw SNR (signal-to-noise ratio) is lower with elevated spatial noise amplification due to acceleration. The proposed training method produced the best results.

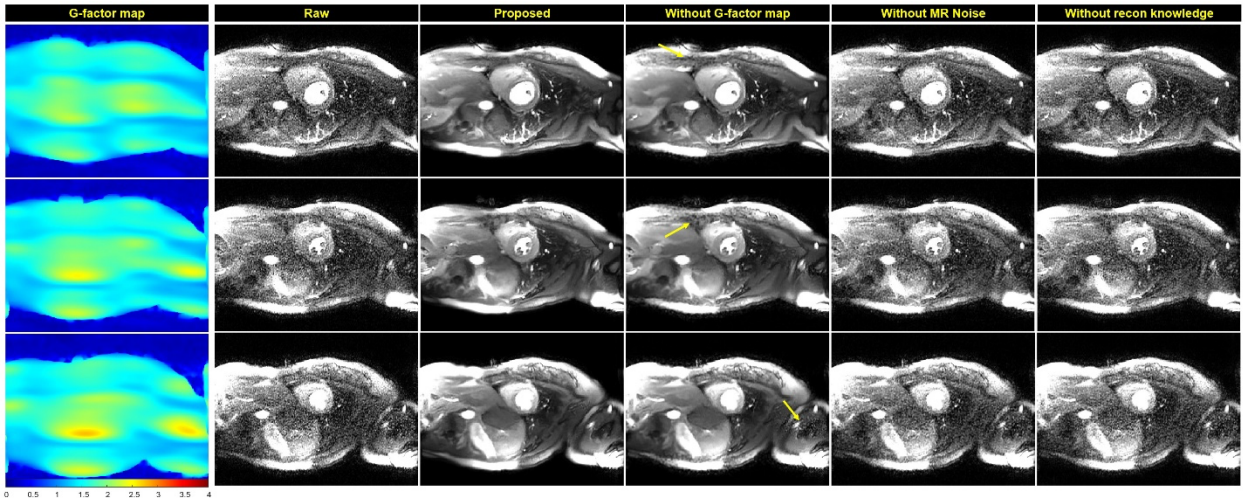


Figure 5: Result for accelerated (R = 4) myocardial perfusion imaging. The contrast passage creates dynamically varying contrast which was not seen in the training dataset. Moreover, the saturation preparation reduced the base SNR. Despite the challenges, the model generalized well to perfusion imaging. Similarly to previous tests, noise reduction is more effective when knowledge about noise distribution is included in training.

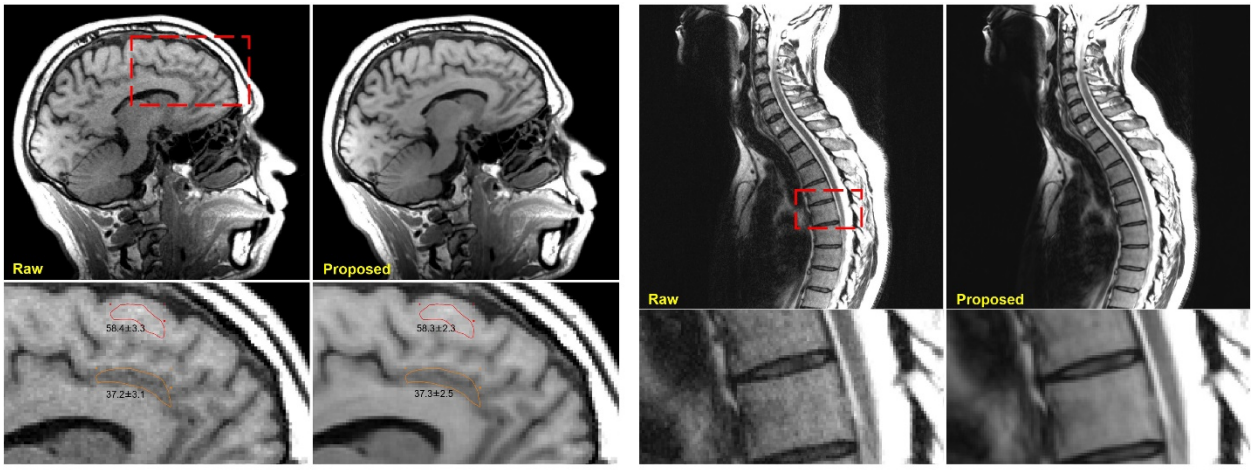
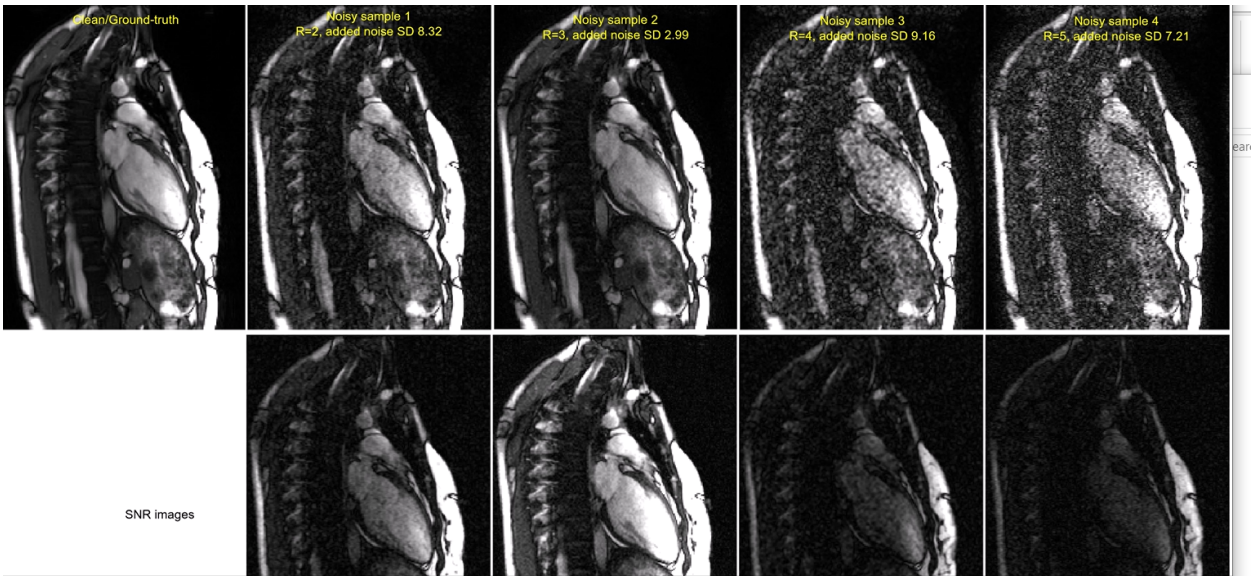


