Supplementary Material for "h-Edit: Effective and Flexible Diffusion-Based Editing via Doob's h-Transform"

Table of Content

A. Theoretical Results

A.1. Derivation of the formula of $h(x_t, t)$

Below, we prove that $h(x_t, t)$ satisfying Eqs. 10, 11 can be expressed as follows:

$$h(x_t, t) = \mathbb{E}_{p(x_0|x_t)}[h(x_0, 0)] \tag{31}$$
$$= \mathbb{E}_{p(x_0|x_t)}[p_{\mathcal{Y}}(x_0)] \tag{32}$$

where $p(x_0|x_t)$ is the transition distribution of the base backward Markov process.

We can quickly verify that Eq. 31 is correct for $t = 1$ since $h(x_1, 1) = \int p(x_0|x_1) h(x_0, 0) dx_0 = \mathbb{E}_{p(x_0|x_1)}[h(x_0, 0)]$ directly from Eqs. 10, 11. Assuming that Eq. 31 has been correct for $t-1$ ($t \geq 2$), we will prove that it is correct for t. The RHS of Eq. 10 can be transformed as follows:

$$h(x_t, t) = \int p(x_{t-1}|x_t) h(x_{t-1}, t-1) dx_{t-1} \tag{33}$$
$$= \int p(x_{t-1}|x_t) \mathbb{E}_{p(x_0|x_{t-1})}[h(x_0, 0)] dx_{t-1} \tag{34}$$
$$= \int p(x_{t-1}|x_t) \left(\int p(x_0|x_{t-1}) h(x_0, 0) dx_0 \right) dx_{t-1} \tag{35}$$
$$= \int \left(\int p(x_0|x_{t-1}) p(x_{t-1}|x_t) dx_{t-1} \right) h(x_0, 0) dx_0 \tag{36}$$
$$= \int p(x_0|x_t) p_{\mathcal{Y}}(x_0) dx_0 \tag{37}$$
$$= \mathbb{E}_{p(x_0|x_t)}[h(x_0, 0)] \tag{38}$$

In Eq. 37, $p(x_0|x_t)$ equals $\int p(x_0|x_{t-1}) p(x_{t-1}|x_t) dx_{t+1}$ because this is the Chapman-Kolmogorov equation [30, 32] for the base backward process. Eq. 38 completes our proof.

A.2. Proof of Proposition 1

First, it can be seen that $p^h(x_{t-1}|x_t)$ is well normalized since according to Eqs. 9, 10, we have:

$$\int p^h(x_{t-1}|x_t) dx_{t-1} = \frac{\int p(x_{t-1}|x_t) h(x_{t-1}, t-1) dx_{t-1}}{h(x_t, t)} \tag{39}$$
$$= \frac{h(x_t, t)}{h(x_t, t)} \tag{40}$$
$$= 1 \tag{41}$$

Thus, $p^h(x_{t-1}|x_t)$ can be viewed as the transition distribution of our bridge. Besides, since x_{t-1} in $p^h(x_{t-1}|x_t)$ only depends on x_t, this bridge is a reverse-time Markov process.

Next, we prove that $p^h(x_t) = \frac{p(x_t) h(x_t, t)}{\mathbb{E}_{p(x_0)}[h(x_0, 0)]}$ for all $t \in [0, T]$. This equation holds for $t = T$ due to our assumption $p^h(x_T) = \frac{p(x_T) h(x_T, T)}{\mathbb{E}_{p(x_0)}[h(x_0, 0)]}$. Assuming that this equation holds for time t, we will prove that it holds for time $t-1$. Since the bridge is a reverse-time Markov process, we can compute $p^h(x_{t-1})$ as follows:

$$p^h(x_{t-1}) = \int p^h(x_{t-1}|x_t) p^h(x_t) dx_t \tag{42}$$
$$= \int p(x_{t-1}|x_t) \frac{h(x_{t-1}, t-1)}{\cancel{h(x_t, t)}} \frac{p(x_t) \cancel{h(x_t, t)}}{\mathbb{E}_{p(x_0)}[h(x_0, 0)]} dx_t \tag{43}$$
$$= \frac{h(x_{t-1}, t-1) \int p(x_{t-1}|x_t) p(x_t) dx_t}{\mathbb{E}_{p(x_0)}[h(x_0, 0)]} \tag{44}$$
$$= \frac{p(x_{t-1}) h(x_{t-1}, t-1)}{\mathbb{E}_{p(x_0)}[h(x_0, 0)]} \tag{45}$$

where Eq. 43 leverages Eq. 9 and the inductive assumption. Eq. 45 completes our proof.

