RelativeFlow: Taming Medical Image Denoising Learning with Noisy Reference

Supplementary Materials

Yuxin Liu[1] Yiqing Dong[4] Wenxue Yu[1] Zhan Wu[1] Rongjun Ge[3] Yang Chen[1*] Yuting He[2*]

[1]School of Computer Science and Engineering, Southeast University, China
[2]Department of Biomedical Engineering, Case Western Reserve University, USA
[3]School of Instrument Science and Engineering, Southeast University, China
[4]Department of Optical Sciences, University of Arizona, USA

Contents

A. Mathematical Derivations of RelativeFlow Framework
A.1 CoT Component Property
A.2 CoT Composition Property
A.3 Simulation-based Velocity Field Construction
A.3.1 Velocity Field from CoT Flow
A.3.2 Expressing Velocity in Terms of Observable Pairs
B. Experimental Details
B.1 Evaluation Metrics
B.2 Baseline Methods
B.3 RelativeFlow Configuration
C. Dataset Details
C.1 Datasets for CT and MR Denoising
C.2 Degradation Operators and Simulation Parameters
D. Additional Denoising Process Visualization

1. Mathematical Derivations of RelativeFlow

This section provides the detailed mathematical derivations for the RelativeFlow framework. We prove the two key properties of Consistent Transport (CoT): the component property (Section 1.1) showing that relative flows are components of the absolute flow, and the composition property (Section 1.2) showing that relative flows progressively compose into the absolute flow. We then derive the Simulation-based Velocity Field (SVF) construction (Section 1.3) that enables practical implementation using only observable noisy reference pairs.

*Corresponding authors.

1.1. CoT Component Property

Theorem 1 (Component Property): Any distribution in a relative flow between p_{t_i} and p_{t_j} is also a component of the absolute flow between p_0 and $p_{+\infty}$.

Proof:

Consider arbitrary quality levels $0 < t_i < t < t_j < +\infty$. Both endpoints lie on the absolute flow path:

$$p_{t_i} = e^{-t_i} p_0 + (1 - e^{-t_i}) p_{+\infty} \tag{1}$$

$$p_{t_j} = e^{-t_j} p_0 + (1 - e^{-t_j}) p_{+\infty} \tag{2}$$

According to the CoT probability path definition (Eq. (3) in the main paper), any intermediate distribution p_t for $t \in (t_i, t_j)$ is given by:

$$p_t = \lambda\, p_{t_i} + (1 - \lambda)\, p_{t_j}, \quad \lambda = \frac{e^{-t} - e^{-t_j}}{e^{-t_i} - e^{-t_j}} \tag{3}$$

Substituting the expressions for p_{t_i} and p_{t_j}:

$$\begin{aligned} p_t &= \lambda \left(e^{-t_i} p_0 + (1 - e^{-t_i}) p_{+\infty}\right) \\ &\quad + (1 - \lambda) \left(e^{-t_j} p_0 + (1 - e^{-t_j}) p_{+\infty}\right) \\ &= \left(\lambda e^{-t_i} + (1 - \lambda) e^{-t_j}\right) p_0 \\ &\quad + \left(\lambda (1 - e^{-t_i}) + (1 - \lambda)(1 - e^{-t_j})\right) p_{+\infty} \end{aligned} \tag{4}$$

Computing the coefficient of p_0:

$$\begin{aligned} &\lambda e^{-t_i} + (1 - \lambda) e^{-t_j} \\ &= \frac{e^{-t} - e^{-t_j}}{e^{-t_i} - e^{-t_j}} e^{-t_i} + \frac{e^{-t_i} - e^{-t}}{e^{-t_i} - e^{-t_j}} e^{-t_j} \\ &= \frac{e^{-t_i} e^{-t} - e^{-t_i} e^{-t_j} + e^{-t_i} e^{-t_j} - e^{-t} e^{-t_j}}{e^{-t_i} - e^{-t_j}} \\ &= \frac{e^{-t}(e^{-t_i} - e^{-t_j})}{e^{-t_i} - e^{-t_j}} = e^{-t} \end{aligned} \tag{5}$$

Similarly, the coefficient of $p_{+\infty}$ equals:

$$\lambda(1-e^{-t_i})+(1-\lambda)(1-e^{-t_j})=1-e^{-t} \tag{6}$$

Therefore:

$$p_t = e^{-t}p_0 + (1-e^{-t})p_{+\infty} \tag{7}$$

This exactly matches the absolute flow formula, confirming that any distribution in a relative flow is a component of the absolute flow. □

Implication: This property ensures that training on diverse noisy references $\{x^i\}_{i=1}^N$ at varying quality levels $\{t_i\}_{i=1}^N$ constructs local relative flows that all lie on the same unified absolute flow path, preventing inconsistencies between flows learned from different reference quality levels.