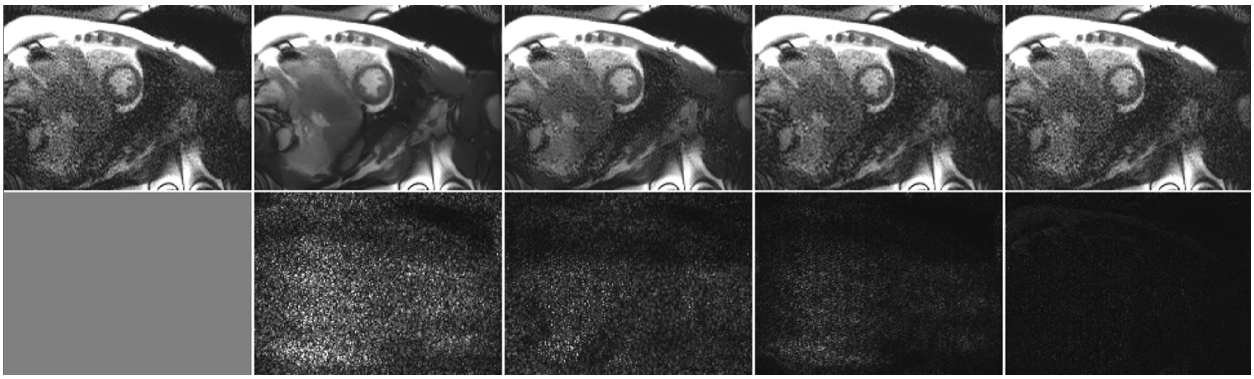
Figure 6: Generalization tests for different anatomies. **(A)** A R = 2 × 2 MPRAGE (Magnetization-Prepared Rapid Gradient Echo) T1 brain scan was acquired, reconstructed and processed with trained model. The training dataset did not include any neuro data, yet the model generalized well, with noticeable SNR (signal-to-noise ratio) improvement and preserved gray-white matter contrast. This test was to check whether models can generalize to new anatomies; further evaluation is needed to test on more datasets with expert image reading. The gray and white matter differentiation appeared. **(B)** A R = 2 T2 TSE (Turbo Spin Echo) spine scan was processed with the trained model. Training data did not include spine scans and did not include the high spatial resolution ($0.76mm^2$) of this acquisition. The model generalized well to this application regardless.

