Supplementary Material of
On Exact Inversion of DPM-Solvers

Seongmin Hong[1], Kyeonghyun Lee[1], Suh Yoon Jeon[1], Hyewon Bae[1], Se Young Chun[1,2,*]
[1]Dept. of ECE, [2]INMC & IPAI, Seoul National University, Republic of Korea
{smhongok, litiphysics, euniejeon, hyewon0309, sychun}@snu.ac.kr
Project page: https://smhongok.github.io/inv-dpm.html

S1. Convergence analysis

S1.1. Forward step method (Alg. 1)

Proposition S1 Convergence guarantee of Alg.1. *Let $\omega \geq 1$, t_{i-1}, C, and $\varnothing$ be appropriate inputs for*

$$\boldsymbol{x}_{\boldsymbol{\theta}}(\cdot, t_{i-1}) = \omega \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\cdot, t_{i-1}, C) - (1-\omega)\bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\cdot, t_{i-1}, \varnothing). \tag{S1}$$

If $\bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\cdot, t_{i-1}, C)$ is monotone and L-Lipschitz with $L \leq \frac{1}{\omega}\frac{\sigma_{t_i}}{\sigma_{t_{i-1}}\alpha_{t_i}(e^{-h_i}-1)}$, then the forward step method converges for all step size ρ such that

$$0 < \rho < 2\frac{\sigma_{t_i}/\sigma_{t_{i-1}} - \alpha_{t_i}(e^{-h_i}-1)\omega L}{(\sigma_{t_i}/\sigma_{t_{i-1}} + \alpha_{t_i}(e^{-h_i}-1)(|\omega| + |1-\omega|)L)^2}. \tag{S2}$$

To prove Proposition S1, we use the following Lemma S1 from page 5 in [S4] and :

Lemma S1. *Let $a, b \in \mathbb{R}$. If $f, g : \mathcal{X} \to \mathcal{Y}$ be L_f, L_g-Lipschitz continuous, respectively, then $af + bg$ is $|a|L_f + |b|L_g$-Lipschitz continuous.*

Now we start the proof of Proposition S1.

Proof of Proposition S1. Define $A : \mathbb{R}^D \to \mathbb{R}^D$ where

$$A(\boldsymbol{z}) := \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}\boldsymbol{z} - \alpha_{t_i}(e^{-h_i}-1)\boldsymbol{z}_{\boldsymbol{\theta}}(\boldsymbol{z}, t_{i-1}) - \hat{\boldsymbol{z}}_{t_i} \tag{S3}$$

for all feasible $\boldsymbol{z} \in \mathbb{R}^D$, so that the forward step method be written as

$$\hat{\boldsymbol{z}}_{t_{i-1}} = \hat{\boldsymbol{z}}_{t_{i-1}} - \rho A(\hat{\boldsymbol{z}}_{t_{i-1}}). \tag{S4}$$

For all feasible $\boldsymbol{z}$ and $\boldsymbol{z}' \in \mathbb{R}^D$,

$$\begin{aligned} \|A(\boldsymbol{z}) - A(\boldsymbol{z}')\| &= \Bigg\| \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}(\boldsymbol{z} - \boldsymbol{z}') \\ &\quad - \alpha_{t_i}(e^{-h_i}-1)(\boldsymbol{z}_{\boldsymbol{\theta}}(\boldsymbol{z}, t_{i-1}) - \boldsymbol{z}_{\boldsymbol{\theta}}(\boldsymbol{z}', t_{i-1}))\Bigg\| \\ &\leq \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}\|\boldsymbol{z} - \boldsymbol{z}'\| \\ &\quad + \alpha_{t_i}(e^{-h_i}-1)\|\boldsymbol{z}_{\boldsymbol{\theta}}(\boldsymbol{z}, t_{i-1}) - \boldsymbol{z}_{\boldsymbol{\theta}}(\boldsymbol{z}', t_{i-1})\|. \end{aligned} \tag{S5}$$

holds by the triangle inequality. Using Lemma S1 and the assumption of Proposition S1, $\boldsymbol{z}_{\boldsymbol{\theta}}(\cdot, t_{i-1})$ is $(|\omega|+|1-\omega|)L$-Lipschitz continuous. Thus, we have

$$\begin{aligned} \|A(\boldsymbol{z}) - A(\boldsymbol{z}')\| &\leq \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}\|\boldsymbol{z} - \boldsymbol{z}'\| \\ &\quad + \alpha_{t_i}(e^{-h_i}-1)(|\omega| + |1-\omega|)L\|\boldsymbol{z} - \boldsymbol{z}'\| \\ &= \left(\frac{\sigma_{t_i}}{\sigma_{t_{i-1}}} + \alpha_{t_i}(e^{-h_i}-1)(|\omega| + |1-\omega|)L\right)\|\boldsymbol{z} - \boldsymbol{z}'\|. \end{aligned} \tag{S6}$$

