A. Proofs

Claim A.1. *Let* $\mathbf{A} \in \mathbb{A}^{m\times n}$ *with* $m \leq n$ *and* $\operatorname{rank}(\mathbf{A}) = m$. *Let* $\mathbf{W} \in \mathbb{R}^{m\times m}$ *such that* $\operatorname{rank}(\mathbf{A}^T\mathbf{W}) = m$. *Then, we have*

$$\mathbf{A}^T\mathbf{W}(\mathbf{A}\mathbf{x}-\mathbf{y}) = \mathbf{0} \iff \mathbf{A}^T(\mathbf{A}\mathbf{x}-\mathbf{y}) = \mathbf{0}.$$

Proof. Since $\operatorname{rank}(\mathbf{A}) = m$ we have that $\mathbf{A}^T(\mathbf{A}\mathbf{x}-\mathbf{y}) = \mathbf{0} \iff \mathbf{A}\mathbf{x}-\mathbf{y} = \mathbf{0}$ (e.g., multiply both sides of $\mathbf{A}^T(\mathbf{A}\mathbf{x}-\mathbf{y}) = \mathbf{0}$ from left by $\mathbf{A}^{T\dagger} = (\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}$). Similarly, since $\operatorname{rank}(\mathbf{A}^T\mathbf{W}) = m$ we have that $\mathbf{A}^T\mathbf{W}(\mathbf{A}\mathbf{x}-\mathbf{y}) = \mathbf{0} \iff \mathbf{A}\mathbf{x}-\mathbf{y} = \mathbf{0}$ (e.g., multiply both sides from left by $(\mathbf{A}^T\mathbf{W})^\dagger$). Thus we get the required result. □

Claim A.2. *Let* $\mathbf{A} \in \mathbb{A}^{m\times n}$ *with* $m \leq n$. *Let* $\mathbf{W} \in \mathbb{R}^{m\times m}$ *be a positive definite matrix that shares eigenbasis with* $\mathbf{A}\mathbf{A}^T$. *Then, there exists a positive definite* $\mathbf{P} \in \mathbb{R}^{n\times n}$ *such that*

$$\mathbf{A}^T\mathbf{W}\mathbf{A} = \mathbf{P}^{1/2}\mathbf{A}^T\mathbf{A}\mathbf{P}^{1/2}.$$

Proof. Let $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$ be the SVD of $\mathbf{A}$, where $\mathbf{\Lambda} \in \mathbb{R}^{m\times n}$ is rectangular diagonal, and $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ are orthogonal matrices. By the assumptions on $\mathbf{W}$ we have $\mathbf{W} = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^T$, where $\mathbf{\Gamma}$ is diagonal and invertible. Thus, we have

$$\mathbf{A}^T\mathbf{W}\mathbf{A} = \mathbf{V}\mathbf{\Lambda}^T\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{V}^T.$$

Pick $\mathbf{P} = \mathbf{V}\tilde{\mathbf{\Gamma}}\mathbf{V}^T$ where $\tilde{\mathbf{\Gamma}} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the first m entries on its diagonal that are the same as $\mathbf{\Gamma}$ and 1's (or any other positive values) in the lower $n-m$ entries. We have

$$\mathbf{P}^{1/2}\mathbf{A}^T\mathbf{A}\mathbf{P}^{1/2} = \mathbf{V}\tilde{\mathbf{\Gamma}}^{1/2}\mathbf{\Lambda}^T\mathbf{\Lambda}\tilde{\mathbf{\Gamma}}^{1/2}\mathbf{V}^T = \mathbf{V}\mathbf{\Lambda}^T\mathbf{\Gamma}\mathbf{\Lambda}\mathbf{V}^T,$$

which concludes the proof. □

Claim A.3. *Denote by* λ_1 *the largest singular value of* $\mathbf{A}$. *Let* $c \leq 1/\lambda_1^2$. *Then, the update* (21) *with* $\mu_t = 1$ *ensures reduction in* (24).

Proof. We begin by showing that under such choice of c, we have that $\mathbf{g}_{\delta_t}(\cdot) = \nabla_{\mathbf{x}}\ell_{WLS,t}(\cdot;\mathbf{y})$ is 1-Lipschitz. We prove it by upper bounding the operator norm of the Hessian $\nabla^2_{\mathbf{x}}\ell_{WLS,t}(\cdot;\mathbf{y})$ by 1:

$$\begin{aligned}\|\nabla^2_{\mathbf{x}}\ell_{\mathbf{W}_t}\| &= \|\mathbf{A}^T\mathbf{W}_t\mathbf{A}\| \\ &\leq (1-\delta_t)\|\mathbf{A}^T(\mathbf{A}\mathbf{A}^T+\eta\mathbf{I}_m)^{-1}\mathbf{A}\| + \delta_t c\|\mathbf{A}^T\mathbf{A}\| \\ &\leq (1-\delta_t)+\delta_t = 1.\end{aligned} \tag{28}$$

where in the first inequality follows from the triangle inequality and the second inequality follows from $\|\mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \eta\mathbf{I}_m)^{-1}\mathbf{A}\| \leq 1$ and $\|\mathbf{A}^T\mathbf{A}\| = 1/\lambda_1^2$.

The claim is a consequence of the descent lemma for the gradient step $\tilde{\mathbf{x}} = \mathbf{x} - \mu_t\nabla_{\mathbf{x}}\ell_{WLS,t}(\mathbf{x};\mathbf{y})$ when the step-size equals 1 over the Lipschitz constant of $\mathbf{g}_{\delta_t} = \nabla_{\mathbf{x}}\ell_{WLS,t}$, which is 1 in our case.

For completeness, we present this well-known result here. To simplify notation we denote $\ell_{WLS,t}$ by ℓ and omit dependency on $\mathbf{y}$. The 1-Lipschitzness of the gradient implies that $\|\nabla_{\mathbf{x}}\ell(\mathbf{x}_2) - \nabla\ell(\mathbf{x}_1)\|_2 \leq \|\mathbf{x}_2 - \mathbf{x}_1\|_2$ for all $\mathbf{x}_2, \mathbf{x}_1$. Equivalently, this implies that for all $\mathbf{x}_2, \mathbf{x}_1$ we have

$$\ell(\mathbf{x}_2) - \ell(\mathbf{x}_1) \leq \nabla\ell(\mathbf{x}_1)^T(\mathbf{x}_2-\mathbf{x}_1) + \frac{1}{2}\|\mathbf{x}_2-\mathbf{x}_1\|_2^2. \tag{29}$$

Recall that $\tilde{\mathbf{x}} = \mathbf{x} - \mu_t\nabla\ell(\mathbf{x})$, so using $\mathbf{x}_1 = \mathbf{x}$ and $\mathbf{x}_2 = \tilde{\mathbf{x}}$ in (29), we get

$$\ell(\tilde{\mathbf{x}}) - \ell(\mathbf{x}) \leq -\mu_t\|\nabla\ell(\mathbf{x})\|_2^2 + \mu_t^2\frac{1}{2}\|\nabla\ell(\mathbf{x})\|_2^2. \tag{30}$$

Finally, substituting $\mu_t = 1$ gives $\ell(\tilde{\mathbf{x}}) - \ell(\mathbf{x}) \leq -\frac{1}{2}\|\nabla\ell(\mathbf{x})\|_2^2 \implies \ell(\tilde{\mathbf{x}}) < \ell(\mathbf{x})$ whenever $\nabla\ell(\mathbf{x}) \neq \mathbf{0}$.

