A. Omitted Proof

A.1. Proof for Denoising

Proof of Theorem 4.1. We first show that gradient descent with sufficiently small learning rate will converge to $\bar{\boldsymbol{x}}$, the locally-optimal solution of Equation (9). Recall the loss function $L(\boldsymbol{x}) := q(\boldsymbol{x}) + \frac{1}{2\sigma^2}\|\boldsymbol{x}-\boldsymbol{y}\|^2$ (we subsume the scaling $\frac{1}{2}$ into $\frac{1}{\sigma^2}$ without loss of generality). Notice in the ball $B_r^d(\boldsymbol{x}^*) := \left\{\boldsymbol{x}\in\mathbb{R}^d \,|\, \|\boldsymbol{x}-\boldsymbol{x}^*\| \le r\right\}$, L is $\left(\mu+\frac{1}{\sigma^2}\right)$ strongly-convex. We next show there is a stationary point $\bar{\boldsymbol{x}} \in B_r^d(\boldsymbol{x}^*)$ of $L(\boldsymbol{x})$.

$$\begin{aligned}\nabla L(\bar{\boldsymbol{x}}) = 0 &\implies \nabla q(\bar{\boldsymbol{x}}) + \frac{1}{\sigma^2}(\bar{\boldsymbol{x}}-\boldsymbol{y}) = 0\\ &\implies \nabla q(\bar{\boldsymbol{x}}) - \nabla q(\boldsymbol{x}^*) = \frac{1}{\sigma^2}(\boldsymbol{y}-\bar{\boldsymbol{x}})\\ &\implies \langle \nabla q(\bar{\boldsymbol{x}}) - \nabla q(\boldsymbol{x}^*), \bar{\boldsymbol{x}}-\boldsymbol{x}^*\rangle\\ &= \frac{1}{\sigma^2}\langle \boldsymbol{y}-\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}-\boldsymbol{x}^*\rangle\end{aligned}$$

From strong convexity of q,

$$\langle \nabla q(\bar{\boldsymbol{x}}) - \nabla q(\boldsymbol{x}^*), \bar{\boldsymbol{x}}-\boldsymbol{x}^*\rangle \ge \mu\|\bar{\boldsymbol{x}}-\boldsymbol{x}^*\|^2.$$

Thus,

$$\begin{aligned}&\frac{1}{\sigma^2}\langle \boldsymbol{y}-\boldsymbol{x}^*, \bar{\boldsymbol{x}}-\boldsymbol{x}^*\rangle\\ &= \frac{1}{\sigma^2}\langle (\boldsymbol{y}-\bar{\boldsymbol{x}}) + (\bar{\boldsymbol{x}}-\boldsymbol{x}^*), \bar{\boldsymbol{x}}-\boldsymbol{x}^*\rangle\\ &= \frac{1}{\sigma^2}\langle \boldsymbol{y}-\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}-\boldsymbol{x}^*\rangle + \frac{1}{\sigma^2}\langle \bar{\boldsymbol{x}}-\boldsymbol{x}^*, \bar{\boldsymbol{x}}-\boldsymbol{x}^*\rangle\\ &= \langle \nabla q(\bar{\boldsymbol{x}}) - \nabla q(\boldsymbol{x}^*), \bar{\boldsymbol{x}}-\boldsymbol{x}^*\rangle + \frac{1}{\sigma^2}\|\bar{\boldsymbol{x}}-\boldsymbol{x}^*\|^2\\ &\ge \mu\|\bar{\boldsymbol{x}}-\boldsymbol{x}^*\|^2 + \frac{1}{\sigma^2}\|\bar{\boldsymbol{x}}-\boldsymbol{x}^*\|^2\\ &= \left(\mu+\frac{1}{\sigma^2}\right)\|\bar{\boldsymbol{x}}-\boldsymbol{x}^*\|^2\end{aligned}$$

Finally, by Cauchy-Schwartz inequality,

$$\langle \boldsymbol{y}-\boldsymbol{x}^*, \bar{\boldsymbol{x}}-\boldsymbol{x}^*\rangle \le \|\boldsymbol{y}-\boldsymbol{x}^*\|\cdot\|\bar{\boldsymbol{x}}-\boldsymbol{x}^*\|.$$

So we get $\|\bar{\boldsymbol{x}}-\boldsymbol{x}^*\| \le \frac{1}{1+\mu\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}^*\| \le \|\boldsymbol{\delta}\| \le r$, in other words, $\bar{\boldsymbol{x}} \in B_r^d(\boldsymbol{x}^*)$.