1.2. CoT Composition Property

Theorem 2 (Composition Property): For any three quality levels $0 < t_1 < t_2 < t_3$, composing flows $\psi_{t_1\to t_2}$ and $\psi_{t_2\to t_3}$ yields the direct flow $\psi_{t_1\to t_3}$, i.e., $\psi_{t_1\to t_3} = \psi_{t_2\to t_3}\circ\psi_{t_1\to t_2}$.

Proof: We prove this by showing that the probability paths are consistent.

Step 1: Express p_{t_2} from flow $\psi_{t_1\to t_2}$. For the relative flow from t_1 to t_2, any distribution p_t at time $t\in[t_1,t_2]$ follows:

$$p_t = \lambda_{12}p_{t_1} + (1-\lambda_{12})p_{t_2}, \quad \lambda_{12} = \frac{e^{-t}-e^{-t_2}}{e^{-t_1}-e^{-t_2}}. \tag{8}$$

Rearranging to solve for p_{t_2}:

$$\begin{aligned} p_{t_2} &= \frac{p_t - \lambda_{12}p_{t_1}}{1-\lambda_{12}} \\ &= \frac{p_t - \frac{e^{-t}-e^{-t_2}}{e^{-t_1}-e^{-t_2}}p_{t_1}}{1-\frac{e^{-t}-e^{-t_2}}{e^{-t_1}-e^{-t_2}}} \\ &= \frac{e^{-t_1}-e^{-t_2}}{e^{-t_1}-e^{-t}}p_t - \frac{e^{-t}-e^{-t_2}}{e^{-t_1}-e^{-t}}p_{t_1}. \end{aligned} \tag{9}$$

Step 2: Express p_{t_2} from flow $\psi_{t_2\to t_3}$. For the relative flow from t_2 to t_3, we have:

$$p_t = \lambda_{23}p_{t_2} + (1-\lambda_{23})p_{t_3}, \quad \lambda_{23} = \frac{e^{-t}-e^{-t_3}}{e^{-t_2}-e^{-t_3}}. \tag{10}$$

Rearranging to solve for p_{t_2}:

$$\begin{aligned} p_{t_2} &= \frac{p_t - (1-\lambda_{23})p_{t_3}}{\lambda_{23}} \\ &= \frac{p_t - \left(1-\frac{e^{-t}-e^{-t_3}}{e^{-t_2}-e^{-t_3}}\right)p_{t_3}}{\frac{e^{-t}-e^{-t_3}}{e^{-t_2}-e^{-t_3}}} \\ &= \frac{e^{-t_2}-e^{-t_3}}{e^{-t}-e^{-t_3}}p_t - \frac{e^{-t_2}-e^{-t}}{e^{-t}-e^{-t_3}}p_{t_3}. \end{aligned} \tag{11}$$

Step 3: Eliminate p_{t_2} to obtain the composed flow. Equating Eq. (9) and Eq. (11):

$$\begin{aligned} &\frac{e^{-t_1}-e^{-t_2}}{e^{-t_1}-e^{-t}}p_t - \frac{e^{-t}-e^{-t_2}}{e^{-t_1}-e^{-t}}p_{t_1} \\ &\quad = \frac{e^{-t_2}-e^{-t_3}}{e^{-t}-e^{-t_3}}p_t - \frac{e^{-t_2}-e^{-t}}{e^{-t}-e^{-t_3}}p_{t_3}. \end{aligned} \tag{12}$$

Collecting terms with p_t:

$$\begin{aligned} &\left(\frac{e^{-t_1}-e^{-t_2}}{e^{-t_1}-e^{-t}} - \frac{e^{-t_2}-e^{-t_3}}{e^{-t}-e^{-t_3}}\right)p_t \\ &\quad = \frac{e^{-t}-e^{-t_2}}{e^{-t_1}-e^{-t}}p_{t_1} - \frac{e^{-t_2}-e^{-t}}{e^{-t}-e^{-t_3}}p_{t_3}. \end{aligned} \tag{13}$$

