OPERATOR-ADAPTIVE MEAN-TARGET GUIDANCE FOR LINEAR INVERSE PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Training-free samplers for inverse problems construct guidance from a residual evaluated at an *evaluation target*, a point along the generative trajectory. Since the measurement is defined from the clean image through a known operator, most existing methods form the residual at a clean-image estimate. Some methods instead map the measurement to the current sampling step and form the residual at the current state. Although existing methods construct guidance in different ways, they typically treat the evaluation target as given rather than derive it. We introduce **Mean-Target Guidance (MTG)**, which derives operator-adaptive evaluation targets. A single frozen model evaluation produces a continuum of virtual targets. The gradient of a loss that aggregates their weighted squared residuals reduces exactly to one residual at their weighted mean, the *mean-target*. Inverse-variance weighting over the virtual targets yields a mean-target for each singular direction. While the mean-target determines where the residual is evaluated, a direction-specific *damping factor* controls how strongly that residual updates the current state. The construction applies to both diffusion and flow-matching backbones. MTG outperforms strong training-free baselines across standard linear inverse problems. An ablation study separates the effects of the mean-target and damping factor. We train the evaluation targets and update strengths, highlighting the importance of where the residual is evaluated and how strongly it acts on the state. Code is available at `https://anonymous.4open.science/r/MTG2027`.

1 INTRODUCTION

Consider the inverse problem of recovering a clean image $x_\star \in \mathbb{R}^d$ from a corrupted measurement $y \in \mathbb{R}^m$, where $x_\star$ follows a natural image distribution p_{data}. Across a wide range of inverse problem tasks, measurements follow the linear observation model,

$$y = Ax_\star + n, \qquad n \sim \mathcal{N}(0, \sigma_y^2 I_m), \tag{1.1}$$

where $A \in \mathbb{R}^{m\times d}$ is a known operator and n is additive Gaussian measurement noise.

In general, Eq. (1.1) is ill-posed (Hansen, 1998; Engl et al., 1996). What A removes from $x_\star$ is absent from y, and only a prior over images can supply it. Training-free samplers take the prior from a pretrained diffusion model such as EDM (Ho et al., 2020; Song et al., 2021; Karras et al., 2022) or a flow-matching model on the straight Optimal-Transport (OT) path (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023). The methods differ in how they incorporate the measurement during sampling. DDNM (Wang et al., 2023) and DDRM (Kawar et al., 2022) set the measurement-consistent part of the state directly, the first through a pseudo-inverse and the second through the singular value decomposition (SVD) of A. DPS (Chung et al., 2023) takes a single gradient step on the residual of a clean-image estimate, while ΠGDM (Song et al., 2023) and OT-ODE (Pokle et al., 2024) weight the residual through the operator's pseudo-inverse. LMAPS (Zhang et al., 2026) and ICTM (Zhang et al., 2024) solve a local maximum-a-posteriori objective around the estimate at each step. MGPS (Moufad et al., 2025) optimizes a variational approximation of the reverse transition, evaluating the potential at an intermediate state along it. DiffPIR (Zhu et al., 2023), PnP-Flow (Martin et al., 2025) and FLOWER (Pourya et al., 2026) alternate a denoising step with an explicit data-consistency update, where the denoiser takes the place of the proximal operator of an image prior. DAPS (Zhang et al., 2025) and PnP-DM (Wu et al., 2024) sample the clean image under an annealed noise level rather than updating the state at each step.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

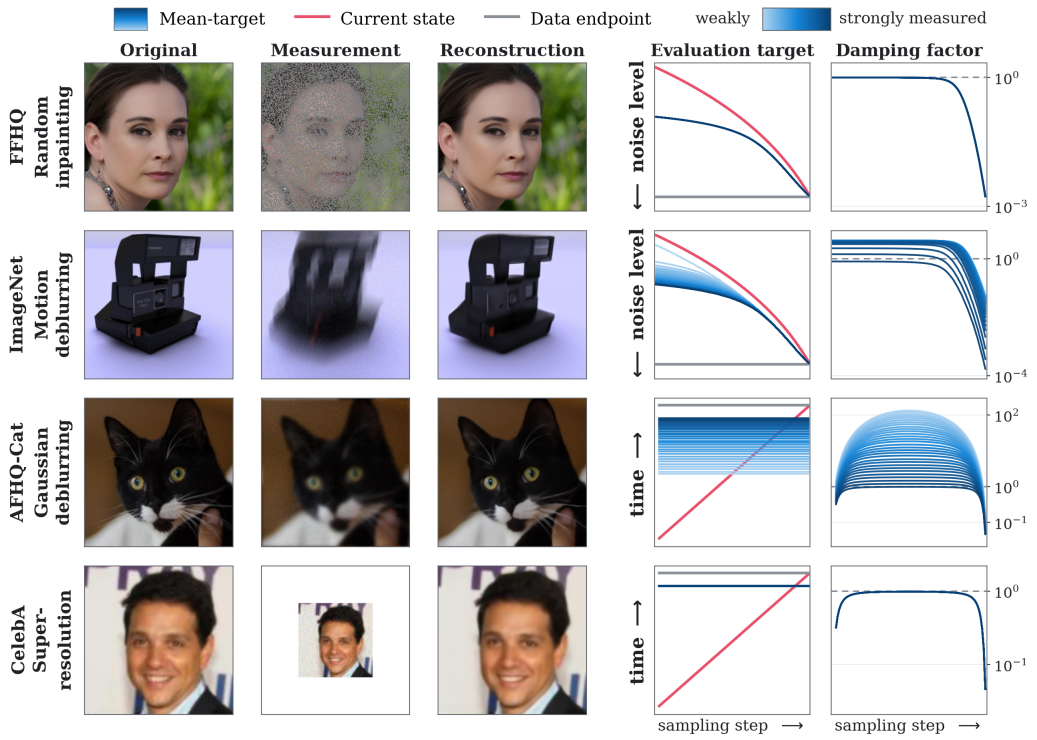

Figure 1: The derived pair at every sampling step for four operators across four datasets. The top two rows run on a diffusion backbone and the bottom two on flow matching. The mean-target and the damping factor both change with the operator. For diffusion, the target is a noise level ranging from 10^2 down to 10^{-2}, whereas for flow matching, it is a time ranging from 0 to 1.

Although the methods above incorporate the measurement in different ways, they all form the residual at a clean-image estimate. The construction of a sampling step shows where it comes from. For the trajectory state x_t at sampling time t, Bayes' rule expresses the score of the posterior $p(x_t \mid y)$ as $\nabla_{x_t} \log p(x_t \mid y) = \nabla_{x_t} \log p(y \mid x_t) + \nabla_{x_t} \log p(x_t)$. The pretrained model supplies the prior score $\nabla_{x_t} \log p(x_t)$, while each sampling step must construct $\nabla_{x_t} \log p(y \mid x_t)$, the *guidance term*. Computing it exactly means averaging the measurement likelihood over the clean images that could have produced x_t, which the sampler cannot do. Under the Gaussian noise model of Eq. (1.1), the likelihood depends on the clean image only through the residual $y - Ax_\star$. Existing methods make this residual tractable in one of two ways. Most methods replace $x_\star$ with a clean-image estimate and form the residual at the data endpoint. FIG (Yan et al., 2025) and FPS (Dou & Song, 2024) instead map the measurement to the current sampling time using Eq. (1.1) and form the residual at the current state. We call the point where the residual is formed the *evaluation target*, and it is the data endpoint or the current state in all these methods.

However, neither Eq. (1.1) nor the exact guidance term prescribes where a computable residual should be evaluated during sampling. The evaluation target influences how the residual guides the state. Even with the current state and model prediction held fixed, changing the target can change the residual and the resulting update. In existing methods, however, the target is determined by the approximation used to make the residual tractable. Therefore, we treat the evaluation target as a main consideration in constructing guidance, allowing the construction itself to determine it. To leave the evaluation target unspecified, we construct a residual over a continuum of points along the trajectory. This residual allows us to derive the evaluation target. The construction proceeds as follows:

1. **The mean-target.** We prove that computing the gradient of the newly defined weighted squared-residual loss over a continuum of virtual targets reduces to evaluating a single residual. The evaluation target is their weighted mean, which we call the *mean-target*.

2. **One mean-target per measured direction.** We model the weight as the inverse of the residual variance. Since the variance depends on the operator spectrum, the resulting mean-targets are operator-adaptive and direction-specific.
3. **One damping factor per measured direction.** After evaluating the residual at each direction's mean-target, we use a *damping factor* to control the magnitude of its contribution to the state update. The factor accounts for the singular value and remaining state uncertainty.

By defining the residual first, we introduce **Mean-Target Guidance (MTG)**, a new formulation of guidance for inverse problems in which the evaluation target itself is derived rather than given. Figure 1 shows how the resulting mean-targets and damping factors vary across operators and measured directions throughout sampling, with the current state and data endpoint shown as reference targets. This guidance with operator-adaptive evaluation targets and update strengths achieves strong reconstruction performance across diverse inverse problems.

2 PRELIMINARIES

Throughout the paper, the forward map in Eq. (1.1) is a known *linear* operator. Accordingly, A preserves linear combinations, $A(\alpha u + \beta w) = \alpha Au + \beta Aw$. This equality does not generally hold for nonlinear forward maps. For clarity, we present the derivation on the OT path in the main text; the corresponding EDM derivation follows the same approach and is given in Appendix D.

The affine path. For flow matching, we use the straight OT path (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023). At sampling time t, the interpolated state on this path is $x_t^{\text{int}} = tx_\star + (1-t)\varepsilon_x$, where $\varepsilon_x \sim \mathcal{N}(0, I_d)$. The path runs from noise at $t = 0$ to the clean image at $t = 1$. A sampler holds the state x_t and one velocity evaluation $v_\theta(x_t, t)$. For the diffusion backbone, we use EDM (Karras et al., 2022). The EDM path is parameterized by noise level σ, with state $x_\sigma = x_\star + \sigma\varepsilon_x$. EDM is not defined as an interpolation between noise and data. Nevertheless, both the OT and EDM paths are affine in their respective coordinates, and applying A preserves this affine dependence. Details of the pretrained prior models are provided in Appendix B.

Directions of the measurement. A linear A also separates what the measurement reaches from what it discards. Its compact SVD $A = U\Sigma V^\top = \sum_{s=1}^{R} \sigma_s u_s v_s^\top$ satisfies $Av_s = \sigma_s u_s$ and $A^\top u_s = \sigma_s v_s$, with $R = \operatorname{rank}(A)$ and $\sigma_s > 0$. The left singular vector $u_s \in \mathbb{R}^m$ is a direction of the measurement, and the corresponding direction is $v_s \in \mathbb{R}^d$. Changing the state by δv_s changes the measurement by $A(\delta v_s) = \delta\sigma_s u_s$, without changing any component along $u_{s'}$ for $s' \neq s$. The measured directions can be analyzed independently. Using $u_s^\top A = \sigma_s v_s^\top$, the projections of the OT state and measurement onto the paired directions v_s and u_s are

$$v_s^\top x_t^{\text{int}} = t\, v_s^\top x_\star + (1-t)\, v_s^\top \varepsilon_x, \qquad u_s^\top y = \sigma_s\, v_s^\top x_\star + u_s^\top n, \tag{2.1}$$

respectively. Since ε_x and n are independent isotropic Gaussians, the projected noise terms are independent across s and between the two noise sources, with $v_s^\top \varepsilon_x \sim \mathcal{N}(0,1)$ and $u_s^\top n \sim \mathcal{N}(0, \sigma_y^2)$. Thus, each direction reduces to a scalar inverse problem. $v_s^\top x_t^{\text{int}}$ interpolates from $v_s^\top \varepsilon_x$ to the clean-image coefficient $v_s^\top x_\star$, while $u_s^\top y$ measures that coefficient with singular value σ_s and additive Gaussian measurement noise n.

Forming the guidance term. Since $p(y \mid x_t) = \int p(y \mid x_\star) p(x_\star \mid x_t) dx_\star$ integrates over the unknown clean image, the guidance term $\nabla_{x_t} \log p(y \mid x_t)$ cannot be evaluated directly. Under the Gaussian noise model of Eq. (1.1), the negative log-likelihood of a clean image is $\|y - Ax_\star\|^2/(2\sigma_y^2)$ up to an additive constant. Existing methods obtain a tractable residual at one of the two evaluation targets introduced above. At the data-endpoint target, $x_\star$ is replaced by the conditional-mean estimate $\hat{x}(x_t) := \mathbb{E}[x_\star \mid x_t] = x_t + (1-t)v_\theta(x_t, t)$, obtained from one velocity evaluation through Tweedie's formula (Efron, 2011). This gives the data-endpoint residual $y - A\hat{x}(x_t)$. At the current-state target, methods define the time-matched measurement $y_t^{\text{int}} = ty + (1-t)A\varepsilon_x$. For the ideal OT state x_t^{int}, Eq. (1.1) gives $y_t^{\text{int}} - Ax_t^{\text{int}} = tn$, so the ideal current-state residual consists only of scaled measurement noise. During sampling, the available state x_t is used in place of x_t^{int}, giving the current-state residual $y_t^{\text{int}} - Ax_t$. Then the two evaluation targets give the squared losses

$$\mathcal{L}_{\text{data}}(x_t) := \|y - A\hat{x}(x_t)\|^2, \qquad \mathcal{L}_{\text{current}}(x_t) := \|y_t^{\text{int}} - Ax_t\|^2. \tag{2.2}$$

More generally, residual-based training-free methods use a guidance update of the form

$$x_t^{\text{new}} = x_t - cf(x_t, t, y, A, \sigma_y, \ldots)\nabla_{x_t}\mathcal{L}(x_t), \tag{2.3}$$

where $\mathcal{L}$ denotes the method's residual loss, f represents method-specific scaling or preconditioning, and c is a tuning parameter that controls the overall guidance strength. Some methods also apply an additional update alongside this gradient term. The sampler applies this update together with the prior update from the pretrained model at each sampling step.

3 METHODOLOGY

We introduce **Mean-Target Guidance (MTG)**, which derives the evaluation target and update strength. Section 3.1 derives the mean-target from a loss over the sampling path. Sections 3.2 and 3.3 derive one mean-target and one damping factor for each measured direction, respectively. Section 3.4 presents the sampling procedure and the corresponding EDM formulation.

3.1 THE MEAN-TARGET

We define a residual at any time τ of the path from the state and the measurement each taken there,

$$\begin{aligned} x_\tau^{\text{ext}} &:= \mathbb{E}[x_\tau^{\text{int}} \mid x_t] = x_t + (\tau - t)\,\text{sg}[v_\theta(x_t, t)], \qquad y_\tau^{\text{ext}} := \mathbb{E}[y_\tau^{\text{int}} \mid y] = \tau y, \\ r^{\text{ext}}(\tau) &:= Ax_\tau^{\text{ext}} - y_\tau^{\text{ext}} = Ax_t - ty + (\tau - t)\big(A\,\text{sg}[v_\theta] - y\big). \end{aligned} \tag{3.1}$$

Beyond x_t, the sampler has one velocity, and keeping it fixed extends the state along the line x_τ^{ext} to any time τ. The $\text{sg}[\cdot]$ in Eq. (3.1) also excludes the velocity from the gradient in x_t. x_τ^{int} and y_τ^{int} depend on $x_\star$ or ε_x, and Eq. (3.1) removes them by taking the conditional mean of each. A residual is then available at every τ, and we call each τ a *virtual target*. The line returns the current state x_t at $\tau = t$ and the data endpoint $\hat{x}(x_t)$ at $\tau = 1$.

Since $r^{\text{ext}}(\tau)$ is not constant in τ, different virtual targets give different residuals. There is no criterion for choosing among them, and a choice would fix the evaluation target. We keep every virtual target instead and attach a nonnegative weight to each.

Definition 3.1. *Let $\mathcal{T} \subseteq [0,1]$ and let $w(\tau) \geq 0$ for every $\tau \in \mathcal{T}$ with $0 < \int_{\mathcal{T}} w < \infty$. The loss is*

$$\mathcal{L}_w(x_t) := \frac{1}{2}\int_{\mathcal{T}} w(\tau)\,\big\|r^{\text{ext}}(\tau)\big\|^2\,d\tau. \tag{3.2}$$

Equation (3.2) is the squared loss of Eq. (2.2) widened from a point to an interval. The definition covers any nonnegative w with finite positive mass, on any $\mathcal{T} \subseteq [0,1]$. The loss uses every residual on $\mathcal{T}$, and the result below shows that its gradient needs only one.

Theorem 3.2. *Fix x_t and the velocity frozen in Eq. (3.1). The gradient of Eq. (3.2) is built from a single residual, formed at the mean-target $\bar{\tau}_w$,*

$$\nabla_{x_t}\mathcal{L}_w = \Big(\int_{\mathcal{T}} w\Big)\,A^\top r^{\text{ext}}(\bar{\tau}_w), \qquad \bar{\tau}_w := \mathbb{E}_{\mu_w}[\tau], \qquad \mu_w := w\Big/\int_{\mathcal{T}} w. \tag{3.3}$$

The gradient depends on $\mathcal{T}$ and w only through the pair $\{\bar{\tau}_w, \int_{\mathcal{T}} w\}$, one virtual target and one positive scalar. Any weight w with the same pair produces the same gradient.