Radiology: Artificial Intelligence

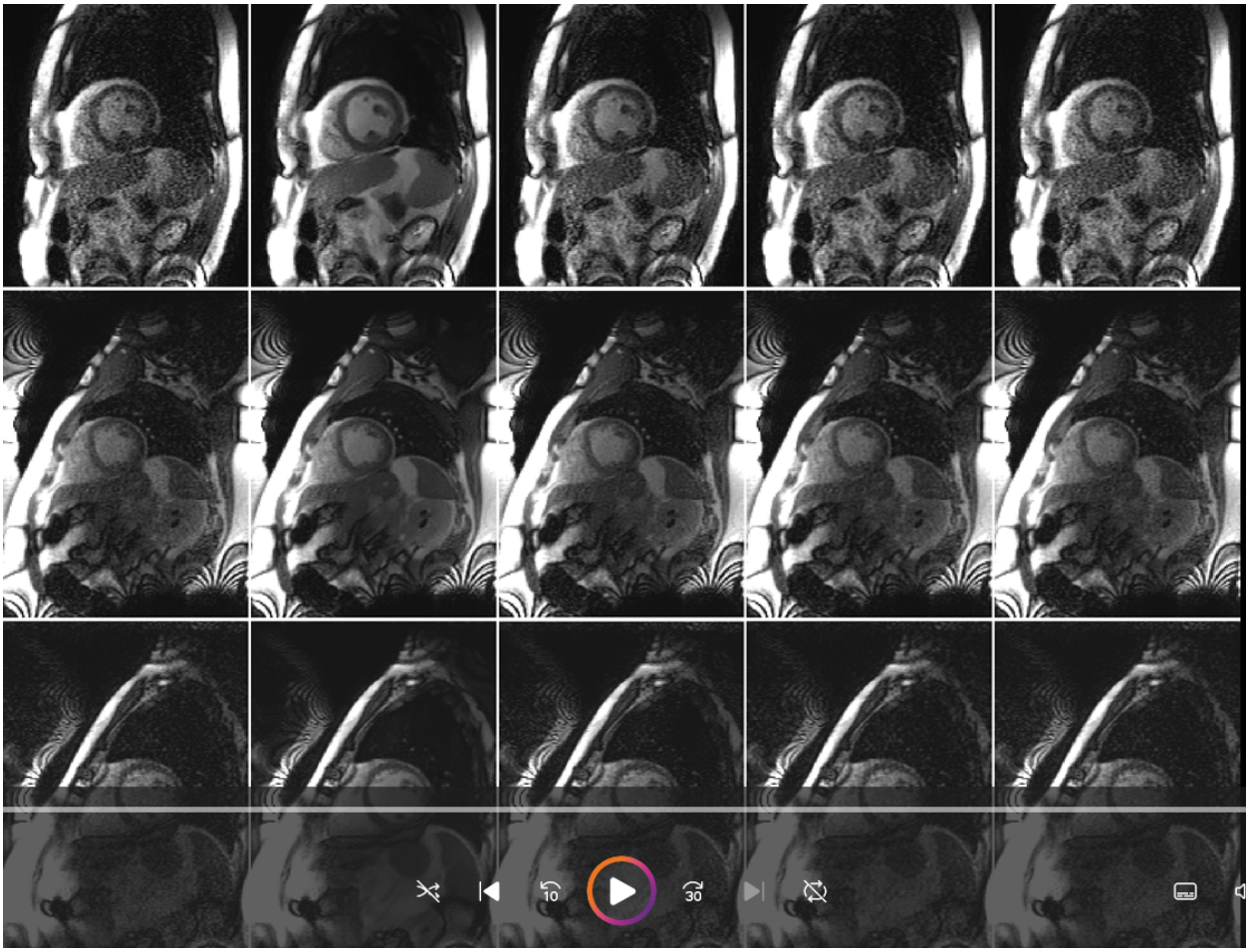
Supplemental Data



Movie 1: The movies correspond to the example in Figure 1b. The reference standard clean image is the single one on the left. The first row are the noisy samples. The second row are the SNR images.

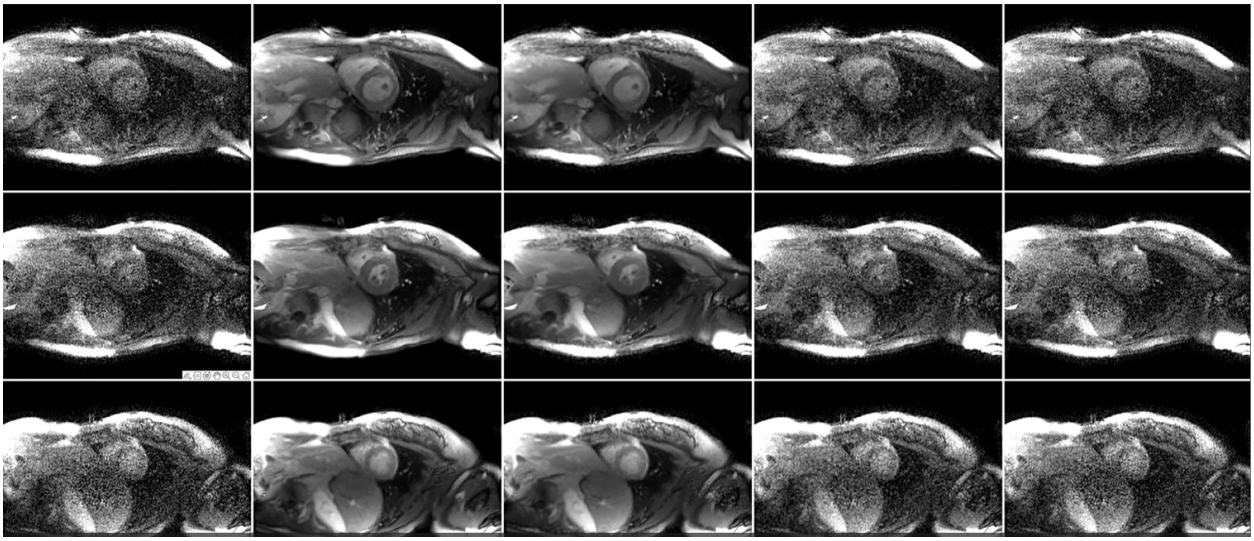


Movie 2: Corresponding movies to Figure 4 are given here.

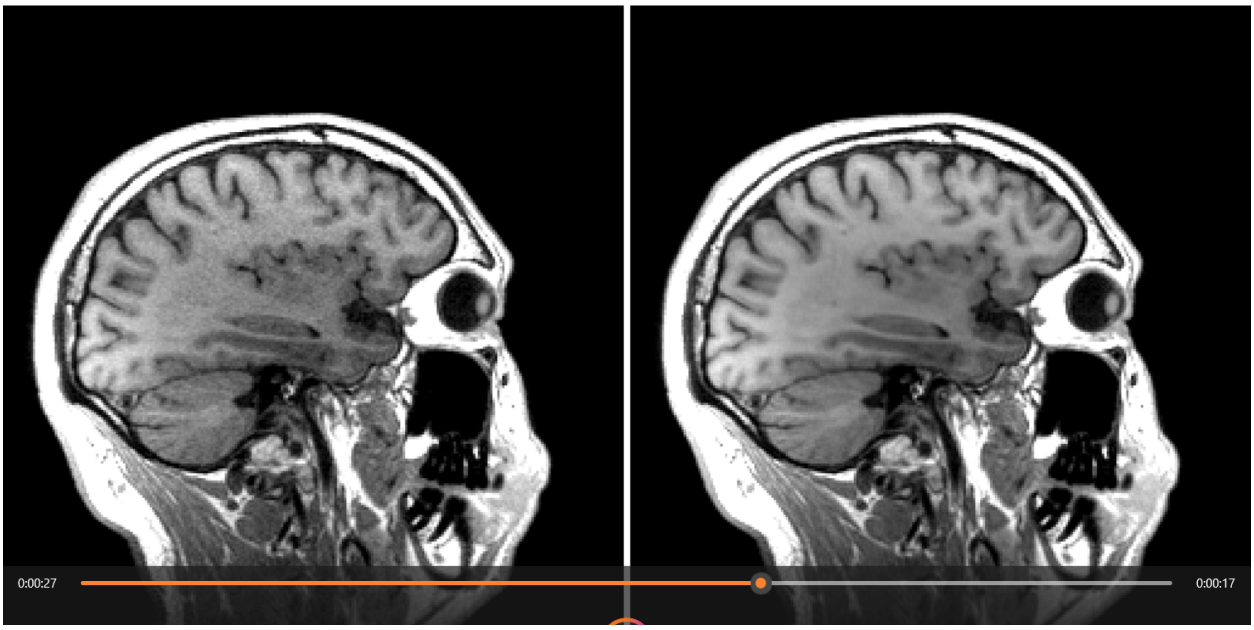


Movie 3: More R=5 real-time cine examples are given here. In all cases, proposed training noticeably improves performance. The leftover noise amplification is very visible without the g-factor map.