Equation (S4) converges if A is β-cocoercive and $\rho \in (0, 2\beta)$ [page 39 in [S4]]. So we should check the cocoercivity of A as follows:

$$\begin{aligned} \langle A(\boldsymbol{z}) - A(\boldsymbol{z}'), \boldsymbol{z} - \boldsymbol{z}'\rangle &= \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2 \\ &\quad - \alpha_{t_i}(e^{-h_i}-1)\Big(\\ &\quad \omega\langle \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}, t_{i-1}, C) - \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}', t_{i-1}, C), \boldsymbol{z} - \boldsymbol{z}'\rangle \\ &\quad + (1-\omega)\langle \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}, t_{i-1}, \varnothing) - \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}', t_{i-1}, \varnothing), \boldsymbol{z} - \boldsymbol{z}'\rangle\Big). \end{aligned} \tag{S7}$$

Since $\bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\cdot, t_{i-1}, C)$ is monotone, $\langle \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}, t_{i-1}, C) - \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}', t_{i-1}, C), \boldsymbol{z} - \boldsymbol{z}'\rangle \geq 0$ and $\langle \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}, t_{i-1}, \varnothing) - \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}', t_{i-1}, \varnothing), \boldsymbol{z} - \boldsymbol{z}'\rangle \geq 0$ holds. Because $\omega \geq 1$,

$$\begin{aligned} \langle A(\boldsymbol{z}) - A(\boldsymbol{z}'), \boldsymbol{z} - \boldsymbol{z}'\rangle &\geq \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2 - \alpha_{t_i} \\ &\quad \cdot (e^{-h_i}-1)\omega\langle \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}, t_{i-1}, C) - \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}', t_{i-1}, C), \boldsymbol{z} - \boldsymbol{z}'\rangle. \end{aligned} \tag{S8}$$

[*] Corresponding author

By the Cauchy-Schwarz inequality,

$$\begin{aligned}&\langle \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}, t_{i-1}, C) - \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}', t_{i-1}, C), \boldsymbol{z} - \boldsymbol{z}' \rangle \\ &\leq \|\boldsymbol{z} - \boldsymbol{z}'\|_2 \|\bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}, t_{i-1}, C) - \bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{z}', t_{i-1}, C)\|_2 \\ &\leq L\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2,\end{aligned} \tag{S9}$$

where the second inequality holds because $\bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\cdot, t_{i-1}, C)$ is L-Lipschitz. Merging Eq. (S6), Eq. (S8) and Eq. (S9), we have

$$\begin{aligned}&\langle A(\boldsymbol{z}) - A(\boldsymbol{z}'), \boldsymbol{z} - \boldsymbol{z}' \rangle \\ &\geq \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2 - \alpha_{t_i}(e^{-h_i} - 1)\omega L\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2 \\ &= \left(\frac{\sigma_{t_i}}{\sigma_{t_{i-1}}} - \alpha_{t_i}(e^{-h_i} - 1)\omega L\right)\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2 \\ &\geq \frac{\left(\frac{\sigma_{t_i}}{\sigma_{t_{i-1}}} - \alpha_{t_i}(e^{-h_i} - 1)\omega L\right)\|A(\boldsymbol{z}) - A(\boldsymbol{z}')\|_2^2}{\left(\frac{\sigma_{t_i}}{\sigma_{t_{i-1}}} + \alpha_{t_i}(e^{-h_i} - 1)(|\omega| + |1-\omega|)L\right)^2},\end{aligned} \tag{S10}$$

which indicates A is $\frac{\sigma_{t_i}/\sigma_{t_{i-1}} - \alpha_{t_i}(e^{-h_i}-1)\omega L}{\left(\sigma_{t_i}/\sigma_{t_{i-1}} + \alpha_{t_i}(e^{-h_i}-1)(|\omega|+|1-\omega|)L\right)^2}$-cocoercive. □

S1.2. Notes on instability of fixed point iteration in large classifier-free guidance

In Section 4.1 of the main paper, we explained why the fixed point iteration (FPI)-based method [S2] loses stability in the context of large classifier-free guidance. Here, we formulate and prove this as Proposition S2. Proposition S2 suggests that FPI may not converge when the classifier-free guidance is large.

Proposition S2 Instability of FPI in large classifier-free guidance. *Let $\omega \geq 1$, t_{i-1}, C, and $\varnothing$ be appropriate inputs for*

$$\boldsymbol{x}_{\boldsymbol{\theta}}(\cdot, t_{i-1}) = \omega\bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\cdot, t_{i-1}, C) - (1-\omega)\bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\cdot, t_{i-1}, \varnothing). \tag{S11}$$

If $\bar{\boldsymbol{x}}_{\boldsymbol{\theta}}(\cdot, t_{i-1}, C)$ is Lipschitz continuous with the constant

$$\frac{1}{|\omega| + |1-\omega|} \cdot \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}\alpha_{t_i}(e^{-h_i} - 1)}, \tag{S12}$$

then

$$F(\cdot) := \frac{\sigma_{t_{i-1}}}{\sigma_{t_i}}\alpha_{t_i}(e^{-h_i} - 1)\boldsymbol{x}_{\boldsymbol{\theta}}(\cdot, t_{i-1}) + \frac{\sigma_{t_{i-1}}}{\sigma_{t_i}}\hat{\boldsymbol{x}}_{t_i} \tag{S13}$$

is nonexpansive (i.e., 1-Lipschitz continuous).

