

NOISE IS ALL YOU NEED: SOLVING LINEAR INVERSE PROBLEMS BY NOISE COMBINATION SAMPLING WITH DIFFUSION MODELS

Xun Su* Hiroyuki Kasai*

WASEDA University, Tokyo, Japan

ABSTRACT

Pretrained diffusion models have demonstrated strong capabilities in zero-shot inverse problem solving by incorporating observation information into the generation process of the diffusion models. However, this presents an inherent dilemma: excessive integration can disrupt the generative process, while insufficient integration fails to emphasize the constraints imposed by the inverse problem. To address this, we propose *Noise Combination Sampling*, a novel method that synthesizes an optimal noise vector from a noise subspace to approximate the measurement score, replacing the noise term in the standard Denoising Diffusion Probabilistic Models process. This enables conditional information to be naturally embedded into the generation process without reliance on step-wise hyperparameter tuning. Our method can be applied to a wide range of inverse problem solvers, including image compression, and, particularly when the number of generation steps T is small, achieves superior performance with negligible computational overhead, significantly improving robustness and stability.

1 INTRODUCTION

Diffusion models have emerged as a powerful class of generative models, achieving state-of-the-art results in high-fidelity image generation, audio synthesis, video modeling, and even language modeling (Ho et al., 2020; Rombach et al., 2022; Podell et al., 2024; Ho et al., 2022; Nie et al., 2025; Gat et al., 2024). Beyond their impressive generative capabilities, researchers have also recognized their strong zero-shot potential for a variety of related tasks, including inpainting, depth estimation, segmentation, and classification (Lugmayr et al., 2022; Tian et al., 2024; Li et al., 2023).

Notably, it has been shown that diffusion models, without any additional training, can effectively solve general linear inverse problems by injecting information during the stochastic denoising steps (Wang et al., 2023; Chung et al., 2023; Cardoso et al.; Dou & Song, 2024; He et al., 2024; Kim et al., 2025). These tasks include denoising, inpainting, and super-resolution. Such powerful plug-and-play, training-free approaches have been adopted in more complex settings, including non-linear and deep learning-guided generation tasks such as style transfer and image editing (Yu et al., 2023; He et al., 2024; Shi et al., 2024; Ye et al., 2024). However, since the imposed guidance drives the trajectory away from the manifold of real data and disrupts the consistency of the generation process, existing methods suffer from sampling instability and rely on complex hyperparameter tuning and long sampling schedules to compensate for the limited influence of the guidance (Wang et al., 2023; Song et al., 2023; Yang et al., 2024; Zhang et al., 2025).

In this paper, we propose *Noise Combination Sampling* (NCS), a novel framework that approximates the measurement score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t)$ using a linear combination of Gaussian noise vectors to replace the noise term in the Denoising Diffusion Probabilistic Models (DDPM) process. By adjusting the sampling trajectory through constructed noise vectors rather than gradients, NCS mitigates the instability inherent in existing sampling-based inverse problem solvers, eliminating the need for carefully-tuned hyperparameters and preserving the denoising behavior of the generative model. Remarkably, the mathematically optimal combination weights can be derived in closed form via the Cauchy-Schwarz inequality, requiring negligible additional computation.

*Emails: suxun_opt@asagi.waseda.jp, hiroyuki.kasai@waseda.jp

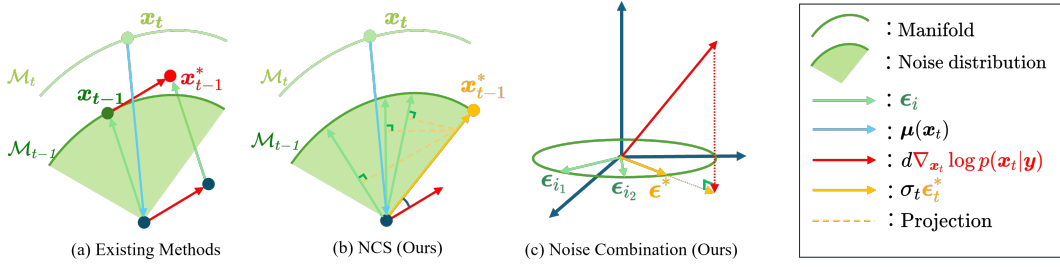


Figure 1: An illustration showing the difference between exact approximation methods and NCS. The intervention, i.e., the **measurement score** in the existing methods pushes the trajectory off the manifold $\mathcal{M}_{t-1}$ of x_{t-1} . In contrast, NCS embeds the intervention into the optimal noise within an ellipsoidal subspace, defined by the span of the noise codebook. This allows NCS to naturally preserve both the position of x_{t-1} on its manifold and the consistency of the diffusion process.

This approach can be seamlessly integrated into various mainstream inverse problem solvers, including Diffusion Posterior Sampling (DPS) (Chung et al., 2023) and Manifold-Preserving Gradient Descent (MPGD) (He et al., 2024), consistently yielding substantial improvements. Notably, a recently proposed impressive generative image compression method—Denoising Diffusion Codebook Models (DDCM) (Ohayon et al., 2025), can be viewed as a special case of NCS, in which the top- m noise vectors ($m = 1$ in DDCM) are selected from a codebook of size K , corresponding to an extremely low compression ratio. To improve compression quality, DDCM employs a greedy search with exponential complexity in m to approximate the measurement score. In contrast, we demonstrate that NCS solution can achieve a competitive quantization result with linear complexity.

Our contributions are summarized as follows:

- We propose the NCS framework, which leverages the noise variables in the DDPM process to approximate the measurement score in inverse problem solving. By optimally combining multiple Gaussian noise vectors, NCS synthesizes a noise sample that closely approximates the desired conditional distribution. We show that this optimization problem admits a closed-form solution via the Cauchy–Schwarz inequality.
- We demonstrate that prominent diffusion-based inverse problem solvers can be naturally reformulated under the NCS framework. Using DPS and MPGD as examples, we show the effectiveness of NCS. We further conjecture that most existing gradient-based approaches admit corresponding NCS formulations, offering a unified and principled perspective.
- We show that the generative image compression method DDCM can be viewed as a special case of NCS. By increasing the number of combined noise vectors and reducing the number of sampling steps, we significantly accelerate both compression and decompression, with negligible quality loss and linear complexity. This is possible because the complexity of the proposed quantization of the optimal noise combination is nearly independent of m .
- We conduct experiments across multiple datasets and inverse problem tasks, showing that NCS consistently outperforms existing methods that rely on careful hyperparameter tuning. In particular, NCS demonstrates remarkable robustness and stability when the number of diffusion steps T is small, achieving high-quality results at reduced sampling cost.

2 BACKGROUND

2.1 DENOISING DIFFUSION PROBABILISTIC MODELS (DDPMs)

Diffusion models define the generative process as the reverse of a predefined noising process. Following the formulation of Song et al. (2021), we describe the forward (noising) process using an Itô stochastic differential equation (SDE), where $x_t \in \mathbb{R}^d$ and $t \in [0, T]$:

$$dx_t = f(x_t, t) dt + g(t) dw_t, \quad (1)$$

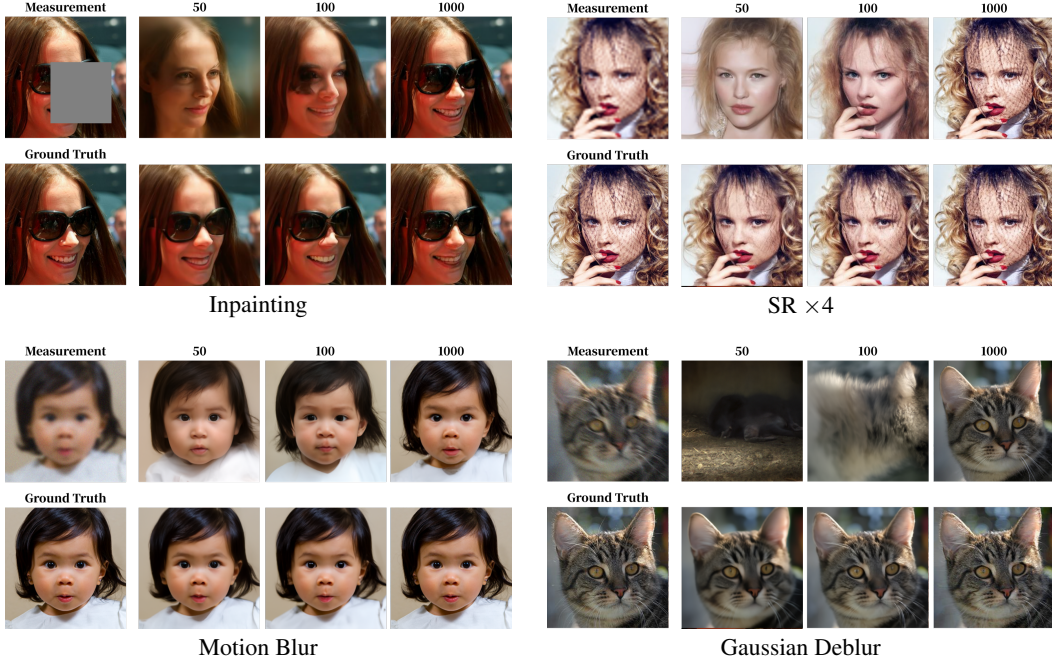


Figure 2: Comparison of DPS and NCS-DPS across four inverse problems under varying sampling steps. NCS-DPS yields clearer details and greater stability, especially at small step counts.

where f and g denote the drift function and diffusion coefficient, respectively, and w_t is a standard Wiener process.