□

Theorem A.4. *Consider the observation model (1) and estimating $\mathbf{x}^*$ via minimization of (2) with $s(\mathbf{x}) = \frac{\beta}{2}\|\mathbf{D}\mathbf{x}\|_2^2$. Assume that: (a) $\mathbf{A}^T\mathbf{A}$ and $\mathbf{D}^T\mathbf{D} \succ 0$ share eigenbasis; (b) the singulars value of $\mathbf{A}$ are in $(0, 1]$, and not all equal (common case); (c) $\eta = 0$ and $c = 1$. Then, $b_{BP} < b_{WLS} < b_{LS}$, and $v_{LS} < v_{WLS} < v_{BP}$.*

Proof. Let us define the singular value decomposition (SVD) of the $m \times n$ matrix $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{U}$ is an $m \times m$ orthogonal matrix whose columns are the left singular vectors, $\mathbf{\Lambda}$ is an $m \times n$ rectangular diagonal matrix with nonzero singular values $\{\lambda_i\}_{i=1}^m$ on the diagonal, and $\mathbf{V}$ is an $n \times n$ orthogonal matrix whose columns are the right singular vectors. The assumptions on $\mathbf{D}$ imply that $\mathbf{D}^T\mathbf{D} = \mathbf{V}\mathbf{\Gamma}^2\mathbf{V}^T \succ 0$, where $\mathbf{\Gamma}^2$ is an $n \times n$ diagonal matrix of nonzero eigenvalues $\{\gamma_i^2\}_{i=1}^n$.

Recall that we consider the cost function

$$f_{WLS}(\mathbf{x}) = \frac{1}{2}\|\mathbf{W}^{1/2}(\mathbf{A}\mathbf{x} - \mathbf{y})\|_2^2 + \frac{\beta}{2}\|\mathbf{D}\mathbf{x}\|_2^2.$$

Due to the (strong) convexity of the cost function, the (unique) minimizer can be obtained simply by equating their gradients to zero

$$\begin{aligned}\nabla f_{WLS}(\tilde{\mathbf{x}}) &= \mathbf{A}^T\mathbf{W}(\mathbf{A}\mathbf{x} - \mathbf{y}) + \beta\mathbf{D}^T\mathbf{D}\mathbf{x} = \mathbf{0} \\ &\Rightarrow \hat{\mathbf{x}}_{WLS} = (\mathbf{A}^T\mathbf{W}\mathbf{A} + \beta\mathbf{D}^T\mathbf{D})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{y}. \end{aligned} \tag{31}$$

Note that $\hat{\mathbf{x}}_{LS}$ and $\hat{\mathbf{x}}_{BP}$ are instances of this formula with $\mathbf{W} = \mathbf{I}_m$ and $\mathbf{W} = (\mathbf{A}\mathbf{A}^T)^{-1}$, respectively. For the WLS under consideration we have $\mathbf{W} = (1-\delta)(\mathbf{A}\mathbf{A}^T)^{-1} + \delta\mathbf{I}_m$. In all these cases we have $\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{U}^T$ where $\mathbf{S}$ is an $m \times m$ diagonal matrix of positive values $\{s_i\}_{i=1}^m$ (eigenvalues of $\mathbf{W}$).

From the conditions of the noise we have that $\mathbb{E}[\mathbf{e}] = \mathbf{0}$ and $\mathbb{E}[\mathbf{e}\mathbf{e}^T] = \sigma_e^2\mathbf{I}_m$. Thus, similarly to the analysis in [38], the MSE (conditioned on $\mathbf{x}^*$) can be expressed as

$$\begin{aligned}\mathbb{E}_{\mathbf{e}}\|\hat{\mathbf{x}}_{WLS} - \mathbf{x}^*\|_2^2 &= \mathbb{E}_{\mathbf{e}}\left\|(\mathbf{A}^T\mathbf{W}\mathbf{A} + \beta\mathbf{D}^T\mathbf{D})^{-1}\mathbf{A}^T\mathbf{W}(\mathbf{A}\mathbf{x}^* + \mathbf{e}) - \mathbf{x}^*\right\|_2^2 \\ &= \left\|(\mathbf{A}^T\mathbf{W}\mathbf{A} + \beta\mathbf{D}^T\mathbf{D})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{A}\mathbf{x}^* - \mathbf{x}^*\right\|_2^2 + \mathbb{E}_{\mathbf{e}}\left[\mathbf{e}^T\mathbf{W}\mathbf{A}(\mathbf{A}^T\mathbf{W}\mathbf{A} + \beta\mathbf{D}^T\mathbf{D})^{-2}\mathbf{A}^T\mathbf{W}\mathbf{e}\right] \\ &= \left\|\left((\mathbf{A}^T\mathbf{W}\mathbf{A} + \beta\mathbf{D}^T\mathbf{D})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{A} - \mathbf{I}_n\right)\mathbf{x}^*\right\|_2^2 + \sigma_e^2\mathrm{Tr}\left((\mathbf{A}^T\mathbf{W}\mathbf{A} + \beta\mathbf{D}^T\mathbf{D})^{-2}\mathbf{A}^T\mathbf{W}^2\mathbf{A}\right) \\ &= \left\|\mathbf{V}\left((\mathbf{\Lambda}^T\mathbf{S}\mathbf{\Lambda} + \beta\mathbf{\Gamma}^2)^{-1}\mathbf{\Lambda}^T\mathbf{S}\mathbf{\Lambda} - \mathbf{I}_n\right)\mathbf{V}^T\mathbf{x}^*\right\|_2^2 + \sigma_e^2\mathrm{Tr}\left(\mathbf{V}(\mathbf{\Lambda}^T\mathbf{S}\mathbf{\Lambda} + \beta\mathbf{\Gamma}^2)^{-2}\mathbf{\Lambda}^T\mathbf{S}^2\mathbf{\Lambda}\mathbf{V}^T\right) \\ &= \sum_{i=1}^n\left(\frac{\lambda_i^2 s_i}{\lambda_i^2 s_i + \beta\gamma_i^2} - 1\right)^2[\mathbf{V}^T\mathbf{x}]_i^2 + \sigma_e^2\sum_{i=1}^n\frac{\lambda_i^2 s_i^2}{(\lambda_i^2 s_i + \beta\gamma_i^2)^2}\end{aligned} \tag{32}$$

where s_i and λ_i with $i > m$ are just used for notation convenience and are in fact zeros.

The first term in (32) is the squared bias and the second term is the variance. These expressions can be specialized to each data-fidelity term by substituting the relevant $\mathbf{S}$. Specifically, we have that the bias terms of the estimators are given by:

$$\begin{aligned}bias_{LS}^2 &= \sum_{i=1}^m\left(\frac{\beta\gamma_i^2}{\lambda_i^2 + \beta\gamma_i^2}\right)^2[\mathbf{V}^T\mathbf{x}^*]_i^2 + \sum_{i=m+1}^n[\mathbf{V}^T\mathbf{x}^*]_i^2, \\ bias_{BP}^2 &= \sum_{i=1}^m\left(\frac{\beta\gamma_i^2}{1 + \beta\gamma_i^2}\right)^2[\mathbf{V}^T\mathbf{x}^*]_i^2 + \sum_{i=m+1}^n[\mathbf{V}^T\mathbf{x}^*]_i^2, \\ bias_{WLS}^2 &= \sum_{i=1}^m\left(\frac{\beta\gamma_i^2}{(1-\delta) + \delta\lambda_i^2 + \beta\gamma_i^2}\right)^2[\mathbf{V}^T\mathbf{x}^*]_i^2 + \sum_{i=m+1}^n[\mathbf{V}^T\mathbf{x}^*]_i^2,\end{aligned} \tag{33}$$

where we used the fact that for $\mathbf{W} = (1-\delta)(\mathbf{A}\mathbf{A}^T)^{-1} + \delta\mathbf{I}_m$ we have that $s_i = (1-\delta)/\lambda_i^2 + \delta$.

By the theorem's assumption $\lambda_i \in (0, 1]$ and not all are equal. Thus, we have that $\lambda_i^2 \leq (1-\delta) + \delta\lambda_i^2 \leq 1$ with strict inequalities at some i. Therefore, to prove $bias_{BP}^2 < bias_{WLS}^2 < bias_{LS}^2$, it suffices to show that the function $f(x) = \left(\frac{a}{x+a}\right)^2$ with $a > 0$ is strictly monotonic decreasing on $[0, 1]$, and this trivially holds.