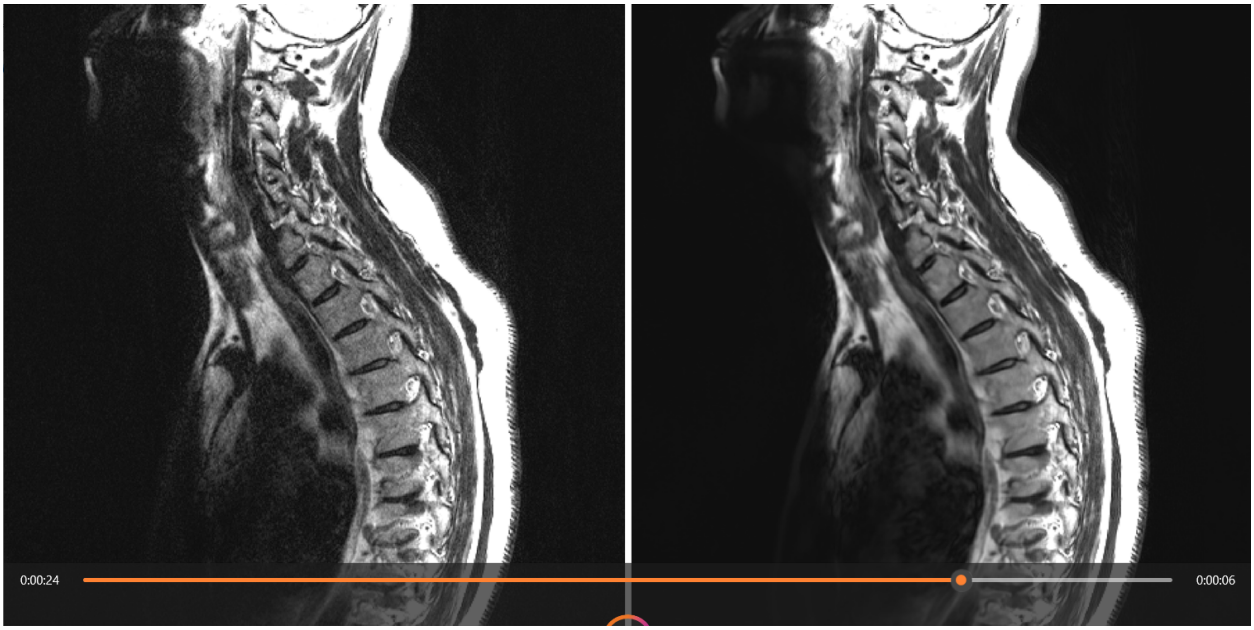
Radiology: Artificial Intelligence



Movie 4: Movies of perfusion denoising corresponding to Figure 5 are presented. Model generalized well to dynamic contrast and low base SNR.



Movie 5: Movie corresponds to Figure 6a for the T1 MPRAGE neuro test.



Movie 6: Movie corresponds to Figure 6b for the T2 TSE spine test.

Supplemental Appendices

Appendix E1. Information for deep learning models

As shown in Figure 2, the model consists of three components: pre-convolution layer, backbone and post-convolution layer. The input tensors are in the shape of [B, C, T/S/D/Z, H, W]. C is 3 for complex inputs (real, imagery and g-factor). Noise in the input images are scaled to 1.0×g-factor, as this setup is consistent with reconstruction outputs.

The pre-convolution layer is a shallow feature extractor (37). It is kept being minimal as a 2D convolution to uplift input channel C to 64, encouraging backbone to take on most heavy lifting and helping generalization. The post-convolution is another CONV layer, converting $C_{backbone}$ after the backbone to required output channels (2 for complex training and 1 for magnitude training).

Two well-known backbone architectures, HRnet and Unet, are implemented and tested in this study. Both architectures utilize the multi-resolution pyramid to balance model size, expressive power and computing cost. The building components include multiple Blocks, downsample and upsample layers, channel-wise concatenation, and skip connection. The HRnet

maintains a longer pipeline on the original tensor size and Unet is smaller in size and less computing expensive.

The input tensors are processed through every block, gaining more channels and reducing

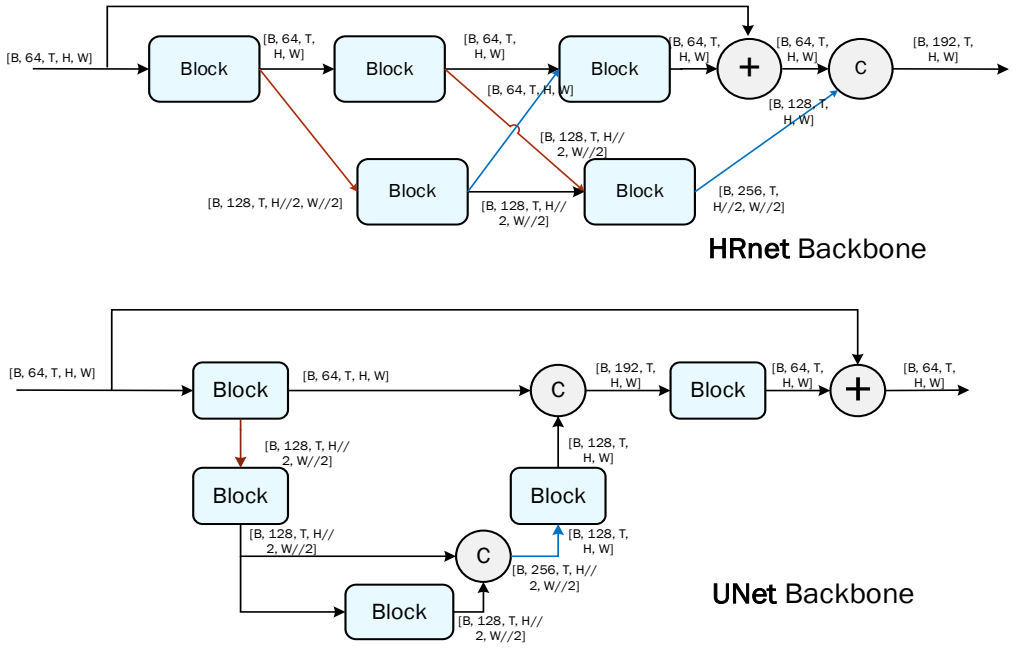


Figure 1. Annotated backbone architectures.

spatial resolution, which is explained by the backbone plots annotated with tensor sizes.