After algebraic simplification (omitted for brevity), we obtain:

$$p_t = \frac{e^{-t}-e^{-t_3}}{e^{-t_1}-e^{-t_3}}p_{t_1} + \frac{e^{-t_1}-e^{-t}}{e^{-t_1}-e^{-t_3}}p_{t_3}. \tag{14}$$

This exactly matches the CoT probability path for the direct flow from t_1 to t_3 with $\lambda_{13} = \frac{e^{-t}-e^{-t_3}}{e^{-t_1}-e^{-t_3}}$, confirming the composition property. □

Implication: This property guarantees that sequences of relative flows learned from heterogeneous noisy references can be composed during inference to construct the complete absolute flow, enabling the model to denoise from arbitrary input quality levels to a unified high-quality target.

1.3. Simulation-based Velocity Field Construction

This subsection provides the detailed mathematical derivation showing how the velocity field can be constructed from observable pairs $(x_{t-\Delta t}, x_t)$ without requiring access to clean endpoints or absolute time labels. This is crucial for practical implementation since we only have access to noisy references at unknown quality levels.

1.3.1. Velocity Field from CoT Flow

Following the CoT-defined flow matching framework, let ψ_t denote the flow map that transports from the noise endpoint at $t=0$ to quality level t. For a sample x_0 drawn from the noise distribution p_0, the CoT path yields:

$$\psi_t(x_0) = e^{-t}x_0 + (1-e^{-t})x_\infty \tag{15}$$

where x_∞ represents the clean endpoint as $t\to+\infty$.

Note on notation: In the main text, we use x_1 to denote the clean target in the context of SimSGL methods for consistency with existing literature. Here in the supplementary, we use x_∞ to emphasize that the clean endpoint is the limiting distribution as $t\to+\infty$ in our continuous-time formulation.

The velocity field is defined as the time derivative of the flow:

$$u_t\big(\psi_t(x_0)\,|\,x_0, x_\infty\big) = \frac{d}{dt}\,\psi_t(x_0)$$
$$= -e^{-t}x_0 + e^{-t}x_\infty \tag{16}$$
$$= e^{-t}(x_\infty - x_0). \tag{17}$$

1.3.2. Expressing Velocity in Terms of Observable Pairs

In practice, we only observe a noisy reference x_t at arbitrary quality level t and its degraded version $x_{t-\Delta t} = D_{\Delta t}(x_t)$ at an earlier quality level. Both samples lie on the same absolute flow path, so they can be expressed using Eq. (15):

$$x_t = e^{-t}x_0 + (1 - e^{-t})x_\infty \tag{18}$$
$$x_{t-\Delta t} = e^{-(t-\Delta t)}x_0 + (1 - e^{-(t-\Delta t)})x_\infty \tag{19}$$

Subtracting the second equation from the first:

$$\begin{aligned} x_t - x_{t-\Delta t} &= \left(e^{-t} - e^{-(t-\Delta t)}\right) x_0 \\ &\quad + \left((1 - e^{-t}) - (1 - e^{-(t-\Delta t)})\right) x_\infty \\ &= \left(e^{-t} - e^{-t}e^{\Delta t}\right) x_0 + \left(e^{-t}e^{\Delta t} - e^{-t}\right) x_\infty \\ &= e^{-t}(1 - e^{\Delta t})x_0 + e^{-t}(e^{\Delta t} - 1)x_\infty \\ &= e^{-t}(e^{\Delta t} - 1)(x_\infty - x_0) \end{aligned} \tag{20}$$

Solving for $(x_\infty - x_0)$:

$$x_\infty - x_0 = \frac{x_t - x_{t-\Delta t}}{e^{-t}(e^{\Delta t} - 1)} = \frac{x_t - x_{t-\Delta t}}{e^{-(t-\Delta t)} - e^{-t}} \tag{21}$$

Substituting this into the velocity field expression from Eq. (17):

$$\begin{aligned} u_t(x_t) &= e^{-t}(x_\infty - x_0) \\ &= e^{-t} \cdot \frac{x_t - x_{t-\Delta t}}{e^{-t}(e^{\Delta t} - 1)} \\ &= \frac{x_t - x_{t-\Delta t}}{e^{\Delta t} - 1} \end{aligned} \tag{22}$$

Note that by the CoT component property (Appendix 1.1), the relative flow between $x_{t-\Delta t}$ and x_t follows the same CoT form as the absolute flow. Therefore, the above velocity $u_t(x_t)$ at the absolute quality level t corresponds to the velocity field in the relative flow with parameter Δt. For notational clarity in the training objective, we rewrite this velocity as a function of the step size Δt and the observable pair:

$$u_{\Delta t}(x_{t-\Delta t}|x_t) = \frac{x_t - x_{t-\Delta t}}{e^{\Delta t} - 1} \tag{23}$$

This expression matches the supervision target in the training loss (Eq. (6) in the main paper), showing that the velocity field can be computed directly from the observable degradation pair $(x_{t-\Delta t}, x_t)$ without requiring knowledge of the absolute endpoints (x_0, x_∞) or the absolute quality level t.