Let us now consider the variances:

$$
\begin{aligned}
var_{LS} &= \sigma_e^2 \sum_{i=1}^{m} \lambda_i^{-2} \frac{\lambda_i^4}{\left(\lambda_i^2 + \beta\gamma_i^2\right)^2}, \\
var_{BP} &= \sigma_e^2 \sum_{i=1}^{m} \lambda_i^{-2} \frac{1}{(1+\beta\gamma_i^2)^2}, \\
var_{WLS} &= \sigma_e^2 \sum_{i=1}^{m} \frac{\lambda_i^2((1-\delta)/\lambda_i^2+\delta)^2}{\left(\lambda_i^2((1-\delta)/\lambda_i^2+\delta)+\beta\gamma_i^2\right)^2} = \sigma_e^2 \sum_{i=1}^{m} \lambda_i^{-2} \frac{((1-\delta)+\delta\lambda_i^2)^2}{\left((1-\delta)+\delta\lambda_i^2+\beta\gamma_i^2\right)^2}
\end{aligned} \tag{34}
$$

Similarly to the way the bias terms where compared, since $\lambda_i^2 \leq (1-\delta)+\delta\lambda_i^2 \leq 1$ with strict inequalities at some, to prove $var_{BP}^2 > var_{WLS}^2 > var_{LS}^2$, it suffices to show that the function $f(x) = \dfrac{x^2}{(x+a)^2} = \dfrac{1}{(1+a/x)^2}$ with $a > 0$ is strictly monotonic increasing on $(0,1]$, and this trivially holds.

□

Claim A.5. *Assume that* $\mathrm{rank}(\mathbf{A}) = m$*, the singular values of* $\mathbf{A}$ *are not all equal,* $\eta = 0$*, and denote by* $\mathbf{V} \in \mathbb{R}^{n\times m}$ *an orthonormal basis for the row-range of* $\mathbf{A}$*. We have that*

$$
\kappa(\mathbf{V}^T\nabla_{\mathbf{x}}^2\ell_{BP}\mathbf{V}) < \kappa(\mathbf{V}^T\nabla_{\mathbf{x}}^2\ell_{WLS}\mathbf{V}) < \kappa(\mathbf{V}^T\nabla_{\mathbf{x}}^2\ell_{LS}\mathbf{V}).
$$

Proof. We can write the *compact* SVD of $\mathbf{A}$ as $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{\Lambda} \in \mathbb{R}^{m\times m}$ is a diagonal matrix with nonzero singular values $\{\lambda_i\}_{i=1}^m$ (indexed in decreasing order), $\mathbf{U} \in \mathbb{R}^{m\times m}$ is an orthogonal matrix and $\mathbf{V} \in \mathbb{R}^{n\times m}$ is the stated partial orthogonal matrix. Note that $\nabla_{\mathbf{x}}^2\ell_{BP} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}$, $\nabla_{\mathbf{x}}^2\ell_{LS} = \mathbf{A}^T\mathbf{A}$, and $\nabla_{\mathbf{x}}^2\ell_{WLS} = (1-\delta)\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A} + \delta c\mathbf{A}^T\mathbf{A}$. Thus, $\kappa(\mathbf{V}^T\nabla_{\mathbf{x}}^2\ell_{BP}\mathbf{V}) = \kappa(\mathbf{I}_m) = 1$. $\kappa(\mathbf{V}^T\nabla_{\mathbf{x}}^2\ell_{LS}\mathbf{V}) = \kappa(\mathbf{\Lambda}^2) = \dfrac{\lambda_1^2}{\lambda_m^2}$. $\kappa(\mathbf{V}^T\nabla_{\mathbf{x}}^2\ell_{WLS}\mathbf{V}) = \kappa((1-\delta)\mathbf{I}_m + \delta c\mathbf{\Lambda}^2) = \dfrac{(1-\delta)+\delta c\lambda_1^2}{(1-\delta)+\delta c\lambda_m^2}$. Clearly, $\kappa(\mathbf{V}^T\nabla_{\mathbf{x}}^2\ell_{WLS}\mathbf{V}) > \kappa(\mathbf{V}^T\nabla_{\mathbf{x}}^2\ell_{BP}\mathbf{V})$. And $\kappa(\mathbf{V}^T\nabla_{\mathbf{x}}^2\ell_{WLS}\mathbf{V}) < \kappa(\mathbf{V}^T\nabla_{\mathbf{x}}^2\ell_{LS}\mathbf{V})$ follows from

$$
\frac{(1-\delta)+\delta c\lambda_1^2}{(1-\delta)+\delta c\lambda_m^2} < \frac{\lambda_1^2}{\lambda_m^2} \iff \lambda_m^2\left((1-\delta)+\delta c\lambda_1^2\right) < \lambda_1^2\left((1-\delta)+\delta c\lambda_m^2\right) \iff \lambda_m^2 < \lambda_1^2.
$$

□

B. Fast Pseudoinverse Implementations

In this section, we show that the pseudoinverse operation $\mathbf{A}^\dagger : \mathbb{R}^m \to \mathbb{R}^n$ can be implemented very efficiently for the cases of image deblurring and image super-resolution (no need to compute and store the SVD of $\mathbf{A}$). We note that there are other cases where this operation can be easily implemented, such as image inpainting, computed tomography, and more. In image inpainting we simply have that $\mathbf{A}^\dagger = \mathbf{A}^T$. In fact, this is the case whenever $\mathbf{A}$ is a *tight-frame* (i.e., when $\mathbf{A}\mathbf{A}^T = \mathbf{I}_m$). In this case, the BP and LS update steps are essentially equivalent, and therefore do not require the special treatment that is considered in the paper. In computed tomography, the pseudoinverse can be implemented via fast (filtered) inverse Radon transform, whose details are out of the scope of this paper. Moreover, as mentioned in the paper, for general $\mathbf{A}$ one can implement the operation $\mathbf{A}^\dagger = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^\dagger$ with low computational complexity by the conjugate gradients methods, where full rank $\mathbf{A}\mathbf{A}^T$ (and $\mathbf{A}\mathbf{A}^T + \eta\mathbf{I}_m$ otherwise) can be "inverted" using few conjugate gradient iterations, which only require applying the operations $\mathbf{A}$ and $\mathbf{A}^T$ and bypass the need of matrix inversion or SVD.

B.1. Image Deblurring

In image deblurring the measurement operator $\mathbf{A} \in \mathbb{R}^{n\times n}$ (note that $m = n$) is a convolution with some blur kernel $\mathbf{k}$, i.e., $\mathbf{A}\mathbf{x} = \mathbf{x} \circledast \mathbf{k}$. Under the assumption of circular convolution (which merely affects boundary pixels and can be addressed by padding), we have that $\mathbf{A}$ is a circulant matrix, and thus can be diagonalized by the discrete Fourier transform. Therefore, this convolution operation can be computed as element-wise multiplication in the discrete Fourier domain, which is efficiently implemented via Fast Fourier Transform (FFT). Specifically, for $\mathbf{z} \in \mathbb{R}^n$ we have that $\mathbf{A}\mathbf{z} = \mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{k})\mathcal{F}(\mathbf{z})\right)$, where $\mathcal{F}$ denotes the FFT. Similarly, $\mathbf{A}^T$, which is convolution with flipped $\mathbf{k}$, can be applied as $\mathbf{A}^T\mathbf{z} = \mathcal{F}^{-1}\left(\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{z})\right)$. Lastly, the operation $\mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \eta\mathbf{I}_n)^{-1}\mathbf{z}$ can be efficiently computed as

$$\mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \eta\mathbf{I}_n)^{-1}\mathbf{z} = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{z})}{|\mathcal{F}(\mathbf{k})|^2 + \eta}\right). \tag{35}$$

As done throughout the paper, we use notation of 1D signal vector for simplification, but the extension to 2D signals, 2D convolutions, and 2D FFT, is straightforward.

B.2. Image Super-Resolution

In image super-resolution the measurement operator $\mathbf{A} \in \mathbb{R}^{m\times n}$ (note that $m = n$) is a composition of convolution with some blur kernel $\mathbf{k}$ and subsampling by some scale factor s, i.e., $\mathbf{A}\mathbf{x} = [\mathbf{x} \circledast \mathbf{k}]\downarrow_s$.