Proof of Proposition S2. By the assumption, Eq. (S11) and *Lemma S1*, $\boldsymbol{x}_{\boldsymbol{\theta}}(\cdot, t_{i-1})$ is $\frac{\sigma_{t_i}}{\sigma_{t_{i-1}}\alpha_{t_i}(e^{-h_i}-1)}$-Lipschitz continuous. By Eq. (S13) and *Lemma S1*, $F(\cdot)$ is nonexpansive. □

Proposition S1 also suggests our method has more generous Lipschitz condition for converge than [20]: $\frac{1}{\omega}\frac{\sigma_{t_i}}{\sigma_{t_{i-1}}\alpha_{t_i}(e^{-h_i}-1)} > \frac{1}{|\omega|+|1-\omega|}\frac{\sigma_{t_i}}{\sigma_{t_{i-1}}\alpha_{t_i}(e^{-h_i}-1)}$.

S2. Algorithm 2

Algorithm 2 Inversion of DPM-Solver++(2M)

```
Require: initial value $\boldsymbol{x}_0$, time steps $\{t_i\}_{i=0}^{M}$, data prediction
    model $\boldsymbol{z}_{\boldsymbol{\theta}}$, UPDATE, $\mathcal{D}^\dagger$ in ??.
Denote $h_i := \lambda_{t_i} - \lambda_{t_{i-1}}$ and $r_i := \frac{h_{i-1}}{h_i}$ for $i = 1, \dots, M$.
$\hat{\boldsymbol{z}}_{t_M} \leftarrow \mathcal{D}^\dagger(\boldsymbol{x}_0)$ if LDM else $\boldsymbol{x}_0$
for $i \leftarrow M$ to 2 do $\hat{\boldsymbol{y}}_{t_i} \leftarrow \hat{\boldsymbol{z}}_{t_i}$
    for $j \leftarrow 1$ to $2J$ do
        $\hat{\boldsymbol{y}}_{t_{i-j/J}} \leftarrow \frac{\sigma_{t_{i-j/J}}}{\sigma_{t_{i-(j-1)/J}}}(\hat{\boldsymbol{y}}_{t_{i-j/J}} + \alpha_{t_{i-j/J}}(e^{-h_{i-j/J}} - 1)\boldsymbol{z}_\theta(\hat{\boldsymbol{y}}_{t_{i-(j-1)/J}}, t_{i-j/J}))$
    end for
    $\hat{\boldsymbol{z}}_{t_{i-1}} \leftarrow \hat{\boldsymbol{y}}_{t_{i-1}}$
    repeat
        $\boldsymbol{d}_i' \leftarrow \boldsymbol{z}_\theta(\hat{\boldsymbol{z}}_{t_{i-1}}, t_{i-1}) + \frac{1}{2r_i}(\boldsymbol{z}_\theta(\hat{\boldsymbol{y}}_{t_{i-1}}, t_{i-1}) - \boldsymbol{z}_\theta(\hat{\boldsymbol{y}}_{t_{i-2}}, t_{i-2}))$
        $\boldsymbol{z}_{t_i}' \leftarrow \frac{\sigma_{t_i}}{\sigma_{t_{i-1}}}\hat{\boldsymbol{z}}_{t_{i-1}} - \alpha_{t_i}\left(e^{-h_i} - 1\right)\boldsymbol{d}_i'$
        UPDATE($\hat{\boldsymbol{z}}_{t_{i-1}}; \hat{\boldsymbol{z}}_{t_i}, \boldsymbol{z}_{t_i}'$ )
    until converged
end for
$\hat{\boldsymbol{z}}_{t_0} \leftarrow \frac{\sigma_{t_0}}{\sigma_{t_1}}\left(\hat{\boldsymbol{z}}_{t_1} + \alpha_{t_1}\left(e^{-h_1} - 1\right)\boldsymbol{z}_\theta(\hat{\boldsymbol{z}}_{t_1}, t_0)\right)$
repeat
    $\boldsymbol{z}_{t_1}' \leftarrow \frac{\sigma_{t_1}}{\sigma_{t_0}}\hat{\boldsymbol{z}}_{t_0} - \alpha_{t_1}\left(e^{-h_1} - 1\right)\boldsymbol{z}_\theta(\hat{\boldsymbol{z}}_{t_0}, t_0)$
    UPDATE($\hat{\boldsymbol{z}}_{t_0}; \hat{\boldsymbol{z}}_{t_1}, \boldsymbol{z}_{t_1}'$ )
until converged
return $\hat{\boldsymbol{z}}_{t_0}$
```

S3. Experimental details

S3.1. Reconstruction

S3.1.1 Pixel-space DPM

For pixel-space DPM, we used gradient descent without momentum, with l_2 loss and a learning rate of 0.1. To dynamically adjust the learning rate, when the minimum loss for the last 5 iterations did not improve, the learning rate was halved, but not below a minimum of 0.001. The number of iterations was a maximum of 500. The convergence criterion was $\|\boldsymbol{x}_{t_i}' - \hat{\boldsymbol{x}}_{t_i}\|_2 < 10^{-3}$. The original model operates in half float (*i.e.*, 16-bit), so it was used as is during gradient descent. We used $J = 100$ in Algorithm 2. We experimented with 100 images.

