# Dual Ascent Diffusion for Inverse Problems

Supplementary Material

## A. Convergence Analysis of DDiff

### Notations

- $X \subset \mathbb{R}^n$.
- For $v \in \mathbb{R}^n$, $\|v\| := \|v\|_2/\sqrt{n}$.
- (Forward operator) $A : \mathbb{R}^n \mapsto \mathbb{R}^m$.
- (Measurement noise level) $\sigma \in \mathbb{R}$.
- (Measurement) $y \in \mathbb{R}^m$.
- (Data fidelity term) $f(x) := -\frac{1}{2\sigma^2}\|y - A(x)\|_2^2$.
- (Diffusion times) $\{t_k\}_{k\geq 0} \in \mathbb{R}_+^{\mathbb{N}}$.
- (Schedule parameters) $\forall k \in \mathbb{N}, \bar{\alpha}_{t_k} \in (0, 1], \sigma_{t_k} \in \mathbb{R}_+$.
- (Score network) $s_\theta : \mathbb{R}^n \times \mathbb{R}_+ \mapsto \mathbb{R}^n$
- (Noise threshold) $t_{\text{TH}} \in \mathbb{R}_+$
- (Step sizes) $\{\gamma_k\}_{k\geq 0} \in \mathbb{R}_+^{\mathbb{N}}$.

### Assumptions

(i) $X$ is compact (e.g., $X = [-1, 1]^n$).
(ii) $\lim_{k\to\infty} t_k = 0$.
(iii) $\lim_{k\to\infty} \bar{\alpha}_{t_k} = 1$.
(iv) $\lim_{k\to\infty} \sigma_{t_k} = 0$.
(v) $s_\theta$ is bounded on $X$, i.e.

$$\exists S \in \mathbb{R}_+ \mid \forall (\tilde{x}, t) \in X \times \mathbb{R}_+, \|s_\theta(\tilde{x}, t)\| \leq S.$$

This is standard in practice, e.g., with ReLU/Tanh final layers on normalized inputs or via runtime clipping to enforce bounded outputs.
(vi) $\lim_{k\to\infty} \gamma_k = 0$.
(vii) $\sum_{k=0}^{\infty} \gamma_k < \infty$
(viii) $\sum_{k=0}^{\infty} \sigma_{t_k} < \infty$
(ix) $\sum_{k=0}^{\infty} \sqrt{1 - \bar{\alpha}_{t_k}} < \infty$

### DDiff Recursive Process

#### Initialization

$(x_0, z_0, u_0) \in X \times X \times \mathbb{R}^n$ (e.g., $x_0 = z_0$ and $u_0 = 0$) and $\tilde{x}_{t_0} \sim \mathcal{N}(0, I)$.

#### Recurrence Relation

For $k \in \mathbb{N}$,

$$\begin{aligned}
&\text{z-update (Tweedie):} && z_{k+1} &&= \frac{1}{\sqrt{\bar{\alpha}_{t_k}}}\Big(\tilde{x}_{t_k} + (1 - \bar{\alpha}_{t_k})\, s_\theta(\tilde{x}_{t_k}, t_k)\Big),\\
&\text{x-update (linearized ADMM):} && x_{k+1} &&= v_k + \gamma_k \nabla f(v_k), \quad v_k := z_{k+1} - u_k,\\
&\text{dual update:} && u_{k+1} &&= u_k + x_{k+1} - z_{k+1}.
\end{aligned}$$

DDIM-style latent reverse step anchored at $(x_{k+1}, u_k)$:

$$\hat{\varepsilon}_{t_k} = \frac{\tilde{x}_{t_k} - \sqrt{\bar{\alpha}_{t_k}}\, x_{k+1}}{\sqrt{1 - \bar{\alpha}_{t_k}}}, \qquad \varepsilon_{t_k} \sim \begin{cases} \mathcal{N}(0, I), & t_k > t_{\text{TH}}, \\ 0, & t_k \leq t_{\text{TH}}. \end{cases}$$

$$\tilde{x}_{t_{k+1}} = \sqrt{\bar{\alpha}_{t_{k+1}}}\, x_{k+1} + \sqrt{\bar{\alpha}_{t_{k+1}}}\, u_k + \sqrt{1 - \bar{\alpha}_{t_{k+1}} - \sigma_{t_k}^2}\, \hat{\varepsilon}_{t_k} + \sigma_{t_k}\, \varepsilon_{t_k} \tag{13}$$

$$= \sqrt{\bar{\alpha}_{t_{k+1}}}\, (x_{k+1} + u_k) + \sqrt{1 - \bar{\alpha}_{t_{k+1}} - \sigma_{t_k}^2}\, \hat{\varepsilon}_{t_k} + \sigma_{t_k}\, \varepsilon_{t_k} \tag{14}$$

The "signal part" of $\tilde{x}_{t_{k+1}}$ in the reverse step is $\sqrt{\bar{\alpha}_{t_{k+1}}}\, w_k := \sqrt{\bar{\alpha}_{t_{k+1}}}\, (x_{k+1} + u_k)$. We say the latent $\tilde{x}_{t_k}$ is "anchored" at $w_k$ scaled by $\sqrt{\bar{\alpha}_{t_{k+1}}}$. The subsequent Tweedie z-update $z_{k+2}$ is then a denoiser applied *around* this same anchor, so $z_{k+2}$ stays near $w_k$ when $t_k$ is small and the schedule is in its deterministic tail, i.e., $t_k \leq t_{\text{TH}}$. This "anchoring" reproduces the ADMM pattern where the denoiser acts on $x_{k+1} + u_k$ rather than on $x_{k+1}$ alone.

In this section, we aim to establish the fixed-point convergence of DDiff, following the key lemmas presented below.

## Lemmas

**Lemma 1** (Bounded diffusion denoiser with vanishing strength)**.**

$$\forall k \in \mathbb{N}, \|z_{k+1} - w_{k-1}\| \;\le\; C_d\, \sigma_{\mathrm{eff},t_k},$$

*where*

$$w_{k-1} := x_k + u_{k-1}$$

*and*

$$\sigma_{\mathrm{eff},t_k} := \sqrt{1-\bar\alpha_{t_k}} \;+\; \sigma_{t_{k-1}} \;+\; (1-\bar\alpha_{t_k})$$

*for some constant $C_d$.*

*Proof.* Write the z-update and subtract the anchor to see the deviation:

$$z_{k+1} - w_{k-1} \;=\; \frac{1}{\sqrt{\bar\alpha_{t_k}}}\Big(\tilde x_{t_k} - \sqrt{\bar\alpha_{t_k}} w_{k-1}\Big) \;+\; \frac{1-\bar\alpha_{t_k}}{\sqrt{\bar\alpha_{t_k}}}\, s_\theta(\tilde x_{t_k}, t_k).$$

The reverse step with anchor $w_{k-1}$ yields

$$\tilde x_{t_k} \;=\; \sqrt{\bar\alpha_{t_k}}\, w_{k-1} \;+\; \delta_{t_k}, \qquad \delta_{t_k} := \sqrt{1-\bar\alpha_{t_k}-\sigma_{t_{k-1}}^2}\,\hat\varepsilon_{t_{k-1}} + \sigma_{t_{k-1}}\varepsilon_{t_{k-1}},$$

where $\hat\varepsilon_{t_k}$ is the network-induced direction with $\|\hat\varepsilon_{t_k}\|/\sqrt{n} \lesssim 1$ on bounded $X$ and $\|\varepsilon_{t_k}\|/\sqrt{n} \approx 1$ for large $n$ almost surely. Hence

$$\frac{1}{\sqrt{n}}\|\tilde x_{t_k} - \sqrt{\bar\alpha_{t_k}} w_{k-1}\| \;=\; \frac{1}{\sqrt{n}}\|\delta_{t_k}\| \;\le\; c_1\sqrt{1-\bar\alpha_{t_k}} + c_2\sigma_{t_{k-1}}$$

for some constants $c_1, c_2 > 0$ and given $\sigma_{t_k} \ge 0$. Using $\|s_\theta(x_{t_k}, t_k)\| \le S_{t_k}$ and boundedness of $X$,

$$\frac{1}{\sqrt{n}}\|z_{k+1} - w_{k-1}\| \;\le\; \frac{c_1}{\sqrt{\bar\alpha_{t_k}}}\sqrt{1-\bar\alpha_{t_k}} + \frac{c_2}{\sqrt{\bar\alpha_{t_k}}}\sigma_{t_{k-1}} + \frac{1-\bar\alpha_{t_k}}{\sqrt{\bar\alpha_{t_k}}}\,\frac{S_{t_k}}{\sqrt{n}} \;\le\; C_d\big(\sqrt{1-\bar\alpha_{t_k}} + \sigma_{t_{k-1}} + (1-\bar\alpha_{t_k})\big),$$

with $C_d$ absorbing constants. For a fixed, finite schedule $\bar\alpha_{\min} := \min_{t_k} \bar\alpha_{t_k} > 0$ and finite $S_{t_k}$, $\sup_k \frac{S_{t_k}}{\sqrt{n}\sqrt{\bar\alpha_{t_k}}} < \infty$; hence both $1/\sqrt{\bar\alpha_{t_k}}$ and $S_{t_k}$ are uniformly bounded and can be absorbed into $C_d$. Concretely, one may take

$$C_d \;\ge\; \max\left\{\frac{\max\{c_1,c_2\}}{\sqrt{\bar\alpha_{\min}}},\; \sup_k \frac{S_{t_k}}{\sqrt{n}\sqrt{\bar\alpha_{t_k}}}\right\}.$$

Define effective denoiser strength

$$\sigma_{\mathrm{eff},t_k} \;:=\; \sqrt{1-\bar\alpha_{t_k}} \;+\; \sigma_{t_{k-1}} \;+\; (1-\bar\alpha_{t_k}).$$

As $t_k \to 0$, $1-\bar\alpha_{t_k} \to 0$ and $\sigma_{t_k} \to 0$ (and at the deterministic tail, we have exactly $\sigma_{t_k} = 0$), so $\sigma_{\mathrm{eff},t_k} \to 0$. □

**Remark 1** (Nonconvex prior is compatible with Lemma 1)**.** *Lemma 1 does* not *require convexity of a prior; it only requires that the z-map be close to the identity when the diffusion noise and Tweedie correction are small (i.e., denoiser $\mathcal{D}_\sigma \to \mathcal{I}$ as $\sigma \to 0$ [5]). Proof of Lemma 1 delivers this by showing $z_{k+1} \approx w_{k-1}$ with an error bounded by $\sigma_{\mathrm{eff},t_k}$ as $t_k \to 0$, even if the learned score (hence implicit prior) is nonconvex.*

**Lemma 2** (Bounded gradient of the data term)**.** *Let $f(x) := -\|y - A(x)\|_2^2/(2\sigma^2)$. There exists $L < \infty$ such that*

$$\sup_{x\in X} \|\nabla f(x)\| \;\le\; L.$$

*Proof.* Assume that $A : X \to \mathbb{R}^m$ is continuously differentiable on the compact set $X = [-1,1]^n$, so its Jacobian $J_A(x)$ exists and is continuous. Define

$$f(x) = -\frac{1}{2\sigma^2}\|y - A(x)\|_2^2.$$

By the chain rule,

$$\nabla f(x) = -\frac{1}{\sigma^2} J_A(x)^\top (y - A(x)).$$

Hence,

$$\|\nabla f(x)\| \;\le\; \frac{1}{\sigma^2}\|J_A(x)^\top\|\,\|y - A(x)\| \;=\; \frac{1}{\sigma^2}\|J_A(x)\|_{\mathrm{op}}\,\|y - A(x)\|,$$

where $\|\cdot\|_{\mathrm{op}}$ denotes the operator (spectral) norm of a matrix.