Under the assumption of circular convolution (which merely affects boundary pixels and can be addressed by padding) and integer $s = n/m$, we have $\mathbf{A} = \mathbf{S}\mathbf{B}$, where $\mathbf{B} \in \mathbb{R}^{n\times n}$ is a circulant matrix and $\mathbf{S} \in \mathbb{R}^{m\times n}$. Therefore, the operation $\mathbf{A} = \mathbf{S}\mathbf{B}$ can be implemented by FFT-based filtering followed by subsampling and the operation $\mathbf{A}^T = \mathbf{B}^T\mathbf{S}^T$ can be implemented by upsampling followed by FFT-based filtering. Moreover, $\mathbf{A}\mathbf{A}^T = \mathbf{S}\mathbf{B}\mathbf{B}^T\mathbf{S}^T$ is circulant and essentially performs filtering with the kernel $\mathbf{k}_0 = \left[\mathcal{F}^{-1}\left(|\mathcal{F}(\mathbf{k})|^2\right)\right]\downarrow_s$. Lastly, the operation $\mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \eta\mathbf{I}_m)^{-1}\mathbf{z}$ can be efficiently computed as

$$\mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \eta\mathbf{I}_m)^{-1}\mathbf{z} = \mathcal{F}^{-1}\left(\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}\left(\left[\mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{z})}{|\mathcal{F}(\mathbf{k}_0)|^2 + \eta}\right)\right]\uparrow_s\right)\right). \tag{36}$$

Again, extension from 1D to 2D is straightforward.

C. More Experimental Details and Results

In this section we present more details on the experiments, and more quantitative and qualitative results, which have not been stated in the main body of the paper due to space limitation. Our code is available at https://github.com/tirer-lab/DDPG.

Table 4. Super-resolution and deblurring PSNR [dB] (↑) and LPIPS (↓) results on CelebA-HQ 1K. N/A marks applicability limitation of: (1) DDNM to noiseless settings and (2) DDRM to settings where the SVD is given and stored. (More details in the text). Note that SwinIR and Restormer are task-specific methods, and are thus not flexible to handle most of the examined tasks.

Task \ Method	SwinIR (SR)	Restormer (Deb.)	DDRM	DPS (1000 NFEs)	DiffPIR	DDNM	IDPG (ours)	DDPG (ours)
Bicub. SRx4 σ_e=0	**33.26** / 0.100	—	31.64 / 0.054	29.39 / 0.065	30.26 / 0.051	31.64 / **0.048**	32.66 / 0.111	31.60 / 0.052
Bicub. SRx4 σ_e=0.05	27.30 / 0.213	—	29.26 / 0.090	27.49 / 0.086	27.44 / **0.085**	N/A	**29.89** / 0.155	29.39 / 0.105
Gauss. Deb. σ_e=0	—	29.32 / 0.100	42.49 / 0.006	31.25 / 0.055	32.97 / 0.041	45.56 / **0.002**	**45.58** / **0.002**	45.46 / **0.002**
Gauss. Deb. σ_e=0.05	—	25.28 / 0.431	30.53 / 0.074	27.75 / 0.084	28.89 / 0.074	N/A	**31.08** / 0.150	30.41 / **0.068**
Gauss. Deb. σ_e=0.1	—	21.67 / 0.652	28.79 / 0.088	26.67 / 0.097	27.59 / 0.083	N/A	**29.28** / 0.146	29.18 / **0.080**
Motion Deb. σ_e=0.05	—	19.03 / 0.530	N/A	19.63 / 0.227	27.96 / 0.102	N/A	**29.73** / 0.134	29.02 / **0.082**
Motion Deb. σ_e=0.1	—	16.32 / 0.813	N/A	19.64 / 0.231	26.23 / 0.132	N/A	**27.86** / 0.166	27.74 / **0.099**

C.1. Hyperparameter setting

As mentioned in Section 3.4, in our experiments we do not modify the denoising diffusion model (DDM) hyperparameters $\{\beta_t\}$ compared to other methods. Specifically, we have that this set is composed of linear scheduling from $\beta_{start} = 0.0001$ to $\beta_{end} = 0.02$. The parameters $\{\bar{\alpha}_t\}$ are determined by $\{\beta_t\}$. As explained in the paper, we use $\{\bar{\alpha}_t\}$ of size $T = 100$ to set $\{\delta_t\}$ via $\delta_t = \bar{\alpha}_t^{\gamma}$, where $\gamma \geq 0$ is a single hyperparameter that we tune. Figure 5 shows the resulting $\{\delta_t\}$ for two values of γ. Note that if $\sigma_e = 0$ we simply set $\delta_t = 0$, so we do not need to tune γ.

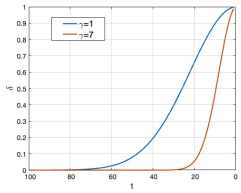

Figure 5. δ_t for $\gamma = \{1, 7\}$.

Another hyperparameter is η, which regularizes the inversion in the operation $\mathbf{A}^T(\mathbf{A}\mathbf{A}^T + \eta\mathbf{I}_m)^{-1}$. We scale it according to the noise level and define: $\eta = \max(1\text{e}{-4}, (2\sigma_e)^2\tilde{\eta})$, where $\tilde{\eta}$ is the hyperparameter that we tune. Note that if $\sigma_e = 0$ we do not need to tune $\tilde{\eta}$. Setting $c = 1$, it is left to set the step-size $\{\mu_t\}$ and, specifically for DDPG, also $\zeta \in [0, 1]$. The step-size that is used is either $\mu_t = 1$ or $\mu_t = (1 - \bar{\alpha}_{t-1})/(1 - \bar{\alpha}_t) := \mu_t^*$, which reduces from 1 significantly only close to the last iterations.

As mentioned in Section 4, the tasks that we consider are common in the literature. For super-resolution, we consider bicubic downsampling with scale factor 4, as in [16, 41]. For deblurring, we consider Gaussian blur kernel with standard deviation 10 clipped to size 5×5, as in [16, 41]. For deblurring, we also consider motion blur kernels generated using the same procedure (with intensity value 0.5) as in [6, 45]. For each observation model we consider different levels of Gaussian noise out of $\{0, 0.05, 0.1\}$.

Let us state the hyperparameters for Section 4.1 (examining the core approach). IDBP is tuned with $\tilde{\eta} = \{32, 6\}$ for deblurring and SR, respectively. For $\sigma_e = 0$, IDPG reduces to IDBP ($\delta_t = 0$), otherwise, for SR with $\sigma_e = 0.05$ it is used with $\tilde{\eta} = 0.2$ and $\gamma = 16$, and for Gaussian deblurring it is used with $\tilde{\eta} = 0.6$ and $\gamma = \{8, 6\}$ for $\sigma_e = \{0.05, 0.1\}$, respectively. In all these cases we use $\mu_t = 1$. Additionally, for motion deblurring in Section 4.2, IDPG is tuned with $\gamma = \{12, 14\}$ and $\tilde{\eta} = \{0.9, 1\}$ (in this case, larger $\tilde{\eta}$ for larger noise allows increasing γ).

Lastly, the hyperparameters of DDPG are listed in Table 5.

Table 5. DDPG hyperparameters.

Task	CelebA-HQ	ImageNet
Bicub. SRx4 σ_e=0	$\zeta = 0.7, \mu_t = 1$	$\zeta = 0.7, \mu_t = 1$
Bicub. SRx4 σ_e=0.05	$\gamma = 10.0, \zeta = 0.8, \tilde{\eta} = 0.3, \mu_t = \mu_t^*$	$\gamma = 6.0, \zeta = 1.0, \tilde{\eta} = 0.3, \mu_t = \mu_t^*$
Gauss. Deb. σ_e=0	$\zeta = 1.0, \mu_t = 1$	$\zeta = 1.0, \mu_t = 1$
Gauss. Deb. σ_e=0.05	$\gamma = 8.0, \zeta = 0.5, \tilde{\eta} = 0.7, \mu_t = \mu_t^*$	$\gamma = 10.0, \zeta = 0.4, \tilde{\eta} = 0.7, \mu_t = \mu_t^*$
Gauss. Deb. σ_e=0.1	$\gamma = 5.0, \zeta = 0.6, \tilde{\eta} = 0.7, \mu_t = \mu_t^*$	—
Motion Deb. σ_e=0.05	$\gamma = 5.0, \zeta = 0.6, \tilde{\eta} = 0.6, \mu_t = \mu_t^*$	$\gamma = 6.0, \zeta = 0.6, \tilde{\eta} = 0.7, \mu_t = \mu_t^*$
Motion Deb. σ_e=0.1	$\gamma = 5.0, \zeta = 0.6, \tilde{\eta} = 0.6, \mu_t = \mu_t^*$	$\gamma = 3.0, \zeta = 0.6, \tilde{\eta} = 0.4, \mu_t = \mu_t^*$

C.2. More quantitative comparisons for deblurring and super-resolution

In this subsection, we report results of more competing methods for the same experimental settings that appear in the main body of the paper.