Notice L is $\left(\mu+\frac{1}{\sigma^2}\right)$ strongly-convex in $B_r^d(\boldsymbol{x}^*)$, which contains the stationary point $\bar{\boldsymbol{x}}$. Therefore $\bar{\boldsymbol{x}}$ is a local minimizer of $L(\boldsymbol{x})$. Also note that we implicitly require q to be twice differentiable, meaning in a compact set $B_r^d(\boldsymbol{x}^*)$ its smoothness is upper bounded by a constant M. Thus gradient descent starting from $\boldsymbol{y} \in B_r^d(\boldsymbol{x}^*)$ with learning rate smaller than $\frac{1}{M}$ will converge to $\bar{\boldsymbol{x}}$ without leaving the (convex) set $B_r^d(\boldsymbol{x}^*)$. □

B. Additional Experimental Results

Here we include experimental results and details not included in the main text. Across all the experiments, we individually tuned the hyperparameters for each method.

B.1. Experimental Details

Dataset. For MNIST, we used the default split of 60,000 training images and 10,000 test images of (LeCun et al., 1998). For CelebA-HQ, we used the split of 27,000 training images and 3,000 test images as provided by (Kingma & Dhariwal, 2018).

During evaluation, the following Python script was used to select 1000 MNIST images and 100 CelebA-HQ images from their respective test sets:

```
np.random.seed(0)
indices_mnist = np.random.choice(
    10000, 1000, False)
np.random.seed(0)
indices_celeba = np.random.choice(
    3000, 100, False)
```

Note that CelebA-HQ images were further resized to 64×64 resolution.

Noise Distributions. For the sinusoidal noise used in the experiments, the standard deviation of the k-th pixel/row is calculated as:

$$\sigma_k = 0.1\cdot\left(\exp\left(\sin(2\pi\cdot\frac{k}{16})\right)-1\right)/(e-1),$$

clamped to be in range $[0.001, 1]$. For Figure 9b, we used vary the coefficient 0.1 to values in $\{0.05, 0.1, 0.2, 0.3, 0.4\}$.

For the radial noise used in the additional experiment below, the standard deviation of each pixel with ℓ_2 distance is d from the center pixel (31, 31) is computed as: $\sigma_k = 0.1\cdot\exp(-0.005\cdot d^2)$, clamped to be in range $[0.001, 1000]$.

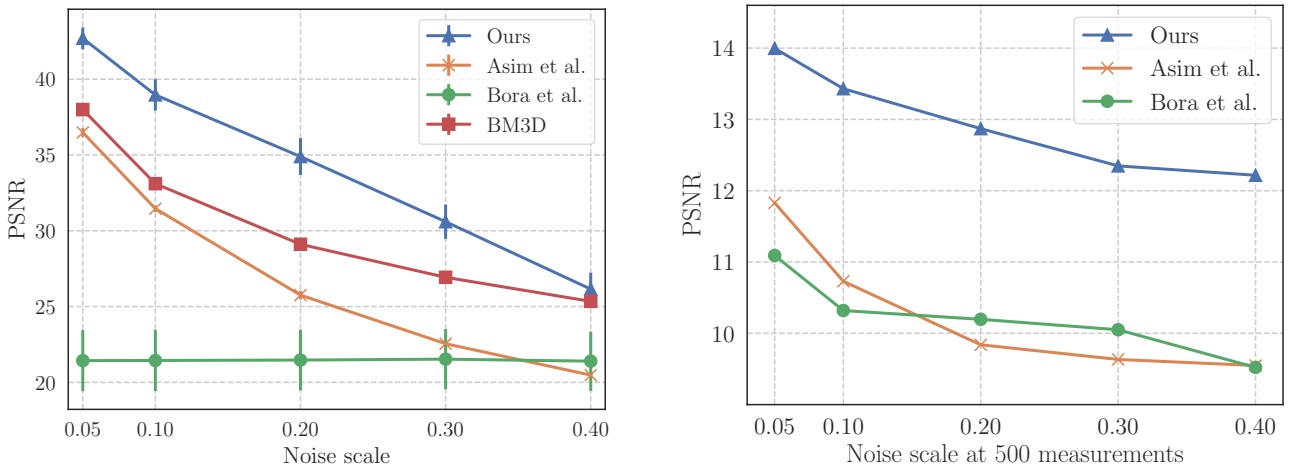
B.2. Additional Result: Removing RADIAL Noise

Consider the measurement process $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\delta}_{\text{radial}}$, where each pixel follows a Gaussian distribution, but with variance that decays exponentially in distance to the center point. For a pixel whose ℓ_2 distance to the center pixel is d, the standard deviation is computed as $\sigma(d) = \exp\left(-0.005\cdot d^2\right)$. See Figure 8 and Figure 9a for reconstructions as well as PSNR plot comparing the methods considered.