S3.1.2 Latent diffusion model (LDM)

For LDM, we generated images with classifier-free guidance of 3.0, and the same prompt was employed for the inversion process. We used the forward step method, with l_2 loss. The initial step size was set as 0.5 for Algorithm 1, and $10/t$ for Algorithm 2 except for the first step ($t = 0$) where it was 1. To dynamically adjust the step size, when the minimum loss for the last 20 iterations did not improve, the step size was halved. For numerical stability, we applied a 20-step warmup (*i.e.*, linearly increasing the step size from 0). We used $J = 10$ in Algorithm 2. The number of iterations was a maximum of 500. The convergence criterion was $\|\boldsymbol{z}'_{t_i} - \hat{\boldsymbol{z}}_{t_i}\|_2 < 10^{-3}$. Every operation was held in float (*i.e.*, 32-bit). We experimented with 100 images.

Decoder inversion We applied decoder inversion for every method for a fair comparison, including naïve DDIM inversion and [S2]. We used Adam with l_2 loss and a learning rate of 0.1. For improved convergence, we applied a cosine learning rate scheduler with 10-step warmup within a total of 100 iterations.

S3.2. Application: Tree-ring watermark

The work of tree-ring watermark in [S5] injects a watermark in the frequency domain of noise and then generates an image with noise obtained by inverse transformation. We tested on detection of watermarks by generating various types of watermarks with the same radius and calculated l_1 difference between the original watermark and the watermark obtained by inversion. The radius was set the same to make the comparison consistent and detection difficult. We tested on three watermarks with the shape of a tree-ring and a radius of 6 pixels. For watermark creation, we set a constant and designated pixel values to be random but close to constant to make the difference scale reliable.

S3.3. Application: Background-preserving editing

In this experiment, we employed the open-source code of [S3] to compare Algorithm 1 with three other methods: (i) The original code utilizing the latents stored during the generating process (*i.e.*, oracle), (ii) the naïve DDIM inversion, and (iii) the naïve DDIM inversion with decoder inversion. We used classifier-free guidance of 7.5 and for decoder inversion, we applied a cosine learning rate scheduler with 50-step warmup within a total of 500 iterations. All other experimental settings remained identical to those of LDM.

S4. Additional results

S4.1. Reconstruction

In Sec. 5.1 of the main paper, we performed the reconstruction of noise and image to evaluate the exact invertibility of the proposed methods. We provide more qualitative results in Fig. S1. In Fig. S1, we addtionally provide the FPI-based method of Pan et al. [S2], namely AIDI-E.

S4.2. Application: Tree-ring watermark

In Sec. 5.2 of the main paper, we demonstrated the improved detection of watermarks [S5] by employing our algorithm, even when the images were generated using high-order DPM-solvers. We provide an additional example of watermark detection / classification in Figs. Fig. S2 and Fig. S3. We experimented with different shaped watermarks and prompts from the main paper.

S4.3. Application: Background-preserving editing

In Sec. 5.3 of the main paper, we experimentally demonstrated our proposed methods enable the background-preserving editing, without the need for the original latents. In Figs. S4 and S5, we show additional results with different prompts.

References

[S1] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv:2211.01095*, 2022. 5

[S2] Zhihong Pan, Riccardo Gherardi, Xiufeng Xie, and Stephen Huang. Effective real image editing with accelerated iterative diffusion inversion. In *ICCV*, pages 15912–15921, 2023. 2, 3, 4

[S3] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. In *ICCV*, 2023. 3

[S4] Ernest K Ryu and Wotao Yin. *Large-scale convex optimization: algorithms & analyses via monotone operators*. Cambridge University Press, 2022. 1

[S5] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-Ring Watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv:2305.20030*, 2023. 3, 5