We examine two representative deep learning methods that are based on per-task supervised learning: SwinIR [17] for super-resolution and Restormer [42] for deblurring. Note though that, as discussed in the paper, we observed that these methods do not generalize well to test sets that are not exactly aligned with their exhaustive training procedure. Specifically, while SwinIR performs well (in terms of PSNR but not in terms of LPIPS) for the noiseless SRx4 with bicubic downsampling, for which it has been exactly trained, it exhibits massive performance drop in the presence of noise. Similarly, we could not managed to get good results with the Restormer, presumably because its training phase considered a specific deblurring dataset. In fact, the behavior of these methods motivates using deep learning for learning the signal prior separately from the observation model, as we discussed in the introduction section.

The results for CelebA-HQ 1K test set are presented in Table 4 (which is an extended version of Table 2). The discussion on the results, as made in the main body of the paper, still carries on. Both our IDPG and DDPG are flexible to the observation model. IDPG presents good PSNR results and DDPG balances it with good LPIPS results (and better perceptual quality). In general, our DDPG demonstrates competitive LPIPS results and better PSNR results than the alternative DDM-based methods. The only reference methods that are as flexible to the observation model as DDPG are DiffPIR [45] and DPS [6]. However, DiffPIR yields significantly lower PSNR and DPS both yields lower PSNR and is also extremely slow.

Applicability issues of DDNM+. As mentioned in Section 4, DDNM+ that was proposed in [41] for handling noisy $\mathbf{y}$, *via SVD* (!), seems to be heavily tied to a specific downsampling task (without bicubic kernel) and does not support the considered tasks. Indeed, when running the official DDNM+ code for bicubic SR with noise we get "not supported" assert, and when running it for deblurring Gaussian kernel with noise level 0.05 (as in Figure 3) it completely fails, e.g., see Figure 6. Thus, DDNM+ cannot be applied to the examined settings (and all the efforts to fix it failed).

Figure 6. Failure of DDNM+ for Gaussian deblurring with noise level 0.05.

Low-noise scenarios. Note that the fact that our approach handles well both noiseless settings and settings with high noise levels implies that it can be readily used for settings with low noise levels. In Table 6 we present the results for $\sigma_e = 0.01$, which show the advantages of our approach also in low noise scenarios. In all these cases we use $\mu_t = 1$. For SR we use $\gamma = 300, \zeta = 1.0, \tilde{\eta} = 1.0$. For Gaussian deblurring we use $\gamma = 11, \zeta = 0.6, \tilde{\eta} = 1.0$. For motion deblurring we use $\gamma = 50, \zeta = 0.5, \tilde{\eta} = 6.0$.

Table 6. PSNR and LPIPS for CelebA-HQ 1K with $\sigma_e = 0.01$. DDRM is not applicable for motion deblur. DDNM(+) is not applicable.

Task \ Method	DDRM	DPS	DiffPIR	IDPG (ours)	DDPG (ours)
Bicub. SRx4 σ_e=0.01	31.09 / 0.066	29.11 / 0.068	29.62 / **0.058**	**31.99** / 0.127	31.81 / 0.092
Gauss. Deb. σ_e=0.01	33.90 / 0.045	30.27 / 0.060	32.01 / 0.060	**34.26** / 0.071	32.20 / **0.044**
Motion Deb. σ_e=0.01	N/A	19.52 / 0.228	31.72 / 0.050	**33.29** / 0.079	32.55 / **0.045**

C.3. Sparse-view computed tomography

In this subsection, we report the performance of our DDPG for sparse-view computed tomography (SV-CT). We compare our method against the recent MCG method [5], which has an official implementation for such task, based on score-SDE model [33], pre-trained on the 2016 American Association of Physicists in Medicine (AAPM) grand challenge dataset resized to 256×256 resolution. As done in [5], the measurement operator $\mathbf{A}$ simulates the CT measurement process with parallel beam geometry with evenly-spaced 180 degrees (essentially, implemented by applying Radon transform on $\mathbf{x}^*$). The test set consists of 100 held-out validation images from the AAPM challenge.

To demonstrate the ease of integrating our approach in SDE-based sampling schemes (and not only in DDPM/DDIM schemes), we make minimal modifications to the MCG implementation, and essentially, merely replace their data-fidelity guidance with our $\mathbf{g}_{\delta_t}$. Specifically, we keep using $T = 2000$ iterations as in MCG (though, this number can be reduced) with the same set of noise levels $\{\tilde{\lambda}_t\} \in (0, 1]$ that decreases along the iterations. Conveniently, we set the step-size $\mu_t = \tilde{\lambda}_t$, and $\delta_t = \left(\frac{1-\tilde{\lambda}_t}{1-\min\tilde{\lambda}}\right)^{\gamma}$. Thus, we can still tune only a scalar γ to determine $\{\delta_t\}$ for our DDPG. No ζ needs to be tune, as the estimated noise in not injected (equivalently $\zeta = 1$). Regarding the regularized back-projection operation (used in $\mathbf{g}_{BP}$), in the context of CT, it is typically being referred to as "filtered back-projection" (FBP) and it is implemented by incorporating

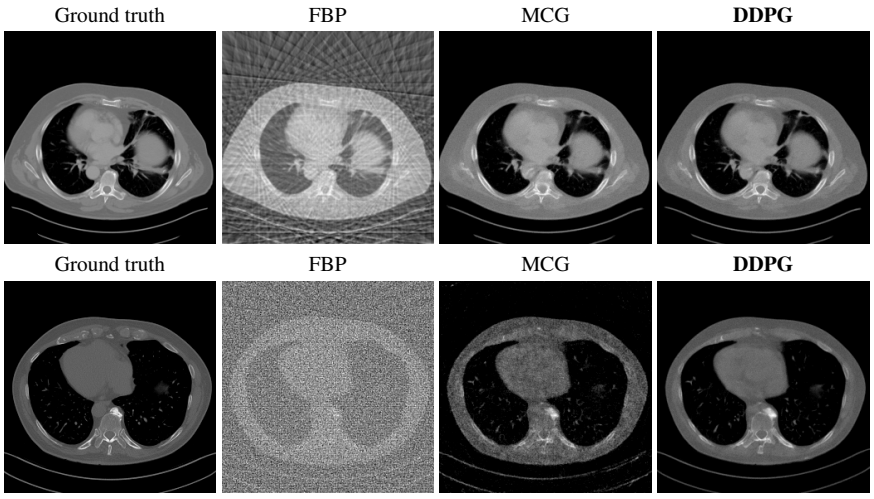

Figure 7. AAPM: Sparse-view CT (30 views). Top: $\sigma_e = 0$; bottom: $\sigma_e = 0.001\|\mathbf{A}\mathbf{x}^*\|_2$.

Ramp filter with the inverse Radon transform. The Ramp filter is triangular in frequency domain with values between 0 and 1 that attenuates low frequencies and thus emphasizes details. We impose the regularization on this BP operation via the hyperparameter η simply by upper bounding the filter in frequency domain by $1/\eta$ (so, e.g., $\eta = 0$ implies no regularization). As for the LS step (used in $\mathbf{g}_{LS}$), the largest eigenvalue of $\mathbf{A}$, denoted by λ_1 in the main body of the paper, is larger than 1 for CT, so we set $c = 1/\lambda_1^2$ instead of $c = 1$. To conclude, we have only two hyperparameters, γ and η, that we manually tune for DDPG.