Downsampling was implemented with patch merging (23) followed by a convolution to format outputs to have the required number of channels. The upsampling was implemented with a linear interpolation followed by a CONV layer.

Backbones consist of several blocks. A block is a container of N cells. Every cell has a classical setup of two skip connections, layer norms and attention or convolution layers. By switching the attention methods (e.g. Swin3D, ViT3D or CNNT etc.), we can instantiate different models for experiments. A pure convolution model was implemented by replacing attention with convolution layers.

Every block in all models, except CNNT-large, has 3 cells. For CNNT-large, a block holds 6 blocks. By inserting more cells or more blocks, the model can be scaled up or down.

As used in other denoising training schemes, models were trained on image patches to encourage models to focus on noise distribution instead of image content. The patch size was

[T/S/D/Z=16, H=64, W=64]. The window size in Swin3D and ViT3D was [16, 8 ,8], where every [2, 2, 2] neighborhood was processed as a token. The CNNT transformer method computed attention between all [H, W] frames without explicit neighborhood tokenization. All convolutions had the kernel size 3 and padding 1. We note that unlike the original Swin and ViT papers, we re-patch and un-patch the tensors before and after every operation, resulting in imaging tensors that can be processed by the normalization and convolutional mixer layers in every cell.

Appendix E2. Information for model training

The inputs for all models were 5D tensors—batch, channel, time or slice or depth, height, and width (B, C, T/S/D, H, W)—providing the flexibility to support different imaging formats. For instance, for the input cine series, the third dimension was time, whereas for a 3D brain scan it was depth or slice. The g-factor map was concatenated to real and imaginary parts of image tensors, so C was 3 for complex training, whereas if only the magnitude image was used, C was 2. The model outputs a tensor with the same shape with a channel dimension of 2 for complex and 1 for magnitude images, respectively.

We evaluated 14 model architectures based on two adapted backbone types: HRnet (21) and Unet (22) (Figure 2B), both of which use multi-resolution pyramids to balance computational complexity with the ability to recover small image features by maintaining a full resolution path. Each network consisted of multiple blocks containing several cells, with every cell including normalization, a computing layer, and a mixer. Different models were instantiated by configuring different computing layers, and both transformer and convolution layers were tested. The transformer layers were inspired by the Swin (23), ViT (24) and more recent CNNT (Convolutional neural network transformer) (11) models, where input tensors are split into patches across T/S/D, H, W and attention was computed over patches. For Swin, we split the input image into patches and applied attention over local and shifted windows; for ViT, attention was global among all patches; and the CNNT cells did not patch the image, instead applying attention in the T/S/D dimension. We also tested convolution layers (“Conv” blocks), which did not patch the image, instead applying standard convolution. All cells included three layers except for CNNT-large, with six layers. The ViT2D and Conv2D models were further trained through 2D patching and attention or 2D convolution, operating over height and width rather than across frames. These configurations enabled us to assess SNRAware over transformer, convolutional models, and 2D/3D models, as well as multiple backbone configurations (Supplemental Appendix E1).

The loss was the sum of Charbonnier loss (25), MRI perpendicular loss (26) designed to match complex values, VGG (Visual Geometry Group) perceptual loss on magnitude (27), and gradient loss, computed as the L1 difference of intensity gradient between reference standard and predicted tensors.

The dataset was split into 95% for training and 5% for validation. A fast second order optimizer, Sophia (28), was used with the one-cycle learning rate scheduler (29) and cosine annealing. The peak learning rate was 1e-5, betas were 0.9 and 0.999, and epsilon was 1e-8. The training lasted 80 epoch, and the final model was selected as the one giving the highest performance on the validation set. All models were implemented using PyTorch (30) and training was performed on a cluster of 128 AMD MI300X GPUs (Graphic Processing Unit), each with 192 GB RAM (Random Access Memory). Data distributed as used GPU cards to speed up training.

Our training and model architecture design were generalized over various data dimensionalities by processing tensors in the shape of batch, channel, time/slice/frame, height and width as [B, C, T/S/D, H, W]. The models denoised each of these formats despite being trained on only 2D + T data, making it practical to combine different training data (e.g., 2D + T, 3D, and multi-slices) into one model training session, potentially improving model generalization.

Appendix E3. Internal and ablation results

Table 2 summarizes the internal test results. Ablation studies were conducted for all 3D and transformer models, but not for 2D models as they were much less competitive. The proposed scheme consistently yielded the best PSNR and SSIM. Removing g-factor or realistic MRI noise from model training degraded performance, with the poorest results observed when reconstruction knowledge was omitted.

Across the various architectures, CNNT-large performed better (as the highest scores in Table 2), with HRnet-CNNT-large achieving the highest PSNR (54.90) and SSIM (0.71). Comparing 3D and 2D versions (Hrnet backbone: ViT3D SSIM = 0.63 and PSNR = 51.75 vs. ViT2D SSIM = 0.48 and PSNR = 46.96, Conv3D SSIM = 0.60 and PSNR = 50.74 vs. Conv2D SSIM = 0.47 and PSNR = 46.67; Unet: ViT3D SSIM = 0.62 and PSNR = 51.25 vs. ViT2D SSIM = 0.47 and PSNR = 46.53, Conv3D SSIM = 0.62 and PSNR = 51.79 vs. Conv2D SSIM = 0.50 and PSNR = 47.43), 3D models showed superior performance, and the model with highest scores, Hrnet-CNNT-large, was used in the generalization tests. Table 3 presents the SNR and CNR results for cardiac generalization tests.

Appendix E4. More discussion about MR noise characteristics

MR noise follows the Gaussian distribution in real and imaginary components, after the noise pre-whitening [1]. The noise corrupts the signal as the additive component. The MR SNR will be biased higher if magnitude detection was applied. This bias decreases with more receiver channels and higher signal strength [2]. But in this study, the processing was in the complex domain, and we studied the noise distribution in the absence of signal as a training augmentation. As a result, the noise distribution was still Gaussian, and simulated noise can be added to the signal.