Proof. Write $r^{\text{ext}}(\tau) = a + b\tau$ with $b := A\,\text{sg}[v_\theta] - y$ and $a := Ax_t - ty - tb$, neither depending on τ. The stop-gradient holds b fixed in the state. The Jacobian $\partial r^{\text{ext}}(\tau)/\partial x_t$ is then A at every τ. With $\int_{\mathcal{T}} \mu_w = 1$ and $\int_{\mathcal{T}} \tau\,\mu_w(\tau)d\tau = \bar{\tau}_w$,

$$\begin{aligned} \nabla_{x_t}\mathcal{L}_w = \frac{1}{2}\int_{\mathcal{T}} w(\tau)\,\nabla_{x_t}\big\|r^{\text{ext}}(\tau)\big\|^2 d\tau \;&=\; \int_{\mathcal{T}} w(\tau)\Big(\frac{\partial r^{\text{ext}}(\tau)}{\partial x_t}\Big)^\top r^{\text{ext}}(\tau)\,d\tau \\ = \int_{\mathcal{T}} w(\tau)\,A^\top r^{\text{ext}}(\tau)\,d\tau \;&=\; \Big(\int_{\mathcal{T}} w\Big)A^\top \int_{\mathcal{T}} \frac{w(\tau)}{\int_{\mathcal{T}} w}\,r^{\text{ext}}(\tau)\,d\tau \\ = \Big(\int_{\mathcal{T}} w\Big)A^\top \int_{\mathcal{T}} \mu_w(\tau)\,(a + b\tau)\,d\tau \;&=\; \Big(\int_{\mathcal{T}} w\Big)A^\top\Big(a + b\int_{\mathcal{T}} \tau\,\mu_w(\tau)d\tau\Big) \\ = \Big(\int_{\mathcal{T}} w\Big)A^\top(a + b\,\bar{\tau}_w) \;&=\; \Big(\int_{\mathcal{T}} w\Big)A^\top r^{\text{ext}}(\bar{\tau}_w). \end{aligned}$$

□

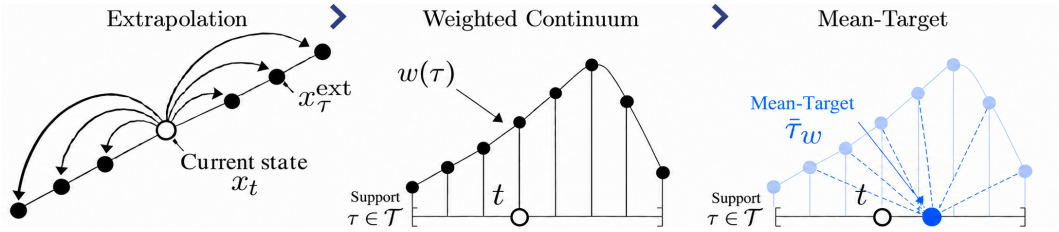

Figure 2: Mean-target construction. A frozen velocity extends the current state x_t to virtual targets x_τ^{ext} at arbitrary times τ (left). Over the support $\mathcal{T}$, each residual at τ is assigned a weight $w(\tau)$ (middle). The gradient of the weighted residual loss reduces exactly to a single residual evaluated at the weighted mean $\bar{\tau}_w$, the mean-target (right).

Figure 2 illustrates the mean-target construction. Importantly, the mean-target is the weighted mean $\bar{\tau}_w$, rather than the maximizer of $w(\tau)$. This reduction relies on the frozen-velocity construction: reevaluating the velocity at each τ breaks the affine dependence of the residual on τ, while differentiating through the velocity makes its Jacobian τ-dependent (see Appendix C.1).

3.2 One mean-target per direction

For the weight, we take the inverse-variance of the residual at each virtual target. Inverse-variance weighting is the standard way to combine measurements of differing precision. Rewriting Eq. (3.1) through x_τ^{int} gives $r^{\text{ext}}(\tau) = (1-\tau)A\varepsilon_x - \tau n + \frac{\tau - t}{1-t}A\big(\hat{x}(x_t) - x_\star\big)$. For the variance model used to construct the weight, we take $\hat{x}(x_t) = x_\star$, neglecting the clean-image estimation error (see Appendix C.2). With ε_x and n independent,

$$\tilde{r}^{\text{ext}}(\tau) := (1-\tau)A\varepsilon_x - \tau n \;\sim\; \mathcal{N}\big(0, C(\tau)\big), \qquad C(\tau) = (1-\tau)^2 AA^\top + \tau^2\sigma_y^2 I_m. \tag{3.4}$$

Since $AA^\top u_s = \sigma_s^2 u_s$, $C(\tau)$ is diagonal in the measurement basis with entries

$$C_s(\tau) = u_s^\top C(\tau) u_s = (1-\tau)^2\sigma_s^2 + \tau^2\sigma_y^2. \tag{3.5}$$

Equation (2.1) takes $u_s^\top y$, and $r_s^{\text{ext}}(\tau) := u_s^\top r^{\text{ext}}(\tau)$ is the same component of the residual. $C_s(\tau)^{-1}$ is its weight at each virtual target. For a support $\mathcal{T} \subseteq [0,1]$, integrating the weighted square over the support and summing over the measured directions gives

$$\boxed{\lambda_s(\tau) := C_s(\tau)^{-1}, \qquad \mathcal{L}_\lambda(x_t) := \frac{1}{2}\sum_{s=1}^{R}\int_{\mathcal{T}} \lambda_s(\tau)\,\big(r_s^{\text{ext}}(\tau)\big)^2\, d\tau.} \tag{3.6}$$

Every direction's integral is Definition 3.1 taken at $w = \lambda_s$, positive and integrable on $\mathcal{T}$ as the definition requires. Therefore, Theorem 3.2 applies once per direction.

Proposition 3.3. *Applying Theorem 3.2 with $w = \lambda_s$, let $Z_s = \int_{\mathcal{T}} \lambda_s(\tau)\, d\tau$ and define the normalized density $\mu_s(\tau) = \lambda_s(\tau)/Z_s$. With $A^\top u_s = \sigma_s v_s$, the gradient of Eq. (3.6) is*

$$\nabla_{x_t}\mathcal{L}_\lambda = \sum_{s=1}^{R} Z_s\,\sigma_s\,\hat{r}_s\,v_s, \qquad \bar{\tau}_s := \mathbb{E}_{\mu_s}[\tau], \qquad \hat{r}_s := r_s^{\text{ext}}(\bar{\tau}_s). \tag{3.7}$$

The operator A determines $\sigma_s v_s$, while the weight λ_s determines the pair $\{\bar{\tau}_s, Z_s\}$. The residual is formed at $\bar{\tau}_s$ and scaled by Z_s.

The operator A enters the computation of the mean-target through σ_s in C_s. A large σ_s makes C_s fall toward $\tau = 1$, and the mean-target sits near the data endpoint. A small one leaves $\tau^2\sigma_y^2$ dominant, and the mean-target is pulled back toward noise. The support sets the range of a mean-target,

$$\bar{\tau}_s \in \text{conv}(\mathcal{T}), \qquad \mathcal{T}_{\text{lf}} = [t,1] \Rightarrow t \le \bar{\tau}_s \le 1, \qquad \mathcal{T}_{\text{full}} = [0,1] \Rightarrow 0 \le \bar{\tau}_s \le 1, \tag{3.8}$$

the look-forward support $\mathcal{T}_{\text{lf}}$ reaching only ahead of the current step and the full support $\mathcal{T}_{\text{full}}$ admitting a mean-target on either side of it (see Appendix C.3 for details and the proof of Proposition 3.3).

Algorithm 1 MTG on the EDM and OT backbones, EDM on the left and OT on the right.

Require: $A = U\Sigma V^\top$, measurement y, measurement noise σ_y, support $\mathcal{T}$, scalar c

EDM	**OT**
Schedule: $\sigma_{\min} = \sigma_0 < \cdots < \sigma_N = \sigma_{\max}$	**Schedule:** $0 < t_0 < t_1 < \cdots < t_N = 1$
Precompute	**Precompute**
1: $\bar{\varsigma}_s, h_s$ for every direction s at every schedule σ	1: $\bar{\tau}_s, g_s$ for every direction s at every schedule t
Sampling	**Sampling**
2: $x_{\sigma_N} \sim \mathcal{N}(0, \sigma_N^2 I)$	2: $x_{t_0} \sim \mathcal{N}(0, I)$
3: **for** $j = N, \dots, 1$ **do**	3: **for** $i = 0, \dots, N-1$ **do**
4: $s_\theta \leftarrow s_\theta(x_{\sigma_j}, \sigma_j)$	4: $v_\theta \leftarrow v_\theta(x_{t_i}, t_i)$
5: $\hat{r}_s \leftarrow u_s^\top [A x_{\sigma_j} \; -y \; + (\sigma_j - \bar{\varsigma}_s)\sigma_j A s_\theta]$	5: $\hat{r}_s \leftarrow u_s^\top [A x_{t_i} \; -\bar{\tau}_s y \; + (\bar{\tau}_s - t_i) A v_\theta]$
6: $x_{\sigma_j}^{\text{new}} \leftarrow x_{\sigma_j} - c \sum_s \sigma_s h_s \hat{r}_s v_s$	6: $x_{t_i}^{\text{new}} \leftarrow x_{t_i} - c \sum_s \sigma_s g_s \hat{r}_s v_s$
7: $x_{\sigma_{j-1}} \leftarrow x_{\sigma_j}^{\text{new}} + \sigma_j(\sigma_j - \sigma_{j-1}) s_\theta$	7: $x_{t_{i+1}} \leftarrow x_{t_i}^{\text{new}} + (t_{i+1} - t_i) v_\theta$
8: **end for**	8: **end for**
9: **return** x_{σ_0}	9: **return** x_{t_N}

3.3 One damping factor per direction

Since $Z_s > 0$ and $\sigma_s > 0$, the gradient of Proposition 3.3 vanishes exactly when every $\hat{r}_s$ does. The state moves to $x_t + \sum_{s=1}^R d_s v_s$, one displacement d_s along each measured direction. By Eq. (2.1), moving the state by $d_s v_s$ changes $\hat{r}_s$ to $\hat{r}_s + \sigma_s d_s$. So, $\mathcal{L}_\lambda$ depends on d_s only through $(\hat{r}_s + \sigma_s d_s)^2$. Setting $\hat{r}_s + \sigma_s d_s$ to zero would take $d_s = -\hat{r}_s/\sigma_s$, which diverges as $\sigma_s \to 0$. It would also move the state by the noise in $\hat{r}_s$, which has variance $C_s(\bar{\tau}_s)$ by Eq. (3.5). For both reasons, we define d_s as the minimizer of a regularized objective with a coefficient $\gamma_s > 0$,

$$\boxed{d_s := \arg\min_{d \in \mathbb{R}} \tfrac{1}{2} \Big\{ \big(\hat{r}_s + \sigma_s d\big)^2 + \gamma_s d^2 \Big\}.} \tag{3.9}$$

Proposition 3.4. *For any $\gamma_s > 0$ the objective in Eq. (3.9) is quadratic in d. Its stationary point is $d_s = -\sigma_s g_s \hat{r}_s$ with $g_s := \big(\sigma_s^2 + \gamma_s\big)^{-1}$. Collecting the displacements gives*

$$x_t^{\text{new}} = x_t - c \sum_{s=1}^R \sigma_s g_s \, \hat{r}_s \, v_s. \tag{3.10}$$

The mean-target $\bar{\tau}_s$ remains unchanged, while g_s replaces Z_s in the state update.

We call g_s the *damping factor*, which decides how much of $\hat{r}_s$ the step acts on. At a guided step $t \in (0,1)$, we set $\gamma_s = \bar{\tau}_s^2 \sigma_y^2 \big(t^2 + (1-t)^2\big) \big/ \big(t^2 (1-t)^2\big)$. This is the ratio of the measurement-noise variance at $\bar{\tau}_s$ to the Gaussian conditional variance of $t x_\star$ given x_t (see Appendix C.4 for details and the proof of Proposition 3.4). Eq. (3.10) is Eq. (2.3) in the form MTG takes on the OT path.

3.4 Sampling with EDM and OT

Algorithm 1 presents the EDM sampler and the OT sampler. The EDM formulation follows the same approach. The main difference is the measurement target: y for EDM and $\bar{\tau}_s y$ for OT, as highlighted in Algorithm 1. EDM is parameterized by a decreasing noise level σ and uses a score s_θ in place of the OT velocity v_θ. Its mean-target and damping factor are denoted by $\bar{\varsigma}_s$ and h_s, corresponding to $\bar{\tau}_s$ and g_s on the OT path. Since the pair is independent of the current state on both backbones, it can be precomputed for every step before sampling starts. Appendix D provides the EDM derivation.

4 Experiments

We run three experiments. Benchmark comparison (Section 4.1) compares MTG against the strongest training-free baselines. A diagonal operator with a graded spectrum (Section 4.2) varies the mean-target and the damping factor independently. Trained-Target Guidance (Section 4.3) trains the evaluation targets and the update strengths. Sections 4.1 and 4.2 validate the look-forward and the full support against each other and tune the additional parameters each task takes. Appendix E provides the experimental details, including the operator setting and tuning parameters.

Table 1: Results on 100 test images from each of FFHQ, ImageNet, AFHQ-Cat, and CelebA. Among the rows shown, best results are **bold**, second-best underlined. Purple/blue cells indicate $\mathcal{T}_{\text{lf}}$ / $\mathcal{T}_{\text{full}}$.

EDM		Motion deblurring			Gaussian deblurring			Super-resolution			Box inpainting			Random inpainting		
	Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
FFHQ	MTG (Ours)	**33.78**	**0.918**	**0.102**	**30.47**	**0.865**	**0.164**	**30.94**	**0.876**	**0.160**	**25.55**	**0.872**	0.152	**35.20**	**0.946**	**0.066**
	FIG	31.26	0.853	0.174	29.59	0.840	0.190	29.52	0.836	0.200	24.72	0.757	0.188	33.35	0.909	0.083
	DAPS	29.66	0.847	0.157	29.19	0.817	0.165	29.07	0.818	0.177	24.07	0.814	**0.133**	31.12	0.844	0.098
	FPS-SMC	27.39	0.826	0.227	26.54	0.773	0.253	28.42	0.813	0.204	24.86	0.823	0.146	28.21	0.823	0.261
	DPS	24.52	0.801	0.246	25.87	0.764	0.219	25.86	0.753	0.269	22.51	0.792	0.209	25.46	0.823	0.203
ImageNet	MTG (Ours)	**29.67**	**0.834**	**0.152**	**26.32**	**0.712**	**0.252**	**26.38**	**0.722**	**0.264**	20.46	**0.773**	0.278	**30.03**	**0.861**	0.129
	FIG	27.43	0.756	0.253	25.60	0.680	0.293	25.55	0.675	0.302	21.13	0.714	0.260	29.51	0.838	**0.118**
	DAPS	27.86	0.766	0.196	26.15	0.684	0.253	25.89	0.694	0.276	21.43	0.725	0.214	28.44	0.775	0.135
	FPS-SMC	24.52	0.647	0.326	23.91	0.601	0.387	24.82	0.703	0.313	**22.16**	0.726	**0.208**	24.52	0.701	0.316
	DPS	18.96	0.629	0.423	20.31	0.598	0.397	21.13	0.489	0.361	18.94	0.722	0.257	23.52	0.745	0.297

OT		Denoising			Gaussian deblurring			Super-resolution			Box inpainting			Random inpainting		
	Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
AFHQ-Cat	MTG (Ours)	32.43	**0.897**	0.043	**29.18**	**0.793**	0.240	**28.59**	**0.805**	0.187	**27.16**	**0.916**	**0.055**	**34.51**	**0.936**	**0.036**
	Flower1-OT	31.69	0.879	0.102	28.64	0.775	0.255	26.23	0.741	0.272	26.19	0.915	0.063	32.97	0.918	0.040
	PnP-Flow	31.18	0.863	0.135	27.87	0.760	0.304	26.94	0.763	**0.171**	26.00	0.897	0.103	33.00	0.918	0.037
	FIG	**32.48**	**0.897**	**0.042**	29.01	0.788	0.250	26.24	0.730	0.185	24.08	0.815	0.140	32.93	0.915	0.045
	Flow-Priors	29.41	0.763	0.153	26.47	0.700	**0.181**	23.51	0.570	0.272	26.20	0.818	0.118	32.37	0.906	0.047
CelebA	MTG (Ours)	**33.01**	**0.923**	**0.016**	35.76	0.954	0.030	**34.05**	**0.943**	0.030	**32.28**	**0.949**	0.023	**35.49**	**0.967**	**0.013**
	Flower1-OT	32.28	0.914	0.034	34.98	0.947	0.026	32.36	0.923	0.034	31.19	0.945	**0.022**	33.08	0.944	0.018
	PnP-Flow	31.80	0.905	0.044	34.48	0.936	0.040	31.09	0.902	0.045	30.47	0.933	0.037	33.05	0.944	0.018
	FIG	**33.01**	0.921	**0.016**	**35.82**	**0.956**	**0.023**	33.00	0.934	**0.020**	27.00	0.858	0.060	33.26	0.950	0.017
	Flow-Priors	29.34	0.768	0.134	31.54	0.858	0.056	28.35	0.713	0.102	30.07	0.858	0.048	32.88	0.871	0.019

4.1 Benchmark comparison

We evaluate five tasks under each backbone. The EDM backbone runs under DAPS's evaluation setup (Zhang et al., 2025) with FFHQ (Karras et al., 2019) and ImageNet (Deng et al., 2009) at 256×256. The OT backbone runs under FLOWER (Pourya et al., 2026) with AFHQ-Cat (Choi et al., 2020) at 256×256 and CelebA (Liu et al., 2015) at 128×128. Operators and noise levels differ between the EDM and OT setups. A shared task name does not imply an identical setting. Appendix E.1.3 draws the mean-target and the damping factor for each operator at every step.

Table 1 reports the four strongest baselines by PSNR for each of EDM and OT, while Appendix E.1.4 provides the full baseline sets and metrics. MTG leads every EDM task on FFHQ and four of five on ImageNet. On both OT datasets, it leads four of five, while denoising and Gaussian deblurring each go to FIG on one of the two. Most of the remaining non-leading results are in LPIPS. Across the full comparison tables, MTG ranks first or second in 27/30 EDM metrics and 25/30 OT metrics.

4.2 Ablation of the mean-target and damping factor

Separating the effects of the mean-target and the damping factor requires both quantities to vary across directions. Therefore, we need an operator with sufficiently varied singular values while keeping all directions observed. Since none of the ten benchmark settings in Section 4.1 provides that (see Appendix E.2), we construct a diagonal measurement operator whose positive entries gradually decrease across the image. Specifically, the measurement is $y = M \odot x_\star + \sigma_y \varepsilon_y$, where the entry of M at each pixel is that direction's singular value. The mask ranges from above the measurement noise level down to below it. The resulting spatially varying measurement is shown in Figure 3.