Assuming the initial data distribution is $x_0 \sim p_{\text{data}}$ and the terminal distribution x_T is Gaussian, i.e., $\mathcal{N}(0, \mathbf{I})$, the goal of the generative (reverse) process is to recover samples from p_{data} by reversing the diffusion trajectory. According to Anderson (1982), this can be achieved by solving the reverse-time SDE:

$$dx_t = (f(x_t, t) - g^2(t) \nabla_{x_t} \log p_t(x_t)) dt + g(t) dw_t, \quad (2)$$

initialized at $x_T \sim \mathcal{N}(0, \mathbf{I})$.

We follow Song et al. (2021)’s definition to choose a Variance-Preserving (VP)-SDE, or DDPM schedule to show the discrete update rule. Researchers use a neural network $s_\theta(x_t, t)$ to approximate the score function $\nabla_{x_t} \log p_t(x_t)$ and makes it possible to use the reverse process to generate the data. We consider the general condition by discretizing the whole process into T bins,

$$x_{t-1} = x_t - f(x_t, t) + g^2(t) s_\theta(x_t, t) + g(t) \epsilon_t \quad (3)$$

where $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$.

Ho et al. (2020) consider the marginal distribution of x_t given x_0 is Gaussian:

$$q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) \mathbf{I}),$$

where the noise schedule is defined via $\beta_t = g(t) = -2f(t)$, $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. This leads to the update rule of the DDPM process:

$$x_{t-1} = \mu(x_t, t) + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad (4)$$

where $\mu(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} s_\theta(x_t, t) \right)$, and $\sigma_t = \sqrt{\beta_t}$ is the variance parameter governing the stochasticity of the reverse process.

The other important tool is Tweedie’s formula in Kadkhodaie & Simoncelli (2021), which can be used to estimate the original signal x_0 from a noisy observation x_t during the denoising process. In practice, DDPMs approximate this expectation using the trained score network, yielding:

$$\tilde{x}_{0|t}(x_t, t) = \mathbb{E}[x_0 | x_t] \approx \frac{1}{\sqrt{\bar{\alpha}_t}} (x_t - \sqrt{1 - \bar{\alpha}_t} s_\theta(x_t, t)). \quad (5)$$

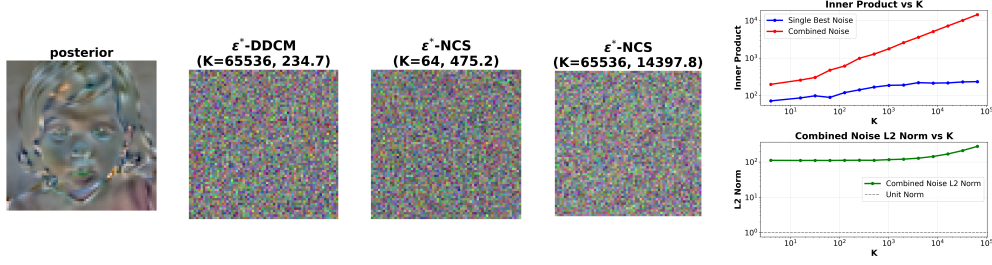


Figure 3: The comparison of the selected noise according to the original image. We choose a downsampled 64x64 image as the target image.

2.2 LINEAR INVERSE PROBLEMS AND CONDITIONAL GENERATION

Conditional generation addresses scenarios where only partial observations or measurements $\mathbf{y} \in \mathbb{R}^n$, derived from the original signal $\mathbf{x}_0 \in \mathbb{R}^d$, are available. The corresponding *inverse problem* is typically formulated as:

$$\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \mathbf{n}, \quad \mathbf{x}_0 \in \mathbb{R}^d, \mathbf{y}, \mathbf{n} \in \mathbb{R}^n, \quad (6)$$

where $\mathcal{A} : \mathbb{R}^d \rightarrow \mathbb{R}^n$ is a known linear degradation operator, and $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ denotes additive white Gaussian noise. This formulation encompasses a wide range of tasks, including inpainting, super-resolution, and deblurring.

Solving the conditional generation problem with a pretrained diffusion model requires replacing the score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ in equation 2 with the conditional score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{y})$.

Under the Bayesian framework, the conditional distribution at t is given by $p(\mathbf{x}_t | \mathbf{y}) = p(\mathbf{y} | \mathbf{x}_t)p(\mathbf{x}_t)/p(\mathbf{y})$, which indicates that $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$. Substituting the $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \mathbf{y})$ into the reverse SDE in equation 2 introduces an additional term that is not learned by the score network. The modified discrete update becomes:

$$\mathbf{x}_{t-1} = \mathbf{x}_t - f(\mathbf{x}_t, t) + g^2(t) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{y}) + g(t) \epsilon_t \quad (7)$$

$$= \mathbf{x}_t - f(\mathbf{x}_t, t) + g^2(t) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + g(t) \epsilon_t + \textcolor{red}{g^2(t) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y} | \mathbf{x}_t)}. \quad (8)$$

The red term in equation 7 captures the influence of the observation $\mathbf{y}$ on the sampling trajectory. Since the exact posterior $\nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t)$ is generally intractable, it typically requires an additional training process, such as classifier guidance or classifier-free guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2022) to be learned. The central challenge in diffusion-based inverse problem solving lies in accurately approximating $\nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t)$. Most methods, including explicit approximations and variational inference techniques Daras et al. (2024), directly modify the sampling trajectory during the denoising process, which can disrupt the consistency of the denoising process. In contrast, sampling-based methods (Dou & Song, 2024; Cardoso et al.; Wu et al., 2023) avoid altering the generative path by selecting candidate samples that satisfy the observation constraint through importance sampling or rejection. However, these approaches often suffer from weak guidance influence and high computational cost.

3 NOISE COMBINATION SAMPLING (NCS)

In the natural diffusion process, for all $t \in [0, T]$, $\mathbf{x}_t$ remains on its corresponding manifold, which is also the manifold on which the neural network $s_\theta(\mathbf{x}_t, t)$ was trained. As illustrated in Figure 1, existing inverse problem solvers typically incorporate the measurement score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t)$ as an external term, adding it directly during the denoising steps. This intervention disrupts the generative consistency of the diffusion process. In contrast, NCS does not alter the assumption or the method for approximating $\nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t)$. Instead, it implicitly embeds the conditional information into the noise component of the DDPM update, preserving the trajectory on the learned manifold. The modified update rule is given by:

$$\mathbf{x}_{t-1} \approx \mathbf{x}_t - f(\mathbf{x}_t, t) + g^2(t) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \textcolor{brown}{g(t) \epsilon_t^*}, \quad (9)$$

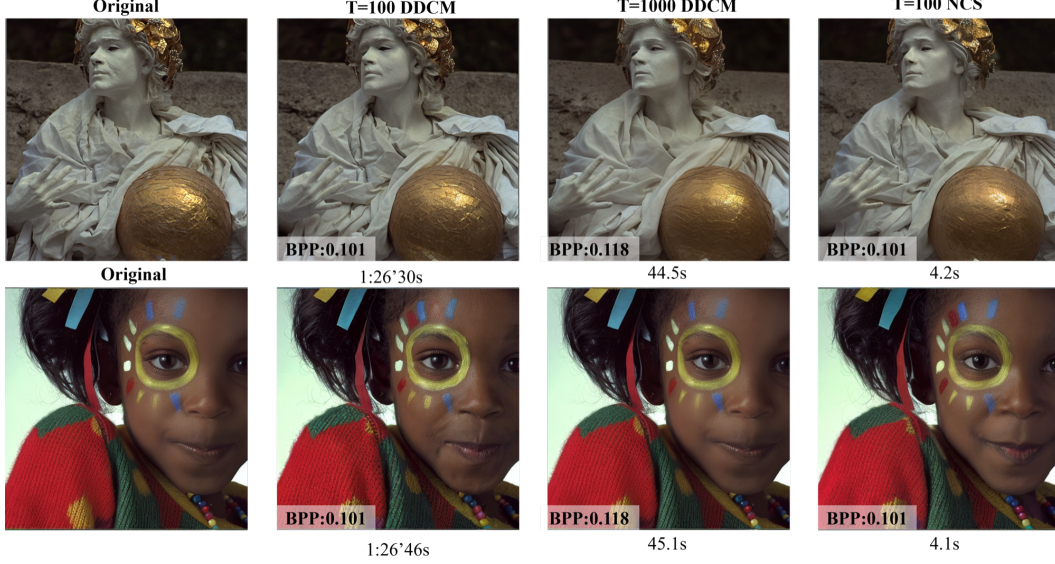


Figure 4: Comparison of the compression efficiency of NCS-MPGD and DDCM. For $T = 1000$, we choose $K = 32768$, $m = 12$, $C = 8$. For $T = 100$, we choose $K = 32768$, $m = 2$, $C = 0$. Our proposal get equivalent performance in fewer time.

where ϵ_t^* is a **constructed noise** vector that approximates the effect of the conditional term. NCS seeks to find an optimal noise vector ϵ_t^* within the span of a finite set of noise vectors, referred to as the noise codebook, and synthesizes it as a linear combination of these basis vectors to align with the target conditional direction. We formalize this process below.