B.3. Additional Result: 1-bit Compressed Sensing

Figure 9b shows the performance of each method at different noise scales for a fixed number of measurements. We

Figure 8. Result of denoising RADIAL noise on CelebA-HQ faces and out-of-distribution images.



(a) Result on denoising RADIAL noise at varying noise rates. Our method achieves the same reconstruction performance even when the noise has approximately $1.5\times$ higher noise scale compared to the best baseline method which is BM3D for this setting.

(b) Result of 1-bit compressed sensing at different noise scale. Our method obtains the best reconstructions, achieving similar PSNR as (Asim et al., 2019) when the noise scale is $8\times$ higher.

Figure 9. RADIAL denoising results (left) and 1-bit compressed sensing results at different noise levels (right).

observe that our method performs consistently better at all noise levels.

C. Model Architecture and Hyperparameters

For the RealNVP models we trained, we used multiscale architecture as was done in (Dinh et al., 2016), with residual networks and regularized weight normalization on convolutional layers. Following (Kingma & Dhariwal, 2018), we used 5-bit color depth for the CelebA-HQ model. Hyperparameters and samples from the models can be found in Table 1 and Figure 10.

Hyperparameter	CelebA-HQ	MNIST
Learning rate	5e−4	1e−3
Batch size	16	128
Image size	$64 \times 64 \times 3$	$28 \times 28 \times 1$
Pixel depth	5 bits	8 bits
Number of epochs	300	200
Number of scales	6	3
Residual blocks per scale	10	6
Learning rate halved every	60 epochs	40 epochs
Max gradient norm	500	100
Weightnorm regularization	1e−5	5e−5

Table 1. Hyperparameters used for RealNVP models.

Figure 10. Samples from the RealNVP models used in our experiments.

Figure 11. Out-of-distribution images used in our experiments. We included different types of out-of-distribution instances including grayscale images and cartoons with flat image areas.

D. Experiment Hyperparameters

Here we list the hyperparameters used for each experiment. We used the Adam optimizer (Kingma & Ba, 2014) for all appropriate methods below.

Denoising MNIST Digits.

- Learning rate: 0.02
- Optimization steps for Ours (MAP) and (Asim et al., 2019): 400
- Optimization steps for Ours (MLE) and (Bora et al., 2017): 1000
- Smoothing parameter for Ours (MAP & MLE): $\beta = 1.0$
- Regularization for (Asim et al., 2019): $\gamma = 0.0$
- Regularization for (Bora et al., 2017): $\lambda = 0.01$

Noisy Compressed Sensing.

- Learning rate: 0.02
- Optimization steps for Ours (MAP) and (Asim et al., 2019): 300
- Optimization steps for (Bora et al., 2017): 1000
- Smoothing parameter for Ours (MAP): $\beta = 100$
- Regularization for (Asim et al., 2019): $\gamma = 10$
- Regularization for (Bora et al., 2017): $\lambda = 0.001$
- Regularization for LASSO: $\lambda = 0.01$

Denoising Sinusoidal Noise.

- Learning rate: 0.02
- Optimization steps for Ours (MAP) and (Asim et al., 2019): 150
- Optimization steps for (Bora et al., 2017): 1000
- Smoothing parameter for Ours (MAP): $\beta = 0.5$
- Regularization for (Asim et al., 2019): $\gamma = 2.0$
- Regularization for (Bora et al., 2017): $\lambda = 0.01$

Noisy 1-bit Compressed Sensing.

- Learning rate: 0.02
- Optimization steps for Ours (MAP) and (Asim et al., 2019): 200
- Optimization steps for (Bora et al., 2017): 1000
- Smoothing parameter for Ours (MAP): $\beta = 1.0$
- Regularization for (Asim et al., 2019): $\gamma = 1.0$
- Regularization for (Bora et al., 2017): $\lambda = 0.01$