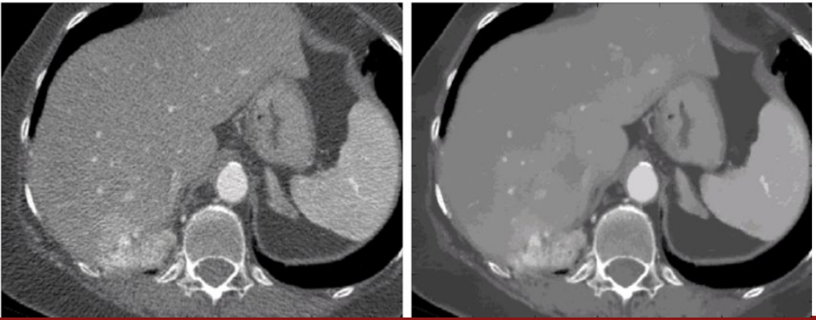
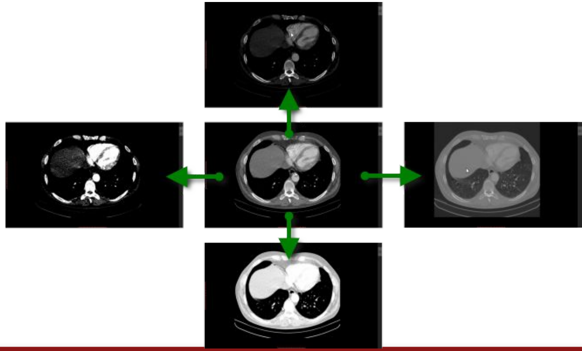
Window-Level is a Strong Denoising Surrogate

Ayaan Haque, Adam Wang, Abdullah-Al-Zubaer Imran





Overview

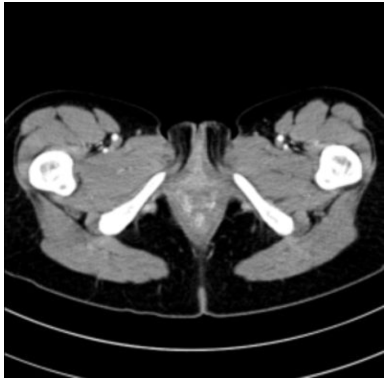
- Clinical and Computational Motivations
- Related Work
- Our Methods
- Results

Clinical Challenges and Motivation

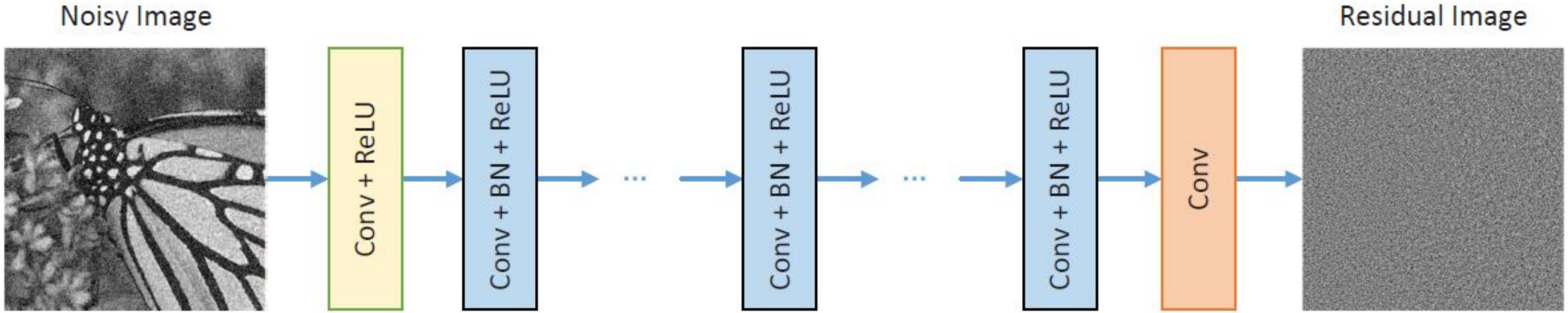
- High radiation dose is required for good quality
  - Using lower dose introduces noise
- High radiation is harmful, so using lower dose is desirable
  - This creates a tradeoff between noise and safety
- Using Deep Learning approaches can help "denoise" low dose scans