2. Experimental Details

2.1. Evaluation Metrics

We evaluate denoising performance using four standard metrics following the main text. Images are normalized to $[0, 1]$ range for computing PSNR, SSIM, and RMSE, and to 3-channel $[0, 255]$ format for LPIPS computation. For clearer numerical comparison, SSIM, RMSE, and LPIPS are reported as their raw scores multiplied by 100.

Peak Signal-to-Noise Ratio (PSNR):

$$\text{PSNR} = 10 \log_{10} \frac{1}{\text{MSE}} \tag{24}$$

where $\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$ is the mean squared error between the denoised image x and ground truth y in $[0, 1]$ range, and N is the total number of pixels.

Structural Similarity Index Measure (SSIM):

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{25}$$

where μ_x, μ_y are local means, σ_x^2, σ_y^2 are local variances, σ_{xy} is local covariance, and $c_1 = (0.01)^2$, $c_2 = (0.03)^2$ are stabilizing constants.

Normalized Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2} \tag{26}$$

This is computed on $[0, 1]$ normalized images. Lower values indicate better performance.

Learned Perceptual Image Patch Similarity (LPIPS):

$$\text{LPIPS}(x, y) = \sum_l w_l \cdot \|\phi_l(x) - \phi_l(y)\|_2^2 \tag{27}$$

where ϕ_l represents features from layer l of a pretrained VGG network, and w_l are learned weights. Images are converted to 3-channel $[0, 255]$ format before computing LPIPS. Lower values indicate better perceptual similarity.

2.2. Baseline Methods

We compare RelativeFlow with 10 baseline methods from three categories under the unified settings summarized in Table 2.

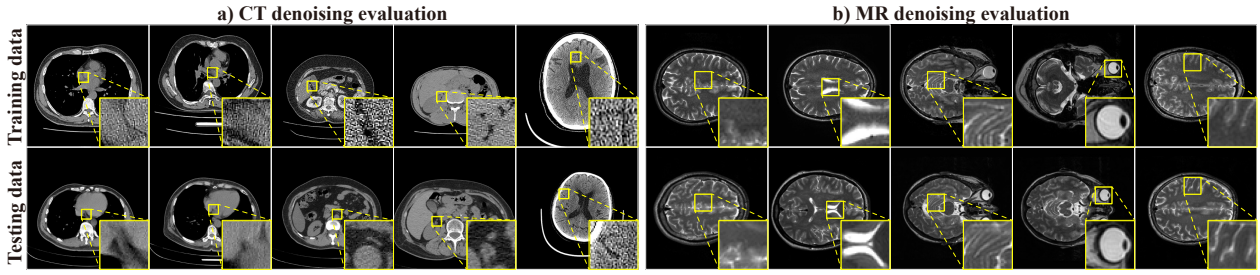

Figure 1. Visual comparison of training and testing data quality distributions for CT (left) and MR (right) datasets. **Top row:** Training data exhibit heterogeneous quality levels with varying degrees of degradation across different acquisition protocols, exemplifying the noisy reference problem. **Bottom row:** Testing data were manually selected to include only the highest-quality reference images for reliable evaluation.

Table 1. RelativeFlow-specific configuration.

Parameter	Value
Initial $\Delta t_{\min}$	0.2
Initial $\Delta t_{\max}$	0.2
Decay factor α	0.9
Sampling steps $\{\Delta t_i\}$	[0.2, 0.1, 0.05]
Architecture	2D Guided U-Net [4]
Loss function	L2 loss (MSE)

Table 2. Unified training configuration for all baseline methods.