In Table 2, the derived mean-target is the best of the three at either setting of the damping factor. The derived damping factor improves every metric at all three targets. MTG's derived pair takes the top of the table. Under EDM, the damping factor already brings the current state near the top. Figure 3

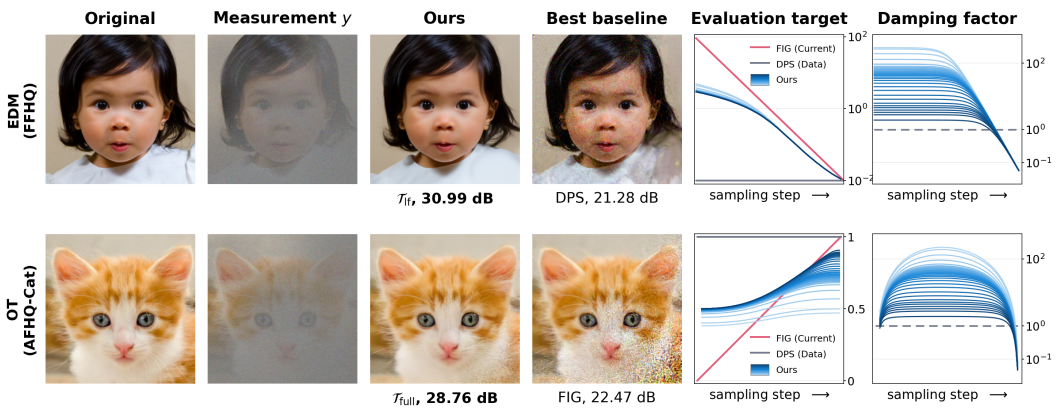

Figure 3: Reconstruction results for the diagonal measurement operator: MTG versus the best baseline among FIG and DPS, with direction-specific evaluation targets and damping factors.

Table 2: Ablation results on 100 test images each from FFHQ and AFHQ-Cat. The six MTG variants combine three evaluation targets with either the derived damping factor g_s or a constant $g_s = 1$. FIG and DPS use their own update rules at the current-state and data-endpoint targets, respectively. Best **bold**, second-best underlined. Purple/blue cells indicate $\mathcal{T}_{\text{lf}}$ / $\mathcal{T}_{\text{full}}$.

Method	Target $\bar{\tau}_s$	Damping g_s	FFHQ (EDM)			AFHQ-Cat (OT)		
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
MTG (Ours)	**Mean-target**	**Derived**	**30.96**	**0.900**	0.119	**28.32**	**0.833**	**0.088**
MTG w/o mean-target	Fixed (Current)	**Derived**	30.63	0.895	**0.118**	21.71	0.626	0.364
MTG w/o mean-target	Fixed (Data)	**Derived**	22.79	0.538	0.335	20.62	0.529	0.267
MTG w/o damping	**Mean-target**	Constant	21.83	0.641	0.296	22.14	0.699	0.196
MTG w/o both	Fixed (Current)	Constant	20.34	0.374	0.446	18.78	0.458	0.400
MTG w/o both	Fixed (Data)	Constant	18.35	0.310	0.479	19.80	0.442	0.349
FIG	Current	Own	19.48	0.419	0.425	23.04	0.694	0.249
DPS	Data	Own	21.57	0.520	0.350	17.71	0.576	0.294

shows the reconstructions against the best baseline. Every method resolves the bright corner. Only MTG stays clean where the intensity falls off. FIG and DPS leave the dark corner undenoised.

Figure 4 reports PSNR across singular values σ_s for the same test runs. Comparing the two rows shows that the derived damping factor improves PSNR at every σ_s for all three targets. For the mean-target, the improvement is largest in the lowest-σ_s band. With derived damping, the mean-target and current-state target achieve similar PSNR at high σ_s (b), but the mean-target gains a substantial advantage as σ_s decreases. With constant damping, target selection also has a clear effect. PSNR decreases monotonically with decreasing σ_s for the current-state and data-endpoint targets. However, the mean-target attains its highest PSNR at intermediate σ_s (c) and maintains relatively high PSNR across the high-to-intermediate σ_s bands (d). These results highlight how the evaluation target and the damping factor each affect reconstruction quality across measured directions, supporting a direction-specific design of both quantities.

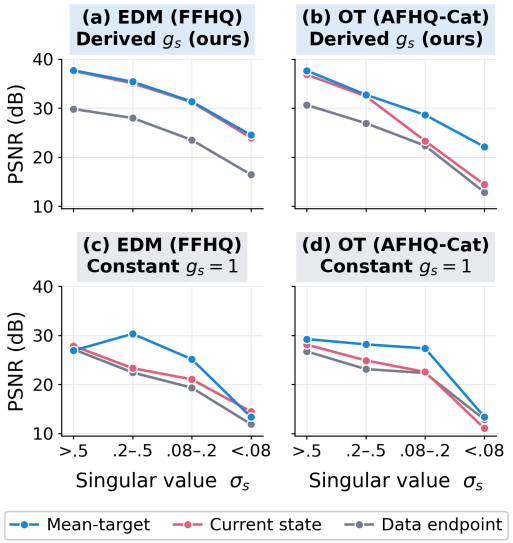

Figure 4: PSNR across the four ranges of σ_s.

	FFHQ	ImageNet
PSNR ↑	26.01	20.60
SSIM ↑	0.889	0.804
LPIPS ↓	0.105	0.224

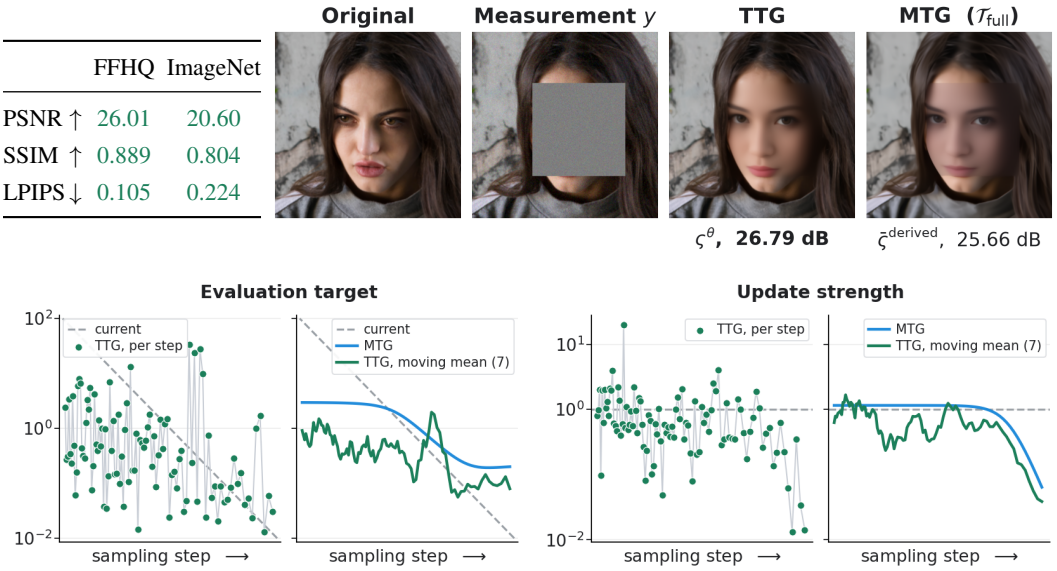

Figure 5: Results of Trained-Target Guidance (TTG) on EDM box inpainting. Top left, averages over the 100 test images, where green marks a metric beating MTG's result in Table 1. Top right, example reconstructions from TTG and MTG. Below, TTG's per-step evaluation targets and update strengths, and their seven-step moving geometric means against MTG.

4.3 Trained-Target Guidance

We introduce **Trained-Target Guidance (TTG)** to examine the evaluation targets and update strengths obtained by directly training these quantities. TTG uses two trainable parameters at each sampling step and no prescribed schedule for either quantity. The parameters are trained on the validation set and evaluated on the test set as in Table 1, using CMA-ES (Hansen & Ostermeier, 2001). To make training feasible, we consider settings whose observed directions share a single singular value. We report results on EDM box inpainting. The details of TTG are in Appendix E.3.

Figure 5 compares the trained pair with MTG's derived pair. Across steps, the trained target moves back and forth between lower and higher noise levels. The trained and derived targets remain on the noisier side over approximately the same range of sampling noise levels. The trained and derived damping factors are comparable in scale, and both decrease toward the end of sampling. These shared trends emerge even though the two quantities are trained directly, without being constrained to the forms derived in Section 3. This correspondence lends empirical support to our analytical construction. TTG also outperforms MTG on this task. These results highlight the importance of both where the residual is formed and how strongly it acts on the state.

5 Conclusion

We introduced Mean-Target Guidance (MTG), which constructs guidance from a residual defined over the generative trajectory and thereby derives its evaluation target. The resulting mean-target adapts to the operator spectrum and varies across measured directions. MTG performs well across various datasets and operators on diffusion and flow-matching backbones. Our ablations demonstrate the importance of both the evaluation target and the update strength. The derived mean-target and damping factor depend on our residual formulation and modeling choices. Therefore, alternative residual formulations or modeling assumptions may lead to different evaluation targets and update strengths. More broadly, MTG suggests a framework for constructing guidance starting from the definition of a residual. Our limitation is that the construction applies to linear operators alone, and that its residual formulation and modeling choices are not guaranteed to be optimal. Future work will extend the construction to nonlinear operators and investigate principled criteria for selecting the residual formulation and modeling choices from which the evaluation target is derived.

AI USE STATEMENT

We used large language models to assist with editing the manuscript for clarity, finding and examining related literature, obtaining feedback on mathematical derivations and experimental analysis, and writing and debugging research code. The authors reviewed the AI-assisted material, checked references against the original sources, verified mathematical arguments under the stated assumptions, and inspected and tested the implementation. The authors made the final decisions on the methodology, interpretation of results, and presentation, and take responsibility for all content of this work, including text, claims, and code developed with AI assistance.

ETHICS STATEMENT

This work studies image reconstruction for linear inverse problems using pretrained generative models and existing image benchmarks. Generative reconstruction can produce plausible details that differ from the original image, and its outputs may reflect biases in the pretrained models and their training data. These considerations are particularly relevant to the face datasets used in our experiments, where reconstructed details may alter perceived identity. The reported benchmark results do not establish fidelity or fairness across all populations and applications.

REPRODUCIBILITY STATEMENT

Section 3, Appendix D and Algorithm 1 describe the MTG construction and sampling procedure. The mathematical assumptions, proofs, and derivations are provided in Section 3 and Appendices C and D. Section 4.1 identifies the benchmark datasets and reference evaluation protocols. Appendix E provides the experimental settings and implementation details for all experiments. Code is available at `https://anonymous.4open.science/r/MTG2027`.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants. In *International Conference on Learning Representations (ICLR)*, 2023.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-Flow: Differentiating through Flows for Controlled Generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3462–3483. PMLR, 2024.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion Posterior Sampling for General Noisy Inverse Problems. In *International Conference on Learning Representations (ICLR)*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Zehao Dou and Yang Song. Diffusion Posterior Sampling for Linear Inverse Problem Solving: A Filtering Perspective. In *International Conference on Learning Representations (ICLR)*, 2024.

Bradley Efron. Tweedie's Formula and Selection Bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Heinz W. Engl, Martin Hanke, and Andreas Neubauer. *Regularization of Inverse Problems*. Mathematics and Its Applications. Kluwer Academic Publishers, 1996.

Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, 1998.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Gradient step denoiser for convergent plug-and-play. *arXiv preprint arXiv:2110.03220*, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the Design Space of Diffusion-Based Generative Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising Diffusion Restoration Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Xiang Li, Soo Min Kwon, Shijun Liang, Ismail R Alkhouri, Saiprasad Ravishankar, and Qing Qu. Decoupled data consistency with diffusion purification for image restoration. *arXiv preprint arXiv:2403.06054*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations (ICLR)*, 2023.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

Ségolène Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. PnP-Flow: Plug-and-Play Image Restoration with Flow Matching. In *International Conference on Learning Representations (ICLR)*, 2025.

Badr Moufad, Yazid Janati, Lisa Bedin, Alain Durmus, Randal Douc, Eric Moulines, and Jimmy Olsson. Variational Diffusion Posterior Sampling with Midpoint Guidance. In *International Conference on Learning Representations (ICLR)*, 2025.

Xinyu Peng, Ziyang Zheng, Wenrui Dai, Nuoqian Xiao, Chenglin Li, Junni Zou, and Hongkai Xiong. Improving Diffusion Models for Inverse Problems Using Optimal Posterior Covariance. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 40347–40370. PMLR, 2024.

Ashwini Pokle, Matthew J. Muckley, Ricky T. Q. Chen, and Brian Karrer. Training-Free Linear Image Inverses via Flows. *Transactions on Machine Learning Research (TMLR)*, 2024. arXiv:2310.04432.

Mehrsa Pourya, Bassam El Rawas, and Michael Unser. Flower: A Flow-Matching Solver for Inverse Problems. In *International Conference on Learning Representations (ICLR)*, 2026.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-Guided Diffusion Models for Inverse Problems. In *International Conference on Learning Representations (ICLR)*, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations (ICLR)*, 2021.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Zihui Wu, Yu Sun, Yifan Chen, Bingliang Zhang, Yisong Yue, and Katherine L. Bouman. Principled Probabilistic Imaging using Diffusion Models as Plug-and-Play Priors. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2405.18782.

Yici Yan, Yichi Zhang, Xiangming Meng, and Zhizhen Zhao. FIG: Flow with Interpolant Guidance for Linear Inverse Problems. In *International Conference on Learning Representations (ICLR)*, 2025.

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving Diffusion Inverse Problem Solving with Decoupled Noise Annealing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20895–20905, 2025. arXiv:2407.01521.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Shaorong Zhang, Rob Brekelmans, and Greg Ver Steeg. Local MAP Sampling for Diffusion Models. In *International Conference on Machine Learning (ICML)*, 2026.

Yasi Zhang, Peiyu Yu, Yaxuan Zhu, Yingshan Chang, Feng Gao, Ying Nian Wu, and Oscar Leong. Flow Priors for Linear Inverse Problems via Iterative Corrupted Trajectory Matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2405.18816.

Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising Diffusion Models for Plug-and-Play Image Restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1219–1229, 2023. arXiv:2305.08995.

A NOTATIONS

Table 3: Notations.

$x_\star$	clean image	$r^{\mathrm{ext}}(\tau)$	residual at a virtual target
y, n	measurement and its noise	r_s^{ext}	$u_s^\top r^{\mathrm{ext}}$, in direction s
σ_y	measurement noise level	$\hat{r}_s$	residual at the mean-target
d, m	image and measurement size	w	an admissible weight
$A = U\Sigma V^\top$	operator and its compact SVD	$\mathcal{T}$	support of the loss integral
σ_s	singular value of direction s	$\mathcal{T}_{\mathrm{lf}}$, $\mathcal{T}_{\mathrm{full}}$	look-forward and full support
u_s, v_s	measurement and state directions	$\mathcal{L}_w$	loss under w
R	number of measured directions	$C(\tau)$	residual variance
$A^\dagger$	pseudo-inverse	C_s	residual variance in direction s
P_{null}	null-space projection	λ_s	inverse variance C_s^{-1}
α, β	affine path coefficients	Z_s, μ_s	mass of λ_s and λ_s/Z_s
ε_x	state noise	$\mathcal{L}_\lambda$	loss under λ_s
t, x_t	OT time and state	$\bar{\tau}_w$	mean-target under w
v_θ	OT velocity	$\bar{\tau}_s$, $\bar{\varsigma}_s$	mean-target on OT and EDM
σ, x_σ	EDM noise level and state	γ_s, g_s	OT coefficient and damping factor
s_θ	EDM score	γ_s^{edm}, h_s	EDM coefficient and damping factor
$\hat{x}$	clean-image estimate	q_t, q_σ	conditional variance of $tx_\star$ and $x_\star$
τ, ς	virtual target on OT and EDM	c, m	guidance scale and mixing coefficient
x_τ^{int}	state on the path at τ	κ	uncertainty scale on EDM
x_τ^{ext}	extrapolation at τ	m_A, δ_A	Fourier-model error and its regularization
		ς_i^θ, h_i^θ, c^θ	trained target, coefficient, and scale

B PRE-TRAINED MODELS

B.1 DIFFUSION: TRAINING AND SAMPLING

A diffusion model is built on a forward process that drives a clean image toward noise. Written as a stochastic differential equation,

$$dx = f(x,t)\,dt + g(t)\,dw, \tag{B.1}$$

with w a Wiener process, f a drift and g a diffusion coefficient. The choice of f and g fixes p_t, the distribution of the state at time t.

Sampling runs that process backward, and what it needs is the score $\nabla_x \log p_t(x)$. A network $s_\theta(x,t)$ is trained to supply it. Although the marginal score has no closed form, the forward transition kernel is Gaussian with an analytic score. Therefore, training minimizes the denoising score matching objective,

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\,x_\star,\,x_t|x_\star}\Big[\omega(t)\big\|s_\theta(x_t,t) - \nabla_{x_t}\log p_{t|0}(x_t \mid x_\star)\big\|^2\Big]. \tag{B.2}$$

Its minimizer is the true score in expectation (Song et al., 2021). Sampling runs the process backward from noise with the trained score s_θ. The reverse process takes either a stochastic or a deterministic form, and both share the marginals p_t (Song et al., 2021; Dhariwal & Nichol, 2021).

B.1.1 THE VARIANCE-PRESERVING AND VARIANCE-EXPLODING SDES

Two choices of f and g in Eq. (B.1) name the variance-preserving (VP) and the variance-exploding (VE) families,

$$dx = -\tfrac{1}{2}\beta_{\mathrm{vp}}(t)x\,dt + \sqrt{\beta_{\mathrm{vp}}(t)}\,dw \;\;\text{(VP)}, \qquad dx = \sqrt{\tfrac{d[\sigma^2(t)]}{dt}}\,dw \;\;\text{(VE)}, \tag{B.3}$$

where $\beta_{\mathrm{vp}}(t)$ and $\sigma(t)$ are schedules that control how fast the process reaches noise. VP recovers DDPM (Ho et al., 2020), and VE recovers score matching with Langevin dynamics (Song et al., 2021).