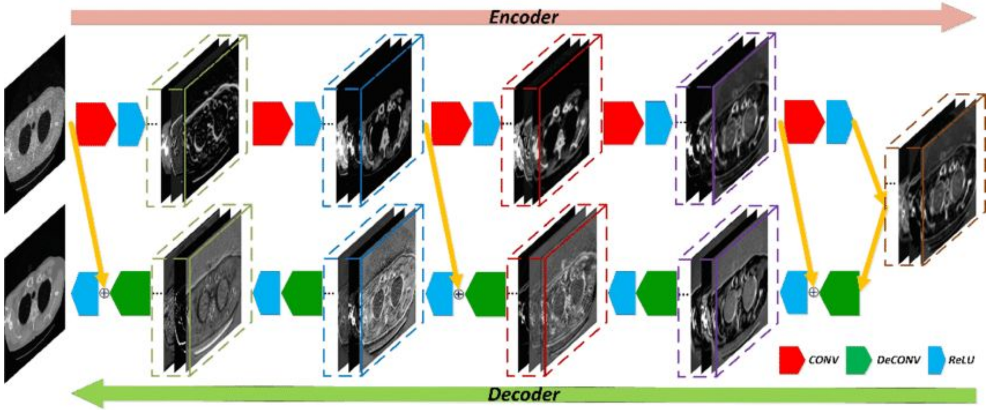
Low Dose

Full Dose



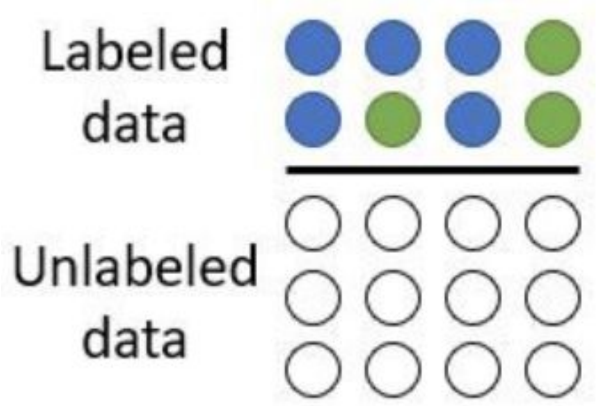
Denoising Methods





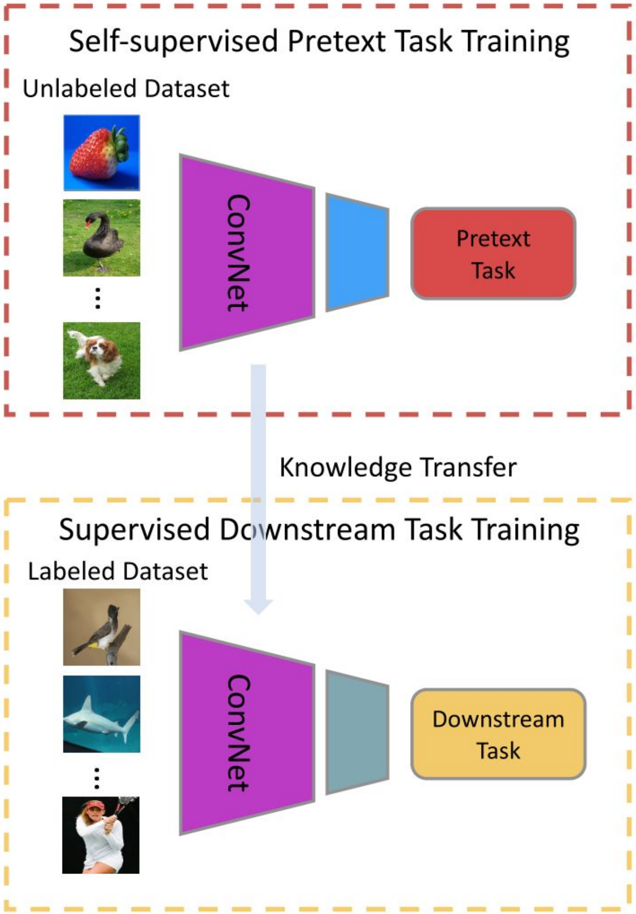
Limited Reference Data

- To train effective deep denoising models, lots of data is required
  - Specifically full-dose reference data
- Collecting additional full-dose scans is challenging
  - can't acquiring two doses of same scan
  - harmful nature of radiation
- Less challenging to have low dose images



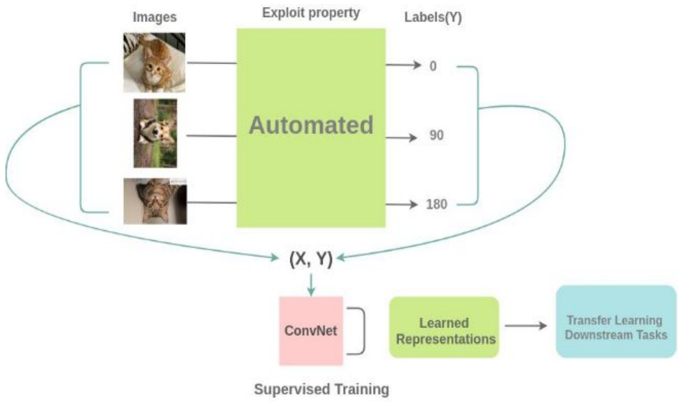
Self-Supervised Learning

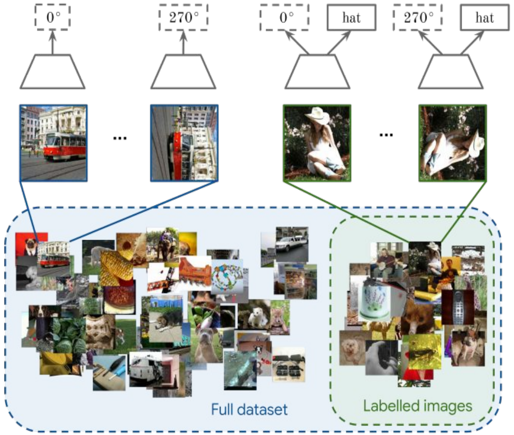
- Self-Supervised Learning is an alternative to to FSL
- In SSL, synthetic labels are generated from the data itself
  - Similar to transfer learning, as both use pre-training
- “Surrogate” and “downstream”
  - back-to-back
- Improved representation learning



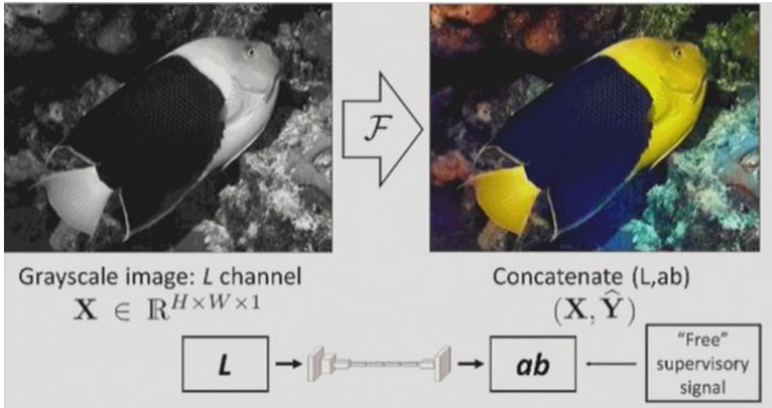
Self-Supervised Learning Cont.

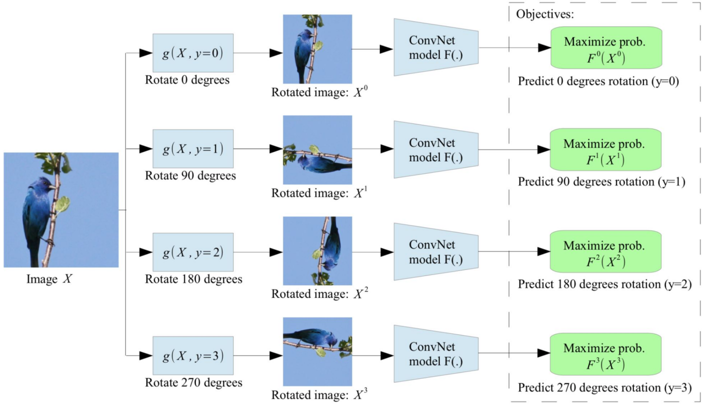
- In SSL, “free” labels are produced by exploiting the data itself
- SSL tasks are not very valuable themselves
  - however, the model learns the data distribution
  - during fine-tuning, it has additional knowledge
- Examples include
  - Rotation prediction
  - colorization/restoration
  - patch prediction