The uncorrelated white noise, however, will become correlated colored noise after MR reconstruction. The g-map noise amplification leads to spatially variant SNR, further deviating the noise distribution from the normal Gaussian.

Although it is tractable to analytically track the change of noise distribution, it is much easier to set up the deep learning training by sampling the white noise and passing them through the same processing steps and adding resulting noise to reduce image SNR. This is the approach used in this study, utilizing the additive nature of MR noise corruption.

The results showed denoising model performance was improved if added noise was augmented with g-factor map and processed with the same filters as in the reconstruction. As the open-source Gadgetron MR reconstruction was used in this study, these requirements were precisely met in the model training, leading to noticeable boost in denoising performance.

To illustrate how the g-factor amplification and other steps alter the noise distribution, we sampled white noise and processed it with a real g-factor map and k-space filters. The noise power spectrum was plotted for the demonstration.

A noise was sampled for a 256×192 matrix. A R=5 g-factor map was sampled to amplify the noise. K-space filters were generated with filter width being 1.0 pixel, 1.5pixel and 2.5 pixel. Partial Fourier filters [3] were tapered Hanning with 6/8 sampling along the phase encoding (in this example, 192 is the phase encoding length) and a transition band of 12, 24 and 48 pixels. The power spectrum was computed by repeating the noise sampling 256 times and taking the average. The 1D profile of power spectrum was plotted here (along with the phase encoding direction at the k-space center) for visualization.

As shown in Figure 2, the g-map amplification and other processing steps changed the noise distribution and altered its appearance. Experiments showed model performance degraded

if model was trained without realistic noise added. By randomly concatenating g-factor maps and filters, the training can see a wide range of combinations of colored noise, helping model differentiate signal from noise.

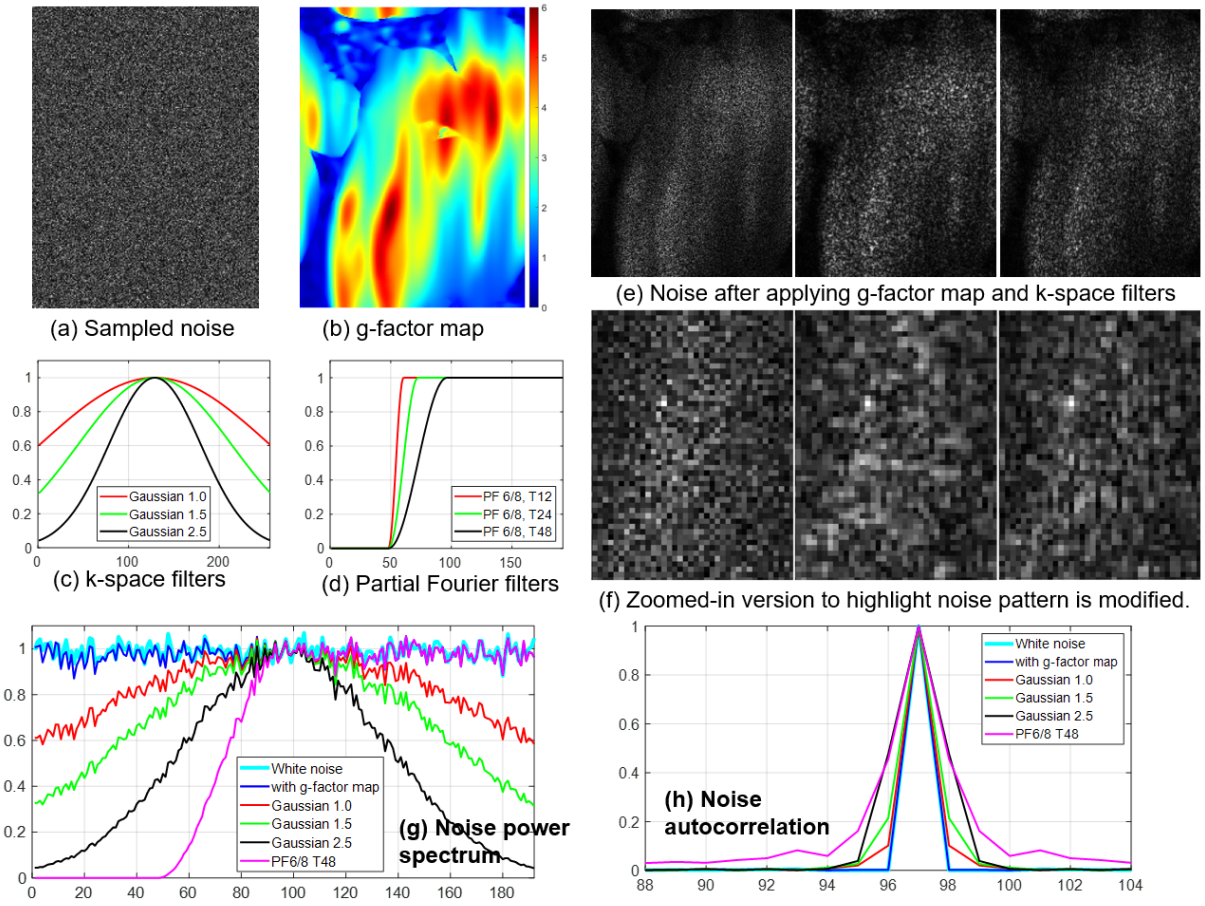


Figure E2. Demonstration of noise distribution after g-factor amplification and k-space filters. (a) The white noise was sampled to a 256x192 matrix. (b) A R=5 g-factor map was used in this demo. (c) A set of k-space filters are plotted for different strength. (d) The partial Fourier filters used in Gadgetron was on side Hanning with different transition bands. (e) Noise pattern after applying g-factor map and k-space filters. From left to right: only applying g-factor map; g-factor map and Gaussian 2.5 filter for both readout and phase encoding; g-factor map and Gaussian 2.5 filter for readout and PF 6/8 T48 filter for phase. (f) Zoomed-in version to visualize differences in noise pattern. (g) Power spectrum of noises. Note the flat white noise spectrum was altered by g-factor map and filters. Different combination of these processing steps will alter the noise differently. (h) Corresponding autocorrelation.