We consider the SV-CT with 30 views (as in [5]). We examine the case where we do not add additional Gaussian noise $\mathbf{e}$ to $\mathbf{A}\mathbf{x}^*$. Yet, we observed that some ground truth images are already noisy and, presumably, this is detrimental for pure BP-based guidance. We also examine the case where the additional noise level is $0.001\|\mathbf{A}\mathbf{x}^*\|_2$. We use $\gamma = 1, \eta = 0$ and $\gamma = 0.1, \eta = 10$ for the two cases, respectively. The quantitative results (PSNR and SSIM metrics) are presented in Table 7. They show that DDPG outperforms MCG. Qualitative results, which are presented in Figure 7, visually demonstrate the superiority of DDPG over MCG in recovering finer details and robustness to noise.

Table 7. Sparse-view CT (30 views): PSNR [dB] (↑) and SSIM (↑) results on AAPM dataset.

Task \ Method	MCG	DDPG (ours)
CT, $\sigma_e = 0$	34.98 / 0.905	**36.01 / 0.913**
CT, $\sigma_e > 0$	23.63 / 0.480	**26.75 / 0.761**

C.4. More qualitative results

In what follows, we present more visual results for the different tasks. In the noiseless cases many of the methods perform well, so we recommend the reader to focus on the results for the noisy settings, which are also the focus of the paper.

Figure 8. CelebA-HQ: SRx4 for noiseless bicubic downsampling.

Figure 9. ImageNet: SRx4 for noiseless bicubic downsampling.

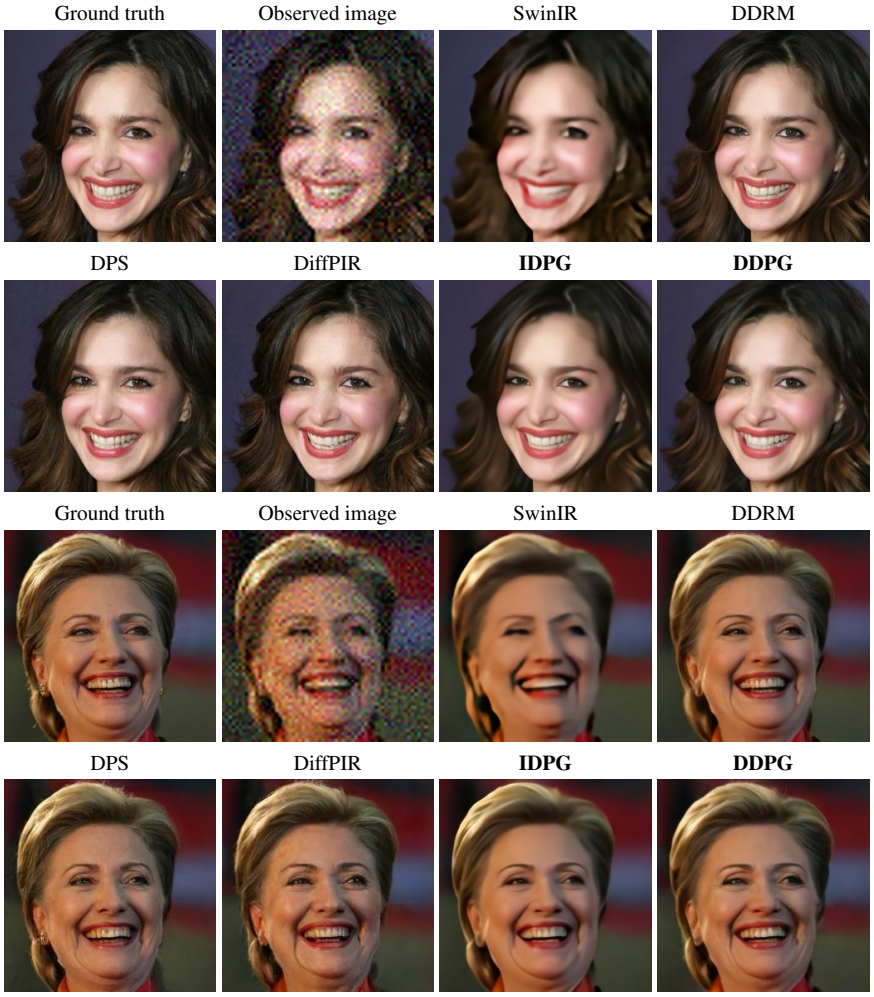

Figure 10. CelebA-HQ: SRx4 for bicubic downsampling with noise level 0.05.

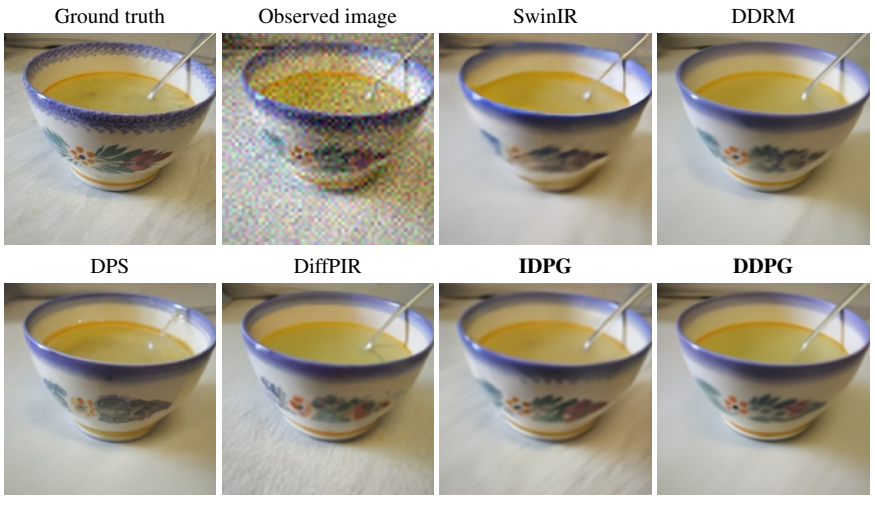

Figure 11. ImageNet: SRx4 for bicubic downsampling with noise level 0.05.

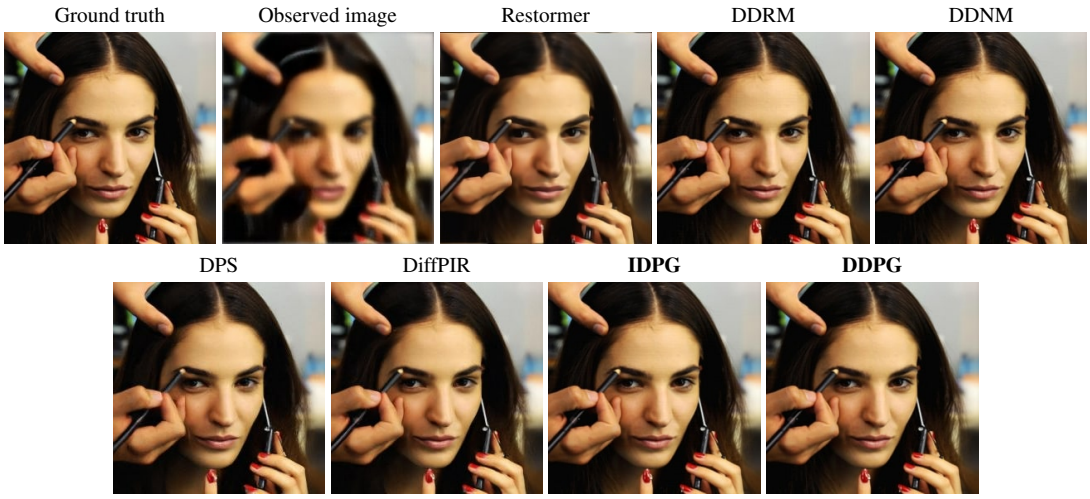

Figure 12. CelebA-HQ: Deblurring for noiseless Gaussian blur.