Theorem 1 (Noise Combination Sampling). *For linear inverse problems, the optimal noise vector ϵ_t^* that best aligns with the conditional score direction is given by:*

$$\epsilon_t^* = \sum_{i=1}^K \gamma_i \epsilon_i, \quad (10)$$

where $\{\epsilon_i\}_{i=1}^K$ are standard Gaussian vectors from a fixed noise codebook, and $\gamma = (\gamma_1, \dots, \gamma_K)$ denotes the combination weights. The optimal weights are obtained by solving the following constrained optimization problem:

$$\gamma^* = \underset{\gamma \in \mathbb{R}^K, \|\gamma\|_2=1}{\operatorname{argmax}} \left\langle \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t), \sum_{i=1}^K \gamma_i \epsilon_i \right\rangle. \quad (11)$$

The synthesized noise vector ϵ_t^* remains standard normal, i.e., $\epsilon_t^* \sim \mathcal{N}(0, \mathbf{I})$, as a consequence of the unit-norm constraint $\|\gamma\|_2 = 1$. A complete proof is provided in Appendix A.

Under the NCS framework, any approximated conditional score, or intervention, can be embedded into the noise term via an optimal linear combination. In the following, we demonstrate how several representative inverse problem solvers can be reformulated as special instances of NCS.

Definition 1 (NCS-DPS). *Chung et al. (2023) approximates the conditional score using the gradient of a likelihood loss defined between the observation $\mathbf{y}$ and the estimated signal $\tilde{\mathbf{x}}_{0|t}$. Specifically,*

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \log \mathcal{N}(\mathbf{y}; \mathbf{A} \tilde{\mathbf{x}}_{0|t}, \sigma_t^2 \mathbf{I}) \\ &= \frac{1}{\sigma_t^2} \left(\frac{\partial \tilde{\mathbf{x}}_{0|t}}{\partial \mathbf{x}_t} \right)^\top \mathbf{A}^\top (\mathbf{y} - \mathbf{A} \tilde{\mathbf{x}}_{0|t}). \end{aligned} \quad (12)$$

Its NCS counterpart is obtained by aligning the synthesized noise vector with this gradient direction:

$$\gamma^* = \underset{\gamma \in \mathbb{R}^K, \|\gamma\|_2=1}{\operatorname{argmax}} \left\langle \left(\frac{\partial \tilde{\mathbf{x}}_{0|t}}{\partial \mathbf{x}_t} \right)^\top \mathbf{A}^\top (\mathbf{y} - \mathbf{A} \tilde{\mathbf{x}}_{0|t}), \sum_{i=1}^K \gamma_i \epsilon_i \right\rangle. \quad (13)$$

Algorithm 1 Noise Combination Sampling for Linear Inverse Problems

Require: Codebooks $\mathcal{C}_t = \{\epsilon_t^1, \dots, \epsilon_t^K\}$ for all t , represented as matrices $\mathbf{E}_t$; observation $\mathbf{y}$; approximate conditional score direction $\mathbf{c}$

Ensure: Reconstructed sample $\mathbf{x}_0$

1: Sample initial latent $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$

2: **for** $t = T$ **to** 1 **do**

3: $\tilde{\mathbf{x}}_{0|t} \leftarrow \text{Tweedie}(\mathbf{x}_t, t)$ (Eq. 5)

4: $\mathbf{c} \leftarrow \nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t)$

5: $\gamma^* \leftarrow \mathbf{c}^\top \mathbf{E}_t / \|\mathbf{c}^\top \mathbf{E}_t\|_2$ (Eq. 20)

6: $\epsilon_t^* \leftarrow \sum_{i=1}^K \gamma_i^* \epsilon_t^i$

7: $\mathbf{x}_{t-1} \leftarrow \mu_\theta(\mathbf{x}_t, t) + \sigma_t \epsilon_t^*$ (Eq. 4)

8: **end for**

9: **return** $\mathbf{x}_0$

Definition 2 (NCS-MPGD). *He et al. (2024) proposes performing updates directly on the estimated signal $\tilde{\mathbf{x}}_{0|t}$ rather than the latent variable $\mathbf{x}_t$. By isolating the additional term introduced in its update rule, we obtain the following approximation of the measurement score:*

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t) &= -\lambda_t \sqrt{\bar{\alpha}_t} \nabla_{\tilde{\mathbf{x}}_{0|t}} \|\mathbf{y} - \mathbf{A} \tilde{\mathbf{x}}_{0|t}\|_2^2 \\ &= 2\lambda_t \sqrt{\bar{\alpha}_t} \mathbf{A}^\top (\mathbf{y} - \mathbf{A} \tilde{\mathbf{x}}_{0|t}), \end{aligned} \quad (14)$$

where λ_t is a time-dependent step size. The corresponding NCS formulation is given by:

$$\gamma^* = \underset{\gamma \in \mathbb{R}^K, \|\gamma\|_2=1}{\operatorname{argmax}} \left\langle \mathbf{A}^\top (\mathbf{y} - \mathbf{A} \tilde{\mathbf{x}}_{0|t}), \sum_{i=1}^K \gamma_i \epsilon_i \right\rangle. \quad (15)$$

Notably, if $\tilde{\mathbf{x}}_{0|t}$ is replaced with $\mathbf{x}_t$, the formulation reduces to Score-Based Annealed Langevin Dynamics (ALD) Jalal et al. (2021). For a comprehensive analysis of the connections between these methods, refer to the survey in (Daras et al., 2024).

Definition 3 (DDCM under the NCS Framework). *DDCM formulates compression as a special case of inverse problems, where $\mathbf{A} = \mathbf{I}$, $\mathbf{n} = 0$, and $\mathbf{y} = \mathbf{x}_0$. The conditional score is approximated based on the difference between the ground-truth signal and the estimated reconstruction:*

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t) \approx \frac{\sqrt{\bar{\alpha}_t}}{1 - \bar{\alpha}_t} (\mathbf{x}_0 - \tilde{\mathbf{x}}_{0|t}). \quad (16)$$

DDCM selects one single noise vector from a predefined codebook via one-hot maximization:

$$\epsilon_t^* = \underset{i \in \{1, \dots, K\}}{\operatorname{argmax}} \langle \mathbf{x}_0 - \tilde{\mathbf{x}}_{0|t}, \epsilon_t^i \rangle, \quad (17)$$

After the reconstruction, DDCM stores these index and can reconstruct the image with index and the generative model. Ohayon et al. (2025) extend this to general linear inverse problems by:

$$\epsilon_t^* = \underset{i \in \{1, \dots, K\}}{\operatorname{argmax}} \langle \mathbf{y} - \mathbf{A} \mathbf{x}_t, \mathbf{A} \epsilon_t^i \rangle. \quad (18)$$

Within the NCS framework, we replace $\mathbf{x}_t$ with the estimated $\tilde{\mathbf{x}}_{0|t}$ and extend the one-hot selection to a full linear combination. This yields the following optimization problem:

$$\gamma^* = \underset{\gamma \in \mathbb{R}^K, \|\gamma\|_2=1}{\operatorname{argmax}} \left\langle \mathbf{y} - \mathbf{A} \tilde{\mathbf{x}}_{0|t}, \mathbf{A} \sum_{i=1}^K \gamma_i \epsilon_i \right\rangle. \quad (19)$$

This is equivalent to NCS-MPGD, as both achieve their optimum when γ^* aligns with the same direction. A formal proof is provided in Theorem 2 and Appendix C. While Ohayon et al. (2025) propose selecting the top- m noise vectors from a codebook of size K , this can be viewed as a special top- m case of NCS-MPGD with restricted support in γ .

In the following, we describe how to compute the optimal combination weights γ^* that define the synthesized noise ϵ_t^* under the NCS formulation.

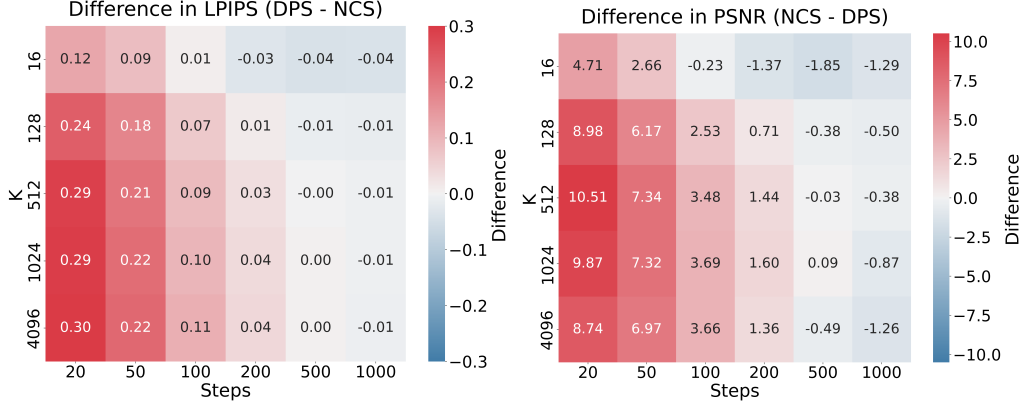


Figure 5: Influence of iteration and codebook size. We conduct the experiment on the inpainting task ($\sigma = 0$). The heatmap shows the improvement of the NCS-DPS compared to the DPS methods.