Integrating the VP drift gives $x_t \mid x_\star \sim \mathcal{N}\big(\bar{\alpha}_t^{1/2}x_\star,\, 1-\bar{\alpha}_t\big)$ with $\bar{\alpha}_t := \exp\big(-\int_0^t \beta_{\mathrm{vp}}(u)\,du\big)$. The squared coefficient on $x_\star$ and the noise variance sum to one, preserving unit marginal variance

when the data have unit variance. The VE drift integrates to $x_t \mid x_\star \sim \mathcal{N}(x_\star, \sigma^2(t))$. Since the schedule $\sigma(t)$ grows without bound, the variance of x_t explodes (Song et al., 2021; Albergo & Vanden-Eijnden, 2023).

B.1.2 Elucidated diffusion model (EDM)

EDM (Karras et al., 2022) is the VE family reindexed by the noise level itself rather than by time. Its state is

$$x_\sigma = x_\star + \sigma\varepsilon_x, \qquad \varepsilon_x \sim \mathcal{N}(0, I_d), \tag{B.4}$$

and its network is trained as a denoiser D_θ whose minimizer is the conditional mean $\mathbb{E}[x_\star \mid x_\sigma]$. For the Gaussian perturbation of Eq. (B.4), Tweedie's formula (Efron, 2011) writes the conditional mean $\mathbb{E}[x_\star \mid x_\sigma]$ as the state shifted along the score of the marginal density of x_σ. Therefore, one trained network supplies both,

$$s_\theta(x_\sigma, \sigma) = \frac{D_\theta(x_\sigma; \sigma) - x_\sigma}{\sigma^2}, \qquad \hat{x}(x_\sigma) = x_\sigma + \sigma^2 s_\theta(x_\sigma, \sigma). \tag{B.5}$$

The estimate $\hat{x}(x_\sigma)$ is the posterior mean of the clean image given the state, so it minimizes the mean squared error among estimates of $x_\star$. Since the deterministic sampler moves from x_σ toward $\hat{x}(x_\sigma)$ as the noise level falls, the estimate also fixes a direction.

B.2 Flow matching: training and sampling

A flow-matching model learns a velocity field $v_\theta(x, t)$ that defines an ordinary differential equation,

$$\frac{dx}{dt} = v_\theta(x, t). \tag{B.6}$$

Its flow transports a noise distribution p_0 at $t = 0$ into the data distribution p_1 at $t = 1$ (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023). Training regresses v_θ on the velocity of a prescribed path between the two endpoints. Sampling integrates Eq. (B.6) from $x \sim \mathcal{N}(0, I_d)$, in practice by Euler steps on the sampling schedule. The path is a modeling choice.

B.2.1 The optimal transport (OT) path and its coupling

Drawing a pair $(\varepsilon_x, x_\star)$ from a coupling π with marginals p_0 and p_1, the straight interpolation between them is

$$x_t^{\text{int}} = t x_\star + (1 - t)\varepsilon_x. \tag{B.7}$$

Since Eq. (B.7) is affine in t, the velocity along the line is constant and equal to $x_\star - \varepsilon_x$. Training minimizes the squared error against that velocity,

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\,(\varepsilon_x, x_\star)\sim\pi}\Big[\big\|v_\theta(x_t^{\text{int}}, t) - (x_\star - \varepsilon_x)\big\|^2\Big]. \tag{B.8}$$

The velocity $x_\star - \varepsilon_x$ is not a function of x_t^{int}. Therefore, the minimizer is its conditional mean $v_\theta = \mathbb{E}[x_\star - \varepsilon_x \mid x_t^{\text{int}}]$, the marginal velocity whose flow takes p_0 to p_1. Eliminating ε_x between the interpolation and the velocity gives $x_\star = x_t^{\text{int}} + (1 - t)(x_\star - \varepsilon_x)$, and conditional expectations of both sides give the clean-image estimate

$$\hat{x}(x_t) = x_t + (1 - t)v_\theta(x_t, t), \tag{B.9}$$

the counterpart of Eq. (B.5).

The coupling π leaves the endpoint marginals untouched but changes the trajectories of the learned ODE. Independent draws let pairs cross. The marginal velocity then averages conflicting directions, and the integral curves bend away from straight lines. The OT coupling admits no crossings, and an iterative reflow procedure approaches the same effect by retraining on the coupling a trained field induces (Liu et al., 2023).

Table 4: Coordinates and affine coefficients α and β for the EDM and OT backbones in Eq. (B.10).

Backbone	Coordinate	α	β
EDM	σ	1	σ
OT	t	t	$1-t$

B.3 Affine representation of EDM and OT

Equations (B.4) and (B.7) put the EDM state and the OT state in one form,

$$x = \alpha\, x_\star + \beta\, \varepsilon_x. \tag{B.10}$$

On the OT path, the coefficients are affine in t, while on EDM, they are affine in σ. Applying A to Eq. (B.10) gives

$$Ax = \alpha\, Ax_\star + \beta\, A\varepsilon_x. \tag{B.11}$$

Since A is linear, the measurement of a state is the same combination of the measurements of the two endpoints. Table 4 collects the two backbones in this form. The OT path sets $\alpha = t$, and EDM keeps $\alpha = 1$.

C Derivations on the OT path

This appendix provides the derivations and modeling choices underlying Section 3. We first derive the additional gradient terms that arise without the frozen-velocity construction in Appendix C.1 and justify the assumption $\hat{x}(x_t) = x_\star$ in Appendix C.2. Appendix C.3 and Appendix C.4 then prove Proposition 3.3 and Proposition 3.4, respectively, with the latter also specifying the coefficient γ_s.

C.1 The frozen velocity

The extrapolation of Eq. (3.1) is the conditional mean of the interpolation, $x_\tau^{\mathrm{ext}} = \mathbb{E}[x_\tau^{\mathrm{int}} \mid x_t]$, and the conditional mean of the transported measurement, $y_\tau^{\mathrm{ext}} = \mathbb{E}[y_\tau^{\mathrm{int}} \mid y]$. Both means remove $x_\star$ and ε_x and allow the sampler to evaluate the residual at every τ. Theorem 3.2 needs two properties of that residual. It has to be affine in τ, and its Jacobian in x_t must not depend on τ. Suppose the velocity is evaluated at each virtual target. The extrapolation then solves

$$x_\tau = x_t + (\tau - t)\, v_\theta(x_\tau, \tau), \tag{C.1}$$

instead of holding $v_\theta(x_t, t)$ fixed as Eq. (3.1) does. Then the residual $Ax_\tau - \tau y$ follows whatever curve the velocity field traces, and it is no longer affine in τ. Theorem 3.2 substitutes $\bar{\tau}_w$ for τ inside the integral, and that step needs an integrand affine in τ. Therefore, with the velocity re-evaluated at each virtual target, the gradient retains every residual on $\mathcal{T}$, and a single virtual target cannot reproduce it.

Suppose the velocity stays at $v_\theta(x_t, t)$ while the stop-gradient is dropped. Although the residual remains affine, its Jacobian now moves with τ. Writing $J_\theta := \partial v_\theta(x_t, t)/\partial x_t$,

$$\frac{\partial r^{\mathrm{ext}}(\tau)}{\partial x_t} = A\big(I_d + (\tau - t)J_\theta\big), \tag{C.2}$$

which depends on τ and cannot leave the integral as a constant factor. Integrating against μ_w with $Z := \int_{\mathcal{T}} w$ and $\bar{\tau}_w = \mathbb{E}_{\mu_w}[\tau]$ gives

$$\frac{1}{Z}\nabla_{x_t}\mathcal{L}_w = \big(I_d + (\bar{\tau}_w - t)J_\theta^\top\big)A^\top r^{\mathrm{ext}}(\bar{\tau}_w) + \mathrm{Var}_{\mu_w}(\tau)\, J_\theta^\top A^\top \big(Av_\theta(x_t, t) - y\big). \tag{C.3}$$

The residual at the mean-target survives, but an extra Jacobian factor multiplies it. Also, a further term appears beside it. The added term is proportional to $\mathrm{Var}_{\mu_w}(\tau)$, and its vector factor $A^\top\big(Av_\theta(x_t, t) - y\big)$ is the rate at which the residual moves with τ rather than its value at any virtual target. The stop-gradient sets J_θ to zero, and both pieces vanish with it. Then the gradient reduces to the residual at a single virtual target, and the network's Jacobian stays out of every step.

C.2 THE ASSUMPTION BEHIND THE INVERSE-VARIANCE WEIGHT

Section 3.2 weights each virtual target by the inverse-variance of its residual, and evaluates that variance under $\hat{x}(x_t) = x_\star$. The extrapolation of Eq. (3.1) moves at the frozen velocity, and the interpolation of Eq. (B.7) moves at $x_\star - \varepsilon_x$,

$$\begin{aligned} x_\tau^{\text{ext}} - x_t &= (\tau - t)\, v_\theta(x_t, t), & \hat{x}(x_t) - x_t &= (1-t)\, v_\theta(x_t, t), \\ x_\tau^{\text{int}} - x_t &= (\tau - t)(x_\star - \varepsilon_x), & x_\star - x_t &= (1-t)(x_\star - \varepsilon_x). \end{aligned} \tag{C.4}$$

Eliminating the velocity from each row gives

$$x_\tau^{\text{ext}} = x_t + \frac{\tau - t}{1-t}\big(\hat{x}(x_t) - x_t\big), \qquad x_\tau^{\text{int}} = x_t + \frac{\tau - t}{1-t}\big(x_\star - x_t\big). \tag{C.5}$$

Their difference is $x_\tau^{\text{ext}} - x_\tau^{\text{int}} = \frac{\tau - t}{1-t}\big(\hat{x}(x_t) - x_\star\big)$. Substituting it into r^{ext} with $y = Ax_\star + n$ rewrites the same residual of Eq. (3.1) through $x_\star$, ε_x and n,

$$\begin{aligned} r^{\text{ext}}(\tau) &= Ax_\tau^{\text{ext}} - \tau y \\ &= Ax_\tau^{\text{int}} - \tau y + \frac{\tau - t}{1-t} A\big(\hat{x}(x_t) - x_\star\big) \\ &= A\big(\tau x_\star + (1-\tau)\varepsilon_x\big) - \tau\big(Ax_\star + n\big) + \frac{\tau - t}{1-t} A\big(\hat{x}(x_t) - x_\star\big) \\ &= (1-\tau)A\varepsilon_x - \tau n + \frac{\tau - t}{1-t} A\big(\hat{x}(x_t) - x_\star\big). \end{aligned} \tag{C.6}$$

Taking $\hat{x}(x_t) = x_\star$ removes the term in $\hat{x}(x_t) - x_\star$ from Eq. (C.6). Then the residual is built from $\varepsilon_x \sim \mathcal{N}(0, I_d)$ and $n \sim \mathcal{N}(0, \sigma_y^2 I_m)$, both fixed before any modeling choice. This leaves $\tilde{r}^{\text{ext}}(\tau)$ of Eq. (3.4). By Eq. (2.1), each measured direction sees two independent Gaussians, and their variances sum to Eq. (3.5).

Keeping the term would require the covariance of $\hat{x}(x_t) - x_\star$, and estimating it is a line of work of its own (Peng et al., 2024; Song et al., 2023; Pokle et al., 2024). Assuming $\hat{x}(x_t) = x_\star$ reduces the variance to $C_s(\tau) = (1-\tau)^2\sigma_s^2 + \tau^2\sigma_y^2$, where $(1-\tau)^2\sigma_s^2$ falls to zero at $\tau = 1$ and $\tau^2\sigma_y^2$ at $\tau = 0$. Therefore, C_s has a minimum inside the interval at $\tau^\star = \sigma_s^2/(\sigma_s^2 + \sigma_y^2)$. The minimum keeps the weight from piling up at an end of the interval. Its position moves with σ_s against σ_y, and therefore differs from direction to direction.

However, an identity measurement operator is the exception. Every direction shares $\sigma_s = 1$ and the same $\tau^\star$. For this operator, we can restore the term in $\hat{x}(x_t) - x_\star$, taking the clean image to be $x_\star \sim \mathcal{N}(0, I_d)$ as OT-ODE (Pokle et al., 2024) does. Under that Gaussian, conditioning gives $\mathrm{Cov}(x_\star \mid x_t) = \rho_t^2 I_d$ with $\rho_t^2 = (1-t)^2/\big(t^2 + (1-t)^2\big)$. Taking the contributions of the residual as uncorrelated, the variance becomes

$$C_s^{\text{Id}}(\tau) = (1-\tau)^2 + \tau^2\sigma_y^2 + \left(\frac{\tau - t}{1-t}\right)^2 \rho_t^2. \tag{C.7}$$

The minimizer of $(1-\tau)^2 + \tau^2\sigma_y^2$ is $\tau^\star = 1/(1+\sigma_y^2)$, which is the same value at every step. Adding $\big((\tau-t)/(1-t)\big)^2\rho_t^2$ moves that minimizer to the current step. The added term $\big((\tau-t)/(1-t)\big)^2\rho_t^2$ vanishes at $\tau = t$ and grows away from it. As a result, the mean-target moves along the run.

In addition, Equation (C.6) builds every residual from the same ε_x and the same n. The residuals at different virtual targets are not independent. Therefore, Eq. (3.6) does not model the virtual targets jointly but weights each one by its own variance.

C.3 THE MEAN-TARGET IN EACH MEASURED DIRECTION

Proof of Proposition 3.3. Fix a measured direction s. In $C_s(\tau) = (1-\tau)^2\sigma_s^2 + \tau^2\sigma_y^2$, the term $(1-\tau)^2\sigma_s^2$ vanishes at $\tau = 1$ and the term $\tau^2\sigma_y^2$ at $\tau = 0$. With $\sigma_s > 0$ and $\sigma_y > 0$, they never vanish together. Then C_s is continuous and strictly positive on $[0, 1]$. Therefore, $\lambda_s = C_s^{-1}$ is bounded on $\mathcal{T}$, which gives $0 < Z_s < \infty$ for any $\mathcal{T}$ of positive measure. As a result, each direction's integral in Eq. (3.6) is given by Definition 3.1 at $w = \lambda_s$. The component $r_s^{\text{ext}}(\tau) = u_s^\top r^{\text{ext}}(\tau)$ inherits the

affine dependence on τ of Eq. (3.1), and the stop-gradient makes $\partial r_s^{\mathrm{ext}}(\tau)/\partial x_t = A^\top u_s = \sigma_s v_s$ at every τ. Write $r_s^{\mathrm{ext}}(\tau) = a_s + b_s\tau$ with $a_s = u_s^\top a$ and $b_s = u_s^\top b$ from the proof of Theorem 3.2, and $\mu_s = \lambda_s / Z_s$. Then

$$\begin{aligned}\nabla_{x_t}\mathcal{L}_\lambda &= \frac{1}{2}\sum_{s=1}^{R}\int_{\mathcal{T}} \lambda_s(\tau)\,\nabla_{x_t}\big(r_s^{\mathrm{ext}}(\tau)\big)^2\,d\tau \;=\; \sum_{s=1}^{R}\int_{\mathcal{T}} \lambda_s(\tau)\, r_s^{\mathrm{ext}}(\tau)\,\frac{\partial r_s^{\mathrm{ext}}(\tau)}{\partial x_t}\,d\tau \\ &= \sum_{s=1}^{R}\sigma_s\, v_s \int_{\mathcal{T}} \lambda_s(\tau)\, r_s^{\mathrm{ext}}(\tau)\,d\tau \;=\; \sum_{s=1}^{R} Z_s\,\sigma_s\, v_s \int_{\mathcal{T}} \mu_s(\tau)\,\big(a_s + b_s\tau\big)\,d\tau \\ &= \sum_{s=1}^{R} Z_s\,\sigma_s\, v_s\big(a_s + b_s\bar\tau_s\big) \;=\; \sum_{s=1}^{R} Z_s\,\sigma_s\, r_s^{\mathrm{ext}}(\bar\tau_s)\, v_s \;=\; \sum_{s=1}^{R} Z_s\,\sigma_s\,\hat r_s\, v_s,\end{aligned}$$

where the Jacobian leaves the integral as a constant factor, $\int_{\mathcal{T}}\mu_s = 1$ and $\int_{\mathcal{T}}\tau\,\mu_s(\tau)\,d\tau = \bar\tau_s$. □

We next establish the support property of the mean-target. Write $\tau_- = \inf\mathcal{T}$ and $\tau_+ = \sup\mathcal{T}$, and let $\mathcal{T}$ hold more than one point. For any admissible weight in Definition 3.1, the mean-target of Theorem 3.2 lies strictly inside the support,

$$\tau_- < \bar\tau_w < \tau_+. \tag{C.8}$$

Both inequalities follow from the distance between $\bar\tau_w$ and each endpoint of $\mathcal{T}$. Since μ_w is a probability density on $\mathcal{T}$,

$$\bar\tau_w - \tau_- = \int_{\mathcal{T}}(\tau - \tau_-)\,\mu_w(\tau)\,d\tau, \qquad \tau_+ - \bar\tau_w = \int_{\mathcal{T}}(\tau_+ - \tau)\,\mu_w(\tau)\,d\tau.$$

Both integrands are nonnegative, and $\tau - \tau_-$ vanishes only at τ_- while $\tau_+ - \tau$ vanishes only at τ_+. Either integral could vanish only if μ_w were zero almost everywhere, which contradicts $\int_{\mathcal{T}}\mu_w = 1$. Proposition 3.3 derives one mean-target per measured direction, and Eq. (C.8) applies to each of them at $w = \lambda_s$.