Category	Parameter	Setting
Training	Loss function	L2 loss
	Optimizer	Adam
	Learning rate	10^{-4}
	Optimizer betas	$\beta_1 = 0.9, \beta_2 = 0.999$
	Training epochs	30
	Inference steps (SimSGL)	3
Hardware	GPUs	8 × NVIDIA V100 (32 GB)
Architecture	SimSDL/SSL	2D U-Net [9]
	SimSGL	2D Guided U-Net [4]

2.3. RelativeFlow Configuration

RelativeFlow-specific hyperparameters:

For SimSDL methods, SwinIR [5] replaces standard convolution with window-based self-attention and shifted window mechanism for hierarchical feature learning. RED-CNN [2] uses symmetric encoder-decoder architecture with shortcut connections between corresponding layers. CNN-DMRI [11] incorporates k-space consistency enforcement to ensure MR reconstruction matches observed measurements.

For SSL methods, Noise2Self [1] implements blind-spot networks where each pixel is predicted from spatially disjoint context via donut masking. Noise2Sim [8] exploits patch redundancy by identifying and averaging structurally similar patches within single images. Deformed2Self [12] enforces deformation consistency through spatial transformer networks.

For SimSGL methods, DDIM [10] uses non-Markovian diffusion process for deterministic sampling with $T = 1000$ training steps. Flow Matching [7] directly regresses velocity fields via optimal transport paths. IPDM [6] applies partial diffusion starting from noisy observations rather than pure noise. R2D2+ [3] performs posterior sampling with predictor-corrector strategy for inverse problems.

3. Dataset Details

3.1. Datasets for CT and MR Denoising

Table 3. Dataset statistics for training, testing, and validation splits (patients / slices).

Split	GBA-LDCT dataset	IXI dataset
Training	263 / 138,440	497 / 129,566
Testing	27 / 12,342	40 / 10,360
Validation	27 / 18,856	42 / 11,064

Both datasets naturally exhibit the noisy reference problem with heterogeneous quality levels across acquisition protocols, as illustrated in Figure 1. For the **training data** (top row), the GBA-LDCT dataset used in CT denoising evaluation contains images acquired from different manufacturers and anatomical regions with varying scanning protocols, including different tube voltages, tube currents, and reconstruction kernels, resulting in substantial quality variations across samples. Similarly, the IXI dataset used in MR denoising evaluation includes images collected from multiple medical centers using different manufacturers' scanners with varying field strengths, pulse sequences, and acquisition parameters, leading to heterogeneous image quality levels. These quality variations naturally present in clinical data exemplify the noisy reference problem addressed

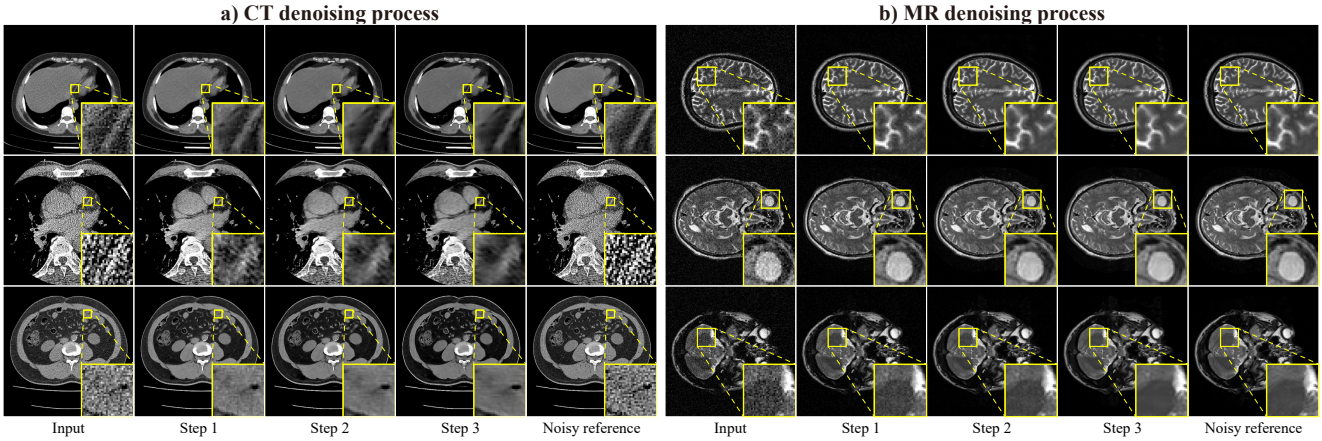

Figure 2. **Denoising process visualization for multiple quality levels.** Left: three CT examples. Right: three MR examples. Each row shows the denoising trajectory with five images: noisy input, three intermediate steps, and the reference. Yellow boxes indicate ROI with zoomed-in views. RelativeFlow achieves consistent high-quality outputs across different noise levels and modalities.

in our work.