Finally, we prove that $p^h(x_t)$ is a well normalized distribution as follows:

$$\int p^h(x_t)\,dx_t = \frac{\int p(x_t)\,h(x_t,t)\,dx_t}{\mathbb{E}_{p(x_0)}[h(x_0,0)]} \tag{46}$$

$$= \frac{\int p(x_t)\,\mathbb{E}_{p(x_0|x_t)}[h(x_0,0)]\,dx_t}{\mathbb{E}_{p(x_0)}[h(x_0,0)]} \tag{47}$$

$$= \frac{\int p(x_t)\left(\int p(x_0|x_t)\,h(x_0,0)\,dx_0\right)dx_t}{\mathbb{E}_{p(x_0)}[h(x_0,0)]} \tag{48}$$

$$= \frac{\int \left(\int p(x_t)\,p(x_0|x_t)\,dx_t\right)h(x_0,0)\,dx_0}{\mathbb{E}_{p(x_0)}[h(x_0,0)]} \tag{49}$$

$$= \frac{\int p(x_0)\,h(x_0,0)\,dx_0}{\mathbb{E}_{p(x_0)}[h(x_0,0)]} \tag{50}$$

$$= 1 \tag{51}$$

The fact that $h(x_t,t) = \mathbb{E}_{p(x_0|x_t)}[h(x_0,0)]$ in Eq. 47 was proven in Section A.1.

A.3. Closed-form expressions for the explicit and implicit h-Edit updates for Stable Diffusion

In this section, we derive closed-form expressions for the explicit and implicit h-Edit updates corresponding to Eq. 15 and Eq. 18, respectively, for Stable Diffusion (SD). First, we can express $\nabla_{x_t}\log h(x_t,t)$ as follows:

$$\nabla_{x_t}\log h(x_t,t) = \nabla_{x_t}\log p^h(x_t) - \nabla_{x_t}\log p(x_t) \tag{52}$$

$$= \frac{-\tilde{\epsilon}_\theta\left(x_t,t,c^{\text{edit}}\right)}{\sigma_t} - \frac{-\tilde{\epsilon}_\theta\left(x_t,t,c^{\text{orig}}\right)}{\sigma_t} \tag{53}$$

$$= \frac{-1}{\sigma_t}\left(\tilde{\epsilon}_\theta\left(x_t,t,c^{\text{edit}}\right) - \tilde{\epsilon}_\theta\left(x_t,t,c^{\text{orig}}\right)\right) \tag{54}$$

$$= \frac{-1}{\sigma_t}\Big(w^{\text{edit}}\epsilon_\theta\left(x_t,t,c^{\text{edit}}\right) + \left(1-w^{\text{edit}}\right)\epsilon_\theta\left(x_t,t,\varnothing\right) - \left(w^{\text{orig}}\epsilon_\theta\left(x_t,t,c^{\text{orig}}\right) + \left(1-w^{\text{orig}}\right)\epsilon_\theta\left(x_t,t,\varnothing\right)\right)\Big) \tag{55}$$

$$= \frac{-1}{\sigma_t}\Big(w^{\text{edit}}\epsilon_\theta\left(x_t,t,c^{\text{edit}}\right) - w^{\text{orig}}\epsilon_\theta\left(x_t,t,c^{\text{orig}}\right) + \left(w^{\text{orig}}-w^{\text{edit}}\right)\epsilon_\theta\left(x_t,t,\varnothing\right)\Big) \tag{56}$$

$$= \frac{-1}{\sigma_t}f(x_t,t) \tag{57}$$

Finding the formula of η in Eq. 15 can be somewhat tricky. The key is to examine the equation $x_{t-1}^{\text{base}} = x_t + \eta\nabla_{x_t}\log p(x_t) + \sqrt{2\eta}z$ in Eq. 14, which can be interpreted as sampling x_{t-1}^{base} from the Gaussian backward transition distribution $p_\theta(x_{t-1}|x_t)$. This implies that if we omit the random term $\sqrt{2\eta}z$, the simplified equation $x_{t-1}^{\text{base}} = x_t + \eta\nabla_{x_t}\log p(x_t)$ corresponds to the mean of $p_\theta(x_{t-1}|x_t)$, as provided in Eq. 4, and rewritten as follows:

$$x_{t-1}^{\text{base}} = \underbrace{\frac{a_{t-1}}{a_t}x_t + \left(\sqrt{\sigma_{t-1}^2-\omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right)\tilde{\epsilon}_\theta\left(x_t,t,c^{\text{orig}}\right)}_{\tilde{\mu}_{\theta,\omega,t,t-1}(x_t,c^{\text{orig}})} \tag{58}$$

$$= \frac{a_{t-1}}{a_t}x_t + \left(\sqrt{\sigma_{t-1}^2-\omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right)\left(w^{\text{orig}}\epsilon_\theta\left(x_t,t,c^{\text{orig}}\right) + \left(1-w^{\text{orig}}\right)\epsilon_\theta\left(x_t,t,\varnothing\right)\right) \tag{59}$$

$$= \frac{a_{t-1}}{a_t}x_t - \left(\sqrt{\sigma_{t-1}^2-\omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right)\sigma_t\nabla_{x_t}\log p(x_t) \tag{60}$$

Eq. 60 suggests that $\eta = -\left(\sqrt{\sigma_{t-1}^2-\omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right)\sigma_t$. One can easily verify that $\eta > 0$. It is worth noting that there is a little mismatch between the coefficients of x_t in Eq. 58 and in $x_{t-1}^{\text{base}} = x_t + \eta\nabla_{x_t}\log p(x_t)$. This is expected because the

standard LMC update assumes a forward diffusion process governed by the SDE $dx_t = \sqrt{2}dw_t$, which lacks a drift term. In contrast, the continuous-time forward process of Stable Diffusion follows the SDE $dx_t = \frac{-\beta_t}{2}x_t dt + \sqrt{\beta_t}dw_t$, which has the drift term $\frac{-\beta_t}{2}x_t$.

It can be inferred that u_t^{orig} mimics the random term $\sqrt{2\eta}z$, with the key difference being that it is precomputed during the forward pass rather than randomly sampled during the backward pass.

According to the above analysis, the explicit h-Edit update for Stable Diffusion is given by:

$$x_{t-1}^{\text{base}} = \underbrace{\tilde{\mu}_{\theta,\omega,t,t-1}\left(x_t^{\text{edit}}, c^{\text{orig}}\right)}_{x_t+\eta\nabla\log p(x_t)} + \underbrace{u_t^{\text{orig}}}_{\sqrt{2\eta}z} \tag{61}$$

$$x_{t-1}^{\text{edit}} = x_{t-1}^{\text{base}} + \underbrace{\left(-\left(\sqrt{\sigma_{t-1}^2 - \omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right)\sigma_t\right)}_{\eta}\underbrace{\frac{-1}{\sigma_t}f\left(x_t^{\text{edit}}, t\right)}_{\nabla\log h(x_t,t)} \tag{62}$$

$$= x_{t-1}^{\text{base}} + \left(\sqrt{\sigma_{t-1}^2 - \omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right) f\left(x_t^{\text{edit}}, t\right) \tag{63}$$

To derive the implicit h-Edit update, we first write Eq. 58 in the implicit form $x_{t-1} = \frac{a_{t-1}}{a_t}x_t + \left(\sqrt{\sigma_{t-1}^2 - \omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right)\tilde{\epsilon}_\theta\left(x_{t-1}, t-1, c^{\text{orig}}\right)$, which reveals that $\gamma = -\left(\sqrt{\sigma_{t-1}^2 - \omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right)\sigma_{t-1}$. Using this, we compute x_{t-1}^{edit} based on the formula in Eq. 18 as follows:

$$x_{t-1}^{\text{edit}} = x_{t-1}^{\text{base}} + \gamma\nabla_{x_{t-1}} h\left(x_{t-1}^{\text{base}}, t-1\right) \tag{64}$$

$$= x_{t-1}^{\text{base}} + \left(-\left(\sqrt{\sigma_{t-1}^2 - \omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right)\sigma_{t-1}\right)\frac{1}{\sigma_{t-1}} f\left(x_{t-1}^{\text{base}}, t-1\right) \tag{65}$$

$$= x_{t-1}^{\text{base}} + \left(\sqrt{\sigma_{t-1}^2 - \omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right) f\left(x_{t-1}^{\text{base}}, t-1\right) \tag{66}$$

where x_{t-1}^{base} is given in Eq. 61.