Since $J_A(x)$ and $A(x)$ are continuous on the compact set $X$, the functions $x \mapsto \|J_A(x)\|_{\mathrm{op}}$ and $x \mapsto \|A(x)\|$ are continuous and therefore bounded by the boundedness theorem. Let

$$M_J := \sup_{x\in X} \|J_A(x)\|_{\mathrm{op}} < \infty, \qquad M_A := \sup_{x\in X} \|A(x)\| < \infty.$$

Then, for all $x \in X$,

$$\|\nabla f(x)\| \;\le\; \frac{1}{\sigma^2} M_J(\|y\| + \|A(x)\|) \;\le\; \frac{1}{\sigma^2} M_J(\|y\| + M_A).$$

Define

$$L := \frac{1}{\sigma^2} M_J(\|y\| + M_A) < \infty.$$

It follows that

$$\sup_{x\in X} \|\nabla f(x)\| \;\le\; L,$$

which proves the lemma. □

**Lemma 3** (Closed-form dual update and vanishing behavior under bounded gradients)**.** *Suppose the iterates $\{v_k\}$ lie in $X$, then*

$$u_{k+1} \;=\; \gamma_k\, \nabla f(v_k),$$

*and, using Lemma 2,*

$$\|u_{k+1}\| \;=\; \gamma_k\, \|\nabla f(v_k)\| \;\le\; \gamma_k\, L \;\xrightarrow[k\to\infty]{}\; 0.$$

*Proof.* By definition,

$$u_{k+1} = u_k + x_{k+1} - z_{k+1}.$$

Using the $x$-update and $v_k := z_{k+1} - u_k$,

$$x_{k+1} = v_k + \gamma_k \nabla f(v_k) \quad\Longrightarrow\quad u_{k+1} = u_k + (v_k + \gamma_k \nabla f(v_k)) - z_{k+1} = \gamma_k \nabla f(v_k).$$

Taking norms and applying Lemma 2 yields the claim. □

**Lemma 4** (u-step increment)**.** *Under Lemma 3,*

$$\|u_{k+1} - u_k\| \;=\; \|x_{k+1} - z_{k+1}\| \;\le\; L(\,\gamma_k \,+\, \gamma_{k-1}).$$

*Proof.* $\|u_{k+1} - u_k\| \le \|u_k\| + \|u_{k+1}\| \le L(\,\gamma_k \,+\, \gamma_{k-1})$, where we used $\|u_k\| \le L\,\gamma_{k-1}$ inductively. □

**Lemma 5** (x-step increment)**.** *With Lemma 1 and Lemma 4, we obtain for $k \ge 2$:*

$$\|x_{k+1} - x_k\| \;\le\; C_d\, \sigma_{\mathrm{eff},t_k} \;+\; L(\,\gamma_k + \gamma_{k-1} + \gamma_{k-2}).$$

*Proof.* Let $v_k = z_{k+1} - u_k$ and $x_{k+1} = v_k + \gamma_k \nabla f(v_k)$ with $f(v) = -\|y - A(v)\|_2^2/2\sigma^2$. Add and subtract $v_k$, we obtain $\|x_{k+1} - x_k\| = \|x_{k+1} - x_k + v_k - v_k\| \le \|x_{k+1} - v_k\| + \|x_k - v_k\|$. We know that $\|x_{k+1} - v_k\| = \gamma_k \|\nabla f(v_k)\| \le \gamma_k L$. To bound $\|x_k - v_k\|$, we decompose $v_k - x_k = (z_{k+1} - (x_k + u_{k-1})) + (u_{k-1} - u_k)$ and then apply Lemma 1 with Lemma 4,

$$\begin{aligned}
\|x_k - v_k\| &\le \|z_{k+1} - (x_k + u_{k-1})\| + \|u_k - u_{k-1}\| \\
&= \|z_{k+1} - w_{k-1}\| + \|u_k - u_{k-1}\| \\
&\le C_d\, \sigma_{\mathrm{eff},t_k} \;+\; L(\,\gamma_{k-1} \,+\, \gamma_{k-2})
\end{aligned}$$

Combining the bounds yields $\|x_{k+1} - x_k\| \le \|x_{k+1} - v_k\| + \|x_k - v_k\| \le C_d\, \sigma_{\mathrm{eff},t_k} \;+\; L(\,\gamma_k + \gamma_{k-1} + \gamma_{k-2})$.

□

**Lemma 6** (z-step increment)**.** *Let $t_k$ be the diffusion index used to form $z_{k+1}$ (so $z_k$ was formed at $t_{k-1}$). Under Lemma 1–5, for $k \geq 3$,*

$$\|z_{k+1} - z_k\| \;\leq\; C_d\,\sigma_{\text{eff},t_k} \;+\; 2C_d\,\sigma_{\text{eff},t_{k-1}} \;+\; L\big(\gamma_{k-1} + 2\gamma_{k-2} + 2\gamma_{k-3}\big).$$

*Proof.* Introduce the anchors $w_{k-1} := x_k + u_{k-1}$ and $w_{k-2} := x_{k-1} + u_{k-2}$. By the triangle inequality,

$$\|z_{k+1} - z_k\| \;\leq\; \|z_{k+1} - w_{k-1}\| \;+\; \|w_{k-1} - w_{k-2}\| \;+\; \|w_{k-2} - z_k\|.$$

By Lemma 1,

$$\|z_{k+1} - w_{k-1}\| \leq C_d\,\sigma_{\text{eff},t_k}, \qquad \|z_k - w_{k-2}\| \leq C_d\,\sigma_{\text{eff},t_{k-1}}.$$

For the anchor increment,

$$\|w_{k-1} - w_{k-2}\| \;\leq\; \|x_k - x_{k-1}\| + \|u_{k-1} - u_{k-2}\|.$$

Apply Lemma 5 at index $k-1$ and Lemma 4 at index $k-2$:

$$\|x_k - x_{k-1}\| \;\leq\; C_d\,\sigma_{\text{eff},t_{k-1}} + L(\gamma_{k-1} + \gamma_{k-2} + \gamma_{k-3}), \qquad \|u_{k-1} - u_{k-2}\| \;\leq\; L(\gamma_{k-2} + \gamma_{k-3}).$$

Combining the bounds yields

$$\|w_{k-1} - w_{k-2}\| \;\leq\; C_d\,\sigma_{\text{eff},t_{k-1}} \;+\; L\big(\gamma_{k-1} + 2\gamma_{k-2} + 2\gamma_{k-3}\big).$$

Returning to the decomposition, we obtain

$$\|z_{k+1} - z_k\| \;\leq\; C_d\,\sigma_{\text{eff},t_k} + C_d\,\sigma_{\text{eff},t_{k-1}} \;+\; \Big[C_d\,\sigma_{\text{eff},t_{k-1}} \;+\; L\big(\gamma_{k-1} + 2\gamma_{k-2} + 2\gamma_{k-3}\big)\Big],$$

which simplifies to the stated inequality. □

## Target Theorem

**Theorem 1** (Fixed-point Convergence of DDiff)**.** *Let $\{(x_k, z_k, u_k)\}_{k\geq 0}$ be generated by the DDiff updates above. Then, the sequences $\{x_k\}$, $\{z_k\}$ and $\{u_k\}$ are Cauchy sequences. In particular,*

$$\|x_{k+1} - x_k\| + \|z_{k+1} - z_k\| + \|u_{k+1} - u_k\| \xrightarrow[k\to\infty]{} 0,$$

*and $(x_k, z_k, u_k) \to (x^\star, z^\star, u^\star)$.*

*Proof.* **Step 1** (increment bounds). Lemma 4 gives

$$\|u_{k+1} - u_k\| \;\leq\; L(\,\gamma_k \;+\; \gamma_{k-1}).$$

Lemma 5 gives

$$\|x_{k+1} - x_k\| \;\leq\; C_d\,\sigma_{\text{eff},t_k} \;+\; L(\,\gamma_k + \gamma_{k-1} + \gamma_{k-2}).$$

Lemma 6 gives

$$\|z_{k+1} - z_k\| \;\leq\; C_d\,\sigma_{\text{eff},t_k} \;+\; 2C_d\,\sigma_{\text{eff},t_{k-1}} \;+\; L\big(\gamma_{k-1} + 2\gamma_{k-2} + 2\gamma_{k-3}\big).$$

**Step 2** (summability). Summing the three bounds,

$$\begin{aligned}\|x_{k+1} - x_k\| + \|z_{k+1} - z_k\| + \|u_{k+1} - u_k\| &\leq 2C_d\,\sigma_{\text{eff},t_k} + 2C_d\,\sigma_{\text{eff},t_{k-1}} \qquad (15)\\ &\quad + L\Big(2\gamma_k + 3\gamma_{k-1} + 3\gamma_{k-2} + 2\gamma_{k-3}\Big)\\ &\leq 2C_d\big(\sigma_{\text{eff},t_k} + \sigma_{\text{eff},t_{k-1}}\big) + 3L\sum_{j=0}^{3}\gamma_{k-j}. \qquad (16)\end{aligned}$$

Following Lemma 1 and the Assumptions, the right-hand side of (16) is summable over $k$ and tends to 0 as $k \to \infty$:

$$\sum_{k=0}^{\infty}\big(\|x_{k+1} - x_k\| + \|z_{k+1} - z_k\| + \|u_{k+1} - u_k\|\big) < \infty, \quad \|x_{k+1} - x_k\| + \|z_{k+1} - z_k\| + \|u_{k+1} - u_k\| \to 0.$$

Absolute summability implies that $\{x_k\}$, $\{z_k\}$, and $\{u_k\}$ are Cauchy, hence converge in $\mathbb{R}^{3n}$ to $(x^\star, z^\star, u^\star)$.