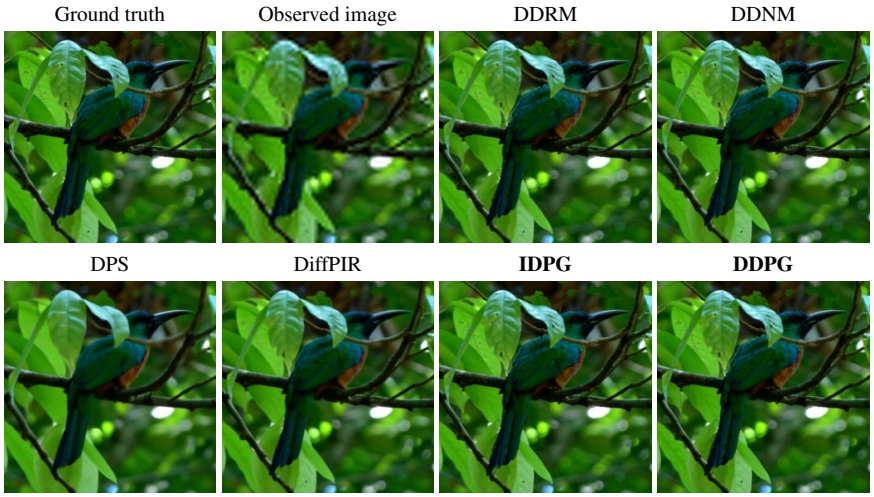

Figure 13. ImageNet: Deblurring for noiseless Gaussian blur.

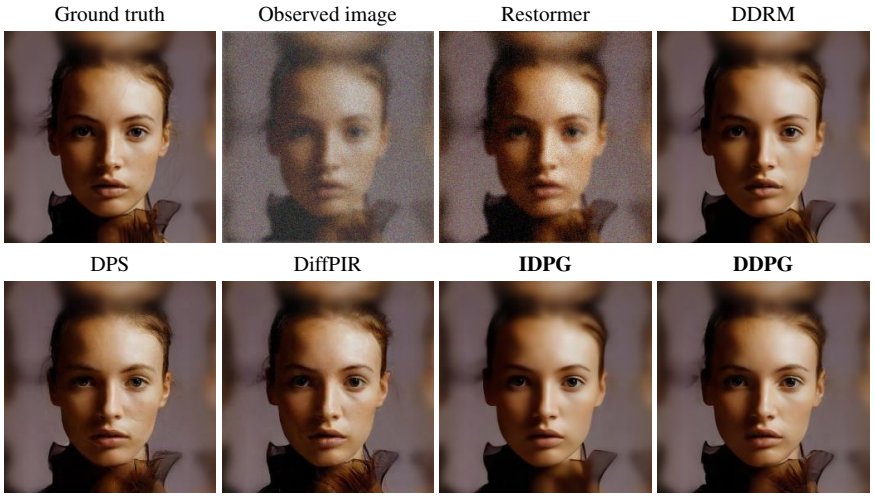

Figure 14. CelebA-HQ: Deblurring for Gaussian blur with noise level 0.05.

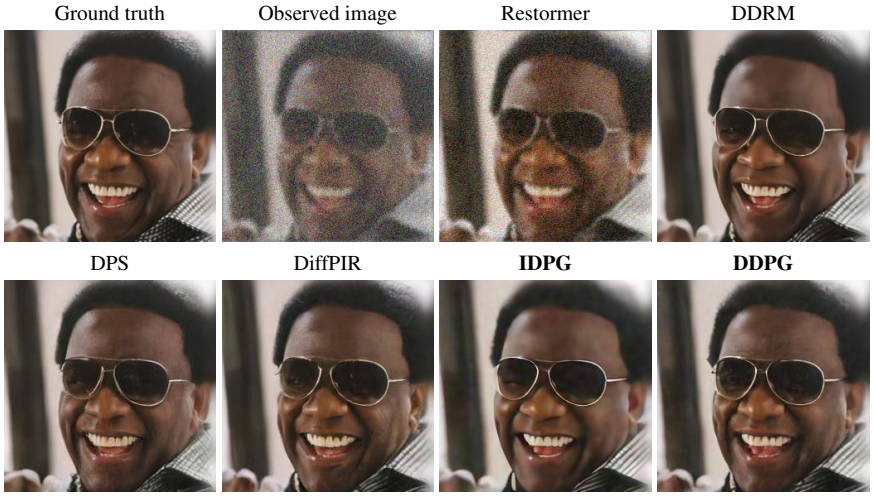

Figure 15. CelebA-HQ: Deblurring for Gaussian blur with noise level 0.1.

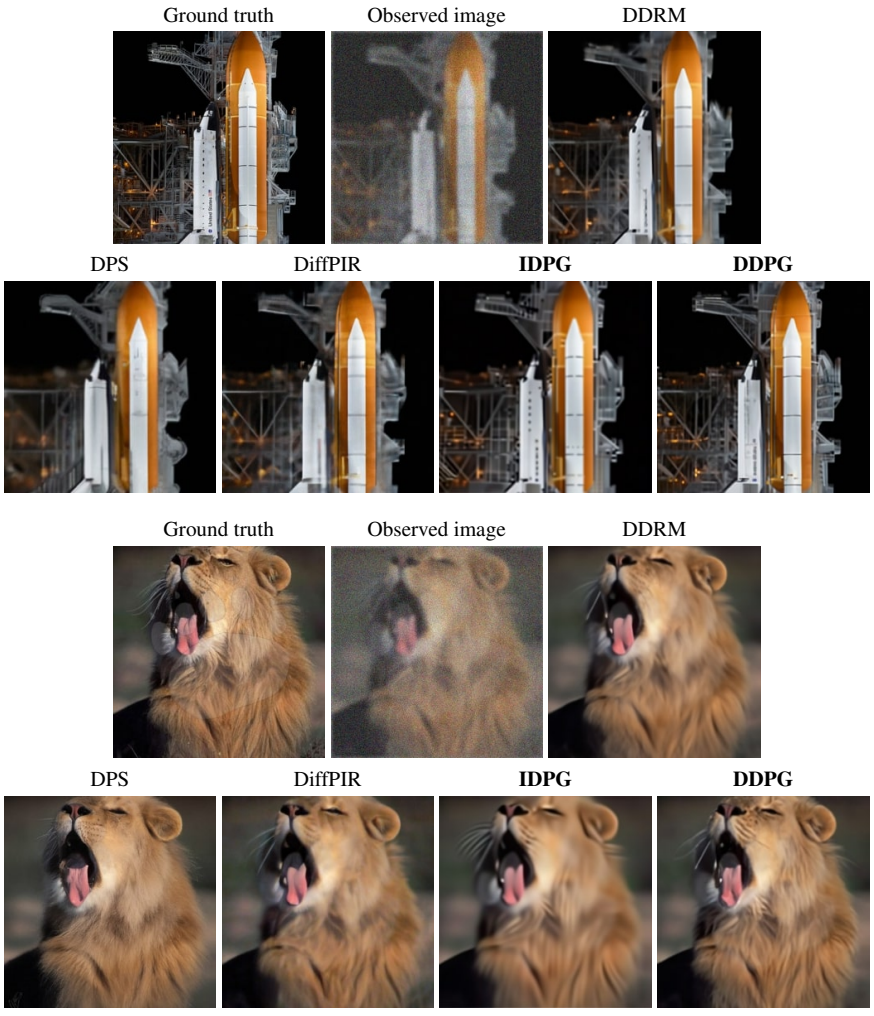

Figure 16. ImageNet: Deblurring for Gaussian blur with noise level 0.05.