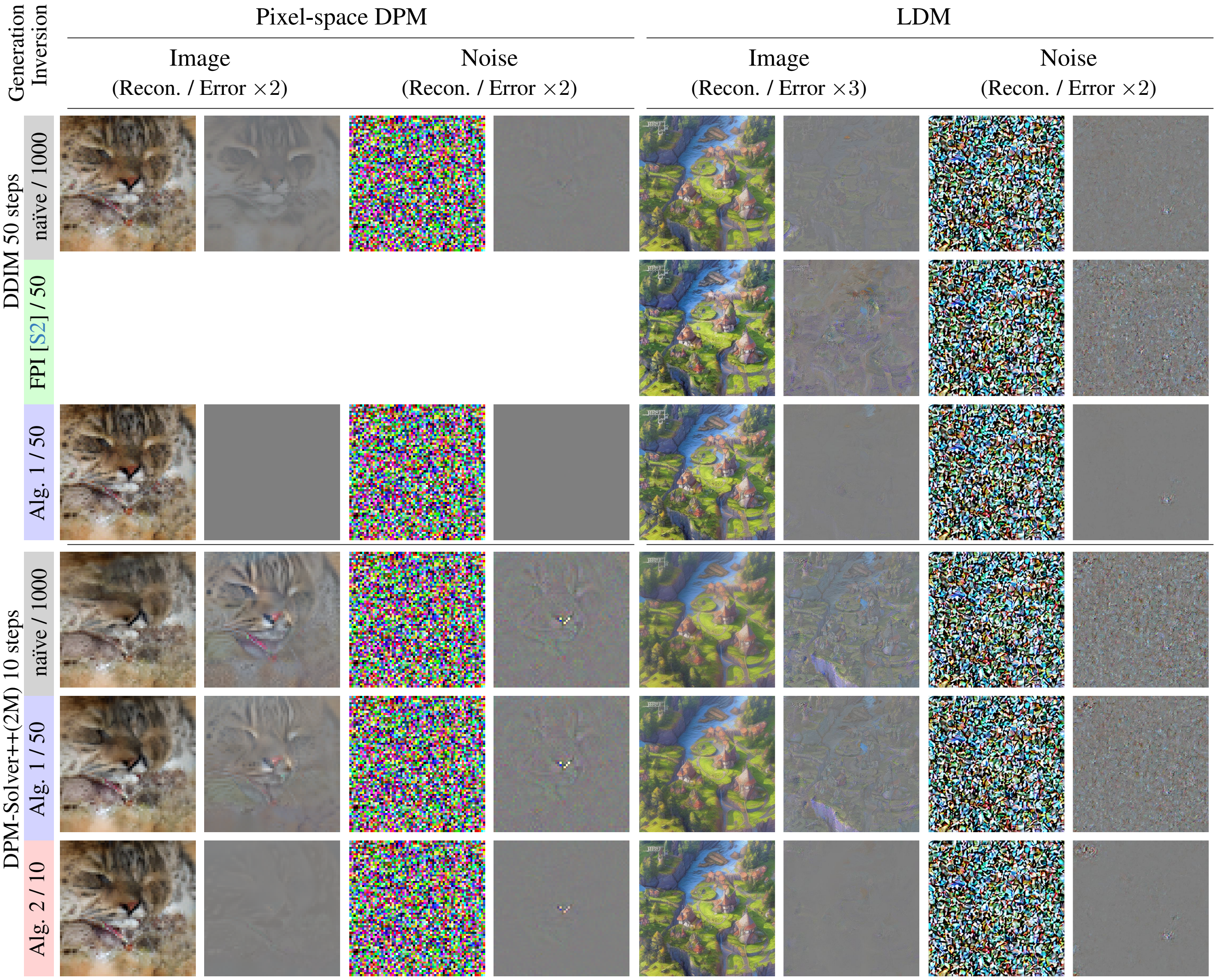


Figure S1. Our Algs. 1 and 2 significantly reduce reconstruction errors, whether it's for images or noise, DDIM or high-order DPM-solvers, or pixel-space DPM or LDM. The generation / inversion method varies for each row, *e.g.*, 'naïve / 1000' indicates that we performed the naïve DDIM inversion for 1000 steps. 'Alg. 1 / 50' and 'Alg. 2 / 10' attempt exact inversion with 50 steps of DDIM and 10 steps of DPM-Solver++(2M), respectively. Achieving exact inversion in LDM is challenging due to information loss from the autoencoder and instability caused by a classifier-free guidance of 3.0. Nonetheless, our algorithm produces good results (extremely low residual error in Alg. 1 for DDIM on the 3rd row and in Alg. 2 for DPM-solver++(2M) on the 6th row) also in LDM.