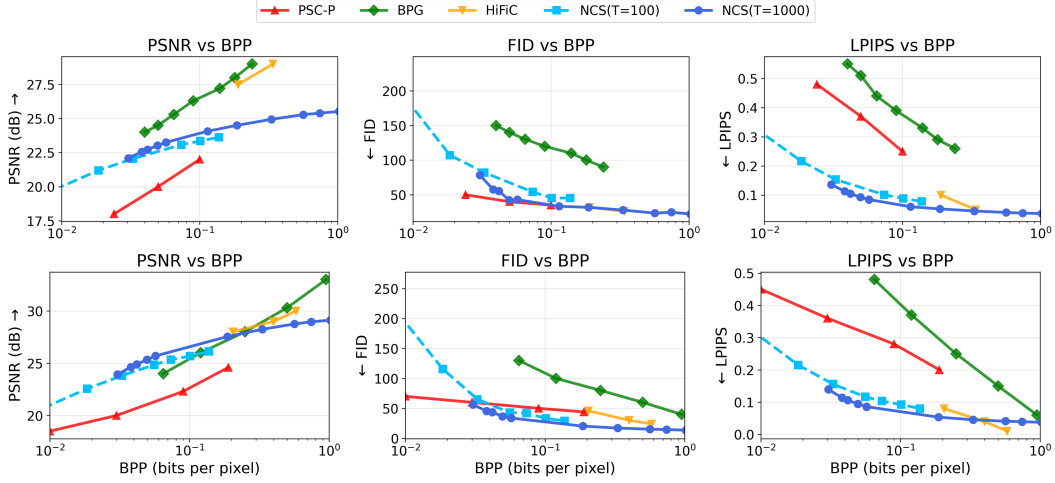


Figure 6: Comparison of compression methods. NCS achieves comparable reconstruction quality while reducing the number of compression steps from 1000 to 100, demonstrating significant efficiency with minimal quality loss.

Theorem 2 (Optimal Noise Combination). *Let $\mathbf{c} = \nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t)$ denote the approximated measurement score, and let $\mathbf{E}_t = [\epsilon_t^1, \dots, \epsilon_t^K] \in \mathbb{R}^{n \times K}$ be the matrix formed by stacking K standard Gaussian noise vectors. The optimal weight vector $\gamma^* \in \mathbb{R}^K$ that maximizes the inner product $\langle \mathbf{c}, \mathbf{E}_t \gamma \rangle$ subject to $\|\gamma\|_2 = 1$ is given by:*

$$\gamma^* = \frac{\mathbf{c}^\top \mathbf{E}_t}{\|\mathbf{c}^\top \mathbf{E}_t\|_2}. \quad (20)$$

By using Cauchy-Schwarz inequality, we can easily obtain the optimal weight vector, the detailed proof is provided in Appendix B. The optimal noise combination can be directly computed based on inner products. This result also extends naturally to the top- m case in DDCM-style problems. It significantly improves the efficiency of noise utilization: As illustrated in Fig. 2.2: combining a few noise vectors can achieve an inner product magnitude comparable to that obtained by DDCM's selection from a large codebook noise vectors.

As shown in Fig. 2.2, the NCS method approximates the measurement score by projecting it onto an ellipsoidal subspace spanned by K noise vectors. As K increases, the expressiveness of the noise combination improves, while the independence among the noise vectors diminishes. The norm of the

synthesized noise grows on the order of $\sqrt{d}$, where d denotes the data dimensionality. This behavior suggests that the synthesized noise may increasingly deviate from the standard noise distribution. In Fig. 5, the influence of K on NCS performance appears quite robust. The NCS method performs well across a broad range of K values, rendering it both practical and easy to deploy.

3.1 COMPRESSION TASKS

DDCM noticed that employing multiple noise vectors can enhance compression quality. However, it requires an efficient strategy for searching and storing the indices and weights per noise. Given the definition of bits per pixel (BPP) as $\text{BPP} = (T - 1)(\log_2(K)m + C(m - 1))/n^2$, where C represents 2^C quantization bins of vectors weights. A greedy search strategy is used to iteratively select the optimal combination of noise vectors across all quantization bins (see Ohayon et al. (2025), Appendix B.5). However, it incurs an overall complexity that grows *exponentially* with C and m , making it computationally prohibitive for large-scale scenarios.

To mitigate this limitation, we propose a method that directly approximates the quantized values using the closed-form solution from Theorem 2, significantly reducing the computational burden relative to the cost of noise inner product evaluations. Our formulation enables increasing both m and C for higher compression fidelity while simultaneously reducing T . As shown in Figures 4 and 6, decreasing T by a factor of 10, while increasing both m and the number of quantization bins, preserves the compression ratio, minimizes reconstruction quality degradation, and achieves a $10\times$ speedup in compression time. Additional implementation details are provided in Appendix D.

4 EXPERIMENTS

Table 1: Quantitative comparison of baseline solvers and their NCS variants on FFHQ dataset (Inpainting and SR $4\times$). Each cell shows PSNR / FID / LPIPS.

Task	Method	PSNR($\uparrow$) / FID($\downarrow$) / LPIPS($\downarrow$)		
		20	100	1000
Inpainting (Box)	DPS	12.52 / 133.9 / 0.497	18.67 / 95.65 / 0.286	22.71 / 57.11 / 0.139
	NCS-DPS	19.16 / 116.0 / 0.323	22.31 / 71.38 / 0.170	23.41 / 39.82 / 0.088
	MPGD	16.84 / 107.3 / 0.220	17.26 / 107.6 / 0.164	13.51 / 241.7 / 0.387
	NCS-MPGD	19.00 / 99.31 / 0.277	20.53 / 62.50 / 0.153	20.96 / 44.42 / 0.101
	DAPS	22.01 / 53.52 / 0.209	22.56 / 46.49 / 0.197	24.20 / 39.79 / 0.168
	NCS-DAPS	22.33 / 49.71 / 0.205	22.47 / 45.01 / 0.195	24.24 / 39.98 / 0.168
Inpainting (Random)	DPS	13.13 / 132.8 / 0.472	19.26 / 96.09 / 0.278	27.35 / 58.85 / 0.126
	NCS-DPS	21.20 / 87.55 / 0.297	27.31 / 42.34 / 0.137	31.57 / 14.28 / 0.042
	MPGD	21.80 / 76.86 / 0.172	25.45 / 45.73 / 0.100	25.05 / 69.68 / 0.214
	NCS-MPGD	20.25 / 102.7 / 0.290	25.21 / 61.82 / 0.126	28.71 / 31.91 / 0.049
	DAPS	14.08 / 259.2 / 0.625	16.31 / 226.4 / 0.543	25.33 / 67.69 / 0.238
	NCS-DAPS	16.83 / 202.8 / 0.557	19.50 / 141.9 / 0.456	25.73 / 63.56 / 0.230
SR $4\times$	DPS	12.87 / 121.9 / 0.480	16.70 / 98.36 / 0.338	23.53 / 69.54 / 0.171
	NCS-DPS	21.07 / 111.5 / 0.290	26.33 / 62.51 / 0.133	26.59 / 31.57 / 0.084
	MPGD	19.35 / 90.92 / 0.246	22.59 / 63.68 / 0.148	20.46 / 84.88 / 0.490
	NCS-MPGD	22.83 / 92.41 / 0.231	25.82 / 49.53 / 0.115	25.85 / 35.83 / 0.161
	DAPS	25.52 / 89.95 / 0.329	26.31 / 79.90 / 0.329	28.22 / 52.80 / 0.200
	NCS-DAPS	25.48 / 91.50 / 0.328	26.22 / 82.45 / 0.328	28.15 / 57.00 / 0.207

4.1 EXPERIMENTAL SETUP

All experiments were conducted on an NVIDIA RTX 4090 GPU. In Fig. 2, we choose pretrained models from Ho et al. (2020), and for experiments on FFHQ (Karras et al., 2019) and ImageNet (Deng et al., 2009) in Table 1 and 2, we choose models from Dhariwal & Nichol (2021). For compression experiments, we choose stable diffusion models 2.0 in Rombach et al. (2022).

4.2 INVERSE PROBLEM SOLVING

We evaluated our method on several challenging inverse problems: image inpainting, super-resolution, gaussian deblurring and motion deblurring. For inpainting, we tested both center box and random mask settings. For super-resolution, we examined 4× and 8×. Except for the experiments in Fig. 2, we choose DPS, MPGD, and DAPS Zhang et al. (2025) as the baseline solvers, inverse problems setting are the same as Chung et al. (2023), all tasks choose $\sigma = 0.05$. Detailed setting is shown in Appendix E. We use **black bold font** to highlight the better result between its baseline and NCS counterpart under the same setting. We use **blue bold font** to highlight the better result among all methods under the same setting. For other experiments, please refer to Appendix F.

4.3 COMPRESSION EXPERIMENTS

We conducted experiments to evaluate the compression performance of our NCS-based approach. We compared our method against several state-of-the-art compression algorithms including PSC-P, BPG, and HiFiC (Albalawi et al., 2015; Elata et al., 2024; Muckley et al., 2023; Mentzer et al., 2020) on Kodak24 dataset (Franzen, 1999) and ImageNet dataset, using the same experimental setup as in DDCM Ohayon et al. (2025). The results demonstrate that our method maintains high fidelity while achieving significant storage savings compared to baseline approaches.

Table 2: Quantitative comparison of baseline solvers and their NCS variants on four inverse problems on ImageNet. Each cell shows PSNR / FID / LPIPS.