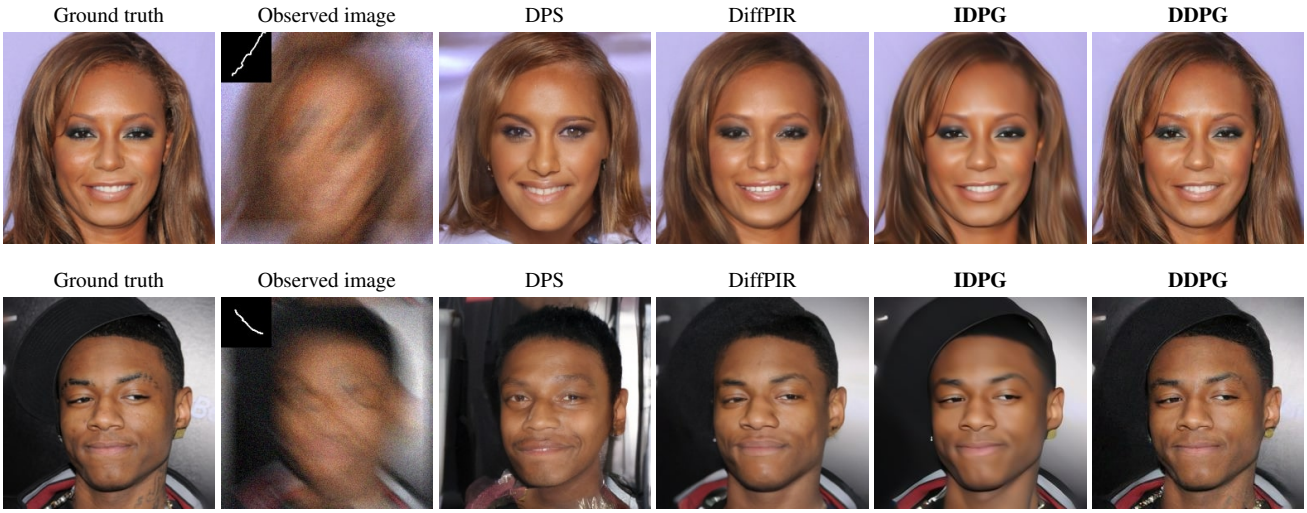

Figure 17. CelebA-HQ: Deblurring for motion blur with noise level 0.05.

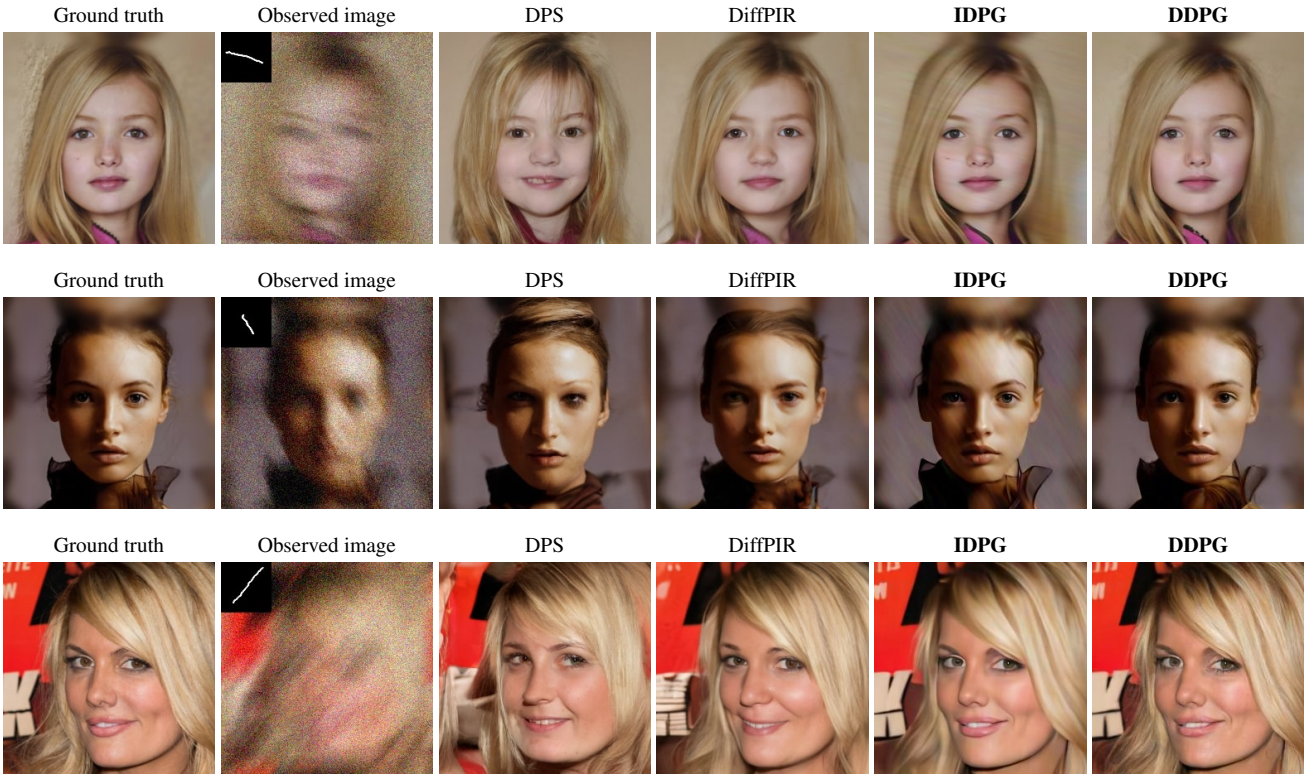

Figure 18. CelebA-HQ: Deblurring for motion blur with noise level 0.1.

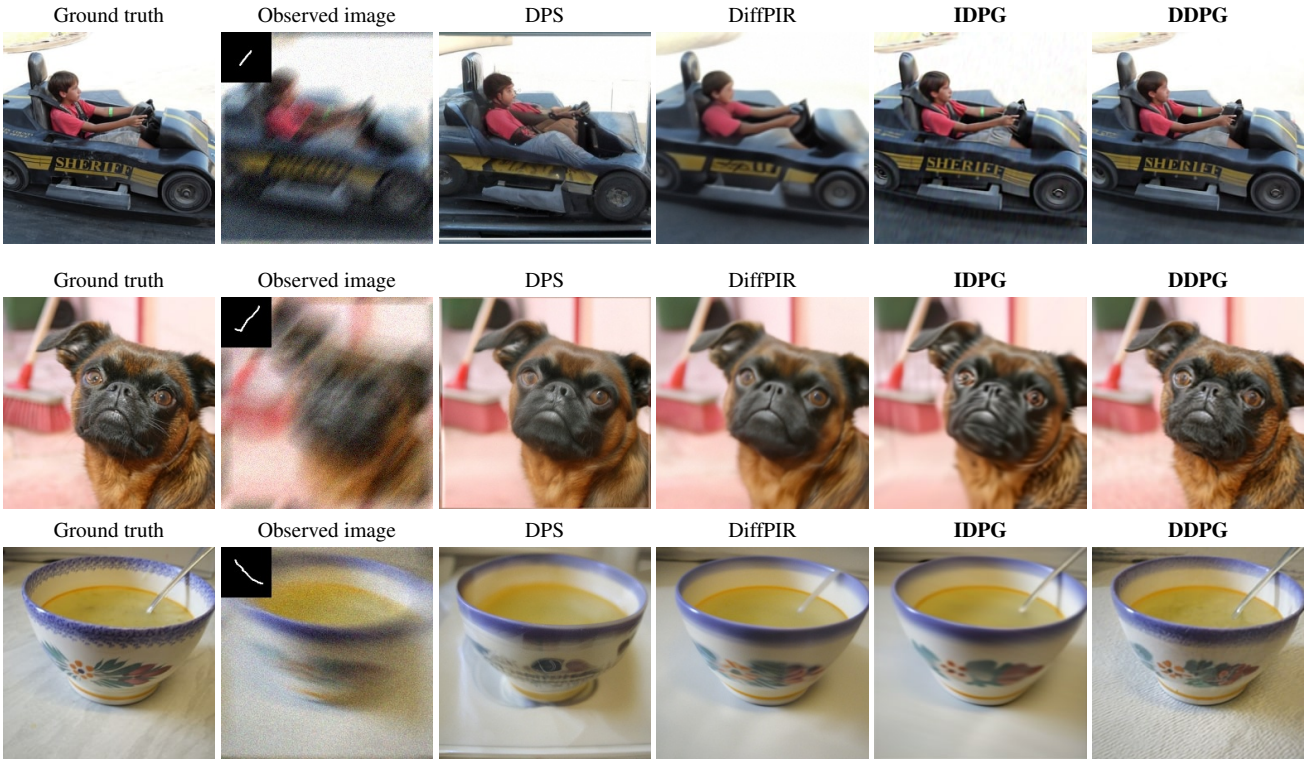

Figure 19. ImageNet: Deblurring for motion blur with noise level 0.05.

Figure 20. ImageNet: Deblurring for motion blur with noise level 0.1.