One advantage of the natural disentanglement in the h-Edit update is that the guidance scales w^{orig} for computing x_{t-1}^{base} in Eq. 59 and w^{orig} for computing $\nabla\log h\left(x_t, t\right)$ in Eq. 56 do *not* need to be the same. This allows w^{orig} in Eq. 59 to follow the guidance scale used in the forward pass, while w^{orig} in Eq. 56 can be chosen arbitrarily. To emphasize this distinction, we denote w^{orig} in Eq. 56 as $\hat{w}^{\text{orig}}$, indicating that it may differ from w^{orig} in Eq. 59. This $\hat{w}^{\text{orig}}$ can be interpreted as a hyperparameter controlling how much of the original image's information is excluded from the editing process. During our experiments, we observed that w^{orig}, $\hat{w}^{\text{orig}}$, and w^{edit} should be chosen such that $0 < w^{\text{orig}} \leq \hat{w}^{\text{orig}} < w^{\text{edit}}$.

B. Algorithms

B.1. h-Edit for Combined Editing

In Algorithms 1 and 2, we provide pseudo-codes for the explicit and implicit versions of h-Edit for combined text-guided and reward-model-based editing.

Algorithm 1 Explicit h-Edit for combined editing, compatible with both deterministic and random inversion, and supporting integration with the P2P [19].

Require: Original image x_0^{orig}, reference image x_0^{ref}, original text c^{orig}, edited text c^{edit}, guidance weights w^{orig}, w^{edit}, $\hat{w}^{\text{orig}}$, external encoder G, external distance loss $\mathcal{L}$, external guidance weight ρ_t.

$\left\{x_t^{\text{orig}}\right\}_{t=1}^{T}, \left\{u_t^{\text{orig}}\right\}_{t=1}^{T} = \text{Inversion}\left(x_0^{\text{orig}}, c^{\text{orig}}\right)$
$x_T^{\text{edit}} = x_T^{\text{orig}}$
for $t = T, \ldots, 1$ **do**
  $x_t = x_t^{\text{edit}}$
  $\tilde{\epsilon}_\theta\left(x_t, t, c^{\text{orig}}\right) = w^{\text{orig}}\epsilon_\theta\left(x_t, t, c^{\text{orig}}\right) + \left(1 - w^{\text{orig}}\right)\epsilon_\theta\left(x_t, t, \varnothing\right)$
  Compute $\tilde{\mu}_{\theta,\omega,t,t-1}\left(x_t, c^{\text{orig}}\right)$ from $\tilde{\epsilon}_\theta\left(x_t, t, c^{\text{orig}}\right)$ via Eq. 2
  $x_{t-1}^{\text{base}} = \tilde{\mu}_{\theta,\omega,t,t-1}\left(x_t, c^{\text{orig}}\right) + u_t^{\text{orig}}$
  if text-guided editing **then**
    if combined with P2P **then**
      Get the attention map M_t^{edit} from $\epsilon_\theta\left(x_t, t, c^{\text{edit}}\right)$
      Get the attention map M_t^{orig} from $\epsilon_\theta\left(x_t^{\text{orig}}, t, c^{\text{orig}}\right)$
      $\hat{M}_t^{\text{edit}} = \text{P2P}\left(M_t^{\text{edit}}, M_t^{\text{orig}}, t\right)$
      Apply the new attention map $\hat{M}_t^{\text{edit}}$ to $\epsilon_\theta\left(x_t, t, c^{\text{edit}}\right)$
    end if
    $f\left(x_t, t\right) = w^{\text{edit}}\epsilon_\theta\left(x_t, t, c^{\text{edit}}\right) - \hat{w}^{\text{orig}}\epsilon_\theta\left(x_t, t, c^{\text{orig}}\right) + \left(\hat{w}^{\text{orig}} - w^{\text{edit}}\right)\epsilon_\theta\left(x_t, t, \varnothing\right)$
    $\hat{x}_{t-1} = x_{t-1}^{\text{base}} + \left(\sqrt{\sigma_{t-1}^2 - \omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right) f\left(x_t, t\right)$
    $\hat{\epsilon}_t = \text{stop\_grad}\left(w^{\text{edit}}\epsilon_\theta\left(x_t, t, c^{\text{edit}}\right) + \left(1 - w^{\text{edit}}\right)\epsilon_\theta\left(x_t, t, \varnothing\right)\right)$
  else
    $\hat{x}_{t-1} = x_{t-1}^{\text{base}}$
    $\hat{\epsilon}_t = \text{stop\_grad}\left(w^{\text{orig}}\epsilon_\theta\left(x_t, t, c^{\text{orig}}\right) + \left(1 - w^{\text{orig}}\right)\epsilon_\theta\left(x_t, t, \varnothing\right)\right)$
  end if
  $x_{0|t} = \dfrac{x_t - \sigma_t\hat{\epsilon}_t}{a_t}$
  $g_t = -\nabla_{x_t}\mathcal{L}\left(G\left(x_{0|t}\right), G\left(x_0^{\text{ref}}\right)\right)$
  $x_{t-1}^{\text{edit}} = \hat{x}_{t-1} + \rho_t g_t$
  if text-guided editing **and** combined with P2P **and** local blending **then**
    $x_{t-1}^{\text{edit}} = \text{local\_blend}\left(x_{t-1}^{\text{edit}}, x_{t-1}^{\text{orig}}\right)$
  end if
end for