Task	Method	PSNR(↑) / FID(↓) / LPIPS(↓)		
		20	100	1000
Inpainting (Box)	DPS	11.67 / 272.9 / 0.682	15.34 / 234.6 / 0.550	18.52 / 134.8 / 0.308
	NCS-DPS	15.77 / 279.2 / 0.623	18.19 / 159.2 / 0.362	19.38 / 88.65 / 0.190
	MPGD	13.94 / 201.0 / 0.364	14.94 / 205.9 / 0.336	14.01 / 218.5 / 0.421
	NCS-MPGD	16.12 / 226.9 / 0.466	16.36 / 152.3 / 0.270	16.57 / 126.1 / 0.226
	DAPS	18.55 / 159.9 / 0.269	26.31 / 140.5 / 0.329	20.83 / 115.8 / 0.231
	NCS-DAPS	17.78 / 152.5 / 0.268	26.22 / 154.8 / 0.328	20.49 / 114.2 / 0.233
Inpainting (Random)	DPS	12.88 / 271.9 / 0.671	16.86 / 231.9 / 0.545	23.96 / 101.2 / 0.278
	NCS-DPS	17.66 / 262.1 / 0.596	23.70 / 110.1 / 0.300	28.69 / 37.77 / 0.097
	MPGD	17.15 / 121.8 / 0.285	19.32 / 170.9 / 0.435	16.38 / 263.9 / 0.770
	NCS-MPGD	17.47 / 238.2 / 0.495	22.05 / 102.0 / 0.222	24.02 / 58.80 / 0.178
	DAPS	25.59 / 60.41 / 0.252	26.31 / 44.64 / 0.329	28.79 / 23.83 / 0.146
	NCS-DAPS	25.97 / 55.28 / 0.240	25.75 / 42.57 / 0.318	28.64 / 25.44 / 0.149
SR 4×	DPS	12.12 / 255.6 / 0.688	14.92 / 248.3 / 0.587	20.40 / 144.6 / 0.365
	NCS-DPS	17.60 / 260.9 / 0.595	22.62 / 112.4 / 0.328	23.78 / 44.12 / 0.195
	MPGD	16.41 / 191.0 / 0.420	18.60 / 96.65 / 0.327	10.43 / 292.1 / 1.109
	NCS-MPGD	19.83 / 165.5 / 0.417	22.35 / 76.57 / 0.223	16.94 / 92.43 / 0.704
	DAPS	23.25 / 135.9 / 0.368	23.02 / 257.4 / 0.358	25.21 / 105.0 / 0.301
	NCS-DAPS	23.19 / 123.9 / 0.369	22.82 / 257.9 / 0.364	25.05 / 108.2 / 0.302

5 CONCLUSION

In this work, we introduced NCS, a principled framework that approximates the measurement score in diffusion models through an optimal linear combination of noise vectors. We derived a closed-form solution via the Cauchy–Schwarz inequality and showed that NCS unifies and generalizes existing inverse problem solvers, achieving strong performance with fewer diffusion steps and improved stability. With the NCS framework and its closed-form solution, we also accelerate DDCM while preserving similar reconstruction quality. As future work, we will explore applying NCS to compression, where its closed-form structure may inspire more efficient quantization schemes.

REFERENCES

- Umar Albalawi, Saraju P. Mohanty, and Elias Kougianos. A hardware architecture for better portable graphics (bpg) compression encoder. In *2015 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)*, pp. 291–296, 2015. doi: 10.1109/iNIS.2015.12.
- Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.
- Gabriel Cardoso, Sylvain Le Corff, Eric Moulines, et al. Monte carlo guided denoising diffusion models for bayesian linear inverse problems. In *The Twelfth International Conference on Learning Representations*.
- Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=OnD9zGAGT0k>.
- Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.
- Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2024.
- Noam Elata, Tomer Michaeli, and Michael Elad. Zero-shot image compression with diffusion-based posterior sampling. *arXiv e-prints*, pp. arXiv–2407, 2024.
- R. Franzen. Kodak lossless true color image suite. <http://r0k.us/graphics/kodak/>, 1999. Accessed: 2025-06-23.
- Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.
- Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=o3BxOLoxml>.
- Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL <https://arxiv.org/abs/2006.11239>.
- Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.
- Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. Robust compressed sensing mri with deep generative priors. *Advances in Neural Information Processing Systems*, 34:14938–14954, 2021.
- Zahra Kadkhodaie and Eero Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Information Processing Systems*, 34:13242–13254, 2021.

- Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.
- Jeongsol Kim, Geon Yeong Park, Hyungjin Chung, and Jong Chul Ye. Regularization by texts for latent diffusion inverse solvers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=TtUh0TolGX>.
- Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2206–2217, 2023.
- Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.
- Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in neural information processing systems*, 33:11913–11924, 2020.
- Matthew J Muckley, Alaaeldin El-Nouby, Karen Ullrich, Hervé Jégou, and Jakob Verbeek. Improving statistical fidelity for neural image compression with implicit local likelihood models. In *International Conference on Machine Learning*, pp. 25426–25443. PMLR, 2023.
- Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025. URL <https://arxiv.org/abs/2502.09992>.
- Guy Ohayon, Hila Manor, Tomer Michaeli, and Michael Elad. Compressed image generation with denoising diffusion codebook models. *arXiv preprint arXiv:2502.01189*, 2025.
- Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=di52zR8xgf>.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL <https://arxiv.org/abs/2112.10752>.
- Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8839–8849, 2024.
- Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. *arXiv preprint arXiv:2307.08123*, 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*, 2021.
- Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco. Diffuse attend and segment: Unsupervised zero-shot segmentation using stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3554–3563, 2024.
- Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*, 2023.
- Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

- Lingxiao Yang, Shutong Ding, Yifan Cai, Jingyi Yu, Jingya Wang, and Ye Shi. Guidance with spherical gaussian constraint for conditional diffusion. In *International Conference on Machine Learning*, pp. 56071–56095. PMLR, 2024.
- Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Y Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models. *Advances in Neural Information Processing Systems*, 37:22370–22417, 2024.
- Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23174–23184, 2023.
- Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20895–20905, 2025.

A PROOF OF GAUSSIANTY AND OPTIMAL WEIGHTS

We first show that, under a unit-norm constraint on the combination weights, the synthesized noise remains standard normal, and then derive the closed-form optimizer for equation 11.

Lemma 1 (Gaussianity of unit-norm combinations). *Let $\{\epsilon_i\}_{i=1}^K$ be mutually independent with $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. For any $\gamma = (\gamma_1, \dots, \gamma_K) \in \mathbb{R}^K$ with $\|\gamma\|_2 = 1$ that is deterministic (or independent of $\{\epsilon_i\}_{i=1}^K$), the linear combination*

$$\epsilon_t^* = \sum_{i=1}^K \gamma_i \epsilon_i$$

satisfies $\epsilon_t^ \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.*

Proof. By linearity of expectation, $\mathbb{E}[\epsilon_t^*] = \sum_{i=1}^K \gamma_i \mathbb{E}[\epsilon_i] = \mathbf{0}$. For the covariance, independence and isotropy give

$$\text{Cov}(\epsilon_t^*) = \sum_{i=1}^K \sum_{j=1}^K \gamma_i \gamma_j \mathbb{E}[\epsilon_i \epsilon_j^\top] = \sum_{i=1}^K \gamma_i^2 \mathbf{I} = \|\gamma\|_2^2 \mathbf{I} = \mathbf{I}.$$

Since $(\epsilon_1, \dots, \epsilon_K)$ is jointly Gaussian and ϵ_t^* is a linear transformation of it, ϵ_t^* is Gaussian with mean $\mathbf{0}$ and covariance $\mathbf{I}$; hence $\epsilon_t^* \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. $\square$

Conclusion. Combining Lemma 1 with Lemma 2, we obtain that the synthesized noise is standard normal whenever the synthesis codebook is independent of the weight computation, i.e., $\epsilon_t^* \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

In specific optimization scenarios, as the number of noise sources increases linearly, the inner product also increases nearly linearly according to $\log(K)$. This approach achieves significantly higher efficiency than selecting a single noise source within K (in DDCM). Furthermore, the magnitude of the optimal noise remains constant over a considerable range (approximately equal to the average magnitude of the noise sources). Specific variations can be observed in Fig. 2.2.

B PROOF OF THEOREM 2

Proof. Define $\mathbf{v} = \mathbf{c}^\top \mathbf{E}_t \in \mathbb{R}^K$. Our goal is to solve:

$$\gamma^* = \underset{\|\gamma\|_2=1}{\text{argmax}} \langle \mathbf{v}, \gamma \rangle.$$

By the Cauchy–Schwarz inequality,

$$\langle \mathbf{v}, \gamma \rangle \leq \|\mathbf{v}\|_2 \|\gamma\|_2 = \|\mathbf{v}\|_2,$$

with equality if and only if γ is aligned with $\mathbf{v}$, i.e., $\gamma = \lambda \mathbf{v}$ for some scalar λ . Enforcing the constraint $\|\gamma\|_2 = 1$ gives $|\lambda| = 1/\|\mathbf{v}\|_2$, and the maximum is attained at:

$$\gamma^* = \frac{\mathbf{v}}{\|\mathbf{v}\|_2} = \frac{\mathbf{c}^\top \mathbf{E}_t}{\|\mathbf{c}^\top \mathbf{E}_t\|_2}.$$

$\square$

C PROOF OF EQUIVALENCE BETWEEN NCS OPTIMIZATION AND NCS-MPGD

Theorem 3 (Equivalence of NCS Formulations). *For NCS optimization problem 11, we propose the following formulation: in which we replaced the $\mathbf{x}_0$ in Ohayon et al. (2025) with the more accurate reconstruction $\tilde{\mathbf{x}}_{0|t}$.*

$$\gamma^* = \underset{\gamma \in \mathbb{R}^K, \|\gamma\|_2=1}{\text{argmax}} \left\langle \mathbf{y} - \mathbf{A} \tilde{\mathbf{x}}_{0|t}, \mathbf{A} \sum_{i=1}^K \gamma_i \epsilon_i \right\rangle, \quad (21)$$

It is equivalent to the NCS-MPGD formulation:

$$\gamma^* = \operatorname{argmax}_{\gamma \in \mathbb{R}^K, \|\gamma\|_2=1} \left\langle -\mathbf{A}^\top (\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}_{0|t}), \sum_{i=1}^K \gamma_i \boldsymbol{\epsilon}_i \right\rangle. \quad (22)$$

That is, both optimization problems have the same optimal solution γ^* .