C.4 The damping factor in each measured direction

Proof of Proposition 3.4. By Eq. (2.1), the measurement sees the state through $u_s^\top A x_t = \sigma_s v_s^\top x_t$. Therefore, moving the state to $x_t + d_s v_s$ sends $\hat r_s$ to $\hat r_s + \sigma_s d_s$ and leaves every other direction where it was. Write ϕ for the objective of Eq. (3.9),

$$\phi(d) = \tfrac{1}{2}\big(\hat r_s + \sigma_s d\big)^2 + \tfrac{\gamma_s}{2}d^2, \qquad \phi''(d) = \sigma_s^2 + \gamma_s > 0,$$

which makes ϕ strictly convex with one minimizer. Setting $\phi'(d) = \sigma_s(\hat r_s + \sigma_s d) + \gamma_s d$ to zero and solving for d gives

$$d_s = -\frac{\sigma_s}{\sigma_s^2 + \gamma_s}\hat r_s = -\sigma_s g_s \hat r_s.$$

Summing $d_s v_s$ over the measured directions and scaling by c gives Eq. (3.10). □

At the minimizer, the direction keeps $\hat r_s + \sigma_s d_s = \gamma_s g_s \hat r_s$ of its residual. Therefore, γ_s decides how much of it the step acts on. For a fixed $\gamma_s > 0$ and a bounded $\hat r_s$, the displacement vanishes as $\sigma_s \to 0$, and a direction the measurement barely sees is barely moved. A direction with $\sigma_s^2 \gg \gamma_s$ has $\gamma_s g_s \approx 0$ and comes close to cancelling its residual outright.

The coefficient γ_s. The coefficient in Eq. (3.9) is a modeling choice specified separately from the loss of Eq. (3.6). We construct it as a ratio of two uncertainty scales: the measurement-noise variance at the mean-target and the conditional variance of $tx_\star$ given x_t.

Since the residual at the mean-target is $\hat r_s = u_s^\top\big(A x_{\bar\tau_s}^{\mathrm{ext}} - \bar\tau_s y\big)$, its measurement side is $\bar\tau_s y$. Equation (1.1) puts $n \sim \mathcal{N}(0, \sigma_y^2 I_m)$ inside y. The noise at the mean-target is $\bar\tau_s n$, of variance

$$\mathrm{Var}\big(\bar\tau_s\, u_s^\top n\big) = \bar\tau_s^2\sigma_y^2 \tag{C.9}$$

in direction s.

For the displacement, the clean image is taken to be $x_\star \sim \mathcal{N}(0, I_d)$, the same assumption Appendix C.2 uses. Under it, $tx_\star$ and x_t are jointly Gaussian with $\mathrm{Cov}(tx_\star) = t^2 I_d$, $\mathrm{Cov}(x_t) = [t^2 + (1-t)^2] I_d$ and $\mathrm{Cov}(tx_\star, x_t) = t^2 I_d$. Then the Gaussian conditioning formula gives

$$\begin{aligned} q_t\, I_d := \mathrm{Cov}(tx_\star \mid x_t) &= t^2 I_d - t^2 I_d \big([t^2 + (1-t)^2] I_d\big)^{-1} t^2 I_d \\ &= \frac{t^2(1-t)^2}{t^2 + (1-t)^2} I_d. \end{aligned} \tag{C.10}$$

Weighing the squared residual against Eq. (C.9) and the squared displacement against q_t returns Eq. (3.9) up to a positive factor,

$$\bar\tau_s^2 \sigma_y^2 \left[\frac{\big(\hat r_s + \sigma_s d\big)^2}{2\,\bar\tau_s^2 \sigma_y^2} + \frac{d^2}{2\,q_t} \right] = \frac{1}{2} \left\{ \big(\hat r_s + \sigma_s d\big)^2 + \frac{\bar\tau_s^2 \sigma_y^2}{q_t}\, d^2 \right\}. \tag{C.11}$$

So the coefficient we choose and the damping factor it produces are

$$\gamma_s = \frac{\bar\tau_s^2 \sigma_y^2}{q_t} = \frac{\bar\tau_s^2 \sigma_y^2 \big[t^2 + (1-t)^2\big]}{t^2(1-t)^2}, \qquad g_s = \frac{t^2(1-t)^2}{\sigma_s^2 t^2 (1-t)^2 + \bar\tau_s^2 \sigma_y^2 \big[t^2 + (1-t)^2\big]}. \tag{C.12}$$

For the identity operator, the coefficient keeps this form; the damping factor comes out the same in every direction,

$$g_s^{\mathrm{Id}} = \frac{t^2(1-t)^2}{t^2(1-t)^2 + \bar\tau_s^2 \sigma_y^2 \big[t^2 + (1-t)^2\big]}, \tag{C.13}$$

where $\bar\tau_s$ is the mean-target that Eq. (C.7) returns.

Our aim in choosing γ_s is to make the damping factor depend on both the mean-target and the current sampling time. The variance-ratio construction above achieves this by incorporating the corresponding measurement and state uncertainties.

D Mean-target guidance on EDM

This appendix provides the EDM counterpart of the derivation in Section 3. Appendix D.1 derives the mean-target from the extrapolated residual, while Appendix D.2 constructs a direction-specific inverse-variance weight. Appendix D.3 then derives the corresponding damping factor.

D.1 The mean-target

Under EDM, the state keeps $\alpha = 1$ and takes $\beta = \sigma$ (see Table 4). Therefore, at a virtual target ς, the state and the measurement are

$$x_\varsigma^{\mathrm{edm}} = x_\star + \varsigma \varepsilon_x, \qquad y_\varsigma^{\mathrm{edm}} = y, \tag{D.1}$$

where the measurement stands unscaled because α does not move with ς.

The state of Eq. (D.1) needs $x_\star$ and ε_x, and the sampler has neither. Taking its conditional mean under the frozen score removes both. With $\hat x(x_\sigma) = x_\sigma + \sigma^2 s_\theta(x_\sigma, \sigma)$ from Eq. (B.5),

$$\begin{aligned} x_\varsigma^{\mathrm{ext}} := \mathbb{E}[x_\varsigma^{\mathrm{edm}} \mid x_\sigma] = \hat x(x_\sigma) + \frac{\varsigma}{\sigma}\big(x_\sigma - \hat x(x_\sigma)\big) = x_\sigma + (\sigma - \varsigma)\sigma\ \mathrm{sg}[s_\theta], \qquad y_\varsigma^{\mathrm{ext}} := y, \\ r^{\mathrm{ext}}(\varsigma) := A x_\varsigma^{\mathrm{ext}} - y_\varsigma^{\mathrm{ext}} = A x_\sigma - y + (\sigma - \varsigma)\sigma A\ \mathrm{sg}[s_\theta]. \end{aligned} \tag{D.2}$$

The extrapolation returns the current state at $\varsigma = \sigma$ and $\hat x(x_\sigma)$ as $\varsigma \to 0$.

Definition D.1. *Let $\mathcal{T} \subseteq [\sigma_{\min}, \sigma_{\max}]$ and let $w(\varsigma) \geq 0$ for every $\varsigma \in \mathcal{T}$ with $0 < \int_{\mathcal{T}} w < \infty$. The loss is*

$$\mathcal{L}_w(x_\sigma) := \frac{1}{2} \int_{\mathcal{T}} w(\varsigma) \left\| r^{\mathrm{ext}}(\varsigma) \right\|^2 d\varsigma. \tag{D.3}$$

Theorem D.2. *Fix x_σ and the score frozen in Eq. (D.2). The gradient of Eq. (D.3) is built from a single residual, formed at the mean-target $\bar{\varsigma}_w$,*

$$\nabla_{x_\sigma}\mathcal{L}_w = \Big(\int_{\mathcal{T}} w\Big) A^\top r^{\mathrm{ext}}(\bar{\varsigma}_w), \qquad \bar{\varsigma}_w := \mathbb{E}_{\mu_w}[\varsigma], \qquad \mu_w := w \Big/ \int_{\mathcal{T}} w. \tag{D.4}$$

Proof. Write $r^{\mathrm{ext}}(\varsigma) = a + b\varsigma$ with $b := -\sigma A\,\mathrm{sg}[s_\theta]$ and $a := Ax_\sigma - y - \sigma b$, neither depending on ς. The stop-gradient holds b fixed in the state. The Jacobian $\partial r^{\mathrm{ext}}(\varsigma)/\partial x_\sigma$ is then A at every ς. With $\int_{\mathcal{T}} \mu_w = 1$ and $\int_{\mathcal{T}} \varsigma\, \mu_w(\varsigma) d\varsigma = \bar{\varsigma}_w$,

$$\begin{aligned}
\nabla_{x_\sigma}\mathcal{L}_w = \frac{1}{2}\int_{\mathcal{T}} w(\varsigma)\, \nabla_{x_\sigma} \big\| r^{\mathrm{ext}}(\varsigma)\big\|^2 d\varsigma \;&=\; \int_{\mathcal{T}} w(\varsigma) \Big(\frac{\partial r^{\mathrm{ext}}(\varsigma)}{\partial x_\sigma}\Big)^\top r^{\mathrm{ext}}(\varsigma)\, d\varsigma \\
&= \int_{\mathcal{T}} w(\varsigma)\, A^\top r^{\mathrm{ext}}(\varsigma)\, d\varsigma \;=\; \Big(\int_{\mathcal{T}} w\Big) A^\top \int_{\mathcal{T}} \frac{w(\varsigma)}{\int_{\mathcal{T}} w}\, r^{\mathrm{ext}}(\varsigma)\, d\varsigma \\
&= \Big(\int_{\mathcal{T}} w\Big) A^\top \int_{\mathcal{T}} \mu_w(\varsigma)\, (a + b\varsigma)\, d\varsigma \;=\; \Big(\int_{\mathcal{T}} w\Big) A^\top \Big(a + b\int_{\mathcal{T}} \varsigma\, \mu_w(\varsigma) d\varsigma\Big) \\
&= \Big(\int_{\mathcal{T}} w\Big) A^\top \big(a + b\,\bar{\varsigma}_w\big) \;=\; \Big(\int_{\mathcal{T}} w\Big) A^\top r^{\mathrm{ext}}(\bar{\varsigma}_w).
\end{aligned}$$

□

D.2 One mean-target per direction

The weight is the inverse-variance of the residual of Eq. (D.2) at each virtual target. The extrapolation moves at the frozen score, and the state of Eq. (D.1) moves at $-\varepsilon_x$,

$$\begin{aligned}
x_\varsigma^{\mathrm{ext}} - x_\sigma &= (\sigma - \varsigma)\sigma\ \mathrm{sg}[s_\theta], & \hat{x}(x_\sigma) - x_\sigma &= \sigma^2\ \mathrm{sg}[s_\theta], \\
x_\varsigma^{\mathrm{edm}} - x_\sigma &= -(\sigma - \varsigma)\,\varepsilon_x, & x_\star - x_\sigma &= -\sigma\,\varepsilon_x.
\end{aligned} \tag{D.5}$$

Eliminating $\mathrm{sg}[s_\theta]$ and ε_x from each row gives

$$x_\varsigma^{\mathrm{ext}} = x_\sigma + \frac{\sigma - \varsigma}{\sigma}\big(\hat{x}(x_\sigma) - x_\sigma\big), \qquad x_\varsigma^{\mathrm{edm}} = x_\sigma + \frac{\sigma - \varsigma}{\sigma}\big(x_\star - x_\sigma\big). \tag{D.6}$$

Their difference is $x_\varsigma^{\mathrm{ext}} - x_\varsigma^{\mathrm{edm}} = \big(1 - \varsigma/\sigma\big)\big(\hat{x}(x_\sigma) - x_\star\big)$, which vanishes at $\varsigma = \sigma$ and grows as ς falls. Substituting it into r^{ext} with $y = Ax_\star + n$ rewrites the residual of Eq. (D.2) through $x_\star$, ε_x and n,

$$\begin{aligned}
r^{\mathrm{ext}}(\varsigma) &= Ax_\varsigma^{\mathrm{ext}} - y \\
&= Ax_\varsigma^{\mathrm{edm}} - y + \Big(1 - \frac{\varsigma}{\sigma}\Big) A\big(\hat{x}(x_\sigma) - x_\star\big) \\
&= A\big(x_\star + \varsigma\varepsilon_x\big) - \big(Ax_\star + n\big) + \Big(1 - \frac{\varsigma}{\sigma}\Big) A\big(\hat{x}(x_\sigma) - x_\star\big) \\
&= \varsigma A\varepsilon_x - n + \Big(1 - \frac{\varsigma}{\sigma}\Big) A\big(\hat{x}(x_\sigma) - x_\star\big).
\end{aligned} \tag{D.7}$$

Appendix C.2 assumes $\hat{x}(x_t) = x_\star$, which removes the term in $\hat{x}(x_t) - x_\star$ from Eq. (C.6). $(1 - \tau)A\varepsilon_x - \tau n$ survives there, and its variance turns inside the interval. The same assumption in Eq. (D.7) leaves $\varsigma A\varepsilon_x - n$. Its variance in direction s is $\varsigma^2\sigma_s^2 + \sigma_y^2$, which increases with ς everywhere. An inverse-variance weight built on it concentrates its mass at $\sigma_{\min}$, identically in every direction. Therefore, we estimate the term in $\hat{x}(x_\sigma) - x_\star$ on EDM.

Estimating it needs $\mathrm{Cov}(x_\star \mid x_\sigma)$, and the clean image is taken to be $x_\star \sim \mathcal{N}(0, I_d)$, as ΠGDM (Song et al., 2023) does. With that assumption, $x_\star$ and $x_\sigma = x_\star + \sigma\varepsilon_x$ are jointly Gaussian, and

$$\mathrm{Cov}(x_\star \mid x_\sigma) = \big(I_d + \sigma^{-2} I_d\big)^{-1} = \frac{\sigma^2}{1 + \sigma^2} I_d. \tag{D.8}$$

Taking Eq. (D.7) in direction s with $u_s^\top A = \sigma_s v_s^\top$, and its contributions as uncorrelated,

$$\begin{aligned} r_s^{\text{ext}}(\varsigma) &= \varsigma\,\sigma_s v_s^\top \varepsilon_x - u_s^\top n + \Big(1 - \frac{\varsigma}{\sigma}\Big)\sigma_s v_s^\top \big(\hat{x}(x_\sigma) - x_\star\big), \\ C_s(\varsigma) &= \varsigma^2\sigma_s^2 + \sigma_y^2 + \Big(1 - \frac{\varsigma}{\sigma}\Big)^2 \sigma_s^2\, v_s^\top \operatorname{Cov}\big(x_\star \mid x_\sigma\big) v_s \\ &= \sigma_y^2 + \varsigma^2\sigma_s^2 + \frac{\sigma^2}{1+\sigma^2}\,\sigma_s^2\Big(1 - \frac{\varsigma}{\sigma}\Big)^2. \end{aligned} \tag{D.9}$$

$\varsigma^2\sigma_s^2$ increases with ς, and the estimated term falls with it. Then C_s has a minimum inside the interval. Define $u(\varsigma)$ by $C_s(\varsigma) = \sigma_y^2 + \sigma_s^2 u(\varsigma)$,

$$u(\varsigma) = \varsigma^2 + \frac{\sigma^2}{1+\sigma^2}\Big(1 - \frac{\varsigma}{\sigma}\Big)^2, \qquad u'(\varsigma) = 2\varsigma - \frac{2(\sigma - \varsigma)}{1+\sigma^2}. \tag{D.10}$$

Hence, $u'(\varsigma) = 0$ reduces to $\varsigma(1+\sigma^2) = \sigma - \varsigma$, whose solution is

$$\varsigma^\star = \frac{\sigma}{2+\sigma^2} \in (0, \sigma) \qquad (\sigma > 0), \tag{D.11}$$

and $u'' = 2(2+\sigma^2)/(1+\sigma^2) > 0$ makes it the unique minimizer. The minimizer $\varsigma^\star$ involves neither σ_s nor σ_y, and what σ_s decides is how sharply the weight concentrates around it. A direction with $\sigma_s \to 0$ has a nearly constant weight, and its mean-target approaches the midpoint of $\mathcal{T}$.