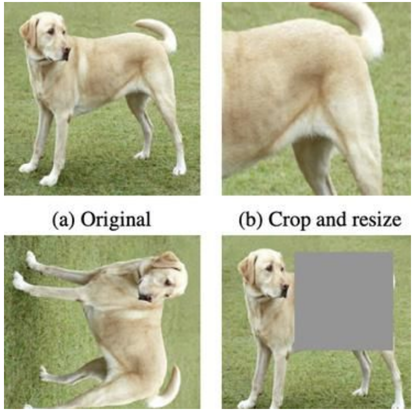




Common Self-Supervised Tasks

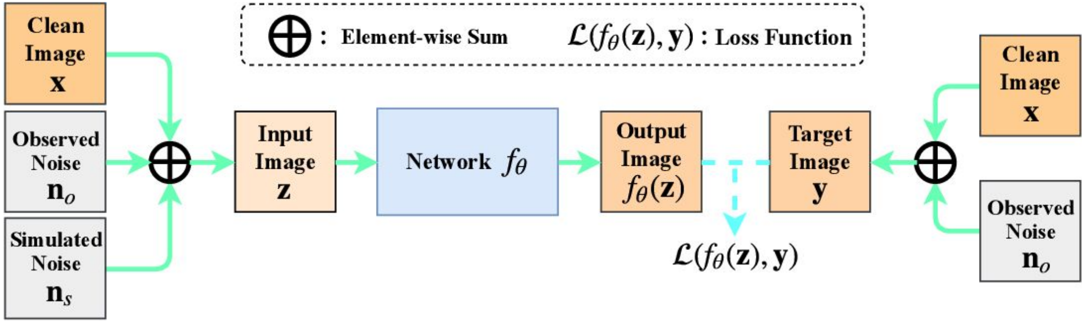


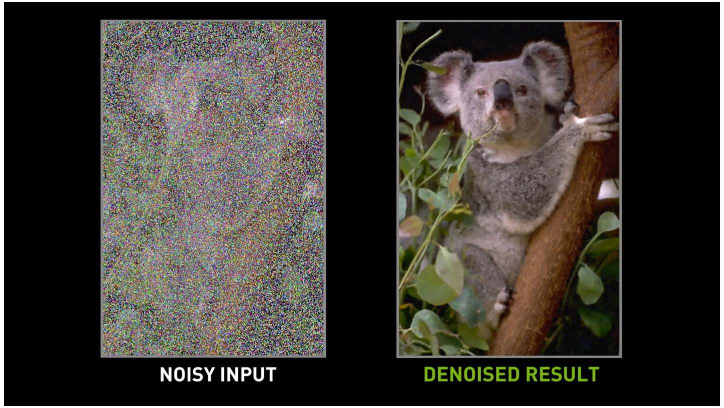




Current Self-Supervised Denoising Methods

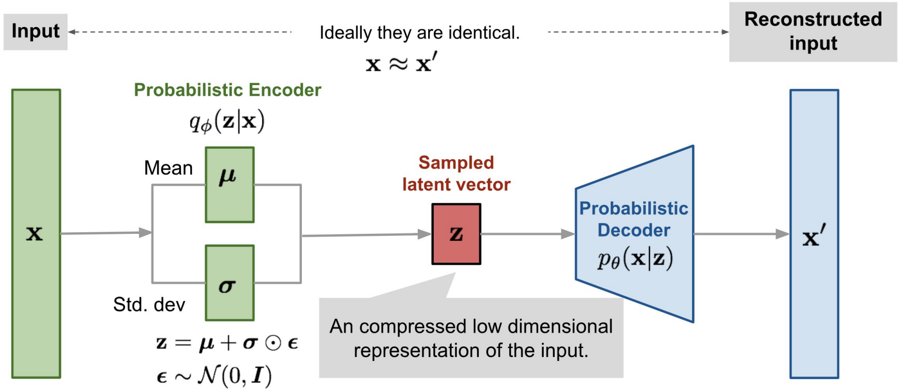
- General-Purpose
  - Self2Self
  - Noise2Same
  - **Noisy-As-Clean**
- CT-Denoising
  - Noise2Noise
  - Noise2Void
  - Half2Half

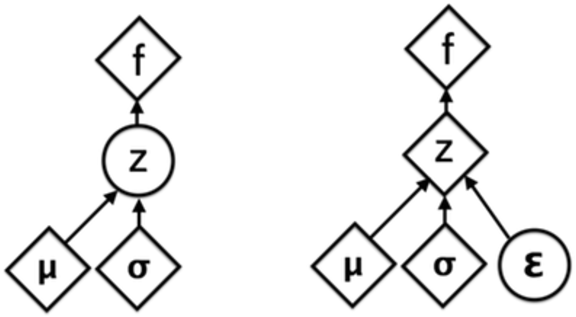




Variational Autoencoders

- Built on autoencoders
  - deconstruct and reconstruct an input the exact same
- Introduces the latent space
  - gaussian noise is introduced
  - Makes the VAE generative
- VAEs have been used in many applications
  - not many for denoising, none for medical image denoising



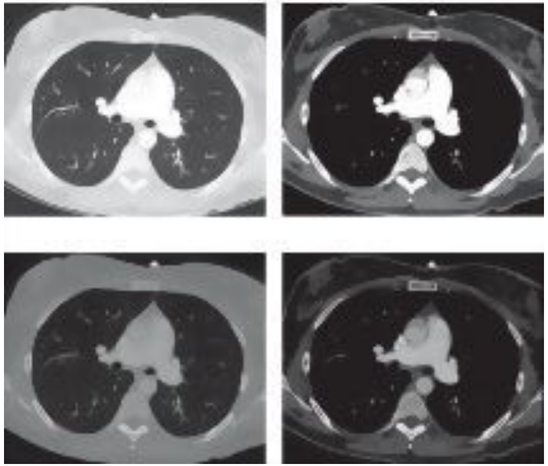


Original

Reparametrized

Window-Leveling

- Window-Leveling is the process of modifying the grayscale of a CT image using the CT numbers
  - purpose is to highlight, brighten, and contrast important structures
- From a deep learning perspective, the images are essentially transformed and recolored