Algorithm 2 Implicit h-Edit for combined editing, compatible with both deterministic and random inversions, and supporting integration with the P2P [19].

Require: Original image x_0^{orig}, reference image x_0^{ref}, original text c^{orig}, edited text c^{edit}, guidance weights w^{orig}, w^{edit}, $\hat{w}^{\text{orig}}$, reconstruction weight λ_t, external encoder G, external distance loss $\mathcal{L}$, external guidance weight ρ_t, number of implicit loops K.

$\left\{x_t^{\text{orig}}\right\}_{t=1}^{T}, \left\{u_t^{\text{orig}}\right\}_{t=1}^{T} = \text{Inversion}\left(x_0^{\text{orig}}, c^{\text{orig}}\right)$
$x_T^{\text{edit}} = x_T^{\text{orig}}$
for $t = T, \ldots, 1$ **do**
  $x_t = x_t^{\text{edit}}$
  $\tilde{\epsilon}_\theta\left(x_t, t, c^{\text{orig}}\right) = w^{\text{orig}}\epsilon_\theta\left(x_t, t, c^{\text{orig}}\right) + \left(1 - w^{\text{orig}}\right)\epsilon_\theta\left(x_t, t, \varnothing\right)$
  Compute $\tilde{\mu}_{\theta,\omega,t,t-1}\left(x_t, c^{\text{orig}}\right)$ from $\tilde{\epsilon}_\theta\left(x_t, t, c^{\text{orig}}\right)$ via Eq. 2
  $x_{t-1}^{\text{base}} = \tilde{\mu}_{\theta,\omega,t,t-1}\left(x_t, c^{\text{orig}}\right) + u_t^{\text{orig}}$
  $x_{t-1}^{(0)} = x_{t-1}^{\text{base}}$
  for $k = 0, \ldots, K-1$ **do**
    if improving reconstruction **then**
      $r_{t-1} = x_{t-1}^{(k)} - x_{t-1}^{\text{base}}$
      $x_{t-1}^{(k)} = x_{t-1}^{(k)} - \lambda_{t-1} r_{t-1}$
    end if
    if text-guided editing **then**
      if combined with P2P **then**
        Get the attention map M_{t-1}^{edit} from $\epsilon_\theta\left(x_{t-1}^{(k)}, t-1, c^{\text{edit}}\right)$
        Get the attention map M_{t-1}^{orig} from $\epsilon_\theta\left(x_{t-1}^{\text{orig}}, t-1, c^{\text{orig}}\right)$
        $\hat{M}_{t-1}^{\text{edit}} = \text{P2P}\left(M_{t-1}^{\text{edit}}, M_{t-1}^{\text{orig}}, t-1\right)$
        Apply the new attention map $\hat{M}_{t-1}^{\text{edit}}$ to $\epsilon_\theta\left(x_{t-1}^{(k)}, t-1, c^{\text{edit}}\right)$
      end if
      $f\left(x_{t-1}^{(k)}, t-1\right) = w^{\text{edit}}\epsilon_\theta\left(x_{t-1}^{(k)}, t-1, c^{\text{edit}}\right) - \hat{w}^{\text{orig}}\epsilon_\theta\left(x_{t-1}^{(k)}, t-1, c^{\text{orig}}\right) + \left(\hat{w}^{\text{orig}} - w^{\text{edit}}\right)\epsilon_\theta\left(x_{t-1}^{(k)}, t-1, \varnothing\right)$