Writing $r_s^{\text{ext}}(\varsigma) := u_s^\top r^{\text{ext}}(\varsigma)$, the precision and the loss it defines are

$$\boxed{\lambda_s(\varsigma) := C_s(\varsigma)^{-1}, \qquad \mathcal{L}_\lambda(x_\sigma) := \frac{1}{2}\sum_{s=1}^{R}\int_{\mathcal{T}} \lambda_s(\varsigma)\,\big(r_s^{\text{ext}}(\varsigma)\big)^2\, d\varsigma.} \tag{D.12}$$

Proposition D.3. *Applying Theorem D.2 with $w = \lambda_s$, let $Z_s = \int_{\mathcal{T}} \lambda_s(\varsigma)\, d\varsigma$ and define the normalized density $\mu_s(\varsigma) = \lambda_s(\varsigma)/Z_s$. With $A^\top u_s = \sigma_s v_s$, the gradient of Eq. (D.12) is*

$$\nabla_{x_\sigma}\mathcal{L}_\lambda = \sum_{s=1}^{R} Z_s\,\sigma_s\,\hat{r}_s\,v_s, \qquad \bar{\varsigma}_s := \mathbb{E}_{\mu_s}[\varsigma], \qquad \hat{r}_s := r_s^{\text{ext}}(\bar{\varsigma}_s). \tag{D.13}$$

Proof. Since C_s is continuous on $[\sigma_{\min}, \sigma_{\max}]$ and bounded below by $\sigma_y^2 + \sigma_{\min}^2\sigma_s^2 > 0$, λ_s is bounded and $0 < Z_s < \infty$ for every $\mathcal{T}$ of positive measure. As a result, each direction's integral in Eq. (D.12) is Definition D.1 at $w = \lambda_s$. The component r_s^{ext} inherits the affine dependence on ς of Eq. (D.2), and the stop-gradient makes $\partial r_s^{\text{ext}}(\varsigma)/\partial x_\sigma = A^\top u_s = \sigma_s v_s$ at every ς. Write $r_s^{\text{ext}}(\varsigma) = a_s + b_s\varsigma$ with $a_s = u_s^\top a$ and $b_s = u_s^\top b$, where $b = -\sigma A\,\mathrm{sg}[s_\theta]$ and $a = Ax_\sigma - y - \sigma b$ as in the proof of Theorem D.2, and $\mu_s = \lambda_s/Z_s$. Then

$$\begin{aligned} \nabla_{x_\sigma}\mathcal{L}_\lambda = \frac{1}{2}\sum_{s=1}^{R}\int_{\mathcal{T}} \lambda_s(\varsigma)\,\nabla_{x_\sigma}\big(r_s^{\text{ext}}(\varsigma)\big)^2\, d\varsigma \;&=\; \sum_{s=1}^{R}\int_{\mathcal{T}} \lambda_s(\varsigma)\, r_s^{\text{ext}}(\varsigma)\,\frac{\partial r_s^{\text{ext}}(\varsigma)}{\partial x_\sigma}\, d\varsigma \\ = \sum_{s=1}^{R}\sigma_s\,v_s\int_{\mathcal{T}} \lambda_s(\varsigma)\, r_s^{\text{ext}}(\varsigma)\, d\varsigma \;&=\; \sum_{s=1}^{R} Z_s\,\sigma_s\,v_s\int_{\mathcal{T}} \mu_s(\varsigma)\,\big(a_s + b_s\varsigma\big)\, d\varsigma \\ = \sum_{s=1}^{R} Z_s\,\sigma_s\,v_s\big(a_s + b_s\bar{\varsigma}_s\big) \;&=\; \sum_{s=1}^{R} Z_s\,\sigma_s\, r_s^{\text{ext}}(\bar{\varsigma}_s)\,v_s \;=\; \sum_{s=1}^{R} Z_s\,\sigma_s\,\hat{r}_s\,v_s, \end{aligned}$$

where the Jacobian leaves the integral as a constant factor, $\int_{\mathcal{T}}\mu_s = 1$ and $\int_{\mathcal{T}}\varsigma\,\mu_s(\varsigma)\,d\varsigma = \bar{\varsigma}_s$. □

We next establish the support property of the mean-target. The supports are $\mathcal{T}_{\text{lf}} = [\sigma_{\min}, \sigma]$, which keeps the mean-target below the current noise level, and $\mathcal{T}_{\text{full}} = [\sigma_{\min}, \sigma_{\max}]$, which allows a mean-target noisier than the current state. The containment argument of Appendix C.3 uses only that μ_w is a probability density on $\mathcal{T}$, and it applies here with ς in place of τ. Therefore, each $\bar{\varsigma}_s$ lies strictly inside the support.

D.3 One damping factor per direction

The step solves the regularized displacement of Eq. (3.9) in each measured direction, with a coefficient $\gamma_s^{\mathrm{edm}} > 0$ of its own.

Proposition D.4. *For any $\gamma_s^{\mathrm{edm}} > 0$ the objective is quadratic in d, with stationary point $d_s = -\sigma_s h_s \hat{r}_s$ and $h_s := \big(\sigma_s^2 + \gamma_s^{\mathrm{edm}}\big)^{-1}$. Collecting the displacements gives*

$$x_\sigma^{\mathrm{new}} = x_\sigma - c \sum_{s=1}^{R} \sigma_s h_s \, \hat{r}_s \, v_s. \tag{D.14}$$

Proof. By Eq. (2.1), the measurement sees the state through $u_s^\top A x_\sigma = \sigma_s v_s^\top x_\sigma$. Therefore, moving the state to $x_\sigma + d_s v_s$ sends $\hat{r}_s$ to $\hat{r}_s + \sigma_s d_s$ and leaves every other direction where it was. Write ϕ for the objective of Eq. (3.9) at γ_s^{edm},

$$\phi(d) = \tfrac{1}{2}\big(\hat{r}_s + \sigma_s d\big)^2 + \tfrac{\gamma_s^{\mathrm{edm}}}{2} d^2, \qquad \phi''(d) = \sigma_s^2 + \gamma_s^{\mathrm{edm}} > 0,$$

which makes ϕ strictly convex with one minimizer. Setting $\phi'(d) = \sigma_s(\hat{r}_s + \sigma_s d) + \gamma_s^{\mathrm{edm}} d$ to zero and solving for d gives

$$d_s = -\frac{\sigma_s}{\sigma_s^2 + \gamma_s^{\mathrm{edm}}} \hat{r}_s = -\sigma_s h_s \hat{r}_s.$$

Summing $d_s v_s$ over the measured directions and scaling by c gives Eq. (D.14). □

At the minimizer, γ_s^{edm} decides how much of the residual the step acts on, exactly as γ_s does in Appendix C.4.

The coefficient γ_s^{edm}. Appendix C.4 builds the coefficient as a ratio of two variances: the measurement noise inside the residual at the mean-target over the conditional variance of the clean image the state holds. The same construction runs here. The measurement enters Eq. (D.7) as $-n$ whatever the virtual target, and the state keeps the clean image unscaled. Therefore, the two variances are

$$\mathrm{Var}\big(u_s^\top n\big) = \sigma_y^2, \qquad q_\sigma I_d := \mathrm{Cov}\left(x_\star \mid x_\sigma\right) = \frac{\sigma^2}{1+\sigma^2} I_d \tag{D.15}$$

by Eq. (D.8). Weighing the squared residual against σ_y^2 and the squared displacement against q_σ returns Eq. (3.9) up to a positive factor,

$$\sigma_y^2 \left[\frac{\big(\hat{r}_s + \sigma_s d\big)^2}{2\,\sigma_y^2} + \frac{d^2}{2\,q_\sigma} \right] = \frac{1}{2} \left\{ \big(\hat{r}_s + \sigma_s d\big)^2 + \frac{\sigma_y^2}{q_\sigma}\, d^2 \right\}. \tag{D.16}$$

So the coefficient we choose and the damping factor it produces are

$$\gamma_s^{\mathrm{edm}} = \frac{\sigma_y^2}{q_\sigma} = \frac{\sigma_y^2\big(1+\sigma^2\big)}{\sigma^2}, \qquad h_s = \frac{\sigma^2}{\sigma_s^2\sigma^2 + \sigma_y^2\big(1+\sigma^2\big)}. \tag{D.17}$$

Our design objective, as in Appendix C.4, is for the damping factor to depend on both the mean-target and the current sampling step. However, Eq. (D.17) depends on σ alone. This difference comes from the measurement of Eq. (D.1), where $y_\varsigma^{\mathrm{edm}} = y$ at every virtual target. To introduce dependence on the mean-target, we model an additional error δ in the component $e := x_\sigma - \hat{x}(x_\sigma)$, with $\mathrm{Cov}(\delta) = \kappa I_d$. Then the extrapolation of Eq. (D.2) is $x_\varsigma^{\mathrm{ext}} = \hat{x}(x_\sigma) + (\varsigma/\sigma)\, e$. At the mean-target, the error enters direction s of the residual scaled by $\bar{\varsigma}_s/\sigma$,

$$u_s^\top A\Big(\frac{\bar{\varsigma}_s}{\sigma}\,\delta\Big) = \frac{\bar{\varsigma}_s}{\sigma}\,\sigma_s v_s^\top \delta, \qquad \mathrm{Var}\Big(\frac{\bar{\varsigma}_s}{\sigma}\,\sigma_s v_s^\top \delta\Big) = \kappa\,\sigma_s^2 \Big(\frac{\bar{\varsigma}_s}{\sigma}\Big)^2, \tag{D.18}$$

and taking it as uncorrelated with the measurement noise,

$$\gamma_s^{\mathrm{edm}} = \frac{\big(\sigma_y^2 + \kappa\sigma_s^2\big(\bar{\varsigma}_s/\sigma\big)^2\big)\big(1+\sigma^2\big)}{\sigma^2}, \qquad h_s = \frac{\sigma^2}{\sigma_s^2\sigma^2 + \big(\sigma_y^2 + \kappa\sigma_s^2\big(\bar{\varsigma}_s/\sigma\big)^2\big)\big(1+\sigma^2\big)}. \tag{D.19}$$

For $\kappa > 0$, the resulting damping factor depends on both the mean-target $\bar{\varsigma}_s$ and the current noise level σ. We select κ on the validation set. Setting $\kappa = 0$ recovers Eq. (D.17).

E Details of experiments

This appendix reports the settings behind every run of Section 4. The support is $\mathcal{T}_{\mathrm{lf}}$ or $\mathcal{T}_{\mathrm{full}}$, which are $[t, 1]$ and $[0, 1]$ on the OT backbone and $[\sigma_{\min}, \sigma]$ and $[\sigma_{\min}, \sigma_{\max}]$ on the EDM backbone. Each operator has its own guidance scale c, and the EDM backbone additionally takes an uncertainty scale κ from $\{0,\ 0.1,\ 1\}$. The parameters are tuned per operator by grid search, and the selection is made on the highest PSNR. On the EDM backbone, the search runs on FFHQ alone, and the selected values are applied to ImageNet unchanged.

Alongside the parameters above, we add a mixing coefficient m for null-space mixing. The updates of Section 3 and Appendix D move the state along the measured directions v_s. Directions with $\sigma_s = 0$ span the null space of A, and directions whose σ_s falls far below the largest singular value behave the same way. Both are given to the prior, which fills them through the sampler's own steps. For an operator with a null space, DDNM (Wang et al., 2023) keeps the measured directions of the state consistent with y and hands the null space to the pretrained model. We follow DDNM in a mixing step that writes the model's own estimate into those directions at each step. The step is

$$x \leftarrow (1 - m)x + mP_{\mathrm{null}}x_{\mathrm{prior}} + mA^{\dagger}Ax, \qquad P_{\mathrm{null}} = I_d - A^{\dagger}A, \qquad m \in [0, 1]. \tag{E.1}$$

x_{prior} is the model's clean-image estimate with noise added back at the state it is mixed into. By the compact SVD of Section 2, the projection $A^{\dagger}A = \sum_s v_s v_s^{\top}$ takes the measured directions, and $P_{\mathrm{null}} = I_d - A^{\dagger}A$ takes the null space. Since the measured part of the state passes through untouched, Eq. (E.1) adds to the updates of Propositions 3.4 and D.4 rather than replacing them. At $m = 0$, the mixing is off. The coefficient m is tuned in the same way as the parameters above.

Appendix E.1 provides the benchmark comparison of Section 4.1, where the support, c, m and κ are all selected per operator. Appendix E.2 provides the ablation of Section 4.2, where c alone is selected, with m and κ fixed at zero. Appendix E.3 provides the trained target of Section 4.3, where the overall scale and the mixing coefficient are trained along with the evaluation targets and the update strengths. Appendix E.4 collects the reconstructions of all three. Across the three experiments, every run takes 100 sampling steps, and we report PSNR to two decimals, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) to three decimals.

E.1 Benchmark comparison

The measurement operators come from two reference protocols without modification. The pretrained EDM and OT checkpoints are the same as those used in each protocol. The random seeds are fixed as each protocol sets them.

On the EDM backbone, we follow DAPS (Zhang et al., 2025). Super-resolution downsamples by $4\times$ with a bicubic kernel. Gaussian deblurring convolves with a 61×61 kernel at $\sigma = 3$. Motion deblurring convolves with a 61×61 kernel at intensity 0.5. Box inpainting removes a 128×128 region, and Random inpainting drops 70% of the pixels. Every operator adds Gaussian measurement noise with $\sigma_y = 0.05$. The baselines are DPS (Chung et al., 2023), DDRM (Kawar et al., 2022), DDNM (Wang et al., 2023), DCDP (Li et al., 2024), DiffPIR (Zhu et al., 2023), FPS-SMC (Dou & Song, 2024), DAPS (Zhang et al., 2025) and FIG (Yan et al., 2025), and a dash marks a task a method does not report.

On the OT backbone, we follow FLOWER (Pourya et al., 2026). Its settings differ between the two datasets. At 256×256 on AFHQ-Cat, Gaussian deblurring convolves with a 61×61 kernel at $\sigma = 3$, Super-resolution downsamples by $4\times$, and Box inpainting removes an 80×80 region. At 128×128 on CelebA, the kernel stays 61×61 at $\sigma = 1$, Super-resolution downsamples by $2\times$, and Box inpainting removes a 40×40 region. On both datasets, Random inpainting drops 70% of the pixels, and Denoising is the identity operator. Every operator adds Gaussian measurement noise on both datasets, at $\sigma_y = 0.2$ for Denoising, $\sigma_y = 0.01$ for Random inpainting, and $\sigma_y = 0.05$ for Gaussian deblurring, Super-resolution, and Box inpainting. The baselines are PnP-GS (Hurault et al., 2021), DiffPIR (Zhu et al., 2023), OT-ODE (Pokle et al., 2024), Flow-Priors (Zhang et al., 2024), D-Flow (Ben-Hamu et al., 2024), PnP-Flow (Martin et al., 2025), FLOWER (Pourya et al., 2026) and FIG (Yan et al., 2025), and a dash marks a task a method does not report.

Table 5: Selected MTG parameters for the benchmark comparison in Section 4.1.

Backbone (dataset)	Task	Support	c	Mix	κ
EDM (FFHQ → ImageNet)	Super-resolution	$\mathcal{T}_{\text{lf}}$	0.255	0.01	0.0
	Gaussian deblurring	$\mathcal{T}_{\text{lf}}$	1.695	0.00	1.0
	Motion deblurring	$\mathcal{T}_{\text{lf}}$	1.285	0.00	1.0
	Random inpainting	$\mathcal{T}_{\text{lf}}$	2.000	0.05	0.1
	Box inpainting	$\mathcal{T}_{\text{full}}$	1.155	0.06	0.0
OT (AFHQ-Cat)	Denoising	$\mathcal{T}_{\text{lf}}$	0.080	0.00	–
	Gaussian deblurring	$\mathcal{T}_{\text{full}}$	1.440	1.00	–
	Super-resolution	$\mathcal{T}_{\text{full}}$	2.345	0.05	–
	Random inpainting	$\mathcal{T}_{\text{lf}}$	1.460	0.05	–
	Box inpainting	$\mathcal{T}_{\text{full}}$	0.250	0.60	–
OT (CelebA)	Denoising	$\mathcal{T}_{\text{lf}}$	0.070	0.00	–
	Gaussian deblurring	$\mathcal{T}_{\text{full}}$	0.280	1.00	–
	Super-resolution	$\mathcal{T}_{\text{full}}$	1.970	0.04	–
	Random inpainting	$\mathcal{T}_{\text{lf}}$	1.380	0.07	–
	Box inpainting	$\mathcal{T}_{\text{full}}$	0.375	0.15	–

E.1.1 Implementation notes

Two points of the implementation need stating for this experiment.

Fourier-diagonal operators. For Gaussian deblurring, Motion deblurring, and Super-resolution on the EDM backbone, the implementation computes σ_s in the discrete Fourier basis, where $AA^\top$ is taken to be diagonal. However, these operators are not exactly diagonal there. The coefficient γ_s^{edm} already suppresses the update as $\sigma_s \to 0$, but it does not account for the difference between the model and the implemented operator. We measure that difference once per operator with a random vector $z \sim \mathcal{N}(0, I_m)$ and form a constant in proportion to it,

$$m_A = \frac{\left\| AA^\top z - \mathcal{F}^{-1}(\boldsymbol{\eta} \odot \mathcal{F}z) \right\|^2}{\|AA^\top z\|^2}, \qquad \delta_A = 2\sqrt{m_A}\, \max_{s'} \sigma_{s'}^2. \tag{E.2}$$

$\boldsymbol{\eta}$ holds the modeled eigenvalues of $AA^\top$ and $\odot$ is entrywise multiplication. We add δ_A to the coefficient γ_s^{edm} to limit residual amplification while keeping the mean-target unchanged,

$$\tilde{\gamma}_s^{\text{edm}} = \gamma_s^{\text{edm}} + \delta_A, \qquad \sigma_s \tilde{h}_s = \frac{\sigma_s}{\sigma_s^2 + \tilde{\gamma}_s^{\text{edm}}} \le \frac{1}{2\sqrt{\tilde{\gamma}_s^{\text{edm}}}}. \tag{E.3}$$

The update applies $\sigma_s \tilde{h}_s$ to the residual in direction s, and the bound is reached when $\sigma_s = \sqrt{\tilde{\gamma}_s^{\text{edm}}}$. The measured m_A is 0.035 for Gaussian deblurring, 0.059 for Motion deblurring, and 0.004 for Super-resolution. Removing δ_A while holding every other parameter fixed made sampling diverge on the deblurring tasks.

Identity operator. Denoising on the OT backbone uses the identity operator. Appendix C.2 derives its modified residual variance. Therefore, OT denoising weights the residual by Eq. (C.7) and damps it by Eq. (C.13).

E.1.2 Selected parameters

Table 5 lists the selected MTG parameters for each task and dataset. We tune FIG (Yan et al., 2025) under the same protocol, and Table 6 lists the selected FIG parameters.

E.1.3 The derived mean-target and damping factor

Figures 6–8 draw both quantities of Propositions 3.4 and D.4 at every guided step, one figure per dataset and one column per task. The top row of each figure holds the evaluation target against the current state and the data endpoint, and the bottom row holds the damping factor. Every cell draws thirty directions, taken at thirty evenly spaced percentiles of the spectrum from its smallest value to its largest. A flat spectrum offers one singular value, and its thirty curves coincide.

Table 6: Selected FIG parameters for the benchmark comparison in Section 4.1.