Challenges

- Denoising with limited reference data
- Task-relevant and domain specific self-supervised surrogates
- New architectures that can be effective at very low dose
- Generalization to different domains

Contributions: SSWL-IDN

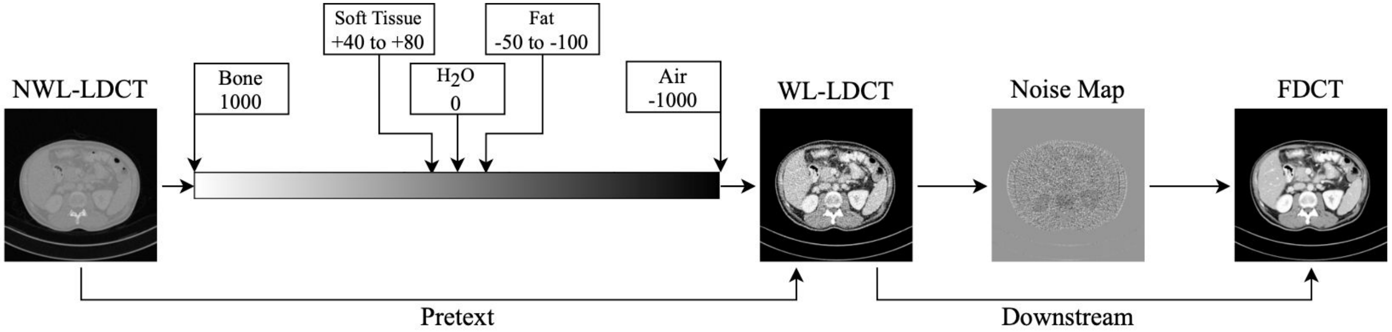
- Task-relevant self-supervised window-level prediction surrogate
- Related to the downstream task

- Residual-based VAE architecture
- Hybrid loss for pixel-wise and perceptual optimization

- Extensive experimentation
- Varied quantities of labeled data on different proposed components
- In- and cross-domain results
- Extremely low dose (5%)

Self-Supervised Window Leveling

- Window-Leveled Images are easily attainable labels
  - DICOM metadata has the width and center values
- NWL are used as inputs, and WL are used as targets
  - model is trained to transform NWL to WL
- This task can be likened to image recolorization
- This task itself is not very useful, but it promotes important feature and representation learning



Window Leveling and Denoising

- Denoising Equation
  - X is LDCT, Y is FDCT, n is noise map
  - train model to remove the noise map
- Window-Leveling Equation
  - X is NWL, Z is WL, a and b are constants
- The two tasks are similar from a computational perspective
  - they are both image transformations
  - other tasks are arbitrary so they may result in worse performance
- Domain-specific task
  - encourages important and relevant feature learning
- end-to-end surrogate