1. Gudbjartsson H, Patz S. The Rician distribution of noisy MRI data. Magn Reson Med. 1995;34(6):910–914. doi: 10.1002/mrm.1910340618.

2. Constantinides CD, Atalar E, McVeigh ER. Signal-to-noise measurements in magnitude images from NMR phased arrays. Magn Reson Med. 1997;38(5):852–857. doi: 10.1002/mrm.1910380524.

3. Mcgibney G, Smith MR, Nichols ST, Crawley A. Quantitative Evaluation of Several Partial Fourier Reconstruction Algorithms Used in MRI. Magn Reson Med. 1993;51–59. https://doi.org/10.1002/mrm.1910300109.

Appendix E5. Extended review for methods using g-factor maps

Previous studies have proposed using g-factor maps in MRI denoising. In a recent study (33), 23 T2 brain scans were collected and used to train a CNN (Convolutional Neural Network) model. G-factor maps were computed and multiplied with the MRI surface coil inhomogeneity maps, and the resulting “noise map” was used to scale the white noise to introduce spatial variance. The main difference when compared with SNRAware is that the previous study used g-factor maps for training data simulation with an unknown noise level; therefore, images were not scaled in the SNR units.

Another study (34) provided g-factor maps as input for training with low SNR. In that study a CNN model was trained with 2000 T2 brain scans and simulated noise. Internal testing was performed against other brain images; however, noise pre-whitening with actual noise readouts was not performed to get a fixed scaling level, rather the authors of that study estimated noise sigma using a wavelet method from the *k*-space data. We trained models on larger datasets for both transformers and convolution architecture, and the impact of noise amplification from the g-factor and noise correlation caused by raw filter and other steps were separately tested with more extensive out of distribution validations. Our study emphasized that a noise-centric view of denoising training can improve the generalization of trained models to unseen imaging applications.

Table 1: Imaging and demographic characteristics for training and test datasets

Category	Imaging Application	Anatomy	Typical Sequence Parameters	Field Strength	No. Samples And Data Format
Training and internal testing	Retro-gated cine	Heart	Data acquisition with breath-holding FOV: 360 × 270mm^2 Acquired matrix size: 256 × 144 Echo time: 1.28 msec Bandwidth: 977 Hz/pixel Readout: SSFP RF Flip angle: 50^o Echo spacing: 2.97 msec Output phases: 30 Acceleration: R = 2	3T	Training: *n* = 7590 patients, 96,605 cine series, 2,885,236 images, 61% male, mean age 54 years Testing: *n* = 231 patients, 3,000 cine series, 89,899 images 2D+T time series Input tensor: [B, 3, T, H, W]
Testing, external	Real-time cine	Heart	Data acquisition with single-shot free-breathing FOV: 360 × 270mm^2 Acquired matrix size: 192 × 110 Echo time: 0.98 msec Echo spacing: 2.27 msec Bandwidth: 1100 Hz/pixel Readout: BSSFP	1.5T	*n* = 10 patients, one slice per patient, 8 males, mean age 52 years 2D+T time series Input tensor: [B, 3, T, H, W]

			RF Flip angle: 50° Imaging duration: 39 msec Acceleration: R = 5		
	Perfusion	Heart	Data acquisition with single-shot free-breathing Contrast injection and dynamic contrast changes Adenosine stress FOV: 360 × 270mm^2 Acquired matrix size: 256 × 108 Echo time: 1.17 msec Single-shot TR: 80 msec Bandwidth: 850 Hz/pixel Readout: BSSFP RF Flip angle: 50° Acceleration: R = 4	1.5T	*n* = 5 patients, each had a stress and a rest scan, 3 slices per scan with 60 heart beats, 2 males, mean age 43 years 2D+T time series Input tensor: [B, 3, T, H, W]
	Neuro	Brain	T1 MPRAGE sequence FOV: 250 × 250mm^2 Acquired matrix size: 256 × 256 Echo time: 7.2 msec Bandwidth: 250 Hz/pixel Readout: Turbo spin echo	1.5T	*n* = 1 male, 45 years old 3D imaging Input tensor: [B, 3, D, H, W]

			Echo spacing: 3.58 msec TI: 200 msec Acceleration: R = 2 × 2		
	Spine	Spine	T2 TSE sequence FOV: 340 × 340mm^2 Acquired matrix size: 448 × 448 Echo time: 89 msec TR: 3000 msec Bandwidth: 260 Hz/pixel Readout: Turbo spin echo Acceleration: R = 2	1.5T	*n* = 1 male, 45 years old 2D imaging for 15 slices Input tensor: [B, 3, SLC, H, W]

Note.—FOV = field of view, SSFP = Steady-state Free Precession, RF = radiofrequency, MPRAGE = Magnetization-Prepared Rapid Gradient Echo, TSE = Turbo Spin Echo.

Table 2: Results of internal tests for two backbone types

HRnet	Number Of Parameters	Structural Similarity Index (SSIM)				Peak Signal-to-noise Ratio (PSNR)			
		Proposed	Without g-factor	Without MR noise	Without recon knowledge	Proposed	Without g-factor	Without MR noise	Without recon knowledge
CNNT-large	54,678,306	0.70	0.56	0.37	0.38	54.90	48.14	41.31	40.48
CNNT	27,485,139	0.68	0.58	0.38	0.38	54.14	49.19	41.51	40.38
Swin3D	54,664,836	0.68	0.59	0.40	0.40	53.78	49.54	42.53	41.23
ViT3D	27,478,404	0.63	0.60	0.55	0.48	51.75	49.89	47.91	44.00