Backbone (dataset)	Task	Iterations	c	Mix
EDM (FFHQ → ImageNet)	Super-resolution	2	0.300	1.00
	Gaussian deblurring	1	0.470	1.00
	Motion deblurring	7	0.050	1.00
	Random inpainting	5	0.025	0.95
	Box inpainting	7	0.070	0.75
OT (AFHQ-Cat)	Denoising	3	7.0	0.00
	Gaussian deblurring	40	12.0	1.00
	Super-resolution	3	70.0	1.00
	Random inpainting	48	50.0	0.95
	Box inpainting	1	350.0	0.20
OT (CelebA)	Denoising	2	5.0	0.00
	Gaussian deblurring	1	100.0	1.00
	Super-resolution	1	35.0	0.90
	Random inpainting	9	40.0	0.90
	Box inpainting	1	180.0	0.20

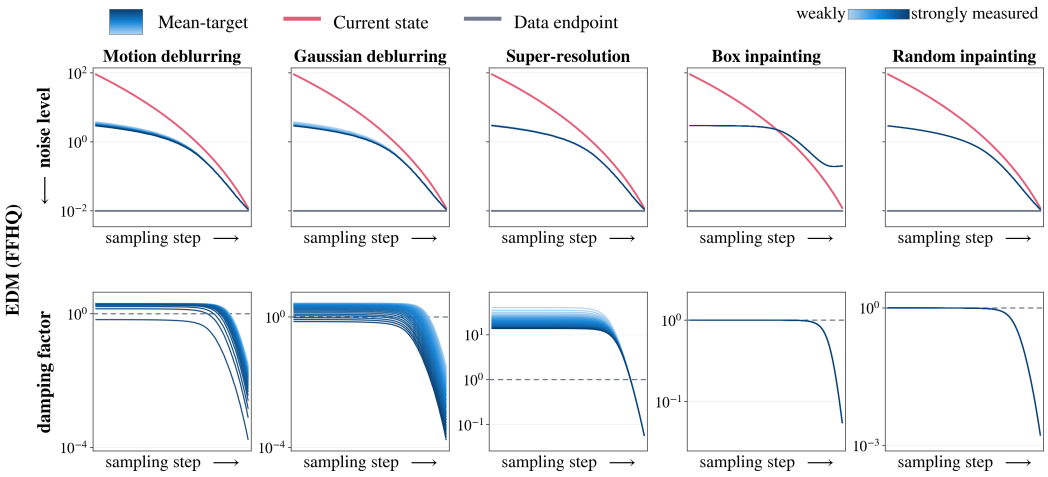

Figure 6: Mean-targets and damping factors across sampling steps for MTG on FFHQ using the EDM backbone. The top row shows the mean-target $\bar{\varsigma}_s$ together with the current-state and data-endpoint targets, and the bottom row shows the damping factor h_s.

E.1.4 Full comparison tables

Except for FIG (Yan et al., 2025), the baseline results in Tables 7 and 8 are taken from DAPS (Zhang et al., 2025) and FLOWER (Pourya et al., 2026), respectively, under their corresponding evaluation protocols.

E.2 Ablation of the mean-target and damping factor

The ablation needs directions where the mean-target and the coefficient $\sigma_s g_s$ both still vary, and none of the ten settings of Appendix E.1 supplies them. Denoising and the two inpainting operators are masks, whose measured directions share one singular value. Then the pair takes one value per step. Gaussian deblurring, Motion deblurring, and Super-resolution keep a spectrum, but most of it lies far below σ_y. The mean-target and the damping factor are set by σ_s against σ_y, and they saturate wherever the spectrum sits on one side of that level. Separating them needs directions whose singular values straddle the measurement noise. Since the operator of Section 4.2 keeps every direction observed and spreads its singular values across σ_y, it is the setting this ablation needs. On the OT

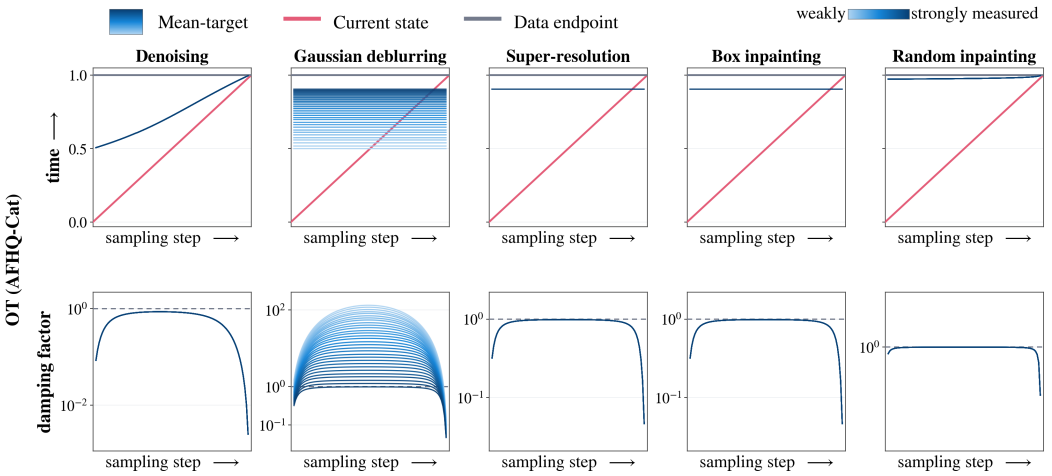

Figure 7: Mean-targets and damping factors across sampling steps for MTG on AFHQ-Cat using the OT backbone. The top row shows the mean-target $\bar{\tau}_s$ together with the current-state and data-endpoint targets, and the bottom row shows the damping factor g_s.

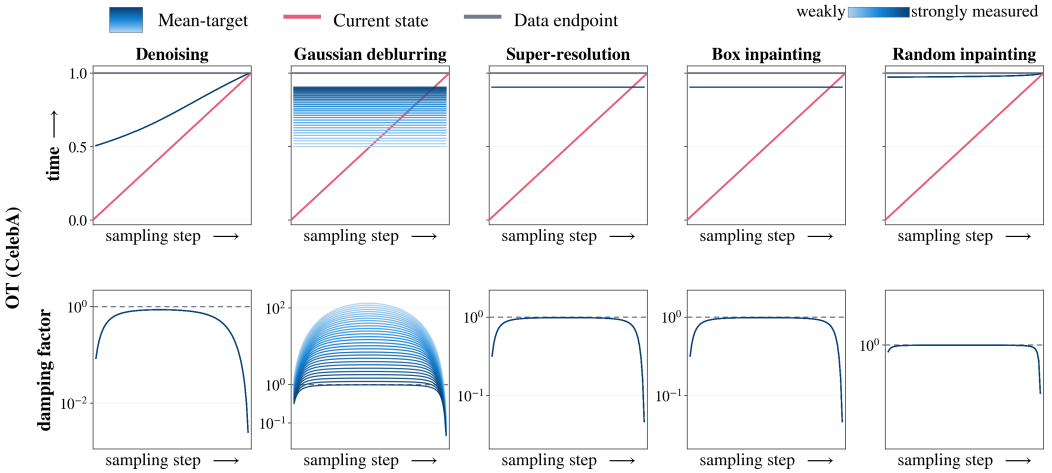

Figure 8: Mean-targets and damping factors across sampling steps for MTG on CelebA using the OT backbone. The top row shows the mean-target $\bar{\tau}_s$ together with the current-state and data-endpoint targets, and the bottom row shows the damping factor g_s.

backbone, we use the flow-matching version of DPS released with FIG (Yan et al., 2025), ported from its implementation without change.

E.2.1 The operator

The operator is $A = \mathrm{diag}(M)$ with a per-pixel weight $M \in (0, 1]$. Then the measurement is $y = M \odot x_\star + \sigma_y \varepsilon_y$, and the direction of each pixel has σ_s equal to that pixel's weight. The mask decays geometrically across the frame, with a slow sinusoidal variation on top. Writing $(k, l) \in [0, 1]^2$ for the pixel's position in the image, with $(0, 0)$ at one corner and $(1, 1)$ at the opposite one, the exponent is

$$\xi(k, l) = \tfrac{1}{2}(k + l) + 0.12 \sin\big(2\pi \cdot 1.3\, k + 0.7\big) \sin\big(2\pi \cdot 1.1\, l + 1.3\big), \tag{E.4}$$

clamped to $[0, 1]$, and the mask is $M(k, l) = m_{\min}^{\xi(k,l)}$ with $m_{\min} = 0.03$ at $\sigma_y = 0.05$. The singular values run from 1 down to 0.03 and cross σ_y inside the frame. Every entry of M is at least $m_{\min}$.

Table 7: Full benchmark results on 100 test images from each of FFHQ and ImageNet using the EDM backbone. For each task, dataset, and metric, the best result is **bold** and the second-best is underlined. Purple/blue cells indicate $\mathcal{T}_{\text{lf}}$ / $\mathcal{T}_{\text{full}}$. A dash indicates an unreported result.

Task	Method	FFHQ			ImageNet		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Super-resolution	DPS	25.86	0.753	0.269	21.13	0.489	0.361
	DDRM	26.58	0.782	0.282	22.62	0.521	0.324
	DDNM	28.03	0.795	0.197	23.96	0.604	0.475
	DCDP	28.66	0.807	0.178	-	-	-
	DiffPIR	26.64	-	0.260	23.18	-	0.371
	FPS-SMC	28.42	0.813	0.204	24.82	0.703	0.313
	DAPS	29.07	0.818	0.177	25.89	0.694	0.276
	FIG	29.52	0.836	0.200	25.55	0.675	0.302
	MTG (Ours)	**30.94**	**0.876**	**0.160**	**26.38**	**0.722**	**0.264**
Box inpainting	DPS	22.51	0.792	0.209	18.94	0.722	0.257
	DDRM	22.26	0.801	0.207	18.63	0.733	0.254
	DDNM	24.47	0.837	0.235	21.64	0.748	0.319
	DCDP	23.89	0.760	0.163	-	-	-
	FPS-SMC	24.86	0.823	0.146	**22.16**	0.726	**0.208**
	DAPS	24.07	0.814	**0.133**	21.43	0.725	0.214
	FIG	24.72	0.757	0.188	21.13	0.714	0.260
	MTG (Ours)	**25.55**	**0.872**	0.152	20.46	**0.773**	0.278
Random inpainting	DPS	25.46	0.823	0.203	23.52	0.745	0.297
	DDNM	29.91	0.817	0.121	**31.16**	0.841	0.191
	DCDP	30.69	0.842	0.142	-	-	-
	FPS-SMC	28.21	0.823	0.261	24.52	0.701	0.316
	DAPS	31.12	0.844	0.098	28.44	0.775	0.135
	FIG	33.35	0.909	0.083	29.51	0.838	**0.118**
	MTG (Ours)	**35.20**	**0.946**	**0.066**	30.03	**0.861**	0.129
Gaussian deblurring	DPS	25.87	0.764	0.219	20.31	0.598	0.397
	DDRM	24.93	0.732	0.239	21.26	0.564	0.443
	DDNM	28.20	0.804	0.216	**28.06**	0.703	0.278
	DCDP	27.50	0.699	0.304	-	-	-
	DiffPIR	27.36	-	0.236	22.80	-	0.355
	FPS-SMC	26.54	0.773	0.253	23.91	0.601	0.387
	DAPS	29.19	0.817	0.165	26.15	0.684	0.253
	FIG	29.59	0.840	0.190	25.60	0.680	0.293
	MTG (Ours)	**30.47**	**0.865**	**0.164**	26.32	**0.712**	**0.252**
Motion deblurring	DPS	24.52	0.801	0.246	18.96	0.629	0.423
	DCDP	25.08	0.512	0.364	-	-	-
	DiffPIR	26.57	-	0.255	24.01	-	0.366
	FPS-SMC	27.39	0.826	0.227	24.52	0.647	0.326
	DAPS	29.66	0.847	0.157	27.86	0.766	0.196
	FIG	31.26	0.853	0.174	27.43	0.756	0.253
	MTG (Ours)	**33.78**	**0.918**	**0.102**	**29.67**	**0.834**	**0.152**

Therefore, every direction remains observed, and none is left to the prior. Since the decay spans the entire frame, every pixel is attenuated. This separates the operator from the inpainting masks, where the visible pixels arrive intact.

E.2.2 Selected parameters

Each of the six MTG variants and the two baselines is tuned separately on each backbone. Table 9 lists the selected parameters.

Table 8: Full benchmark results on 100 test images from each of AFHQ-Cat and CelebA using the OT backbone. For each task, dataset, and metric, the best result is **bold** and the second-best is underlined. Purple/blue cells indicate $\mathcal{T}_{\text{lf}}$ / $\mathcal{T}_{\text{full}}$. A dash indicates an unreported result.

Task	Method	AFHQ-Cat			CelebA		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Denoising	PnP-GS	**32.58**	0.894	0.072	32.64	0.910	0.035
	DiffPIR	30.58	0.835	0.189	31.20	0.885	0.060
	OT-ODE	30.03	0.815	0.076	30.54	0.859	0.032
	D-Flow	26.13	0.574	0.175	26.04	0.607	0.092
	Flow-Priors	29.41	0.763	0.153	29.34	0.768	0.134
	PnP-Flow	31.18	0.863	0.135	31.80	0.905	0.044
	Flower1-OT	31.69	0.879	0.102	32.28	0.914	0.034
	FIG	32.48	**0.897**	**0.042**	**33.01**	0.921	**0.016**
	MTG (Ours)	32.43	**0.897**	0.043	**33.01**	**0.923**	**0.016**
Gaussian deblurring	PnP-GS	27.91	0.753	0.349	34.03	0.924	0.041
	DiffPIR	27.56	0.728	0.342	32.77	0.912	0.060
	OT-ODE	27.06	0.713	**0.123**	33.01	0.921	0.029
	D-Flow	27.82	0.721	0.164	31.25	0.854	0.038
	Flow-Priors	26.47	0.700	0.181	31.54	0.858	0.056
	PnP-Flow	27.87	0.760	0.304	34.48	0.936	0.040
	Flower1-OT	28.64	0.775	0.255	34.98	0.947	0.026
	FIG	29.01	0.788	0.250	**35.82**	**0.956**	**0.023**
	MTG (Ours)	**29.18**	**0.793**	0.240	35.76	0.954	0.030
Super-resolution	PnP-GS	24.15	0.632	0.362	31.31	0.892	0.064
	DiffPIR	23.65	0.624	0.402	31.52	0.895	0.033
	OT-ODE	25.91	0.716	**0.108**	31.46	0.907	0.025
	D-Flow	24.64	0.601	0.190	30.47	0.843	0.026
	Flow-Priors	23.51	0.570	0.272	28.35	0.713	0.102
	PnP-Flow	26.94	0.763	0.171	31.09	0.902	0.045
	Flower1-OT	26.23	0.741	0.272	32.36	0.923	0.034
	FIG	26.24	0.730	0.185	33.00	0.934	**0.020**
	MTG (Ours)	**28.59**	**0.805**	0.187	**34.05**	**0.943**	0.030
Random inpainting	PnP-GS	29.42	0.836	0.126	29.22	0.875	0.070
	DiffPIR	31.70	0.881	0.062	31.74	0.917	0.025
	OT-ODE	29.40	0.839	0.090	28.68	0.871	0.051
	D-Flow	32.20	0.894	0.040	33.67	0.943	0.015
	Flow-Priors	32.37	0.906	0.047	32.88	0.871	0.019
	PnP-Flow	33.00	0.918	0.037	33.05	0.944	0.018
	Flower1-OT	32.97	0.918	0.040	33.08	0.944	0.018
	FIG	32.93	0.915	0.045	33.26	0.950	0.017
	MTG (Ours)	**34.51**	**0.936**	**0.036**	**35.49**	**0.967**	**0.013**
Box inpainting	PnP-GS	-	-	-	-	-	-
	DiffPIR	-	-	-	-	-	-
	OT-ODE	24.62	0.875	0.085	29.40	0.920	0.038
	D-Flow	26.26	0.842	0.077	30.70	0.899	0.026
	Flow-Priors	26.20	0.818	0.118	30.07	0.858	0.048
	PnP-Flow	26.00	0.897	0.103	30.47	0.933	0.037
	Flower1-OT	26.19	0.915	0.063	31.19	0.945	**0.022**
	FIG	24.08	0.815	0.140	27.00	0.858	0.060
	MTG (Ours)	**27.16**	**0.916**	**0.055**	**32.28**	**0.949**	0.023

E.3 Trained-Target Guidance

Section 4.3 trains the evaluation target and the update strength directly at every sampling step, without deriving either. Training returns one value per step for each quantity, and those values do not state which assumptions would produce them. They cannot point to a better assumption either. Therefore, this experiment only asks which evaluation target and update strength come out of training for solving inverse problems, with neither quantity constrained.

Table 9: Selected parameters for the mean-target and damping factor ablation in Section 4.2, including the MTG variants and the FIG and DPS baselines. Constant denotes $g_s = 1$.

Backbone (dataset)	Method	Target $\bar{\tau}_s$	Damping g_s	Support	Iterations	c
EDM (FFHQ)	**MTG (Ours)**	**Mean-target**	**Derived**	$\mathcal{T}_{\text{lf}}$	—	0.155
	MTG w/o mean-target	Fixed (Current)	**Derived**	—	—	0.290
	MTG w/o mean-target	Fixed (Data)	**Derived**	—	—	0.055
	MTG w/o damping	**Mean-target**	Constant	$\mathcal{T}_{\text{full}}$	—	2.285
	MTG w/o both	Fixed (Current)	Constant	—	—	2.695
	MTG w/o both	Fixed (Data)	Constant	—	—	1.975
	FIG	Current	Own	—	8	0.095
	DPS	Data	Own	—	—	1.945
OT (AFHQ-Cat)	**MTG (Ours)**	**Mean-target**	**Derived**	$\mathcal{T}_{\text{full}}$	—	0.120
	MTG w/o mean-target	Fixed (Current)	**Derived**	—	—	0.105
	MTG w/o mean-target	Fixed (Data)	**Derived**	—	—	0.045
	MTG w/o damping	**Mean-target**	Constant	$\mathcal{T}_{\text{full}}$	—	1.620
	MTG w/o both	Fixed (Current)	Constant	—	—	2.695
	MTG w/o both	Fixed (Data)	Constant	—	—	0.675
	FIG	Current	Own	—	32	1.735
	DPS	Data	Own	—	—	4.995

E.3.1 The optimizer

Training minimizes the mean squared error between the sampler's output and the clean image. One evaluation runs the full 100-step sampler, and a gradient in the parameters would have to pass through the score network at every step. We trained the parameters with Adam and with SGD, and neither converged. Therefore, training uses CMA-ES (Hansen & Ostermeier, 2001), which needs only the value of the objective. It keeps a Gaussian over the parameter vector and samples a population of candidates from it each generation. The candidates are ranked by objective, and the mean moves toward the better half. Then the covariance widens along the directions in which the selected candidates have been moving. We run CMA-ES for 100 generations at a population of 32, from an initial step size of 0.5 on every coordinate, with its own random seed fixed. We use the `pycma` package.[1]

E.3.2 Parameterization

TTG retains the form of the EDM update in Eq. (D.14) and replaces the derived mean-target and damping factor with trainable per-step parameters. At sampling step i with noise level σ_i, one score evaluation fixes the frozen residual of Eq. (D.2), and TTG applies

$$x^{\text{new}} = x_{\sigma_i} - c^\theta h_i^\theta A^\top \big[A x_{\sigma_i} - y + (\sigma_i - \varsigma_i^\theta)\, \sigma_i\, s_\theta(x_{\sigma_i}, \sigma_i) \big]. \tag{E.5}$$

The trained ς_i^θ decides where the residual is formed, and $c^\theta h_i^\theta$ scales what the state receives. Where every observed direction has the same σ_s, one h_i^θ per step covers the whole of h_s, and the task trained here has a single observed singular value.