By Lemma 3, we have

$$u_{k+1} = \gamma_k \nabla f(v_k), \quad v_k := z_{k+1} - u_k.$$

Since $\|\nabla f(v_k)\|$ is bounded and $\gamma_k \to 0$, it follows that $u_{k+1} \to 0$, hence $u^\star = 0$. Moreover,

$$x_{k+1} - z_{k+1} = u_{k+1} - u_k \to 0,$$

so $x^\star = z^\star$.

Taking $k \to \infty$ in the DDiff updates and noting that $\sigma_{\mathrm{eff},t_k} \to 0$, $\gamma_k \to 0$, and $(x_k, z_k, u_k) \to (x^\star, z^\star, 0)$, we obtain:

- By Lemma 1, $z_{k+1} - (x_k + u_{k-1}) \to 0$;
- From the $x$-update, $x_{k+1} - (z_{k+1} - u_k) \to 0$;
- The dual update $u_{k+1} = u_k + x_{k+1} - z_{k+1}$ holds by definition.

In the limit, these relations imply

$$z^\star = x^\star, \quad x^\star = z^\star, \quad \text{and} \quad u^\star = 0,$$

confirming that $(x^\star, z^\star, u^\star) = (x^\star, x^\star, 0)$ is a fixed point of the asymptotic DDiff updates. □

**Remark 2** (Finite iterations, step-size policy, and ADMM–diffusion coupling)**.** *We run a finite number of iterations $k = 0, \ldots, K$, setting $K = T$ equal to the number of diffusion time steps. Each diffusion reverse step $(t_k \to t_{k-1})$ is paired with one ADMM update, so that the dual variable $u_k$ is updated in synchrony with the denoising process. The convergence analysis assumes a vanishing step-size schedule $(\gamma_k \to 0)$. In practice, we adopt a nonincreasing, step-down policy for $\gamma_k$ to control measurement update magnitudes; the exact schedule and hyperparameters are provided in Sec. H. Empirically, as $\gamma_k$ becomes sufficiently small across iterations, the iterates are stable and reconstruction quality improves, as evidenced by lower LPIPS and higher SSIM (often with competitive PSNR).*

## B. Residual Metric

We report the residual

$$r(x) = \|y - \mathcal{A}(x)\|_2^2 - \sigma^2,$$

which measures the deviation of the reconstruction from the noisy measurement after accounting for the expected noise energy $\sigma^2 = \mathbb{E}\big[\|\varepsilon\|_2^2\big]$ under the forward model $y = \mathcal{A}(x) + \varepsilon$. An ideal posterior sample $\tilde{x} \sim p(x|y)$ satisfies $\mathbb{E}_{x\sim p(x), y\sim p(y|x), \tilde{x}\sim p(x|y)}[r(\tilde{x})] = 0$. Hence, $r(x)$ quantifies *data consistency*: small positive values indicate faithful reconstructions that match the measurement statistics, whereas excessively negative values (i.e., $\|y - \mathcal{A}(x)\|_2^2 \ll \sigma^2$) correspond to over-smoothed or overly noise-fitting solutions, and large positive values indicate hallucination or loss of fidelity to $y$.

We therefore interpret residual magnitudes near zero as optimal, while deviations in either direction (too small or too large) signal a mismatch between data fidelity and prior regularization. Throughout, we use the residual only as a diagnostic for data consistency and report it alongside perceptual and distortion metrics (PSNR, SSIM, LPIPS).

## C. DDiff with Latent Diffusion Models

Pretrained diffusion models can operate in the latent space of an autoencoder instead of directly in the pixel space to reduce dimensionality and improve computational efficiency [29]. An encoder $\mathcal{E} : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^d$ maps an image $\mathbf{x}$ to a compact latent representation $\mathbf{z} = \mathcal{E}(\mathbf{x})$, and a decoder $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}^{H\times W\times C}$ reconstructs an image $\tilde{\mathbf{x}} = \mathcal{D}(\mathbf{z})$. Diffusion training and denoising are then performed in the latent space $\mathbf{z}$ using a latent score network $\mathbf{s}_\theta(\mathbf{z}, t)$, which estimates $\nabla_{\mathbf{z}} \log p_t(\mathbf{z})$. Given a pretrained LDM, our goal is to adapt DDiff to operate with such encoder–decoder architectures.

**Algorithm 2** LatentDDiff (latent-space DDiff with an encoder–decoder LDM)

**Require:** $T$; forward operator $\mathcal{A}(\cdot)$; schedules $\{\sigma_t\}_{t=1}^T$, $\{\bar{\alpha}_t\}_{t=1}^T$; latent score $\mathbf{s}_\theta$; measurements $\mathbf{y}$; step sizes $\{\gamma_t\}_{t=1}^T$; noise threshold $t_0$; encoder $\mathcal{E}$, decoder $\mathcal{D}$; MODE $\in \{$LATENT-DC, PIXEL-DC$\}$

Initialize latent variable $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, dual variable $\mathbf{u} = \mathbf{0}$.
**for** $t = T-1$ **to** $0$ **do**
  $\tilde{\mathbf{z}} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{z}_t + (1-\bar{\alpha}_t)\,\mathbf{s}_\theta(\mathbf{z}_t, t))$ ▷ Denoising in latent space
  *Data-consistency update (choose one by* MODE*)*
  **if** MODE = LATENT-DC **then**
    $\hat{\mathbf{x}} \leftarrow \tilde{\mathbf{z}} - \mathbf{u} - \gamma_t \nabla_{\mathbf{v}=\tilde{\mathbf{z}}-\mathbf{u}} \left\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{v}))\right\|_2^2$ ▷ Gradient flows through $\mathcal{D}$: $\nabla_{\mathbf{v}}\|\cdot\|^2$ involves $J_{\mathcal{D}}(\mathbf{v})$
    $\mathbf{x} \leftarrow \mathcal{E}(\mathcal{D}(\hat{\mathbf{x}}))$ ▷ Re-Encode
  **else**
    $\tilde{\mathbf{x}} \leftarrow \mathcal{D}(\tilde{\mathbf{z}})$ ▷ PIXEL-DC
    $\hat{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} - \mathbf{u} - \gamma_t \nabla_{\mathbf{v}=\tilde{\mathbf{x}}-\mathbf{u}} \left\|\mathbf{y} - \mathcal{A}(\mathbf{v})\right\|_2^2$
    $\mathbf{x} \leftarrow \mathcal{E}(\hat{\mathbf{x}})$ ▷ Encode to latent
  **end if**
  $\hat{\epsilon} \leftarrow \frac{1}{\sqrt{1-\bar{\alpha}_t}}(\mathbf{z}_t - \sqrt{\bar{\alpha}_t}\,\mathbf{x})$
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ **if** $t > t_0$ **else** $\epsilon = 0$
  $\mathbf{z}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\mathbf{x} + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\,\hat{\epsilon} + \sigma_t\,\epsilon + \sqrt{\bar{\alpha}_{t-1}}\,\mathbf{u}$ ▷ Reverse diffusion in latent space
  $\mathbf{u} \leftarrow \mathbf{u} + \mathbf{x} - \tilde{\mathbf{z}}$ ▷ Dual update
**end for**
**return** $\mathbf{x}_0 \leftarrow \mathcal{D}(\mathbf{z}_0)$

**Latent-space diffusion updates.** Analogous to the pixel-space formulation, each DDiff iteration in the latent space performs (i) a denoising step using the latent score and (ii) a data-consistency (DC) correction. Let $\mathbf{z}_t$ denote the latent variable at timestep $t$ and $\bar{\alpha}_t$ the corresponding noise schedule. The denoising prediction is given by

$$\tilde{\mathbf{z}} = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{z}_t + (1-\bar{\alpha}_t)\,\mathbf{s}_\theta(\mathbf{z}_t, t))\,, \tag{17}$$

after which a DC update is applied before the reverse diffusion step.

**Data-consistency in latent diffusion.** Incorporating measurement consistency within latent diffusion can be done in two ways, depending on where the update is applied:

**(1) LATENT-DC: data consistency in latent space.** This approach enforces the measurement constraint directly in the latent space by minimizing the following objective:

$$\hat{\mathbf{x}} \leftarrow \operatorname*{argmin}_{\hat{\mathbf{x}}} \frac{1}{2\sigma^2}\left\|\mathbf{y} - \mathcal{A}(\mathcal{D}(\hat{\mathbf{x}}))\right\|_2^2 + \frac{\rho}{2}\left\|\hat{\mathbf{x}} - \tilde{\mathbf{z}} + \mathbf{u}\right\|_2^2, \tag{18}$$

where the first term enforces fidelity to the measurements and the second term couples the current estimate $\hat{\mathbf{x}}$ with the diffusion variable $\tilde{\mathbf{z}}$ and dual variable $\mathbf{u}$. In practice, rather than solving (18) exactly, a single gradient step is performed to approximate the proximal update, which is well established in the plug-and-play literature as an efficient first-order method

for nonlinear forward operators [39]. The gradient of the data-fidelity term is back-propagated through the decoder $\mathcal{D}(\cdot)$, enforcing consistency directly on the latent variable. The update reads

$$\hat{\mathbf{x}} = \tilde{\mathbf{z}} - \mathbf{u} - \gamma_t \nabla_{\mathbf{v}=\tilde{\mathbf{z}}-\mathbf{u}} \left\| \mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{v})) \right\|_2^2, \tag{19}$$

where the gradient involves the Jacobian $J_{\mathcal{D}}(\mathbf{v})$ of the decoder, i.e., $\nabla_{\mathbf{v}} \|\mathbf{y} - \mathcal{A}(\mathcal{D}(\mathbf{v}))\|^2 = J_{\mathcal{D}}(\mathbf{v})^\top \nabla_{\mathcal{D}(\mathbf{v})} \|\mathbf{y} - \mathcal{A}(\cdot)\|^2$.

After the update in (19), we perform a subsequent *Re-Encode* step, $\mathbf{x} \leftarrow \mathcal{E}(\mathcal{D}(\hat{\mathbf{x}}))$, following the strategy used in DCDP [24]. This re-encoding step ensures that the updated latent remains on the manifold learned by the LDM, since the encoder $\mathcal{E}$ is nonlinear and the model is trained only on latents of the form $\mathbf{z} = \mathcal{E}(\mathbf{x})$ corresponding to valid images.