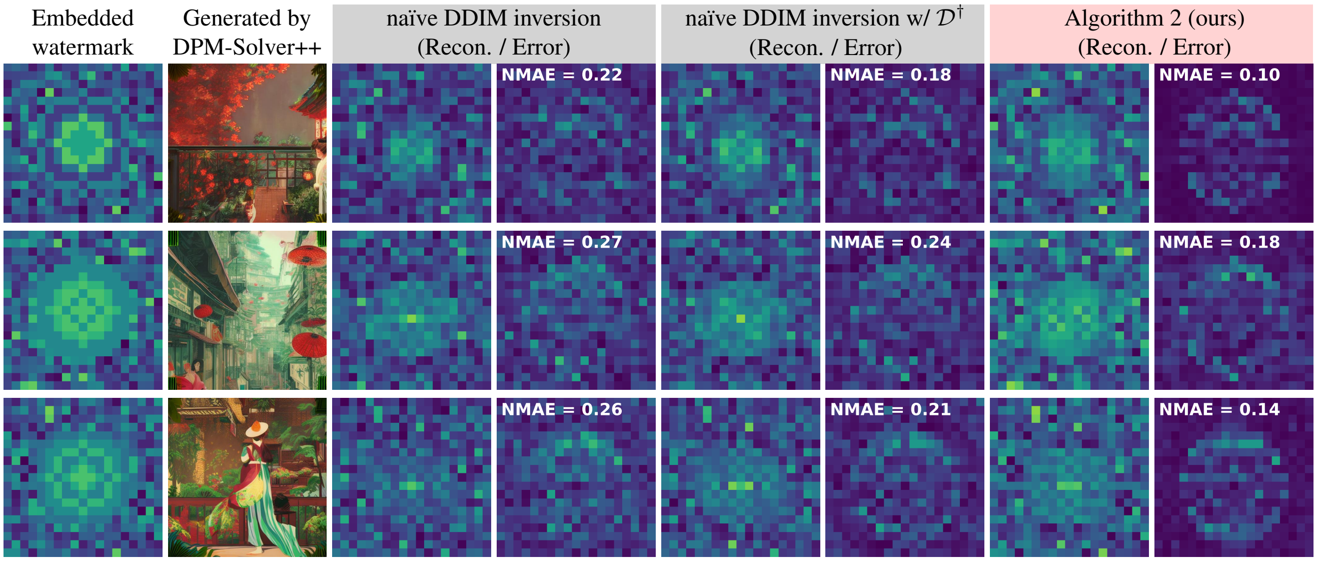
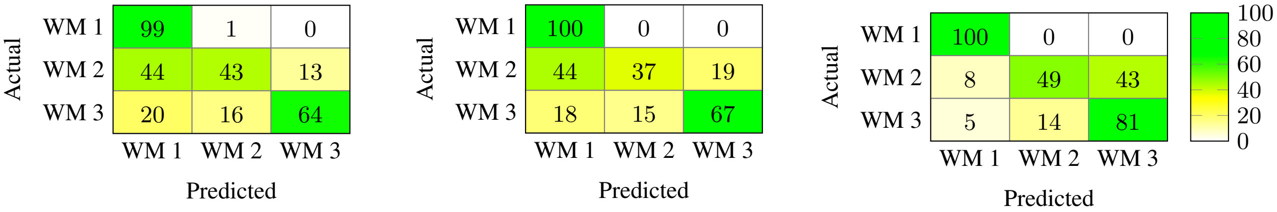

Embedded watermark
Generated by DPM-Solver++
naïve DDIM inversion (Recon. / Error)
naïve DDIM inversion w/ $\mathcal{D}^\dagger$ (Recon. / Error)
Algorithm 2 (ours) (Recon. / Error)
NMAE = 0.22
NMAE = 0.18
NMAE = 0.10
NMAE = 0.27
NMAE = 0.24
NMAE = 0.18
NMAE = 0.26
NMAE = 0.21
NMAE = 0.14

Figure S2. Our Algorithm 2 enables accurate reconstruction of Tree-ring watermarks [S5] in the Fourier space of the initial noise (z_T). The Tree-ring watermark is embedded in the Fourier space of the initial noise in the shape of tree-rings and can be utilized for copyright tracing (column 1). Then, the image is generated starting from the watermarked noise. The practical approach is to accelerate image generation using methods like DPM-Solver++(2M) [S1] (column 2). NMAEs are shown on each error map. Using Algorithm 2 (columns 7-8) for watermark reconstruction results in lower errors compared to employing naïve DDIM inversion (columns 3-6), achieving nearly 50% reduction in NMAE.



(a) Naïve DDIM inversion

(b) naïve DDIM inversion w/ $\mathcal{D}^\dagger$ in Sec. 4.1

(c) Algorithm 2 (ours)

Figure S3. Our algorithm's strong reconstruction performance allows for the classification of tree-ring watermarks as well. For copyright tracing, it is possible to generate images by embedding different unique watermarks. Three distinct watermarks (WM 1,2, and 3) are displayed in the first column of Fig. S2. In the confusion matrices, 'Predicted' corresponds to the watermark with the smallest l_1 difference among the three watermarks. In Figs. S3a and S3b, the naïve DDIM inversion encounters difficulties in detecting WM 2. In contrast (Fig. S3c), our Algorithm 2 performs well in detecting WM 2.