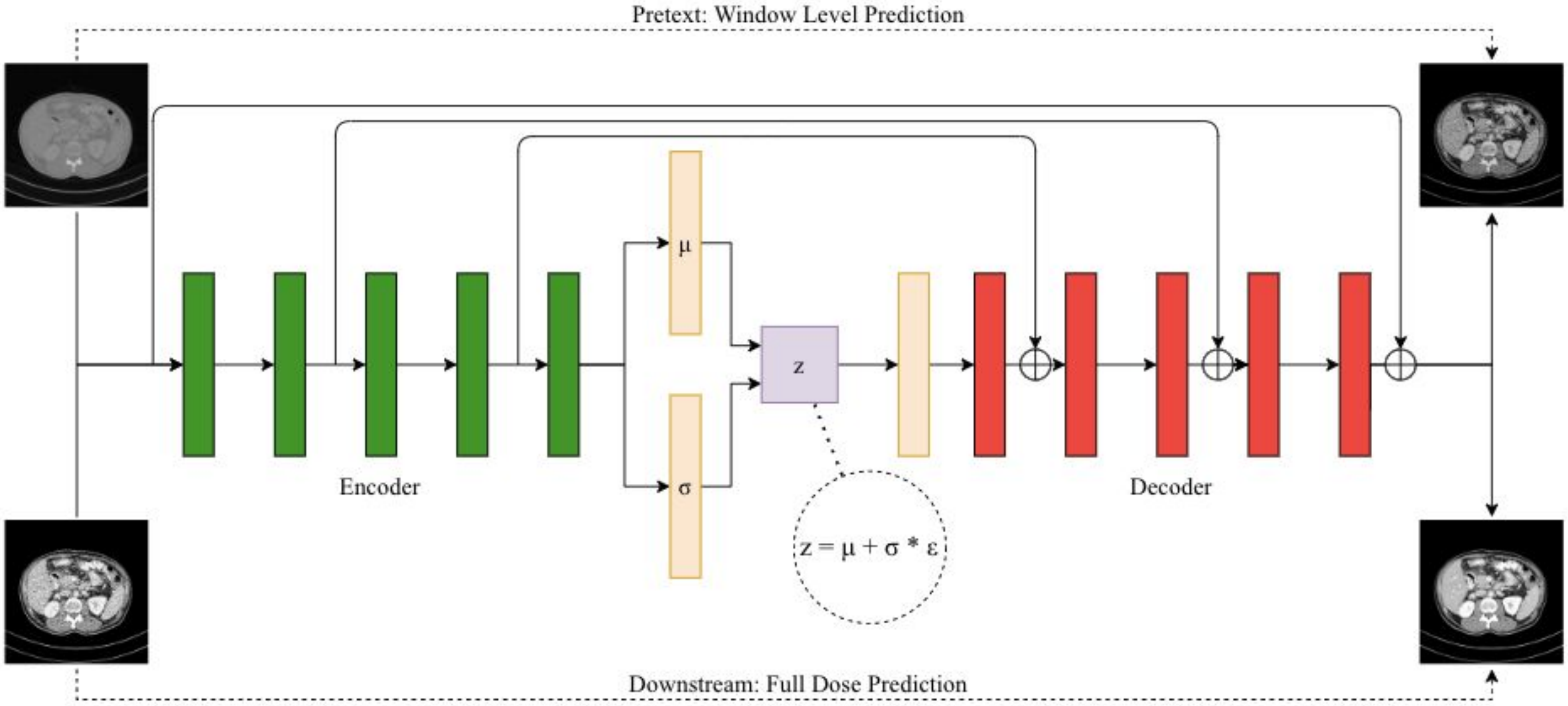
Denoising

$$X = Y + n$$

Window-Leveling

$$Z = aX + b$$

Residual-VAE



Architectural Table

Layer	Encoder		Layer	Decoder	
	Feature maps (input)	Feature maps (output)		Feature maps (input)	Feature maps (output)
Conv Layer - 1	$M \times 256 \times 256 \times 1$	$M \times 252 \times 252 \times 96$	Upsample	$M \times 4 \times 4 \times 96$	$M \times 236 \times 236 \times 96$
ReLU	$M \times 252 \times 252 \times 96$	$M \times 252 \times 252 \times 96$	Transpose Conv Layer - 1	$M \times 236 \times 236 \times 1$	$M \times 240 \times 240 \times 96$
Conv Layer - 2	$M \times 252 \times 252 \times 96$	$M \times 248 \times 248 \times 96$	ReLU	$M \times 240 \times 240 \times 96$	$M \times 240 \times 240 \times 96$
ReLU	$M \times 248 \times 248 \times 96$	$M \times 248 \times 248 \times 96$	Residual (Conv 4)	$M \times 240 \times 240 \times 96$	$M \times 240 \times 240 \times 96$
Conv Layer - 3	$M \times 248 \times 248 \times 96$	$M \times 244 \times 244 \times 96$	Transpose Conv Layer - 2	$M \times 240 \times 240 \times 96$	$M \times 244 \times 244 \times 96$
ReLU	$M \times 244 \times 244 \times 96$	$M \times 244 \times 244 \times 96$	ReLU	$M \times 244 \times 244 \times 96$	$M \times 244 \times 244 \times 96$
Conv Layer - 4	$M \times 244 \times 244 \times 96$	$M \times 240 \times 240 \times 96$	Transpose Conv Layer - 3	$M \times 244 \times 244 \times 96$	$M \times 248 \times 248 \times 96$
ReLU	$M \times 240 \times 240 \times 96$	$M \times 240 \times 240 \times 96$	ReLU	$M \times 248 \times 248 \times 96$	$M \times 248 \times 248 \times 96$
Conv Layer - 5	$M \times 240 \times 240 \times 96$	$M \times 240 \times 240 \times 96$	Residual (Conv 2)	$M \times 248 \times 248 \times 96$	$M \times 248 \times 248 \times 96$
ReLU	$M \times 236 \times 236 \times 96$	$M \times 236 \times 236 \times 96$	Transpose Conv Layer - 4	$M \times 248 \times 248 \times 96$	$M \times 252 \times 252 \times 96$
Global Average Pooling	$M \times 236 \times 236 \times 96$	$M \times 4 \times 4 \times 96$	ReLU	$M \times 252 \times 252 \times 96$	$M \times 252 \times 252 \times 96$
Linear - Mu	$M \times 4 \times 4 \times 96$	$M \times 1 \times 256$	Transpose Conv Layer - 5	$M \times 252 \times 252 \times 96$	$M \times 256 \times 256 \times 1$
Linear - Logvar	$M \times 4 \times 4 \times 96$	$M \times 1 \times 256$	Residual (Input)	$M \times 256 \times 256 \times 1$	$M \times 256 \times 256 \times 1$
Linear - Decoder	$M \times 1 \times 256$	$M \times 4 \times 4 \times 96$	ReLU	$M \times 256 \times 256 \times 1$	$M \times 256 \times 256 \times 1$

Hybrid Loss

- Hybrid Loss combining MSE and Perceptual Loss
- Perceptual loss encourages high-level feature matching
- MSE provides pixel-based accuracy
- KL Divergence loss for the VAE component of the model

Perceptual Loss

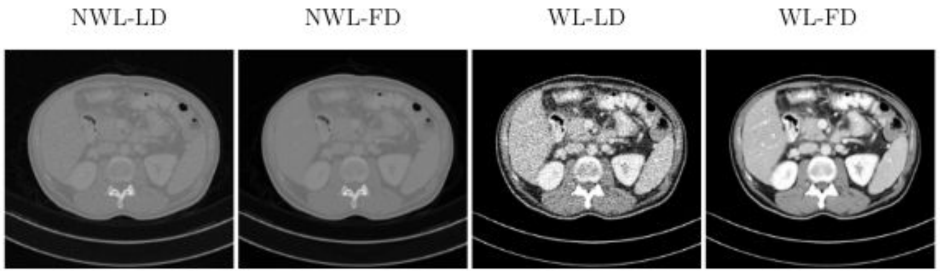
$$\mathcal{L}_{perceptual} = \frac{1}{M}\sum_{i=1}^{M}\left\|\mathcal{F}(\hat{y}_i) - \mathcal{F}(y_i)\right\|^2$$

Final Loss Objective

$$L(y, \hat{y}, \mu, \sigma) = L_{MSE}(\hat{y}, y) + \beta\mathcal{L}_{perceptual}(\mathcal{F}(\hat{y}), \mathcal{F}(y)) + \alpha\mathcal{L}_{KL}(\mu, \sigma)$$

Datasets: Overview

- We used the Mayo CT Low-Dose Dataset
  - Abdomen scans
  - simulated quarter dose scans
- 15 abdomen scans, 10 for training, 5 for testing