Proof. We prove this equivalence by showing that the two objective functions are identical for any feasible γ .

Let $\mathbf{c} = \mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}_{0|t}$ and $\boldsymbol{\varepsilon} = \sum_{i=1}^K \gamma_i \boldsymbol{\epsilon}_i$. The standard NCS objective can be written as:

$$\langle \mathbf{c}, \mathbf{A}\boldsymbol{\varepsilon} \rangle = \mathbf{c}^\top (\mathbf{A}\boldsymbol{\varepsilon}) \quad (23)$$

$$= (\mathbf{c}^\top \mathbf{A}) \boldsymbol{\varepsilon} \quad (24)$$

$$= \boldsymbol{\varepsilon}^\top (\mathbf{c}^\top \mathbf{A})^\top \quad (25)$$

$$= \boldsymbol{\varepsilon}^\top \mathbf{A}^\top \mathbf{c} \quad (26)$$

$$= \langle \mathbf{A}^\top \mathbf{c}, \boldsymbol{\varepsilon} \rangle. \quad (27)$$

The NCS-MPGD objective is:

$$\langle -\mathbf{A}^\top \mathbf{c}, \boldsymbol{\varepsilon} \rangle = \left\langle -\mathbf{A}^\top \mathbf{c}, \sum_{i=1}^K \gamma_i \boldsymbol{\epsilon}_i \right\rangle \quad (28)$$

$$= \sum_{i=1}^K \gamma_i \langle -\mathbf{A}^\top \mathbf{c}, \boldsymbol{\epsilon}_i \rangle. \quad (29)$$

Since the two inner products are identical for every $\boldsymbol{\varepsilon}$ (and thus for every γ), and the feasible set $\{\gamma \in \mathbb{R}^K \mid \|\gamma\|_2 = 1\}$ is the same for both problems, their maxima and argmax sets are identical.

Furthermore, since each objective is a linear functional of γ , the maximum over the unit sphere occurs when γ is aligned with the functional's direction. For the standard NCS formulation, this gives:

$$\gamma^* = \frac{(\mathbf{A}^\top \mathbf{c})^\top \mathbf{E}_t}{\|(\mathbf{A}^\top \mathbf{c})^\top \mathbf{E}_t\|_2}, \quad (30)$$

where $\mathbf{E}_t = [\boldsymbol{\epsilon}_1, \dots, \boldsymbol{\epsilon}_K]$ is the matrix of noise vectors.

For the NCS-MPGD formulation, the optimal solution is:

$$\gamma^* = \frac{(-\mathbf{A}^\top \mathbf{c})^\top \mathbf{E}_t}{\|(-\mathbf{A}^\top \mathbf{c})^\top \mathbf{E}_t\|_2} = \frac{-(-\mathbf{A}^\top \mathbf{c})^\top \mathbf{E}_t}{\|(\mathbf{A}^\top \mathbf{c})^\top \mathbf{E}_t\|_2}. \quad (31)$$

The negative sign in the NCS-MPGD formulation is due to the maximization of the negative inner product, which is equivalent to minimizing the positive inner product. However, since we are maximizing the absolute value of the alignment, both formulations yield the same optimal direction (up to a sign, which is normalized out by the unit norm constraint).

Therefore, the two optimization problems are not merely equivalent—they are literally the same problem written in two different notations, and they achieve their maximum when the optimal γ^* is the same (up to normalization). $\square$

D QUANTIZATION BY NCS CLOSED-FORM SOLUTION

To utilize extra noise vectors to approximate the measurement score, Ohayon et al. (2025) proposes to use greedy search to find the noise vector that maximize the inner product between the measurement score and the noise vector, and to search the optimal quantization parameters on the selected noise vector and the next one. This process is computationally expensive. It will cost hours to search the optimal quantization parameters. If $C = 10$ and $m = 10$ $K = 1024$, it will cost around 10 hours to search the optimal quantization parameters.

Problem. Let $\{\epsilon_i\}_{i=1}^N$ be given noise vectors and let c be the target. Define $b_i := \langle \epsilon_i, c \rangle$ and (optionally) align signs so $b_i \geq 0$ by replacing $\epsilon_i \leftarrow \text{sgn}(b_i)\epsilon_i$. We form a mixture

$$m(\gamma) = \sum_{i=1}^m \gamma_i \epsilon_{(i)}, \quad \gamma = (\gamma_1, \dots, \gamma_m)^\top, \quad \sum_{i=1}^m \gamma_i^2 = 1,$$

where (i) indexes an ordered subset of m atoms (e.g., the Top- m by $|b_i|$). The objective is to maximize alignment, i.e.

$$\max_{\gamma} \langle m(\gamma), c \rangle = \max_{\gamma} \sum_{i=1}^m \gamma_i b_{(i)} \quad \text{s.t.} \quad \sum_{i=1}^m \gamma_i^2 = 1.$$

Closed-form (continuous) solution. Without quantization, the optimal coefficients on a fixed support are

$$\gamma^* = \frac{b_S}{\|b_S\|_2}, \quad b_S = (b_{(1)}, \dots, b_{(m)})^\top,$$

by Cauchy–Schwarz. If the support S is free, it is optimal to take the m indices with largest $|b_i|$.

Definition (quantization via ℓ_2 stick-breaking). We quantize by parameterizing γ through a stick-breaking map using

$$u_i \in \mathcal{Q} \subset (0, 1], \quad i = 1, \dots, 2^C - 1,$$

where $\mathcal{Q}$ is a finite grid with $2^{C-1} - 1$ elements (e.g., if $m = 3$, then if selected (not 0), $\mathcal{Q} = \{0.33, 0.66, 1\}$). The coefficients are then

$$\gamma_1 = \sqrt{u_1}, \tag{32}$$

$$\gamma_i = \left(\prod_{j=1}^{i-1} \sqrt{1 - u_j} \right) \sqrt{u_i}, \quad i = 2, \dots, m-1, \tag{33}$$

$$\gamma_m = \prod_{j=1}^{m-1} \sqrt{1 - u_j}. \tag{34}$$

By construction $\sum_{i=1}^m \gamma_i^2 = 1$ for any choices of $\{u_i\}$, so no final normalization is required. The inverse map (from any feasible γ with $\sum \gamma_i^2 = 1$) is

$$u_1 = \gamma_1^2, \tag{35}$$

$$u_i = \frac{\gamma_i^2}{1 - \sum_{t=1}^{i-1} \gamma_t^2}, \quad i = 2, \dots, m-1. \tag{36}$$

Using the closed-form to obtain a quantized solution. We obtain a quantized solution directly from γ^* in two simple steps:

1. **Project γ^* into stick-breaking space.** Compute $\{u_i^*\}_{i=1}^{m-1}$ from γ^* using the inverse map above. (When $b_{(1)} \geq b_{(2)} \geq \dots$, $\gamma_i^* \propto b_{(i)}$ is non-increasing, which matches the stick-breaking order.)
2. **Quantize and reconstruct.** Independently quantize each stage by nearest-neighbor projection onto the grid,

$$\hat{u}_i = \arg \min_{u \in \mathcal{Q}} |u - u_i^*|, \quad i = 1, \dots, m-1,$$

then form $\hat{\gamma}$ from $\{\hat{u}_i\}$ via the forward stick-breaking map.

This yields $\hat{\gamma}$ in $O(m)$ time and preserves $\sum_i \hat{\gamma}_i^2 = 1$ by construction.

Remark (stage-wise closed form and exact discrete refinement). If one optimizes stage-wise in the *continuous* domain, the optimal fraction at stage i has the closed form

$$u_i^* = \frac{b_{(i)}^2}{b_{(i)}^2 + v_{i+1}^2}, \quad v_m = b_{(m)}, \quad v_i = b_{(i)}\sqrt{u_i^*} + v_{i+1}\sqrt{1 - u_i^*}.$$

A discretized variant replaces u_i^* by the nearest grid point in $\mathcal{Q}$ at each stage (still $O(m)$). For the *exact* discrete optimum on $\mathcal{Q}$ one can use a 1D dynamic program:

$$v_m = b_{(m)}, \quad v_i = \max_{u \in \mathcal{Q}} \left\{ b_{(i)}\sqrt{u} + v_{i+1}\sqrt{1 - u} \right\},$$

which selects $\hat{u}_i \in \mathcal{Q}$ per stage and remains $O(m|\mathcal{Q}|)$.

E INVERSE PROBLEMS SETTING

We choose the same setting of inverse problems with Chung et al. (2023). Detailed setting is shown in Table 3 and 4.