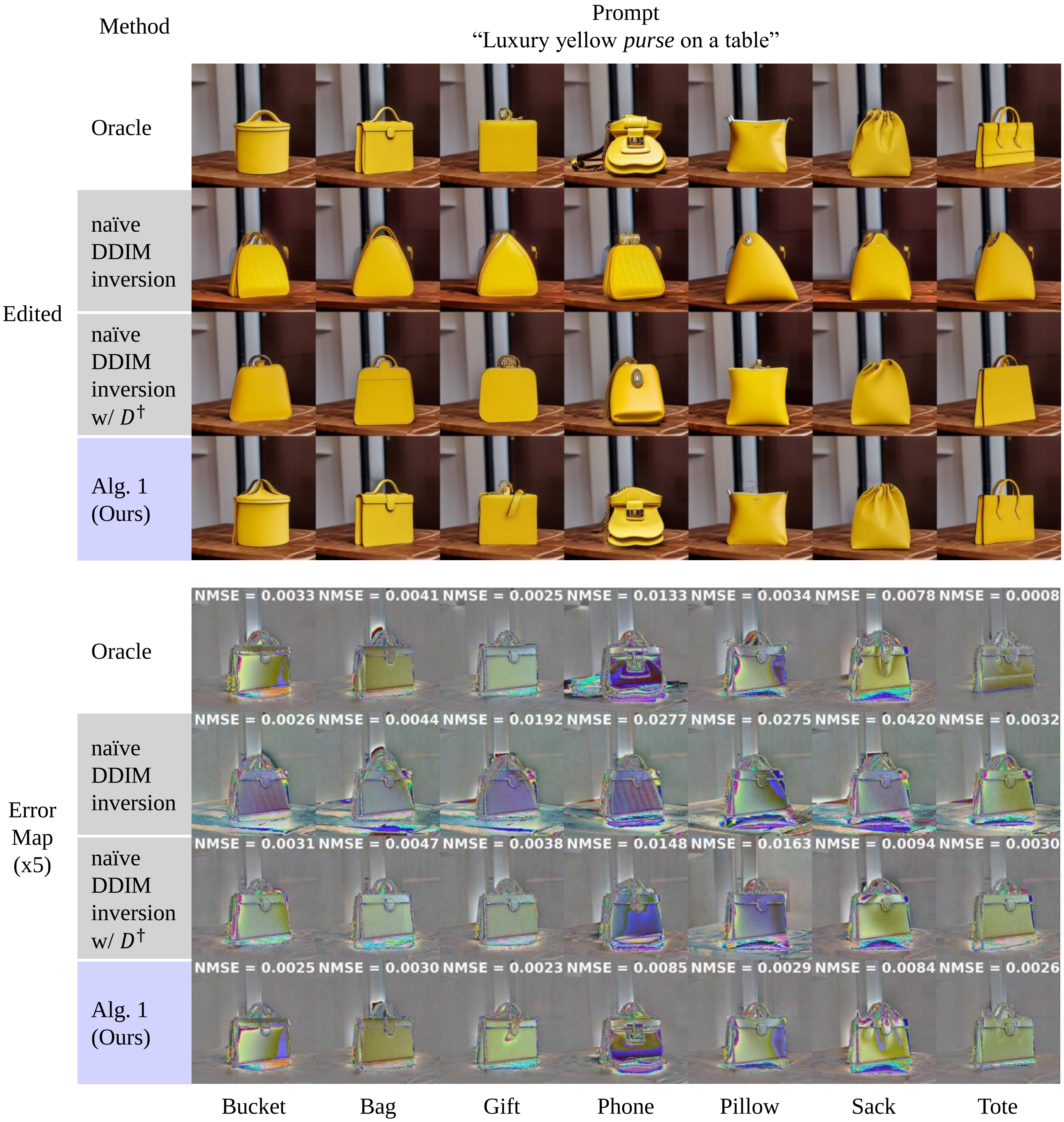

Method
Prompt
“Luxury yellow *purse* on a table”
Edited
Oracle
naïve DDIM inversion
naïve DDIM inversion w/ $D^\dagger$
Alg. 1 (Ours)
Error Map (x5)
Oracle
NMSE = 0.0033 NMSE = 0.0041 NMSE = 0.0025 NMSE = 0.0133 NMSE = 0.0034 NMSE = 0.0078 NMSE = 0.0008
naïve DDIM inversion
NMSE = 0.0026 NMSE = 0.0044 NMSE = 0.0192 NMSE = 0.0277 NMSE = 0.0275 NMSE = 0.0420 NMSE = 0.0032
naïve DDIM inversion w/ $D^\dagger$
NMSE = 0.0031 NMSE = 0.0047 NMSE = 0.0038 NMSE = 0.0148 NMSE = 0.0163 NMSE = 0.0094 NMSE = 0.0030
Alg. 1 (Ours)
NMSE = 0.0025 NMSE = 0.0030 NMSE = 0.0023 NMSE = 0.0085 NMSE = 0.0029 NMSE = 0.0084 NMSE = 0.0026
Bucket
Bag
Gift
Phone
Pillow
Sack
Tote

Figure S4. Additional experiment result on background-preserving editing. The *emphasized* word indicates the object changed in the generated image, replaced with different words at the bottom of the figure (e.g., *purse* becomes ‘bucket’). Our Alg. 1 preserves the background and allows diverse editing, even when the original image’s path is unknown (i.e., $(\boldsymbol{z}_{t_i})_{i=0}^{M}$). The first row (Oracle) shows results with the full trajectory, while subsequent rows use only the generated image (i.e., $\boldsymbol{x}_0$). In these cases, we estimate the trajectory through each inversion method and edit based on the inversion results. Estimating the original trajectory using the basic DDIM inversion (rows 2-3) fails to keep the background (background in the error map is not gray) compared to the Oracle and doesn’t consistently edit the ‘purse’ as prompted. In contrast, using our Alg. 1 (row 4) preserves the background similarly to the Oracle (with the background in the error map being gray) while consistently editing the ‘purse’ as prompted. Background NMSEs are inset.