**(2) PIXEL-DC: data consistency in pixel space.** Alternatively, the latent is first decoded to the pixel domain, and the correction is applied directly on the reconstructed image. This is done by minimizing the following objective:

$$\hat{\mathbf{x}} \leftarrow \underset{\hat{\mathbf{x}}}{\operatorname{argmin}} \frac{1}{2\sigma^2} \left\| \mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}) \right\|_2^2 + \frac{\rho}{2} \left\| \hat{\mathbf{x}} - \tilde{\mathbf{x}} + \mathbf{u} \right\|_2^2, \tag{20}$$

where $\tilde{\mathbf{x}} = \mathcal{D}(\tilde{\mathbf{z}})$. As in LATENT-DC, (20) can be solved via a single gradient step:

$$\hat{\mathbf{x}} = \tilde{\mathbf{x}} - \mathbf{u} - \gamma_t \nabla_{\mathbf{v}=\tilde{\mathbf{x}}-\mathbf{u}} \left\| \mathbf{y} - \mathcal{A}(\mathbf{v}) \right\|_2^2. \tag{21}$$

The corrected image $\hat{\mathbf{x}}$ is then encoded via $\mathbf{x} = \mathcal{E}(\hat{\mathbf{x}})$ for the next diffusion iteration.

Compared to LATENT-DC, the PIXEL-DC variant is considerably more efficient since it avoids back-propagation through the deep decoder $\mathcal{D}$. In practice, both methods are effective across various inverse problems; however, we find that PIXEL-DC yields consistently higher-quality reconstructions. Consequently, all quantitative results reported in Table 3 correspond to the PIXEL-DC variant. We provide the full algorithm in Algorithm 2.

Table 3. **Quantitative evaluation (latent diffusion).** Comparing latent-space methods across 5 linear and 2 nonlinear tasks. We use 100 validation images and report the average metric value. The best and second-best results are distinguished by **bold** and underlined marks, respectively. All tasks are run with a noise of standard deviation $\sigma = 0.05$. LatentDDiff achieves competitive performance relative to prior latent methods.

| | | FFHQ | | | | ImageNet | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Task** | **Method** | PSNR (↑) | SSIM (↑) | LPIPS (↓) | Residual (↓) | PSNR (↑) | SSIM (↑) | LPIPS (↓) | Residual (↓) |
| Super Resolution 4× | LatentDDiff | 26.67 | 0.688 | 0.324 | 0.0040 | **28.65** | 0.664 | 0.411 | **0.0043** |
| | LatentDAPS | **27.35** | **0.800** | **0.185** | **0.0038** | 25.01 | 0.669 | **0.284** | 0.0058 |
| | PSLD | 24.24 | 0.639 | 0.289 | 0.0060 | 25.40 | **0.689** | 0.362 | 0.0056 |
| | ReSample | 23.13 | 0.588 | 0.401 | 0.0065 | 22.57 | 0.567 | 0.382 | 0.0065 |
| Inpainting (Box) | LatentDDiff | 20.83 | 0.617 | 0.208 | 0.0192 | 19.10 | 0.551 | 0.337 | **0.0162** |
| | LatentDAPS | 23.86 | 0.799 | 0.192 | 0.0146 | 17.15 | 0.645 | 0.341 | 0.0372 |
| | PSLD | **24.15** | **0.800** | **0.163** | **0.0095** | **20.08** | **0.687** | 0.479 | 0.0390 |
| | ReSample | 19.56 | 0.621 | 0.190 | 0.0131 | 18.13 | 0.625 | **0.270** | 0.0214 |
| Inpainting (Random) | LatentDDiff | 26.81 | 0.643 | 0.239 | 0.0035 | 29.15 | 0.665 | 0.341 | **0.0038** |
| | LatentDAPS | **30.52** | **0.804** | 0.153 | **0.0033** | 27.31 | **0.769** | 0.177 | 0.0058 |
| | PSLD | 30.14 | 0.776 | 0.228 | 0.0037 | **30.23** | 0.760 | 0.340 | 0.0042 |
| | ReSample | 29.24 | 0.730 | **0.145** | 0.0036 | 27.03 | 0.721 | **0.155** | 0.0095 |
| Gaussian Deblurring | LatentDDiff | 27.56 | 0.725 | 0.252 | 0.0027 | **26.06** | **0.850** | 0.310 | **0.0027** |
| | LatentDAPS | **27.58** | **0.753** | **0.248** | **0.0026** | 24.67 | 0.637 | 0.356 | 0.0067 |
| | PSLD | 22.98 | 0.600 | 0.331 | 0.0064 | 25.03 | 0.656 | 0.411 | 0.0058 |
| | ReSample | 25.56 | 0.697 | 0.278 | 0.0045 | 25.45 | 0.683 | **0.267** | 0.0049 |
| Motion Deblurring | LatentDDiff | 26.11 | 0.593 | 0.286 | 0.0028 | 25.54 | 0.637 | 0.330 | 0.0038 |
| | LatentDAPS | 26.88 | 0.799 | 0.299 | 0.0027 | 26.34 | **0.721** | 0.304 | 0.0034 |
| | PSLD | 21.81 | 0.642 | 0.357 | 0.0036 | 20.23 | 0.566 | 0.525 | 0.0067 |
| | ReSample | **27.21** | **0.801** | **0.214** | **0.0025** | **26.65** | 0.706 | **0.244** | **0.0030** |
| Nonlinear Deblurring | LatentDDiff | 27.06 | 0.683 | 0.234 | **0.0050** | **26.80** | **0.711** | 0.265 | **0.0041** |
| | LatentDAPS | 27.87 | 0.695 | 0.250 | 0.0056 | 25.04 | 0.612 | 0.346 | 0.0057 |
| | ReSample | **28.04** | **0.722** | **0.199** | 0.0052 | 25.89 | 0.621 | **0.245** | 0.0044 |
| High Dynamic Range | LatentDDiff | 22.68 | 0.648 | 0.204 | **0.0145** | 21.69 | **0.662** | **0.201** | **0.0360** |
| | LatentDAPS | **25.66** | **0.737** | 0.255 | 0.0154 | 23.32 | 0.589 | 0.280 | 0.0366 |
| | ReSample | 25.21 | 0.702 | **0.197** | 0.0161 | **24.97** | 0.616 | 0.208 | 0.0381 |

## D. Statistical Significance

Building on the main results table, we report 95% confidence intervals for DDiff and DAPS in Table 4, highlighting the statistical significance of our improvement, compared to DAPS [43]. These statistics also serve as empirical evidence for the practical uniqueness of the fixed point established in Theorem 1; the narrow confidence intervals across all tasks indicate that independent initializations $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ yield tightly concentrated reconstructions.

Table 4. **Quantitative evaluation with confidence intervals.** We show the average metric values over 100 validation images and the corresponding 95% confidence intervals for DDiff and DAPS.

| | | FFHQ | | | | ImageNet | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Task** | **Method** | PSNR (↑) | SSIM (↑) | LPIPS (↓) | Residual (↓) | PSNR (↑) | SSIM (↑) | LPIPS (↓) | Residual (↓) |
| Super Resolution 4× | DDiff (ours) | **30.07**±0.41 | **0.824**±0.008 | 0.211±0.009 | (**2.85**±0.06)·$10^{-3}$ | **25.81**±0.72 | **0.656**±0.029 | 0.396±0.029 | (**3.83**±0.29)·$10^{-3}$ |
| | DAPS | 29.34±0.33 | 0.783±0.006 | **0.190**±0.006 | (2.97±0.06)·$10^{-3}$ | 25.44±0.56 | 0.636±0.017 | **0.295**±0.011 | (4.77±0.36)·$10^{-3}$ |
| Inpainting (Box) | DDiff (ours) | **24.88**±0.50 | **0.831**±0.005 | **0.110**±0.005 | (**7.75**±9.21)·$10^{-3}$ | 21.15±0.68 | **0.743**±0.009 | 0.240±0.011 | (**1.19**±0.29)·$10^{-2}$ |
| | DAPS | 24.52±0.40 | 0.742±0.006 | 0.174±0.006 | (9.86±6.76)·$10^{-3}$ | **21.22**±0.70 | 0.714±0.007 | **0.230**±0.011 | (1.50±0.24)·$10^{-2}$ |
| Inpainting (Random) | DDiff (ours) | **33.08**±0.37 | **0.877**±0.004 | **0.050**±0.003 | (**2.05**±0.15)·$10^{-2}$ | **28.39**±0.76 | **0.758**±0.019 | **0.136**±0.018 | (**2.41**±0.35)·$10^{-2}$ |
| | DAPS | 30.76±0.27 | 0.801±0.005 | 0.156±0.003 | (2.93±0.16)·$10^{-2}$ | 27.32±0.63 | 0.725±0.013 | 0.189±0.010 | (7.88±0.70)·$10^{-2}$ |
| Gaussian Deblurring | DDiff (ours) | 28.87±0.43 | **0.800**±0.010 | **0.119**±0.005 | (**2.60**±0.07)·$10^{-3}$ | 22.29±0.84 | 0.471±0.039 | 0.415±0.035 | (**4.68**±0.39)·$10^{-3}$ |
| | DAPS | **29.63**±0.36 | 0.789±0.006 | 0.177±0.005 | (2.67±0.06)·$10^{-3}$ | **25.90**±0.64 | **0.658**±0.020 | **0.269**±0.011 | (8.46±0.31)·$10^{-3}$ |
| Motion Deblurring | DDiff (ours) | 28.24±0.38 | 0.785±0.009 | **0.129**±0.005 | (**5.82**±0.27)·$10^{-3}$ | 24.16±0.65 | 0.585±0.027 | 0.242±0.014 | (**7.97**±0.60)·$10^{-3}$ |
| | DAPS | **29.17**±0.36 | **0.797**±0.007 | 0.186±0.005 | (5.91±0.28)·$10^{-3}$ | **26.61**±0.65 | **0.710**±0.020 | **0.241**±0.011 | (8.58±0.25)·$10^{-3}$ |
| Phase Retrieval | DDiff (ours) | **29.94**±0.88 | **0.816**±0.025 | **0.120**±0.019 | (**4.02**±0.22)·$10^{-3}$ | 18.54±1.23 | **0.494**±0.045 | **0.262**±0.029 | (**5.92**±0.24)·$10^{-3}$ |
| | DAPS | 29.60±0.88 | 0.768±0.020 | 0.182±0.018 | (4.94±0.25)·$10^{-3}$ | **20.23**±1.27 | 0.449±0.044 | 0.397±0.032 | (8.53±0.25)·$10^{-3}$ |
| Nonlinear Deblurring | DDiff (ours) | **31.48**±0.29 | **0.873**±0.005 | **0.120**±0.006 | (**2.74**±0.07)·$10^{-3}$ | **29.68**±0.62 | **0.805**±0.016 | **0.207**±0.019 | (**3.55**±0.18)·$10^{-3}$ |
| | DAPS | 28.45±0.38 | 0.764±0.007 | 0.188±0.006 | (4.16±0.43)·$10^{-3}$ | 27.28±0.62 | 0.718±0.017 | 0.213±0.011 | (4.87±0.33)·$10^{-3}$ |
| HDR | DDiff (ours) | 26.05±0.68 | **0.873**±0.007 | **0.129**±0.009 | (**4.59**±0.27)·$10^{-2}$ | **26.50**±0.56 | 0.800±0.018 | **0.108**±0.014 | (**5.41**±0.27)·$10^{-2}$ |
| | DAPS | **27.39**±0.56 | 0.846±0.010 | 0.163±0.012 | (5.05±0.18)·$10^{-2}$ | 26.10±0.79 | **0.825**±0.021 | 0.171±0.019 | (7.17±0.75)·$10^{-2}$ |