As for the NCS method, because the strength of the measurement would influence the dimension of the noise space that is available for us to approximate, we set the K for different tasks as follows:

- Super-resolution $\times 4$ (ImageNet/FFHQ): $K = 512$
- Super-resolution $\times 8$ (ImageNet/FFHQ): $K = 64$
- Inpainting box (ImageNet/FFHQ): $K = 64$
- Inpainting random (ImageNet/FFHQ): $K = 512$
- Gaussian deblurring (ImageNet/FFHQ): $K = 256$
- Motion deblurring (ImageNet/FFHQ): $K = 512$
- Phase retrieval (ImageNet/FFHQ): $K = 512$

F ADDITIONAL EXPERIMENTS OUTCOMES

We record experiments on Super-resolution $\times 8$, deblurring, gaussian blurring, in Table. 3 and 4

G COMPRESSION BY NCS-DPS

Since we have unified various methods under the NCS framework, this implies that—aside from DDCM, which is a special case equivalent to NCS-MPGD, NCS-DPS can also be employed for compression tasks. To evaluate this, we conducted experiments on several images from the Kodak24 dataset, using the Stable Diffusion 2 model as the pre-trained backbone (Rombach et al., 2022). We adopted a codebook of size 1024 but restricted the selection to a single optimal noise vector to maintain consistency with the experimental settings of the DDCM paper.

As shown in Fig. 11, the compression efficiency of NCS-DPS is inferior to that of DDCM. The compressed images appear overly smooth, exhibiting a significant loss of fine details. This degradation may stem from inaccuracies in the gradient used during the reverse computation. Nonetheless, when using fewer denoising steps, the images compressed by NCS-DPS appear to retain richer semantic structures.

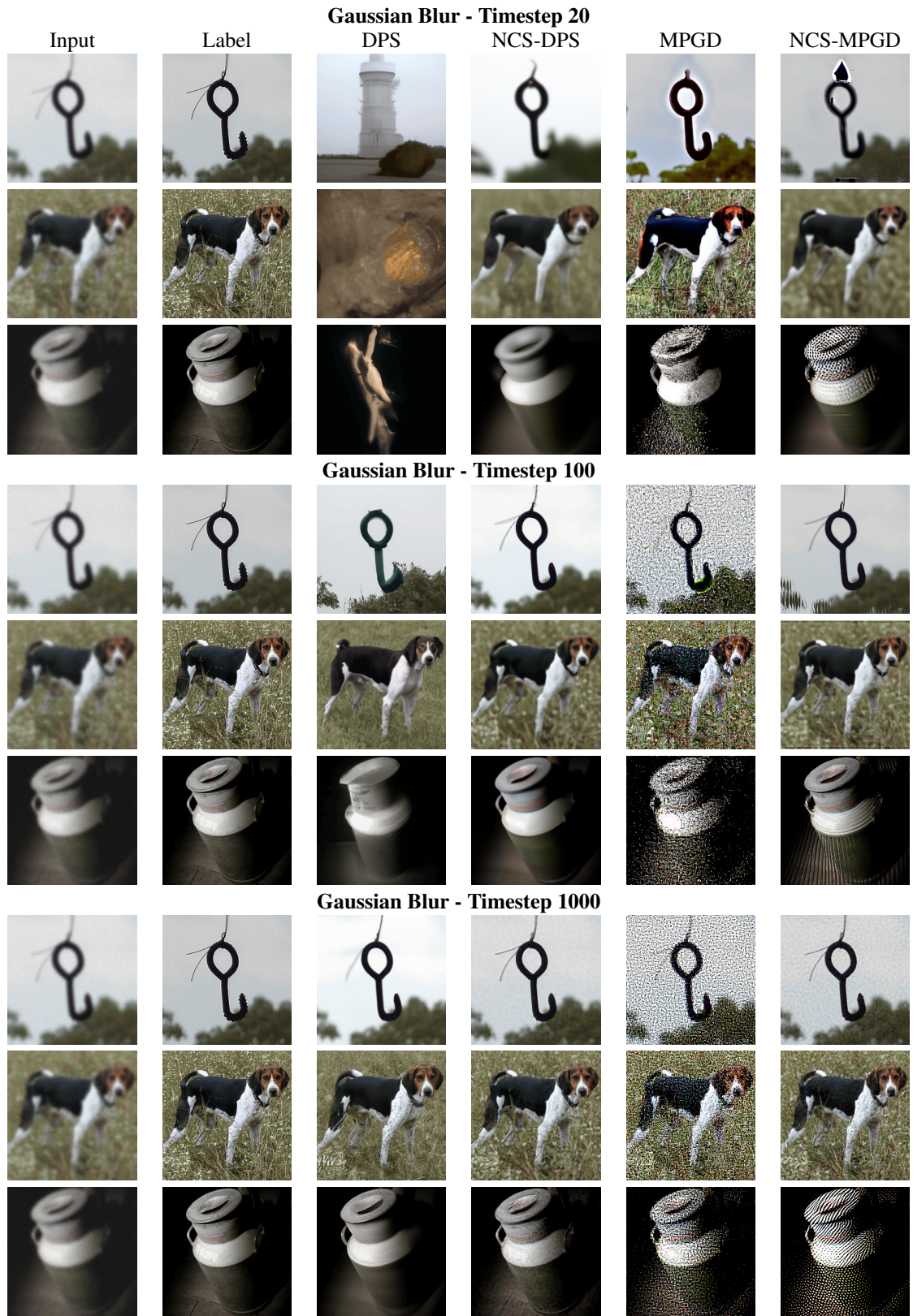


Figure 7: Visual comparison on Gaussian Blur task. Each row shows results for one image across different methods. The three sections correspond to timesteps 20, 100, and 1000 respectively.

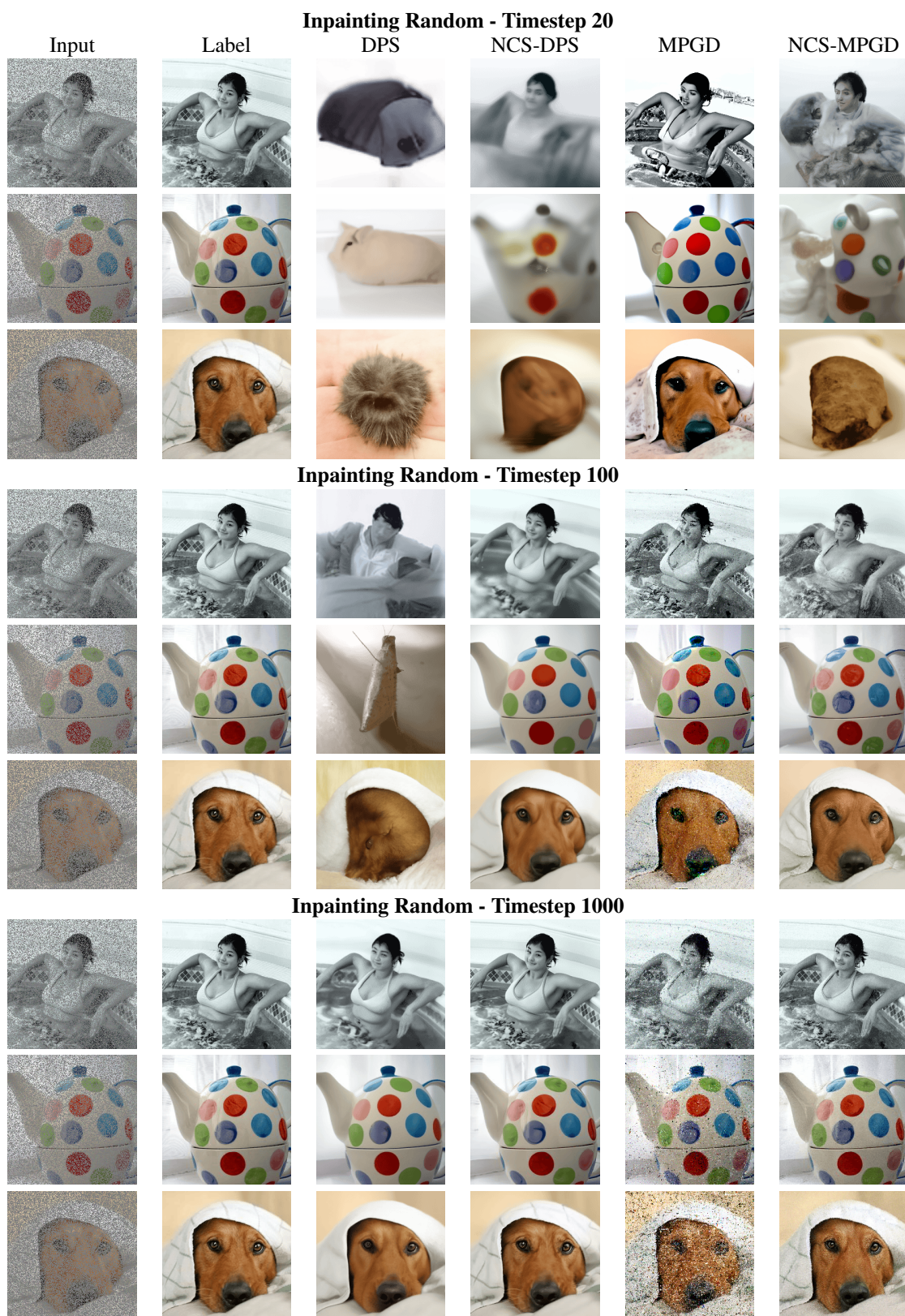


Figure 8: Visual comparison on Inpainting Random task. Each row shows results for one image across different methods. The three sections correspond to timesteps 20, 100, and 1000 respectively.