In contrast, the **testing data** (bottom row) for both datasets were manually selected to include only the highest-quality reference images, ensuring reliable evaluation of denoising performance. The complete dataset statistics are summarized in Table 3.

3.2. Degradation Operators and Simulation Parameters

3.2.1. CT Degradation Operator

For CT images, the degradation operator models quantum noise and electronic noise following Poisson-Gaussian distribution. In the sinogram domain where x represents normalized measurements, we define:

$$D_{\Delta t}(x) = \frac{\text{Poisson}(\alpha(\Delta t) \cdot I_0 \cdot x) + \epsilon(\Delta t)}{\alpha(\Delta t) \cdot I_0} \tag{28}$$

where $\alpha(\Delta t) = \exp(-\lambda_{\text{CT}} \Delta t)$ models dose reduction, I_0 is the incident photon count, and $\epsilon(\Delta t) \sim \mathcal{N}(0, \sigma_e^2(\Delta t))$ represents electronic noise with $\sigma_e^2(\Delta t) = \beta_{\text{CT}} \Delta t$.

Parameters: $\lambda_{\text{CT}} = 5$, $\beta_{\text{CT}} = 400$, $I_0 = 10^6$ photons.

3.2.2. MR Degradation Operator

For MR images, the degradation operator models thermal noise and dielectric losses following Rician distribution in the magnitude domain:

$$D_{\Delta t}(x) = \sqrt{(x + n_1(\Delta t))^2 + n_2^2(\Delta t)} \tag{29}$$

where $n_1(\Delta t), n_2(\Delta t) \sim \mathcal{N}(0, \sigma^2(\Delta t))$ are independent Gaussian noise components with variance $\sigma^2(\Delta t) = \gamma_{\text{MR}} \Delta t$.

Parameters: $\gamma_{\text{MR}} = 0.005$.

4. Denoising Process Visualization

To further demonstrate the effectiveness of RelativeFlow, we provide additional visualizations of the iterative denoising process on both CT and MR images with varying noise levels. As shown in Figure 2, for inputs with different noise levels, RelativeFlow progressively improves the image quality through three intermediate denoising steps, ultimately achieving consistently high-quality results that closely match the reference images. Each row displays the complete denoising trajectory: from the noisy input (left), through three successive refinement steps, to the final high-quality reference (right). The yellow-highlighted regions of interest (ROI) and their zoomed-in views clearly illustrate the progressive noise reduction and detail preservation across different modalities and quality levels.

References

[1] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International Conference on Machine Learning*, pages 524–533. PMLR, 2019. 4

[2] Hu Chen, Yi Zhang, Mannudeep K Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. Low-dose ct with a residual encoder-decoder convolutional neural network. *IEEE Transactions on Medical Imaging*, 36(12):2524–2535, 2017. 4

[3] Hyungjin Chung, Eunha Lee, and Jong Chul Ye. Mr image denoising and super-resolution using regularized reverse diffusion. *IEEE Transactions on Medical Imaging*, 42(4): 922–934, 2023. 4

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, pages 8780–8794, 2021. 4

[5] Jingyun Liang, Jie Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration

using swin transformer. In *IEEE/CVF International Conference on Computer Vision Workshops*, pages 1833–1844, 2021. 4

[6] Feiyang Liao, Yufei Tang, Qiang Du, Jiping Wang, Ming Li, and Jian Zheng. Domain progressive low-dose ct imaging using iterative partial diffusion model. *IEEE Transactions on Medical Imaging*, 43(12):4449–4461, 2024. 4

[7] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 4

[8] Chuang Niu, Mengzhou Li, Fenglei Fan, Weiwen Wu, Xiaodong Guo, Qing Lyu, and Ge Wang. Noise suppression with similarity-based self-supervised deep learning. *IEEE Transactions on Medical Imaging*, 41(11):3142–3152, 2022. 4

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[10] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 4

[11] Prabhat Chandra Tripathi and Soumya Bag. Cnn-dmri: A convolutional neural network for denoising of magnetic resonance images. *Pattern Recognition Letters*, 135:57–63, 2020. 4

[12] Junshen Xu and Elfar Adalsteinsson. Deformed2self: Self-supervised denoising for dynamic medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 25–35. Springer, 2021. 4