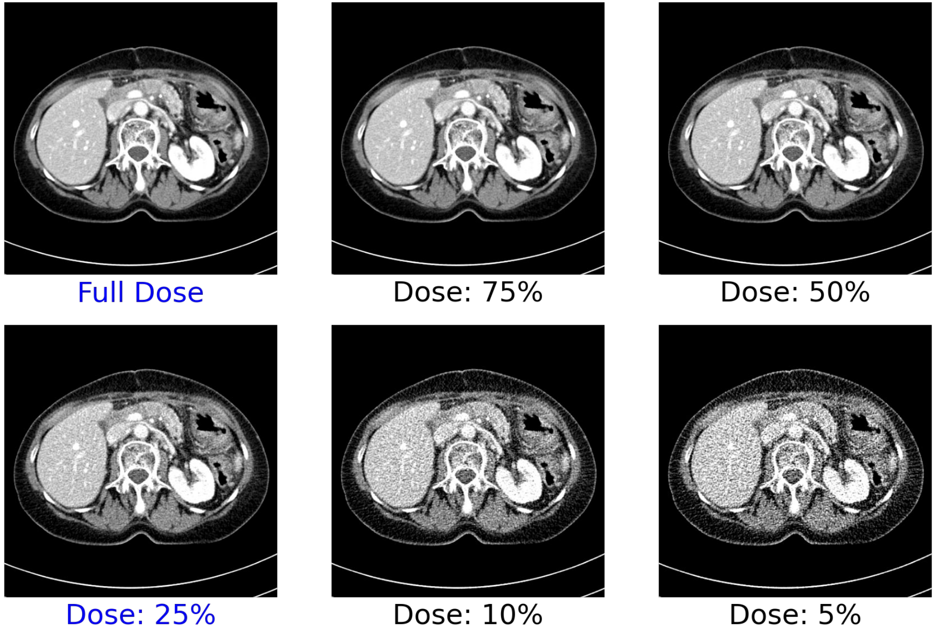
Dataset Details

mode	Dataset	Total	Train	Test
in-domain	Abdomen	2,196	1,533	663
cross-domain	Chest	1,061	–	1,061

Datasets: Dose

- We simulate 5% dose
- 5% dose is for thorough evaluation of the denoising model
  - Clinically, it is better to use 25% dose and receive better predictions
  - Computationally, aggressive denoising is desirable

Different Dose Levels



Implementation Details

- Training
  - PyTorch, NVIDIA K80 GPU 12 GB RAM
  - Varying levels of supervision
    - 250, 500, 1000, Full
  - Each experiment was performed 5 times, averages were reported
  - Inputs normalized and resized to 256x256x1
- Hyperparameters
  - Adam Optimizer, $\beta = 0.6$
- Evaluation
  - PSNR, SSIM, MSE, NRMSE
- Baselines
  - Architectures: AE, VAE, U-Net, DnCNN, RED-CNN
  - Loss: L1, MSE, Perceptual, L1-Smooth, L1+Perceptual
  - SSL Tasks: No pre-training, Reconstruction, Noisy-As-Clean

Results: Architecture Comparisons

Model	$\lvert\mathcal{D}_l\rvert$	In-Domain				Cross-Domain			
		PSNR	SSIM	MSE	NRMSE	PSNR	SSIM	MSE	NRMSE
LDCT	—	20.179	0.7554	0.0107	0.3082	16.876	0.6725	0.0207	0.4832
AE	250	15.098	0.4623	0.0312	0.5344	14.814	0.3900	0.0331	0.6126
	500	16.026	0.5716	0.0251	0.4794	15.284	0.5053	0.0297	0.5803
	1000	16.407	0.5929	0.0230	0.4587	15.251	0.5285	0.0299	0.5822
	Full	16.509	0.5964	0.0224	0.4538	15.289	0.5399	0.0297	0.5802
U-Net	250	17.325	0.5685	0.0201	0.4204	16.434	0.4781	0.0234	0.5186
	500	18.445	0.6606	0.0144	0.3629	17.385	0.6094	0.0187	0.4647
	1000	20.930	0.7697	0.0083	0.2746	18.458	0.6883	0.0147	0.4110
	Full	21.900	0.7918	0.0067	0.2470	18.843	0.7095	0.0135	0.3933
VAE	250	17.278	0.4656	0.0192	0.4173	16.318	0.3741	0.0238	0.5245
	500	19.241	0.5849	0.0121	0.3324	17.329	0.4859	0.0188	0.4671
	1000	20.771	0.6873	0.0086	0.2802	18.094	0.5634	0.0148	0.4266
	Full	21.547	0.7510	0.0073	0.2573	18.364	0.5971	0.0145	0.4091
DnCNN	250	19.814	0.7031	0.0112	0.3171	16.861	0.6132	0.0207	0.4777
	500	20.430	0.7372	0.0101	0.2981	17.192	0.6554	0.0202	0.4740
	1000	21.693	0.7460	0.0071	0.2533	18.702	0.6343	0.0157	0.4217
	Full	21.186	0.7907	0.0088	0.2749	18.925	0.6584	0.0136	0.3662
RED-CNN	250	23.414	0.8433	0.0054	0.2155	18.625	0.6971	0.0139	0.4319
	500	23.612	0.8522	0.0051	0.2078	18.919	0.7264	0.0132	0.4019
	1000	24.097	0.8538	0.0052	0.1990	19.005	0.7327	0.0129	0.3819
	Full	24.115	0.8616	0.0047	0.1979	19.005	0.7467	0.0124	0.3745
RVAE	250	24.829	0.8483	0.0053	0.1889	19.033	0.7438	0.0131	0.3861
	500	26.139	0.8538	0.0050	0.1889	19.193	0.7467	0.0127	0.3789
	1000	26.286	0.8626	0.0046	0.1878	19.259	0.7497	0.0125	0.3763
	Full	**26.574**	**0.8646**	**0.0045**	**0.1817**	**19.288**	**0.7490**	**0.0122**	**0.3690**

Results: Hybrid Loss Comparisons

Model	Dataset Size	PSNR	SSIM	MSE	NRSME
L1	250	20.331	0.7676	0.0100	0.2991
	500	20.928	0.7774	0.0091	0.2820
	1000	22.349	0.7970	0.0067	0.2412
	Full	22.588	0.8205	0.0063	0.2333
MSE	250	23.414	0.8433	0.0054	0.2155
	500	23.612	0.8522	0.0051	0.2078
	1000	24.097	0.8538	0.0052	0.1990
	Full	24.115	0.8616	0.0047	0.1979
Perceptual	250	19.246	0.7739	0.0126	0.3361
	500	19.242	0.7901	0.0125	0.3354
	1000	22.633	0.8322	0.0321	0.2323
	Full	23.623	0.8516	0.0049	0.2063
L1-Smooth	250	22.145	0.7715	0.0106	0.2859
	500	23.862	0.7952	0.0076	0.2382
	1000	24.863	0.8187	0.0064	0.2159
	Full	25.532	0.8362	0.0056	0.2001
L1 + Perceptual	250	23.858	0.8555	0.0050	0.2068
	500	24.069	0.8580	0.0047	0.2017
	1000	24.151	0.8626	0.0046	0.1777
	Full	24.413	0.8656	0.0045	0.1752
MSE + Perceptual	250	24.726	0.8568	0.0046	0.1828
	500	26.530	0.8581	0.0046	0.1804
	1000	26.517	0.8649	0.0045	0.1780
	Full	**26.743**	**0.8660**	**0.0044**	**0.1752**