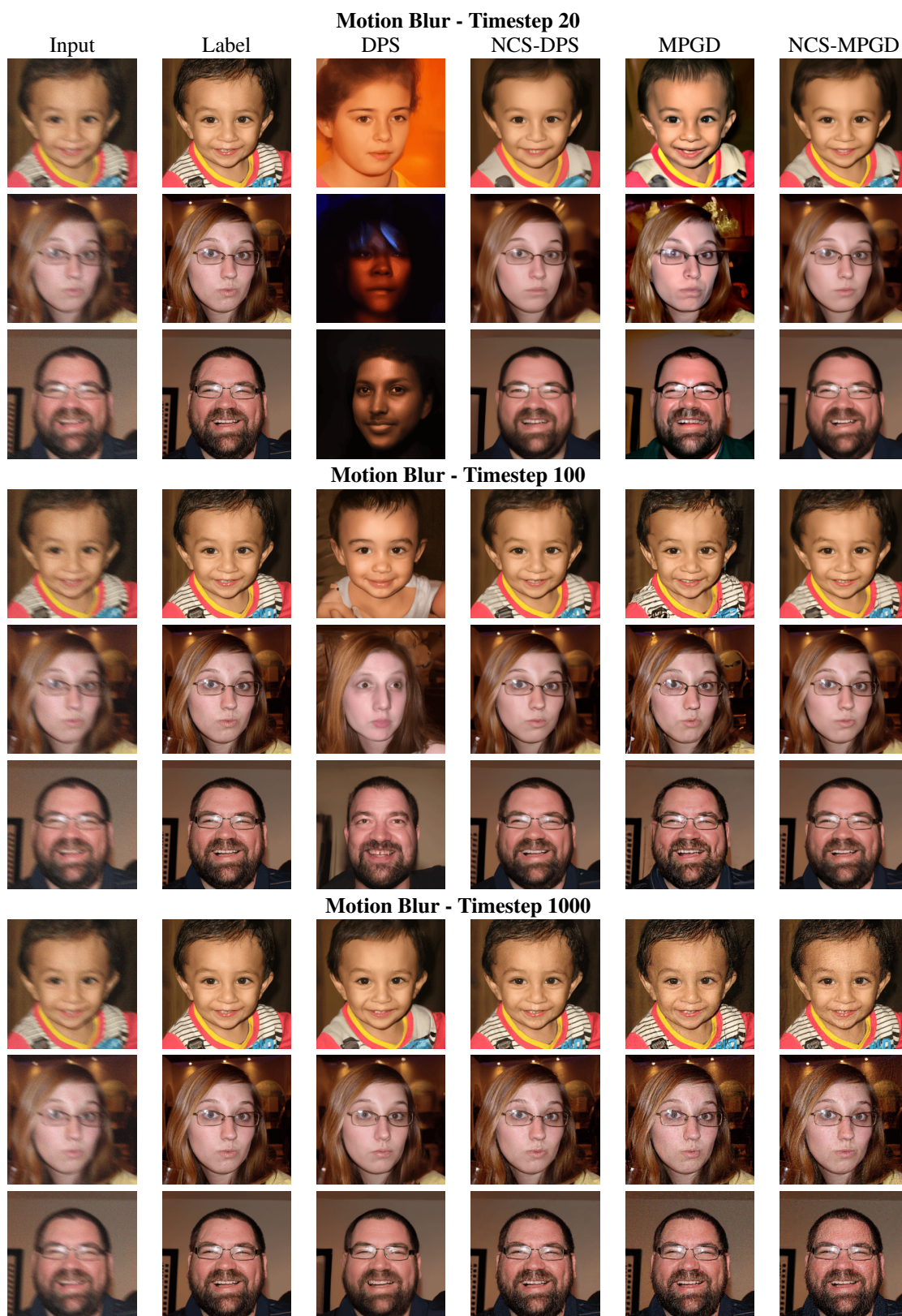


Figure 9: Visual comparison on Motion Blur task. Each row shows results for one image across different methods. The three sections correspond to timesteps 20, 100, and 1000 respectively.



Figure 10: Visual comparison on Super Resolution x4 task. Each row shows results for one image across different methods. The three sections correspond to timesteps 20, 100, and 1000 respectively.

Table 3: Quantitative comparison of baseline solvers and their NCS variants on the FFHQ dataset (Inpainting and SR 4 $\times$). Each cell shows PSNR / FID / LPIPS. Bold indicates the better result between a baseline and its NCS counterpart under the same setting. Values are formatted for visual alignment.

Task	Method	PSNR($\uparrow$) / FID($\downarrow$) / LPIPS($\downarrow$)		
		20	100	1000
SR 8 $\times$	DPS	11.84/118.3/0.506	15.05/102.1/0.387	20.87/76.90/ 0.220
	NCS-DPS	20.83/109.4/0.290	24.09/67.34/0.154	21.97/49.35/0.266
	MPGD	17.60/93.26/0.303	20.08/77.28/0.213	18.08/87.93/0.560
	NCS-MPGD	22.10/92.13/0.246	23.29/65.00/0.170	21.67/60.28/0.352
	DAPS	23.66/202.8/0.387	24.02/270.5/0.337	25.20/188.4/ 0.278
	NCS-DAPS	23.74/199.9/0.382	24.14/273.6/0.334	25.23/188.6/0.279
Gaussian Deblur	DPS	12.12/139.4/0.495	20.02/88.92/0.253	24.81/60.57/0.130
	NCS-DPS	21.59/115.1/0.311	26.68/52.14/0.115	27.02/49.90/0.083
	MPGD	21.36/78.46/0.206	23.85/64.22/0.133	24.80/74.67/0.130
	NCS-MPGD	25.42/74.20/0.184	26.58/48.04/0.110	26.36/86.73/0.166
	DAPS	25.80/85.06/0.319	26.63/71.26/0.267	28.38/50.10/0.187
	NCS-DAPS	25.80/ 83.67/0.318	26.57/72.99/0.268	28.34/51.49/0.189
Motion Deblur	DPS	12.21/140.3/0.493	20.19/88.61/0.251	25.87/56.65/0.118
	NCS-DPS	22.05/114.9/0.303	28.28/47.25/0.097	29.50/28.41/0.044
	MPGD	21.86/76.79/0.191	24.88/53.05/0.107	25.54/54.78/0.142
	NCS-MPGD	26.22/64.51/0.153	27.76/39.31/0.078	28.17/41.91/0.083
	DAPS	26.25/66.91/0.289	27.77/57.02/0.232	30.52/34.34/0.139
	NCS-DAPS	26.19/ 65.73/0.290	27.68/ 54.91/0.233	30.51/ 33.59/0.139



Figure 11: Compression results on Kodak24 images. Each row corresponds to a different image. Columns: Original, DDCM (100 steps), DDCM (1000 steps), NCS-DPS (100 steps), NCS-DPS (1000 steps).

Table 4: Quantitative comparison of baseline solvers and their NCS variants on four inverse problems on ImageNet. Each cell shows PSNR / FID / LPIPS. Bold indicates the better result between a baseline and its NCS counterpart under the same setting. Values are formatted for visual alignment.

Task	Method	PSNR($\uparrow$) / FID($\downarrow$) / LPIPS($\downarrow$)		
		20	100	1000
SR 8 $\times$	DPS	11.16/257.5/0.702	13.78/246.6/0.616	18.45/179.9/ 0.415
	NCS-DPS	17.54/254.3/0.583	21.02/118.0/0.328	18.79/84.54/0.453
	MPGD	15.46/226.4/0.500	16.98/145.9/0.383	10.86/315.1/1.037
	NCS-MPGD	19.26/176.1/0.461	19.74/118.3/0.371	14.64/172.8/0.798
	DAPS	21.43/202.8/ 0.368	21.32/270.5/0.472	25.21/188.4/0.301
	NCS-DAPS	21.50/199.9/0.369	21.24/273.6/0.472	22.31/188.6/0.424
Gaussian Deblur	DPS	12.02/265.5/0.675	17.32/213.8/0.493	21.01/105.2/0.298
	NCS-DPS	13.18/247.3/0.656	22.29/87.41/0.298	23.87/72.03/0.215
	MPGD	16.53/196.2/0.413	14.72/261.9/ 0.253	10.69/317.4/1.079
	NCS-MPGD	22.06/114.6/0.376	22.73/93.45/0.258	16.97/206.9/0.720
	DAPS	23.42/135.2/0.370	26.62/120.6/0.267	24.89/102.6/0.275
	NCS-DAPS	23.38/ 127.5/0.371	26.57/123.3/0.268	24.85/106.5/0.281
Motion Deblur	DPS	12.05/265.5/0.676	17.39/218.4/0.502	22.75/87.47/0.268
	NCS-DPS	13.39/250.8/0.648	24.13/77.85/0.255	27.25/34.19/0.103
	MPGD	16.99/165.3/0.375	14.81/242.0/0.253	13.99/227.4/0.886
	NCS-MPGD	22.83/90.49/0.283	23.90/62.78/0.169	17.46/157.6/0.648
	DAPS	24.54/80.48/0.297	25.21/68.35/0.272	27.93/43.42/0.181
	NCS-DAPS	24.42/ 80.41/0.298	24.83/69.94/0.278	27.85/44.81/0.185