Each ς_i^θ is stored as an unconstrained θ_i and mapped through a sigmoid log-uniformly onto the sampler's own noise range $[\sigma_{\min}, \sigma_{\max}]$, which keeps a target behind the current state reachable and every trained value one the derived sampler could also express. The coefficient is $c^\theta \exp(\theta_i^h)$ with $c^\theta = \text{softplus}(\theta^c) > 0$, where c^θ adds a degree of freedom the per-step terms already cover. Overall magnitude is the dominant direction of variation, and one coordinate hands it to the optimizer instead of making it find a coordinated move across 97 axes. The mixing coefficient passes through a sigmoid into $[0, 1]$. Guidance is active at $i \in \{2, \ldots, 98\}$ out of 100 steps, which gives 97 targets, 97 coefficients, one global scale and one mixing coefficient, 196 parameters in all.

Algorithm 2 shows how TTG is trained. The target is initialized at the midpoint of the log noise range, while the update strength, global scale $c = 1.155$, and mixing coefficient $m = 0.06$ are

[1] https://github.com/CMA-ES/pycma

Algorithm 2 Training TTG.

Require: A, measurements $\{y^{(n)}\}_{n=1}^{B}$, schedule $\sigma_{\min} = \sigma_0 < \cdots < \sigma_N = \sigma_{\max}$, active steps $\mathcal{I}$, clean images $\{x_\star^{(n)}\}_{n=1}^{B}$, candidate $\theta = \{\theta_i, \theta_i^h\}_{i\in\mathcal{I}} \cup \{\theta^c, \theta^m\}$

$\varsigma_i^\theta \leftarrow \sigma_{\min}(\sigma_{\max}/\sigma_{\min})^{\mathrm{sigmoid}(\theta_i)}$
$c^\theta \leftarrow \mathrm{softplus}(\theta^c), \quad h_i^\theta \leftarrow \exp(\theta_i^h), \quad m \leftarrow \mathrm{sigmoid}(\theta^m)$
for $n = 1, \ldots, B$ **do**
  $x \sim \mathcal{N}(0, \sigma_N^2 I)$
  for $i = N, \ldots, 1$ **do**
    $s_\theta \leftarrow s_\theta(x, \sigma_i)$
    if $i \in \mathcal{I}$ **then**
      $x \leftarrow x - c^\theta h_i^\theta A^\top \left[Ax - y^{(n)} + (\sigma_i - \varsigma_i^\theta)\sigma_i s_\theta\right]$
      $x_{\text{prior}} \leftarrow x + \sigma_i^2 s_\theta + \sigma_i \epsilon, \quad \epsilon \sim \mathcal{N}(0, I_d)$
      $x \leftarrow (1-m)x + mP_{\text{null}}x_{\text{prior}} + mA^\dagger Ax$
    end if
    $x \leftarrow x + \sigma_i(\sigma_i - \sigma_{i-1})s_\theta$
  end for
  $\ell_n \leftarrow \|x - x_\star^{(n)}\|^2/d$
end for
return $\frac{1}{B}\sum_{n=1}^{B} \ell_n$



Figure 9: MTG reconstructions on FFHQ using the EDM backbone. From top to bottom, the rows show original images, measurements, and reconstructions.

initialized to their MTG values. It returns $c^\theta = 0.769$ and $m = 0.126$ after 100 generations, and one run takes about 17 hours on one GPU.

E.4 Reconstructions

Figures 9–12 show MTG's output on each of the five operators of Appendix E.1, one figure per dataset, with every column drawn from a different test image. Figure 13 puts the eight configurations of Appendix E.2 on four test problems. Figure 14 puts the trained pair of Appendix E.3 beside the derived one on five test problems. The test images shown in these figures were randomly selected from the respective test sets.

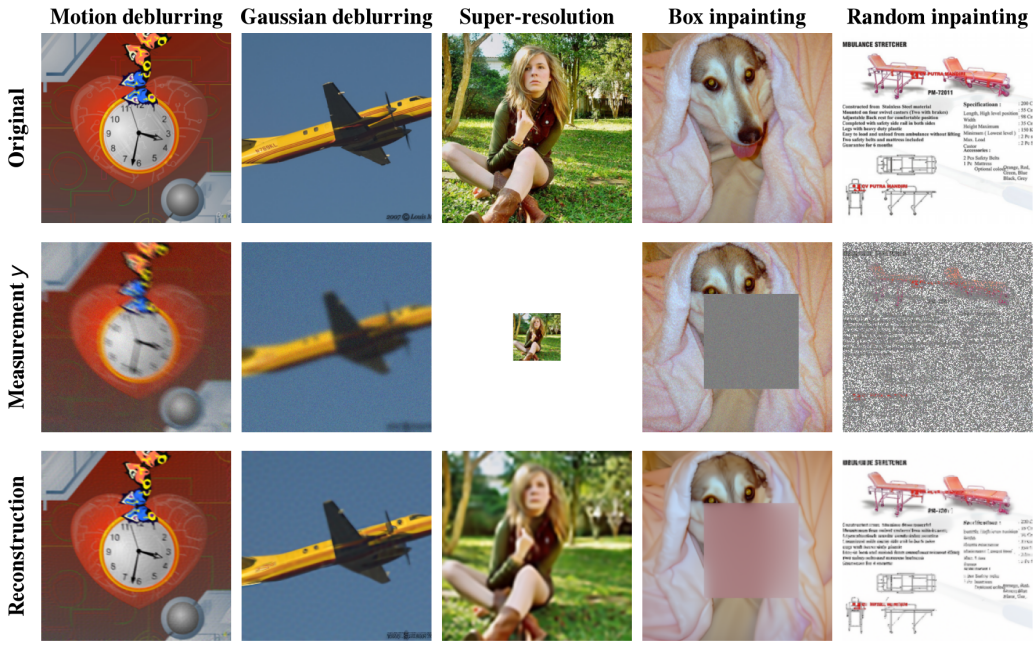

Figure 10: MTG reconstructions on ImageNet using the EDM backbone. From top to bottom, the rows show original images, measurements, and reconstructions.

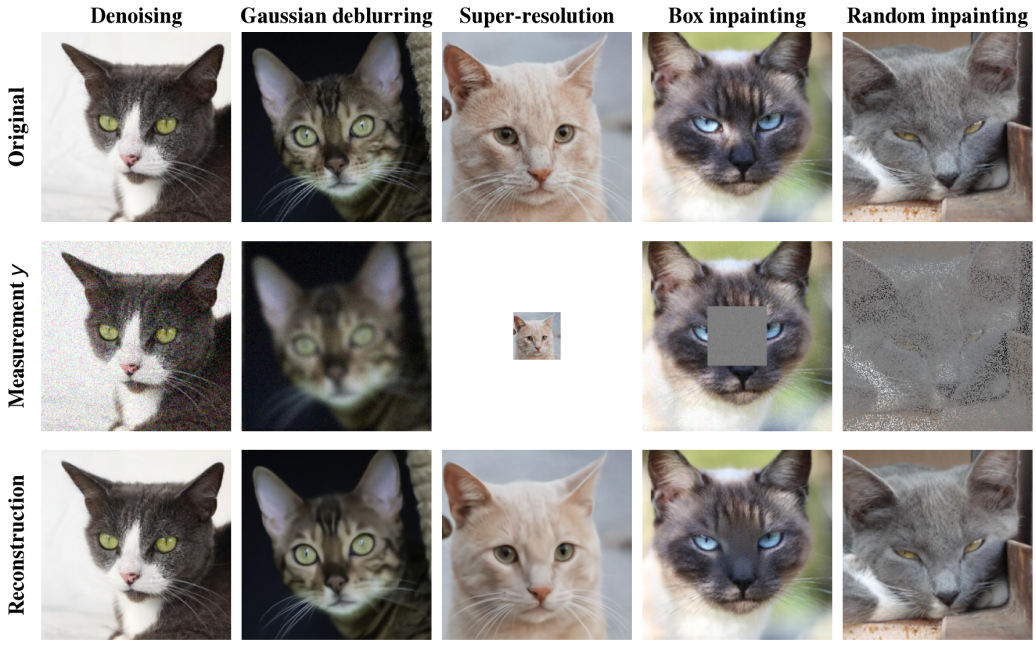

Figure 11: MTG reconstructions on AFHQ-Cat using the OT backbone. From top to bottom, the rows show original images, measurements, and reconstructions.

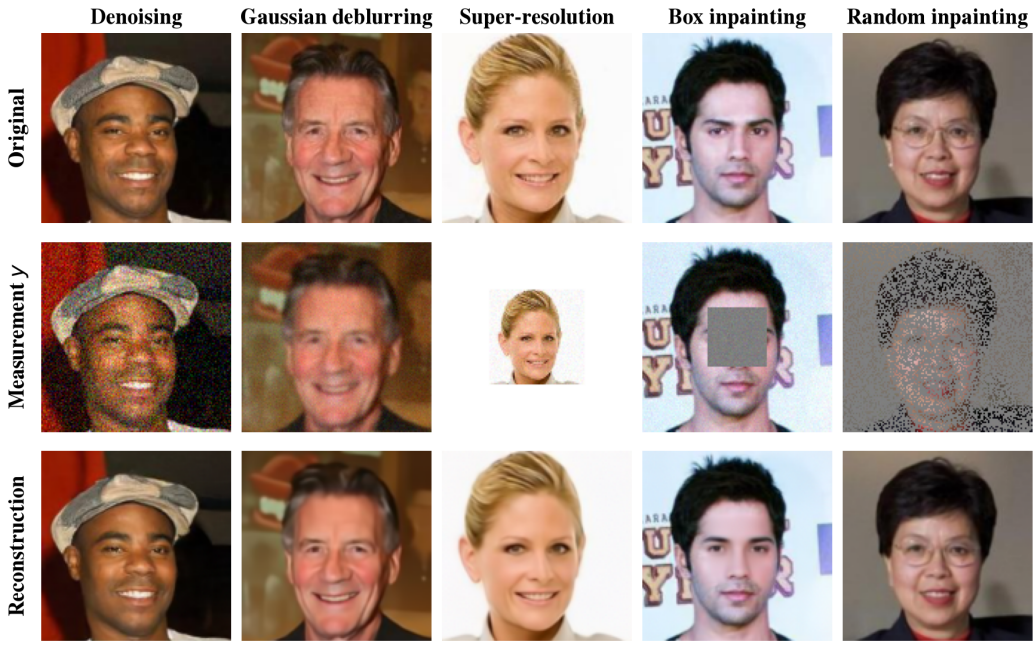

Figure 12: MTG reconstructions on CelebA using the OT backbone. From top to bottom, the rows show original images, measurements, and reconstructions.

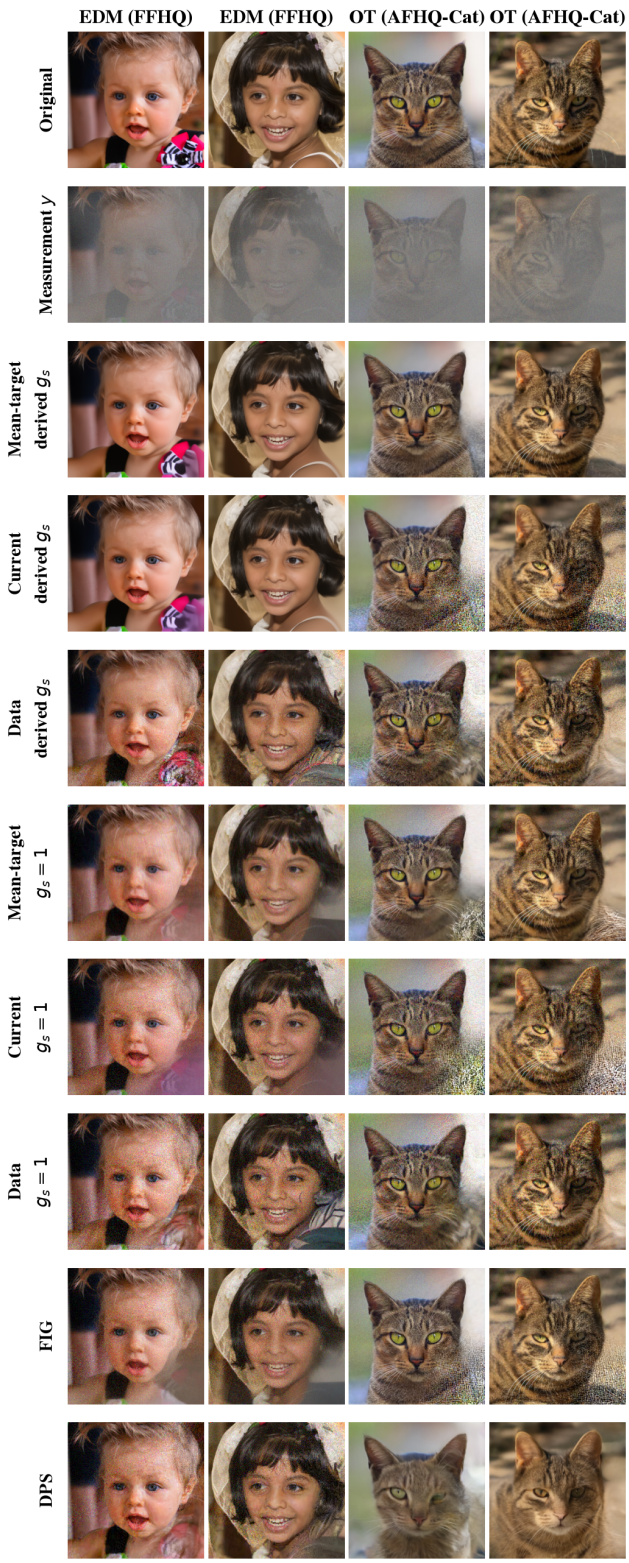

Figure 13: Reconstruction results for the target and damping ablation in Section 4.2 on four test problems, with FIG and DPS as baselines.

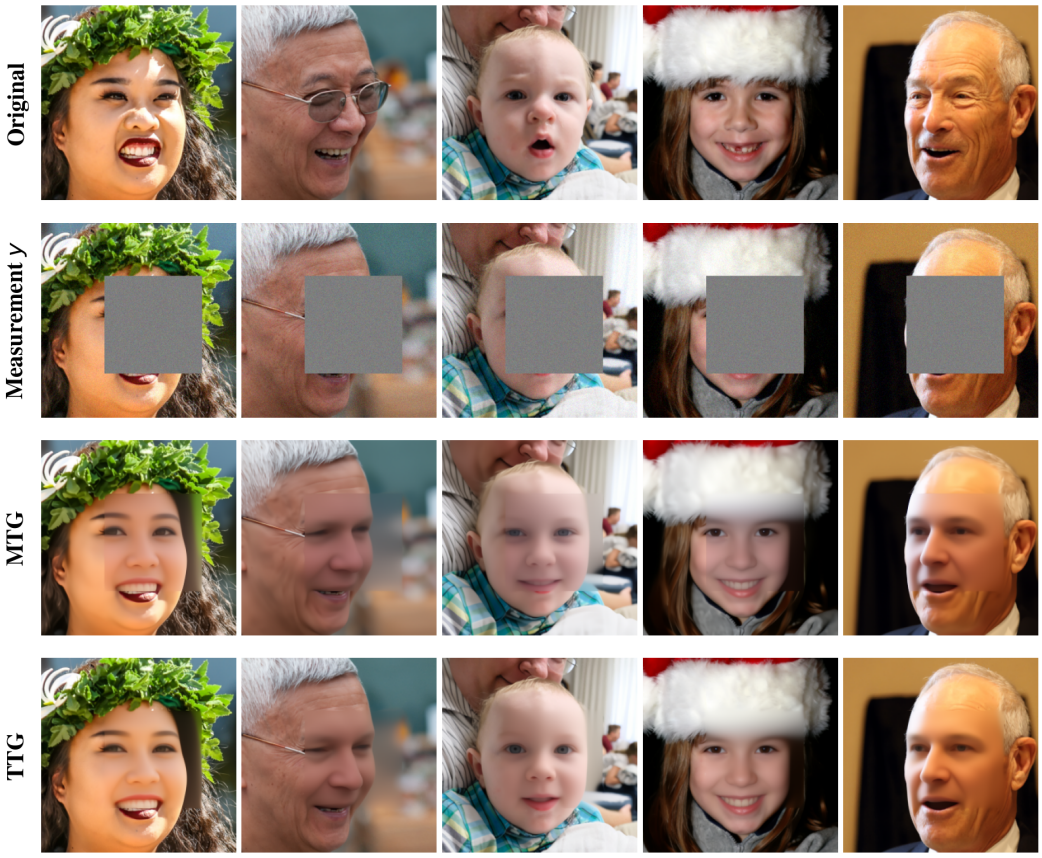

Figure 14: MTG and TTG reconstructions for Box inpainting on FFHQ using the EDM backbone.