Results: Self-Supervised Learning Comparisons

Model	$\lvert\mathcal{D}_l\rvert$	In-Domain				Cross-Domain			
		PSNR	SSIM	MSE	NRMSE	PSNR	SSIM	MSE	NRMSE
RED-CNN + REC	250	23.414	0.8433	0.0054	0.2155	19.351	0.7551	0.0122	0.3726
	500	23.612	0.8522	0.0051	0.2078	19.430	0.7572	0.0120	0.3691
	1000	24.097	0.8538	0.0052	0.1990	19.499	0.7523	0.0117	0.3644
	Full	24.115	0.8616	0.0047	0.1979	19.546	0.7555	0.0120	0.3678
RVAE + REC	250	23.795	0.8522	0.0051	0.2120	19.417	0.7583	0.0120	0.3694
	500	24.088	0.8598	0.0050	0.2067	19.544	0.7602	0.0117	0.3640
	1000	24.071	0.8650	0.0048	0.2007	19.570	0.7598	0.0116	0.3632
	Full	24.301	0.8669	0.0047	0.1958	19.609	0.7613	0.0115	0.3614
RED-CNN + NAC	250	23.989	0.8375	0.0050	0.2050	19.122	0.7527	0.0124	0.3750
	500	24.033	0.8541	0.0048	0.2015	19.401	0.7565	0.0127	0.3703
	1000	24.219	0.8549	0.0045	0.1961	19.494	0.7577	0.0118	0.3660
	Full	24.422	0.8611	0.0045	0.1958	19.579	0.7588	0.0116	0.3625
RVAE + NAC	250	23.890	0.8612	0.0052	0.2029	19.313	0.7545	0.0123	0.3739
	500	24.168	0.8579	0.0047	0.1996	19.338	0.7532	0.0123	0.3726
	1000	24.019	0.8647	0.0047	0.1980	19.439	0.7580	0.0122	0.3686
	Full	24.186	0.8649	0.0046	0.1956	19.520	0.7596	0.0118	0.3650
RED-CNN + SSWL	250	26.119	0.8509	0.0050	0.1890	19.550	0.7611	0.0117	0.3635
	500	26.300	0.8542	0.0050	0.1865	19.460	0.7614	0.0120	0.3679
	1000	26.710	0.8627	0.0046	0.1786	19.520	0.7617	0.0120	0.3652
	Full	26.747	0.8626	0.0045	0.1764	19.547	0.7619	0.0117	0.3642
RVAE + SSWL	250	26.150	0.8612	0.0051	0.1900	19.566	0.7632	0.0116	0.3630
	500	26.464	0.8659	0.0048	0.1820	19.619	0.7634	0.0115	0.3607
	1000	26.799	0.8669	0.0043	0.1781	19.549	0.7619	0.0117	0.3505
	Full	26.844	0.8701	0.0044	0.1774	19.617	0.7624	0.0115	0.3692
SSWL-IDN	250	26.581	0.8649	0.0046	0.1793	19.660	0.7645	0.0109	0.3556
	500	26.778	0.8723	0.0045	0.1783	19.854	0.7680	0.0107	0.3530
	1000	27.018	0.8744	0.0043	0.1732	20.016	0.7706	0.0105	0.3505
	Full	**27.800**	**0.8815**	**0.0042**	**0.1701**	**20.178**	**0.7739**	**0.0104**	**0.3458**

Results: SSL Visual Performance

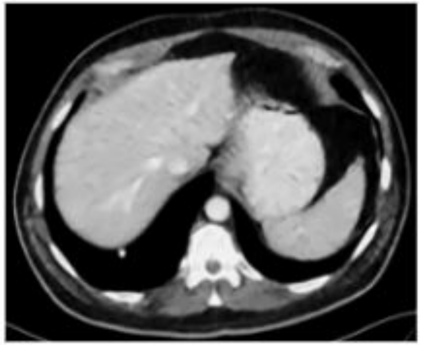
LDCT (0.7586)

FDCT

RED-CNN (0.8790)

RED-CNN+SSWL (0.8901)

RVAE (0.8851)



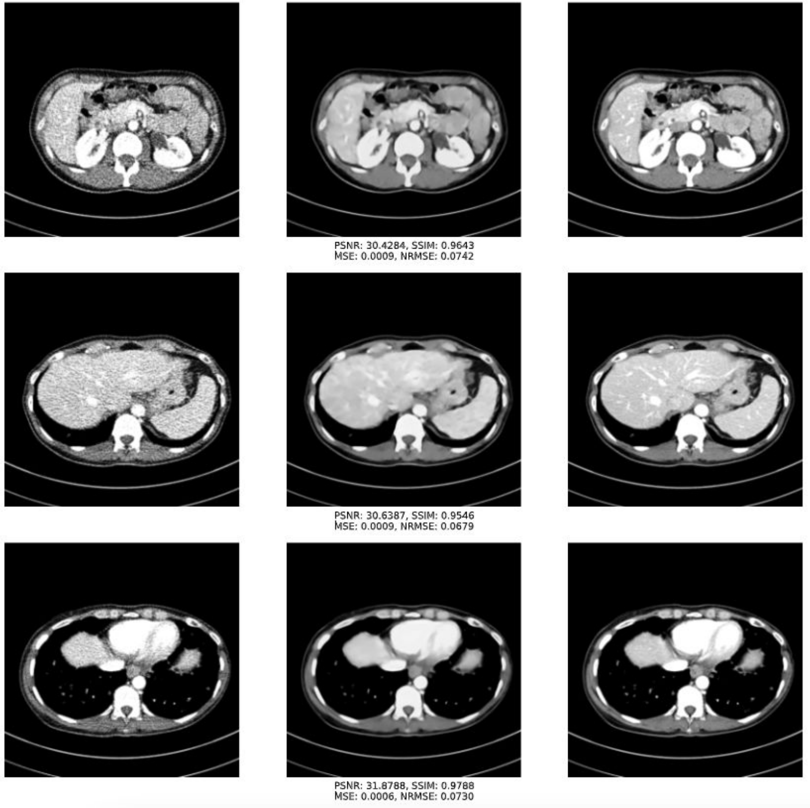
RVAE+SSWL (0.9023)