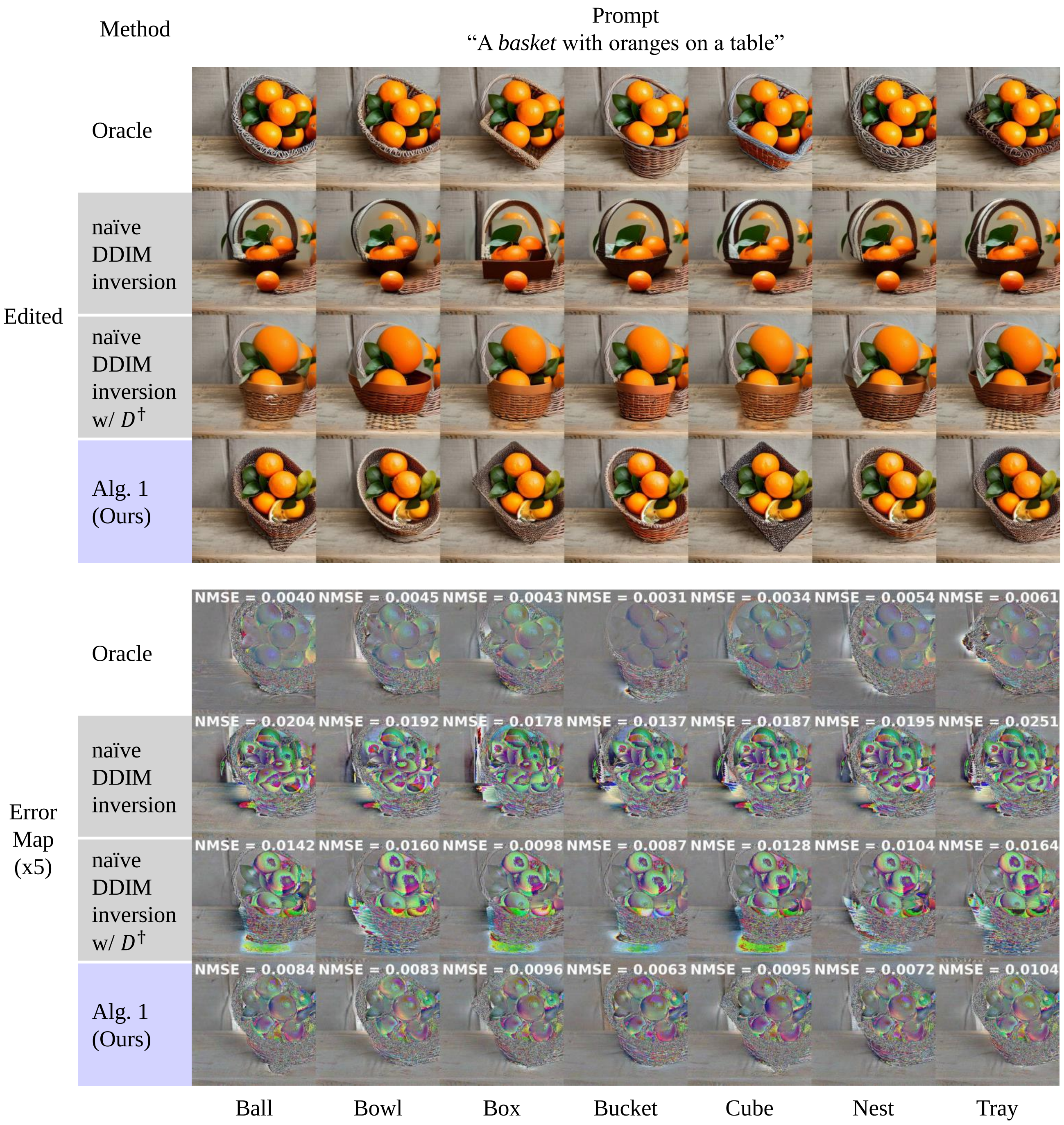

Method
Prompt
"A *basket* with oranges on a table"
Edited
Oracle
naïve DDIM inversion
naïve DDIM inversion w/ $D^\dagger$
Alg. 1 (Ours)
Error Map (x5)
Oracle
NMSE = 0.0040 NMSE = 0.0045 NMSE = 0.0043 NMSE = 0.0031 NMSE = 0.0034 NMSE = 0.0054 NMSE = 0.0061
naïve DDIM inversion
NMSE = 0.0204 NMSE = 0.0192 NMSE = 0.0178 NMSE = 0.0137 NMSE = 0.0187 NMSE = 0.0195 NMSE = 0.0251
naïve DDIM inversion w/ $D^\dagger$
NMSE = 0.0142 NMSE = 0.0160 NMSE = 0.0098 NMSE = 0.0087 NMSE = 0.0128 NMSE = 0.0104 NMSE = 0.0164
Alg. 1 (Ours)
NMSE = 0.0084 NMSE = 0.0083 NMSE = 0.0096 NMSE = 0.0063 NMSE = 0.0095 NMSE = 0.0072 NMSE = 0.0104
Ball
Bowl
Box
Bucket
Cube
Nest
Tray

Figure S5. Additional experiment result on background-preserving editing using prompt "A *basket* with oranges on a table". This experiment has the same setting with Fig. S4. As we attempt to edit the *basket*, the oranges should be treated as part of the background and should not be changed. Our Alg. 1 (row 4) preserves the oranges well, similar to the Oracle (row 1). In contrast, the naïve DDIM inversion indicates significant changes to the appearance of oranges (rows 2-3).