## E. Relating DDiff and DiffPIR

Here we show the equivalence of DDiff and DiffPIR (see Algorithm 3) under specific parameter choices. DiffPIR introduces $\zeta$ as a hyperparameter controlling the stochasticity of the algorithm and $\bar{\sigma}_t$ as a parameter controlling the strength of the likelihood step. Our algorithm is equivalent to the linearized version of DiffPIR when:

i. $\sigma_t = \sqrt{\zeta \cdot (1 - \bar{\alpha}_{t-1})}$;
ii. $\gamma_t = \frac{\bar{\sigma}_t^2}{2\lambda\sigma^2}$ where $\sigma$ is the measurement noise standard deviation;
iii. We remove early the relaxation parameter $t_0$;
iv. We remove the dual updates.

**Algorithm 3** DiffPIR

**Require:** $T$, $\mathcal{A}(\cdot)$, $\{\bar{\sigma}_t\}_{t=1}^T$, $\{\bar{\alpha}_t\}_{t=1}^T$, $\mathbf{s}_\theta$, $\mathbf{y}$, $\zeta$, $\lambda$, $\sigma$

Initialize $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
**for** $t = T-1$ **to** $0$ **do**
  $\mathbf{x}_0^{(t)} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t + (1 - \bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t)\right)$
  $\hat{\mathbf{x}}_0^{(t)} \leftarrow \mathbf{x}_0^{(t)} - \frac{\bar{\sigma}_t^2}{2\lambda\sigma^2} \nabla_{\mathbf{x}_0^{(t)}} \|\mathbf{y} - \mathcal{A}(\mathbf{x}_0^{(t)})\|^2$
  $\hat{\epsilon} \leftarrow \frac{1}{\sqrt{1-\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0^{(t)}\right)$
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_0^{(t)} + \sqrt{1 - \bar{\alpha}_{t-1}}(\sqrt{1-\zeta}\hat{\epsilon} + \sqrt{\zeta}\epsilon_t)$
**end for**
**return** $\mathbf{x}_0$

## F. DDiff Variants Algorithms

This section presents the precise algorithms corresponding to the three variants of DDiff discussed in the ablation study. Note that DDiff-HQS 6 is equivalent to DiffPIR 3 if we satisfy (i.-iii.) from Appendix E.

**Algorithm 4** Diff-PnP-HQS (no noise, no $\mathbf{u}$)

**Require:** $T, \mathcal{A}(\cdot), \{\sigma_t\}_{t=1}^T, \{\bar{\alpha}_t\}_{t=1}^T, \mathbf{s}_\theta, \mathbf{y}, \{\gamma_t\}_{t=1}^T, t_0$

Initialize $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
**for** $t = T-1$ **to** $0$ **do**
  $\mathbf{z} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t)\right)$
  $\mathbf{x}_{t-1} \leftarrow \mathbf{z} - \gamma_t \nabla_{\mathbf{z}} \|\mathbf{y} - \mathcal{A}(\mathbf{z})\|^2$
**end for**
**return** $\mathbf{x}_0$

**Algorithm 5** Diff-PnP-ADMM (no noise, with $\mathbf{u}$)

**Require:** $T, \mathcal{A}(\cdot), \{\sigma_t\}_{t=1}^T, \{\bar{\alpha}_t\}_{t=1}^T, \mathbf{s}_\theta, \mathbf{y}, \{\gamma_t\}_{t=1}^T, t_0$

Initialize $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \mathbf{u} = \mathbf{0}$.
**for** $t = T-1$ **to** $0$ **do**
  $\mathbf{z} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t + \mathbf{u} + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t + \mathbf{u}, t)\right)$
  $\mathbf{x}_{t-1} \leftarrow \mathbf{z} - \mathbf{u} - \gamma_t \nabla_{\mathbf{v}=\mathbf{z}-\mathbf{u}} \|\mathbf{y} - \mathcal{A}(\mathbf{v})\|^2$
  $\mathbf{u} \leftarrow \mathbf{u} + \mathbf{x}_{t-1} - \mathbf{z}$
**end for**
**return** $\mathbf{x}_0$

**Algorithm 6** DDiff-HQS (with noise, no $\mathbf{u}$)

**Require:** $T, \mathcal{A}(\cdot), \{\sigma_t\}_{t=1}^T, \{\bar{\alpha}_t\}_{t=1}^T, \mathbf{s}_\theta, \mathbf{y}, \{\gamma_t\}_{t=1}^T, t_0$

Initialize $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
**for** $t = T-1$ **to** $0$ **do**
  $\mathbf{z} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t)\right)$
  $\mathbf{x} \leftarrow \mathbf{z} - \gamma_t \nabla_{\mathbf{z}} \|\mathbf{y} - \mathcal{A}(\mathbf{z})\|^2$
  $\hat{\epsilon} \leftarrow \frac{1}{\sqrt{1-\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}\right)$
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ **if** $t > t_0$ **else** $\epsilon = 0$
  $\mathbf{x}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \cdot \mathbf{x} + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \hat{\epsilon} + \sigma_t \epsilon$
**end for**
**return** $\mathbf{x}_0$

## G. Inverse Problems Setup

Most inverse problems are implemented using the same approach described in [7], except for the HDR task which follows the setup in [43]. We set a fixed random seed for inpainting, motion deblurring, and nonlinear deblurring for fair comparison. Specific parameters are defined as follows:

- Super resolution: $4\times$ downsampling factor
- Inpainting: $128 \times 128$ box mask and $70\%$ random mask
- Gaussian & motion deblurring: $61 \times 61$ kernel size with standard deviations of $3.0$ and $0.5$, respectively
- Phase retrieval: oversampling with ratio $k/n$ where $k = 2$ and $n = 8$
- Nonlinear deblurring: blur kernel generated using [38]
- HDR: $2\times$ dynamic range

## H. Hyperparameter Choices

The step size $\gamma_t$ for the measurement step is defined by a step function below:

$$\gamma_t = \gamma_0 \cdot f(t_\gamma) \text{ where } f(t_\gamma) = \begin{cases} a & \text{for } t > t_\gamma, \\ b & \text{for } t \leq t_\gamma. \end{cases}$$

To enhance sample quality, the parameter $\gamma_t$ is reduced at time step $t_\gamma$. This adjustment encourages the traversal to relax toward the prior manifold while diverging from the likelihood manifold. As a result, the generated samples exhibit reduced noise and improved visual fidelity. For all pixel-space diffusion tasks, $a = 3.3$ and $b = 0.1$ worked well empirically. For latent-space diffusion tasks, we report the exact values in Table 5 and 6.

We used two different choices of $\sigma_t$ in the main experiments.

- $\sigma_t = \sqrt{(1-\bar{\alpha}_{t-1})/(1-\bar{\alpha}_t)}\sqrt{1-\bar{\alpha}_t/\bar{\alpha}_{t-1}}$ (this choice allows the generative process to become a DDPM) was used for Gaussian deblurring, motion deblurring, nonlinear deblurring, and HDR.
- $\sigma_t = \sqrt{1-\bar{\alpha}_{t-1}}$ was used for phase retrieval, inpainting (box and random), and super resolution.

The time threshold $t_0$ controls the reverse diffusion step to switch from stochastic to deterministic.

We report the hyperparameters used in the main experiments in Table 5 and 6.

Table 5. **DDiff hyperparameter settings** for different tasks on FFHQ.

| Algorithm | Task | Super Res. 4× | Inpaint (Box) | Inpaint (Rand.) | Gaussian Deblur | Motion Deblur | Phase Retrieval | Nonlinear Deblur | HDR |
|---|---|---|---|---|---|---|---|---|---|
| DDiff | $\gamma_0$ | 18 | 30 | 50 | 2.9 | 2.9 | 38 | 2.5 | 3.5 |
| | $t_\gamma$ | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 |
| | $t_0$ | 1 | 1 | 1 | 50 | 80 | 1 | 120 | 120 |
| LatentDDiff | $\gamma_0$ | 100 | 200 | 400 | 60 | 60 | - | 500 | 150 |
| | $t_\gamma$ | 90 | 600 | 200 | 90 | 150 | - | 90 | 600 |
| | $t_0$ | 500 | 1 | 800 | 200 | 200 | - | 200 | 1 |
| | $a$ | 3.3 | 25 | 30 | 3.3 | 3.3 | - | 3.3 | 30 |
| | $b$ | 0.1 | 0.15 | 0.1 | 0.1 | 0.1 | - | 0.1 | 0.15 |

Table 6. **DDiff hyperparameter settings** for different tasks on ImageNet.

| Algorithm | Task | Super Res. 4× | Inpaint (Box) | Inpaint (Rand.) | Gaussian Deblur | Motion Deblur | Phase Retrieval | Nonlinear Deblur | HDR |
|---|---|---|---|---|---|---|---|---|---|
| DDiff | $\gamma_0$ | 18 | 50 | 50 | 1.8 | 1.5 | 38 | 2.5 | 3.8 |
| | $t_\gamma$ | 90 | 500 | 90 | 90 | 90 | 90 | 90 | 90 |
| | $t_0$ | 1 | 1 | 1 | 50 | 80 | 1 | 120 | 100 |
| LatentDDiff | $\gamma_0$ | 150 | 400 | 400 | 150 | 100 | - | 100 | 250 |
| | $t_\gamma$ | 600 | 600 | 600 | 600 | 90 | - | 90 | 600 |
| | $t_0$ | 1 | 1 | 1 | 1 | 200 | - | 200 | 200 |
| | $a$ | 50 | 30 | 50 | 50 | 3.3 | - | 3.3 | 25 |
| | $b$ | 0.15 | 0.1 | 0.15 | 0.15 | 0.1 | - | 0.1 | 0.15 |

## I. Baseline Details

We follow the baseline configurations reported in DAPS [43] for all methods we compare against: DPS [7], DDRM [22], DCDP [24], DiffPIR [46], RED-diff [27], PSLD [30], and ReSample [32]. For DAPS and DMPlug [40], we use the authors' official implementations with default hyperparameters. All baselines use the same pretrained models (as specified in Sec. 4) and forward operators (Sec. G).