Results: ROI Comparisons

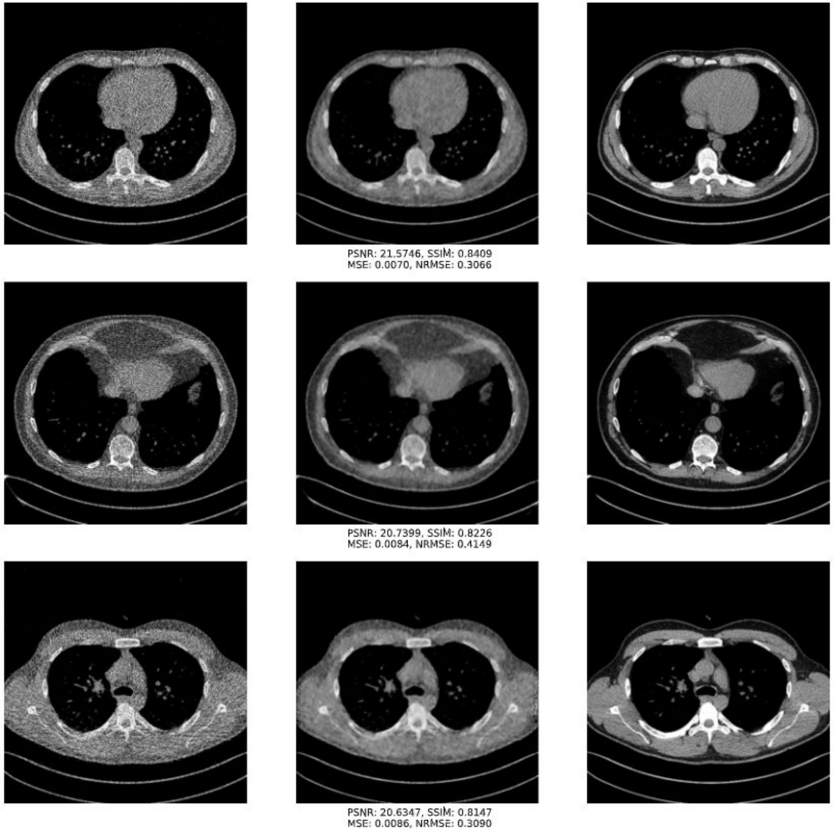

Image GT ROI LDCT (0.7675) REC (0.8756) NAC (0.8802) SSWL (0.8884)

Results: Additional Predictions

In-Domain

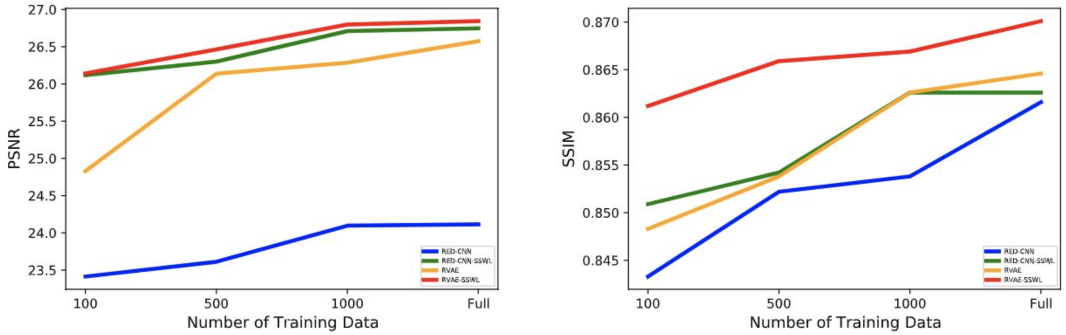


Cross-Domain

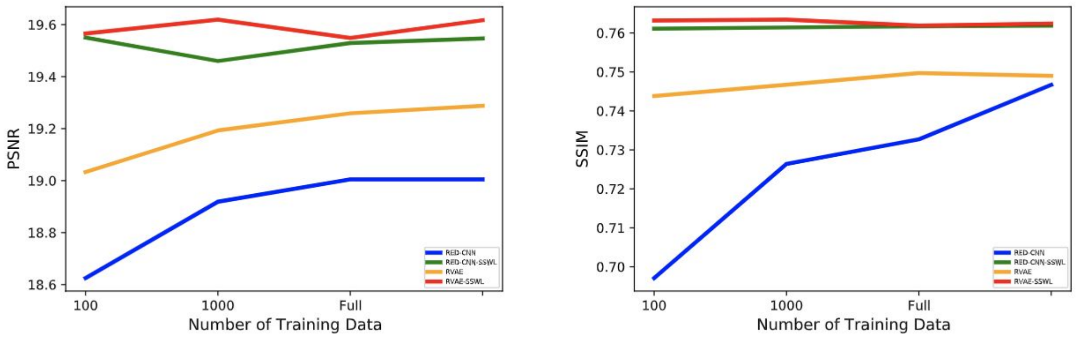


Results: Consistency Plots

In-Domain



Cross-Domain



Strengths/Weaknesses

- Strengths
  - Novel, task-relevant and explainable methods
  - Significantly improved performance of architecture over SOTA
  - SSL task outperforms baselines and SOTA
  - Strong performance in ROIs, aggressive denoising
- Weaknesses
  - Promising cross-domain performance, but can be improved
  - MSE and N-RMSE can be improved over other methods

Acknowledgements

- Dr. Abdullah-Al-Zubaer Imran

- Professor Adam Wang

- This was a great learning experience for me to learn a lot about CT
  - How CT works
  - Challenges in CT from the AI perspective

Research Challenges

- Original training of SSWL
  - Images were not normalized, so results were off
- Tried other surrogate tasks, and none worked out
  - DICOM metadata predictions
  - Learned why certain methods did not work theoretically
- Hard and quick deadline
  - started project in late December, entirely new, so learning curve was steep

Conclusions

- We introduce SSWL-IDN
- Novel, Task-Relevant Self-Supervised Learning
  - Window-Level Prediction, provides efficient and related representation learning
- Residual-VAE
  - Residual learning improves performance
  - Using the reparameterization trick
  - perceptual and MSE loss for pixel-wise and perceptual learning
- Results
  - Strong performance against baselines and other related methods
  - Good predictions in ROIs, promising cross-domain performance
- Future Work
  - Cascaded and joint surrogate learning
  - 3D architectures and models
  - 25% dose evaluation

Questions?

Window-Level is a Strong Denoising Surrogate

Ayaan Haque, Adam Wang, Abdullah-Al-Zubaer Imran