Conv3D	22,815,891	0.60	0.57	0.57	0.46	50.74	49.06	48.73	41.52
ViT2D	17,746,308	0.48	—	—	—	46.96	—	—	—
Conv2D	20,382,867	0.47	—	—	—	46.67	—	—	—
Unet	Number of parameters	Structural Similarity Index (SSIM)				Peak signal-to-noise ratio (PSNR)			
		Proposed	Without g-factor	Without MR noise	Without recon knowledge	Proposed	Without g-factor	Without MR noise	Without recon knowledge
CNNT-large	48,880,418	0.70	0.55	0.38	0.37	54.70	47.65	41.53	40.32
CNNT	25,226,195	0.67	0.57	0.38	0.38	54.09	48.93	41.51	40.49
Swin3D	49,309,316	0.63	0.48	0.51	0.40	51.59	45.23	46.47	41.22
ViT3D	25,661,828	0.62	0.60	0.58	0.48	51.25	50.49	49.31	44.25
Conv3D	18,787,475	0.62	0.59	0.49	0.44	51.79	50.02	46.15	40.96
ViT2D	15,487,364	0.47	—	—	—	46.53	—	—	—
Conv2D	16,206,995	0.50	—	—	—	47.43	—	—	—

Note.—Proposed: training with g-factor map augmentation, realistic noise with the signal-noise-ratio (SNR) unit scaling; Without g-factor: training and inference without inputting geometry-factor (g-factor) maps; Without MR noise: training with white noise, but still adding g-factor maps; Without recon knowledge: training without g-factor maps and adding white noise, the SNR unit scaling was not used. $P < .001$ for proposed method against three ablation tests for CNNT-large, CNNT, Swin3D, ViT3D and Conv3D. SSIM = Structural Similarity Index, PSNR = peak signal-to-noise ratio, CNNT = Convolutional neural network transformer, SWIN = shifted window transformer, ViT = vision transformer, Conv = convolution.

Table 3: Results for real-time cine and perfusion generalization tests

Real-time Cine, R = 5						
Measurements	SNR				CNR	
ROIs	Blood Pool G-factor 4.05 ± 1.04		Myocardium G-factor 3.95 ± 0.80		Blood Pool And Myocardium	
Proposed	70.04 ± 11.70		20.40 ± 3.22		49.65 ± 9.80	
Raw	13.47 ± 4.85	*P* value: < 0.001	5.81 ± 2.20	< 0.001	7.67 ± 2.98	*P* value: < 0.001
Without G-factor	22.19 ± 7.88	< 0.001	7.39 ± 2.70	< 0.001	14.80 ± 5.49	< 0.001
Without MR noise	19.33 ± 6.78	< 0.001	8.09 ± 2.91	< 0.001	11.24 ± 4.18	< 0.001
Without recon knowledge	18.43 ± 5.98	< 0.001	7.64 ± 2.73	< 0.001	10.79 ± 3.81	< 0.001
Perfusion, R = 4						
Measurements	SNR				CNR	
ROIs	Blood pool G-factor 1.91 ± 0.52		Myocardium G-factor 1.87 ± 0.44		Blood pool and Myocardium	
Proposed	74.05 ± 26.32		16.69 ± 5.86		57.37 ± 24.63	
Raw	24.54 ± 14.81	*P* value: < 0.001	4.51 ± 2.69	< 0.001	20.03 ± 12.42	*P* value: <.001

Without G-factor	70.10 ± 31.54	0.01	12.50 ± 6.25	< 0.001	56.60 ± 27.28	0.84
Without MR noise	59.05 ± 32.05	< 0.001	9.96 ± 5.53	< 0.001	49.09 ± 27.50	0.001
Without recon knowledge	46.54 ± 23.58	< 0.001	8.98 ± 5.21	< 0.001	37.56 ± 19.43	< 0.001

Note.—Statistical significance tests were the proposed method against the raw and three ablation tests. The alpha level for significance is adjusted to be 0.05/4 = 0.0125 to count for four tests. The reported format is mean ± SD. Paired *t* test was performed between proposed method and three ablations. SNR = signal to noise ratio, CNR = contrast to noise ratio, ROI = region of interest.

Radiology: Artificial Intelligence

RSNA

SNRAware: Improved Deep Learning MRI Denoising with Signal-to-Noise Ratio Unit Training and G-factor Map Augmentation

Key Result

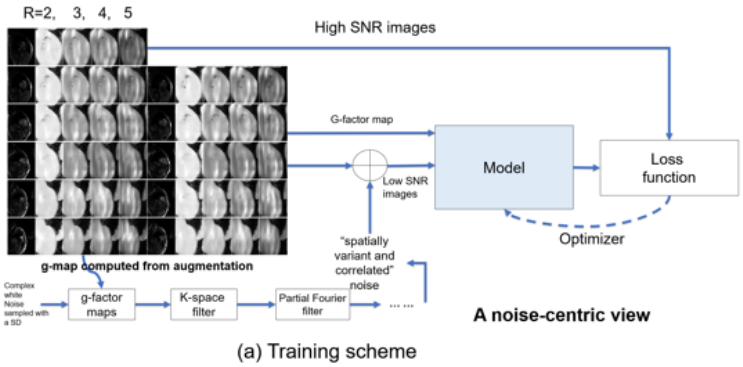
SNRAware, a model-agnostic approach for training MRI denoising models that leverages information from the image reconstruction process, improved performance and enhanced generalization to unseen imaging applications.

Methods:

- 14 model architectures were trained on an extensive dataset (2885236 images from 96605 cardiac MRI cine series), and ablation experiments were conducted to assess the impact of g-factor augmentation, realistic MRI noise, and SNR-based training.
- Model generalization was assessed across a range of imaging contrasts, sequences, field strengths, and anatomies.

Results:

- The proposed SNRAware training scheme leveraged MRI reconstruction knowledge to enhance denoising by simulating diverse synthetic datasets and providing quantitative noise distribution information.
- SNRAware improved performance in internal testing on a hold-out dataset of 3000 cine series and enabled strong generalization.



(a) Training scheme

Model trained with only cine data denoised cardiac perfusion MRI, demonstrating the strong generalization of SNRAware training.

Xue H et al. Published Online: October 22, 2025
https://doi.org/10.1148/ryai.250227

Radiology: Artificial Intelligence