In our experiments, we use DDiff-1k for all inverse problems. In contrast, DAPS uses DAPS-1k for linear tasks and DAPS-4k for nonlinear tasks by default, and we retain these defaults (including hyperparameters) in our comparisons. Under these settings, DDiff attains higher reconstruction quality (PSNR/LPIPS; see Table 1) while requiring roughly half the sampling time on linear tasks and about one-sixth the sampling time on nonlinear tasks (see Fig. 5).

## J. More Qualitative Results

We provide additional qualitative results in Fig. 6 to 13.

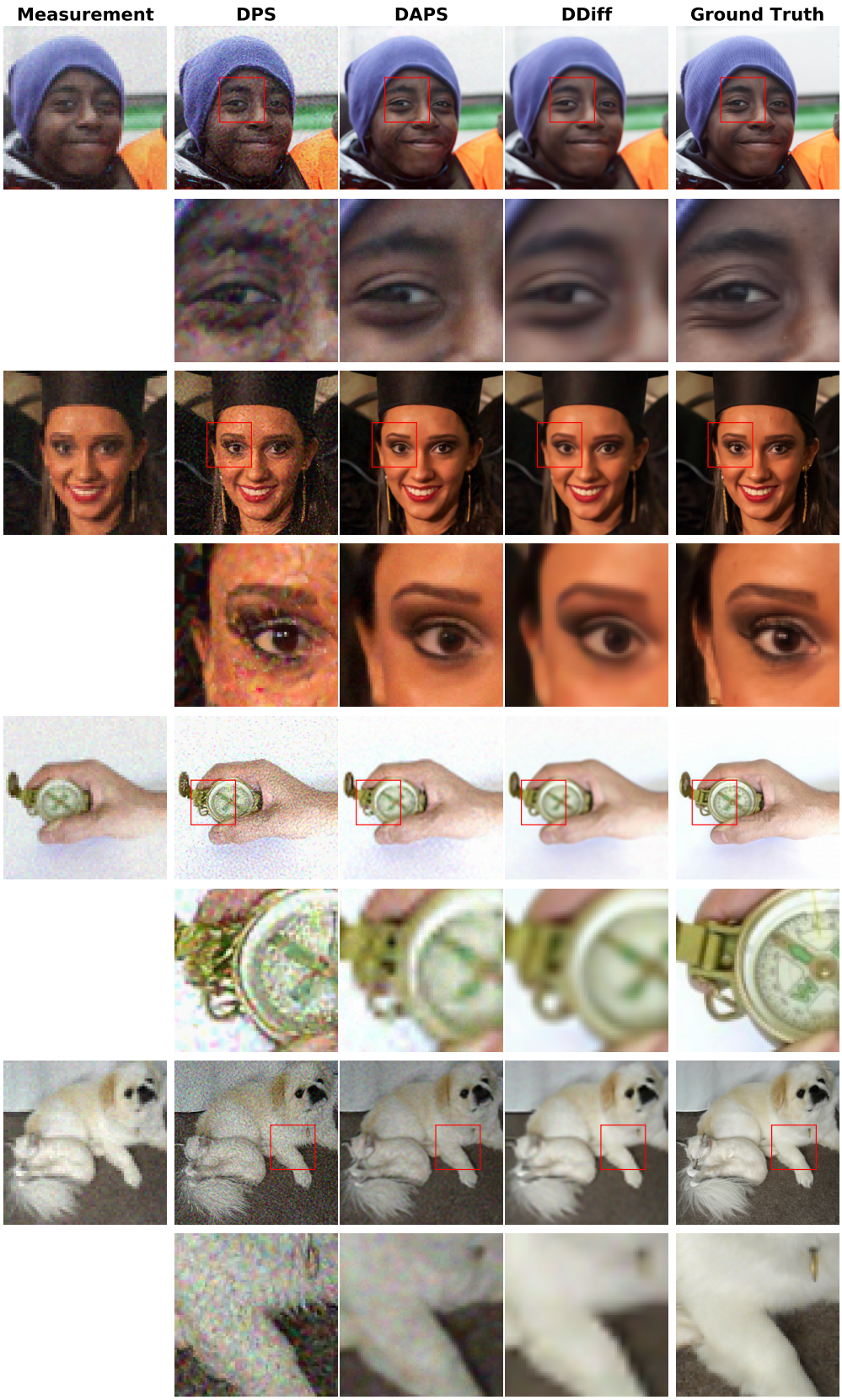


Figure 6. **Visual comparison of DDiff and baselines** on super resolution task.

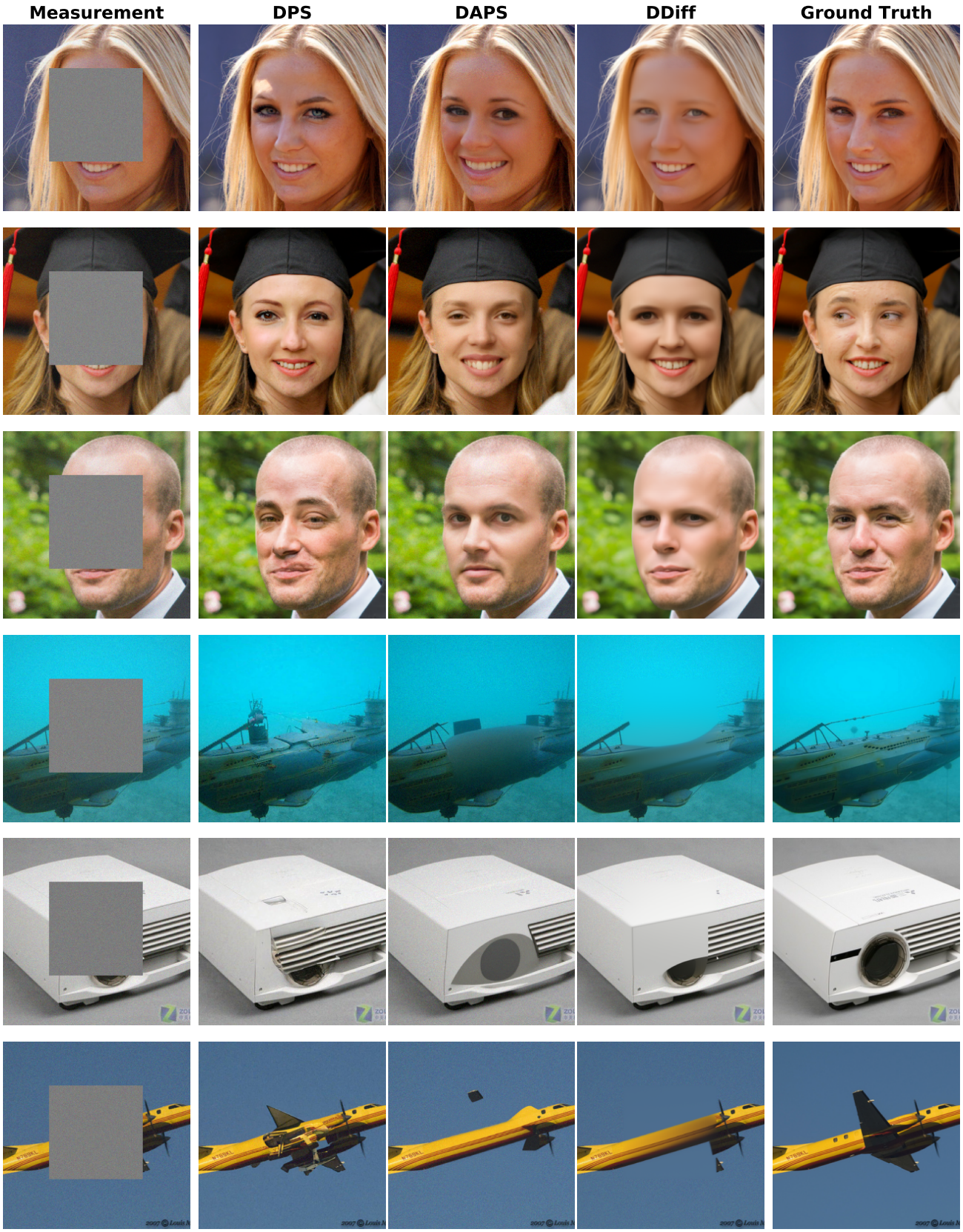


Figure 7. **Visual comparison of DDiff and baselines** on box inpainting task.

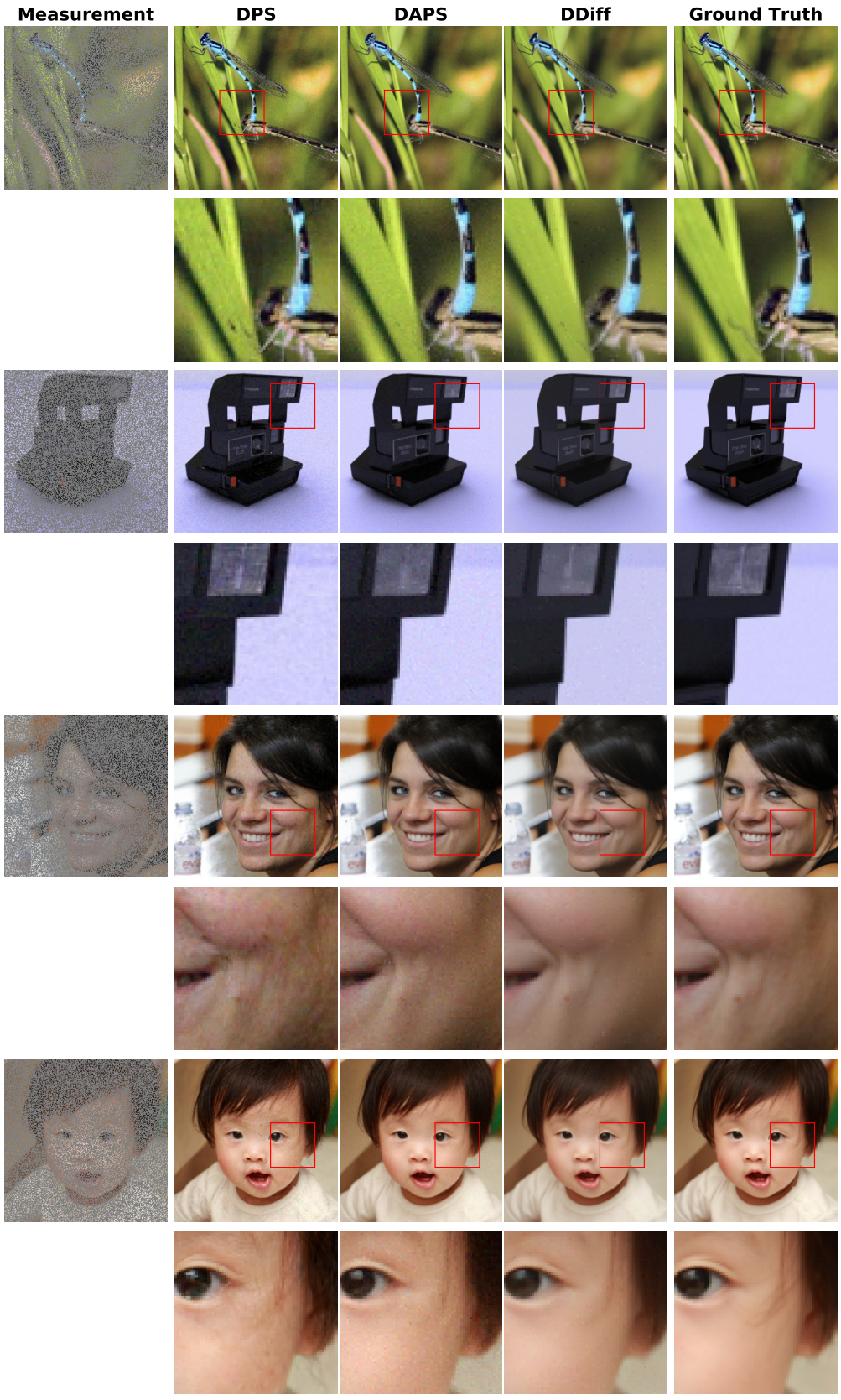


Figure 8. **Visual comparison of DDiff and baselines** on random inpainting task.

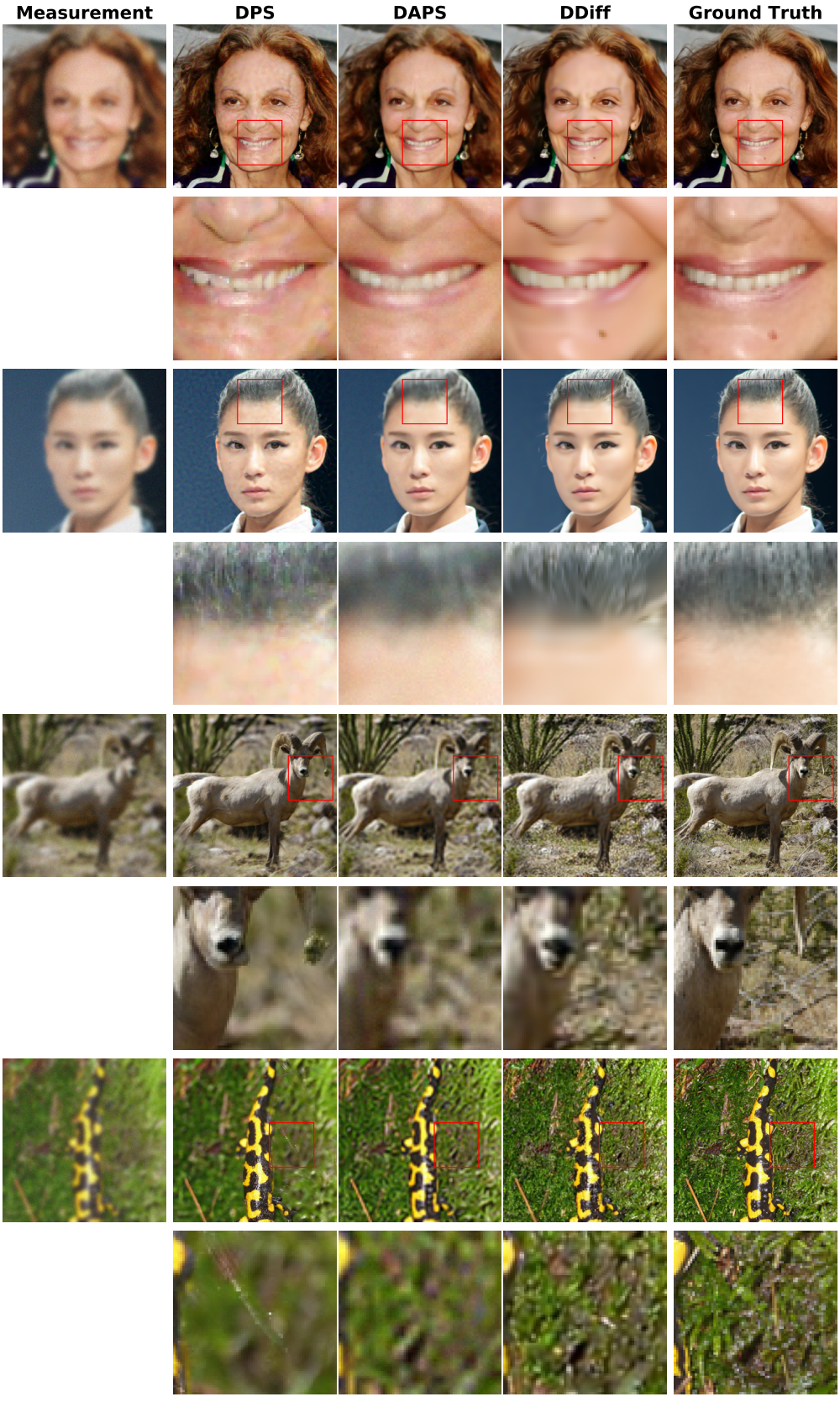


Figure 9. **Visual comparison of DDiff and baselines** on Gaussian deblurring task.

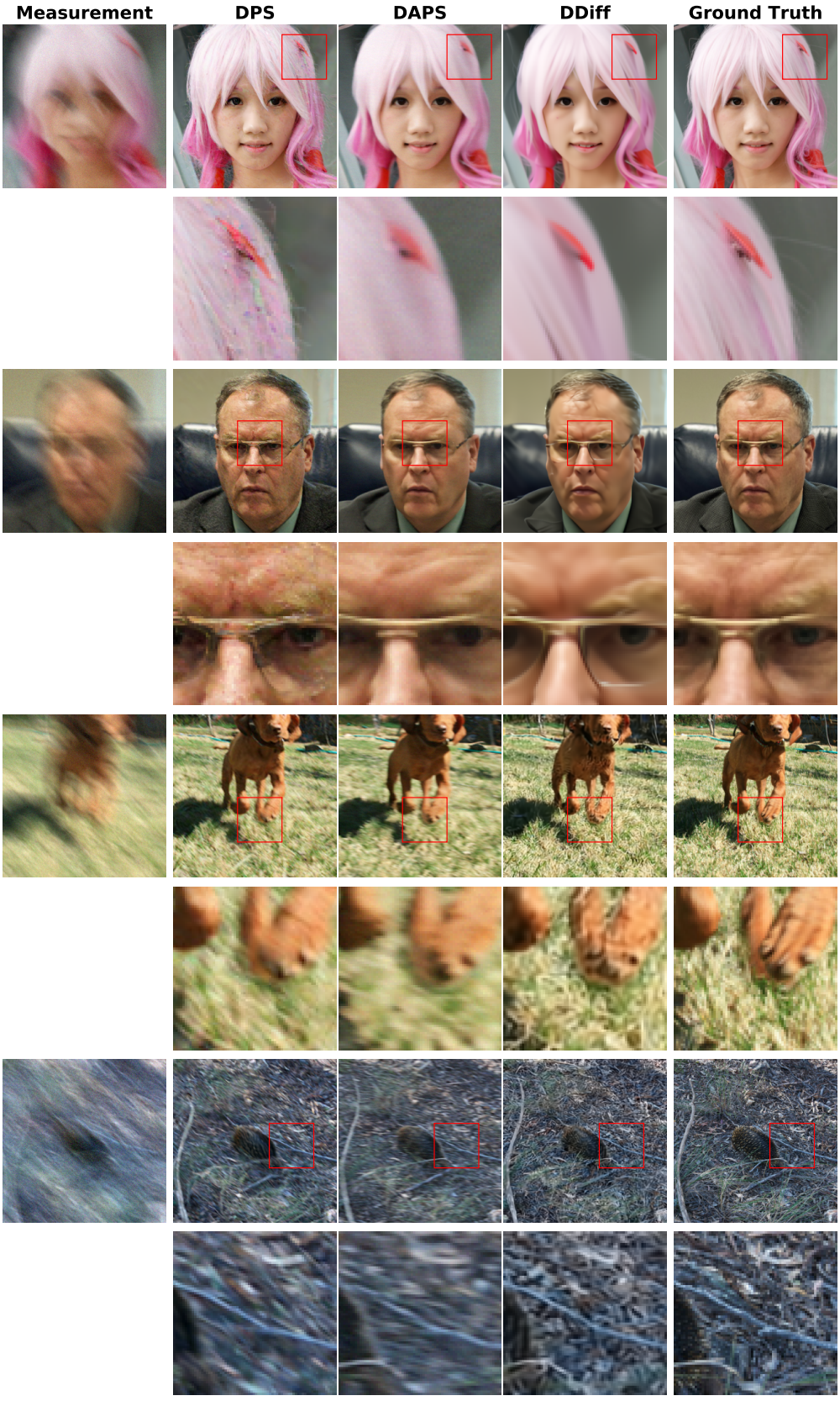


Figure 10. **Visual comparison of DDiff and baselines** on motion deblurring task.

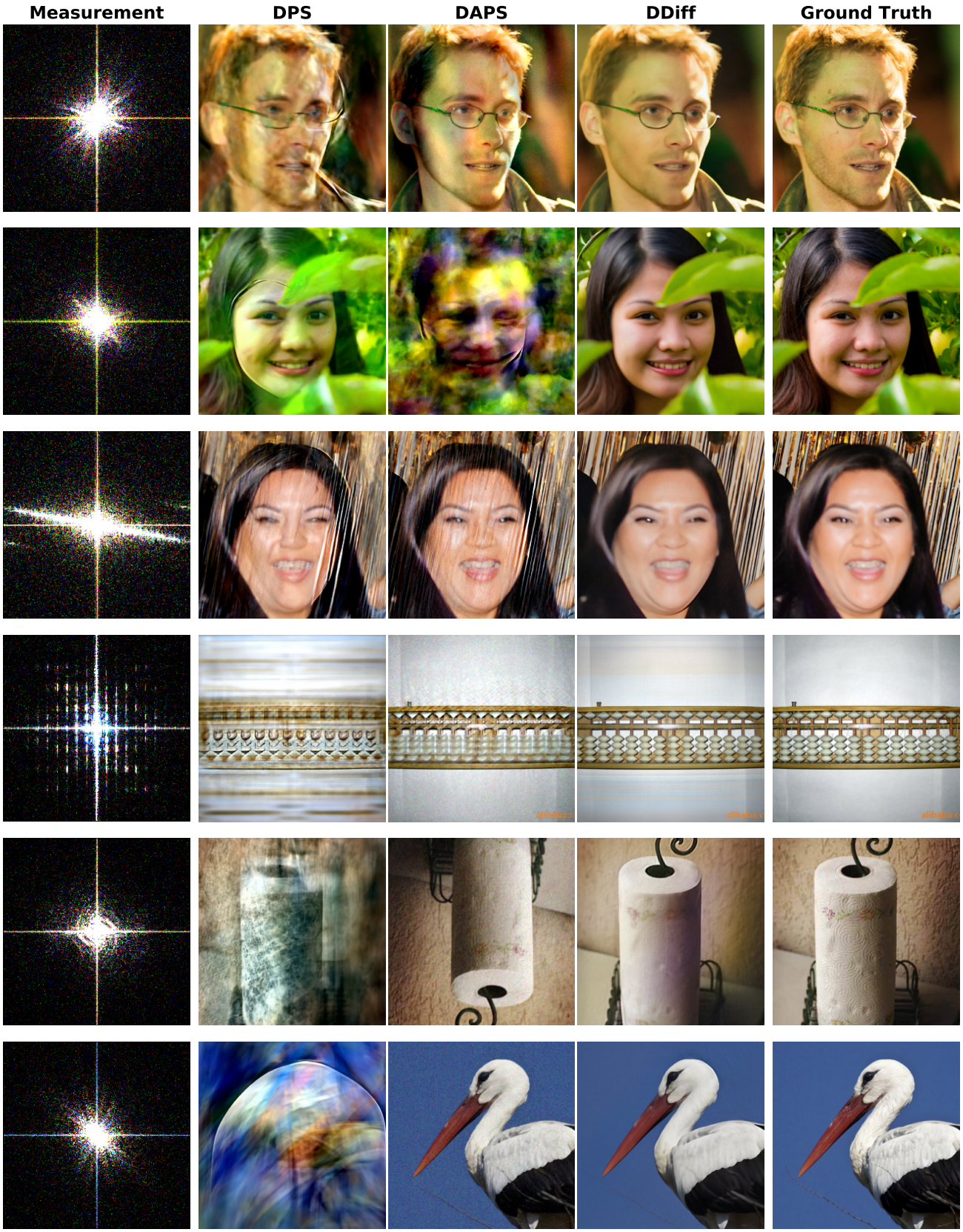


Figure 11. **Visual comparison of DDiff and baselines** on phase retrieval task.

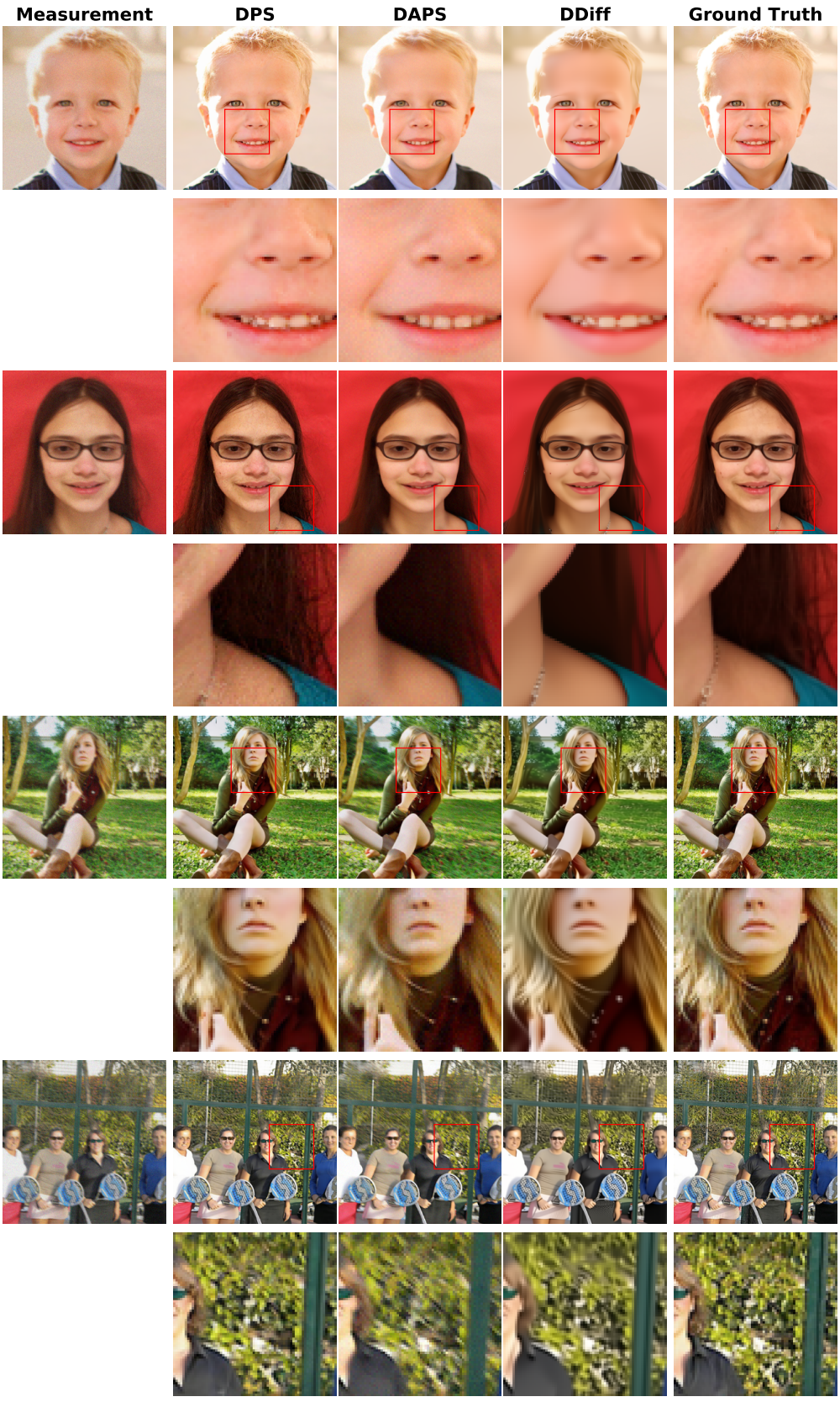


Figure 12. **Visual comparison of DDiff and baselines** on nonlinear deblurring task.

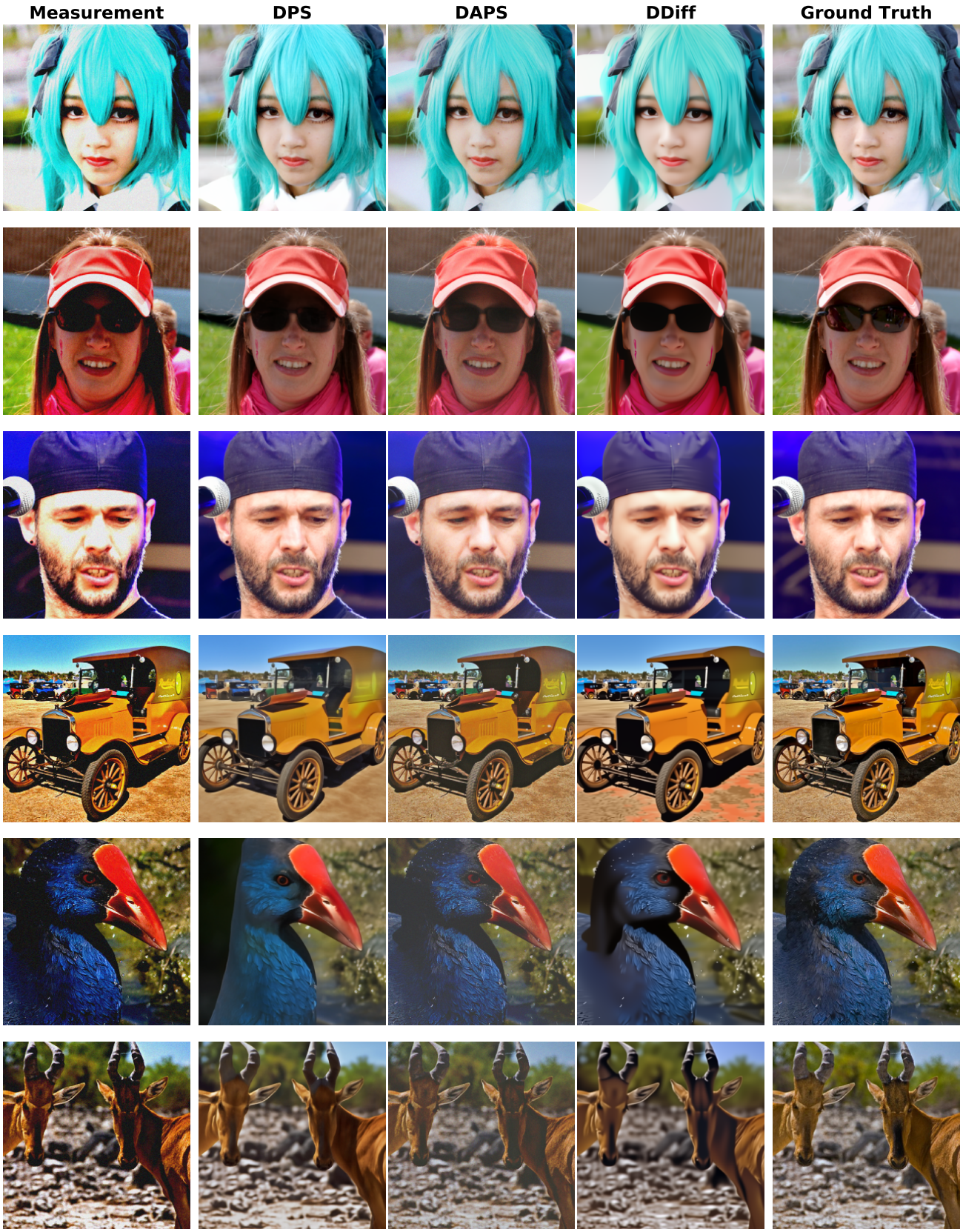


Figure 13. **Visual comparison of DDiff and baselines** on HDR task.