      $\hat{x}_{t-1} = x_{t-1}^{(k)} + \left(\sqrt{\sigma_{t-1}^2 - \omega_{t,t-1}^2} - \frac{\sigma_t a_{t-1}}{a_t}\right) f\left(x_{t-1}^{(k)}, t-1\right)$
      $\hat{\epsilon}_{t-1} = \text{stop\_grad}\left(w^{\text{edit}}\epsilon_\theta\left(x_{t-1}^{(k)}, t-1, c^{\text{edit}}\right) + \left(1 - w^{\text{edit}}\right)\epsilon_\theta\left(x_{t-1}^{(k)}, t-1, \varnothing\right)\right)$
    else
      $\hat{x}_{t-1} = x_{t-1}^{(k)}$
      $\hat{\epsilon}_{t-1} = \text{stop\_grad}\left(w^{\text{orig}}\epsilon_\theta\left(x_{t-1}^{(k)}, t-1, c^{\text{orig}}\right) + \left(1 - w^{\text{orig}}\right)\epsilon_\theta\left(x_{t-1}^{(k)}, t-1, \varnothing\right)\right)$
    end if
    $x_{0|t-1} = \dfrac{\hat{x}_{t-1} - \sigma_{t-1}\hat{\epsilon}_{t-1}}{a_{t-1}}$
    $g_{t-1} = -\nabla_{\hat{x}_{t-1}}\mathcal{L}\left(G\left(x_{0|t-1}\right), G\left(x_0^{\text{ref}}\right)\right)$
    $x_{t-1}^{(k+1)} = \hat{x}_{t-1} + \rho_{t-1} g_{t-1}$
  end for
  $x_{t-1}^{\text{edit}} = x_{t-1}^{(K)}$
  if text-guided editing **and** combined with P2P **and** local blending **then**
    $x_{t-1}^{\text{edit}} = \text{local\_blend}\left(x_{t-1}^{\text{edit}}, x_{t-1}^{\text{orig}}\right)$
  end if
end for

B.2. Edit Friendly for Combined Editing

In this work, we extend Edit Friendly [24] to combined text-guided and reward-model-based editing tasks by combining it with the technique in [79]. The pseudo-code for this extension is provided in Algorithm 3. This extension serves as a baseline for our method in the combined editing setting.

Algorithm 3 Edit Friendly for combined editing, supporting integration with the P2P [19].

Require: Original image x_0^{orig}, reference image x_0^{ref}, original text c^{orig}, edited text c^{edit}, guidance weights w^{orig}, w^{edit}, external encoder G, external distance loss $\mathcal{L}$, external guidance weight ρ_t.

$x_T^{\text{orig}}, \left\{u_t^{\text{orig}}\right\}_{t=1}^{T} = \text{RandomInversion}(x_0^{\text{orig}}, c^{\text{orig}})$
$x_T^{\text{edit}} = x_T^{\text{orig}}$
for $t = T, \ldots, 1$ **do**
  $x_t = x_t^{\text{edit}}$
  if text-guided editing **then**
    if combined with P2P **then**
      Get the attention map M_t^{edit} from $\epsilon_\theta\left(x_t, t, c^{\text{edit}}\right)$
      Get the attention map M_t^{orig} from $\epsilon_\theta\left(x_t^{\text{orig}}, t, c^{\text{orig}}\right)$
      $\hat{M}_t^{\text{edit}} = \text{P2P}\left(M_t^{\text{edit}}, M_t^{\text{orig}}, t\right)$
      Apply the new attention map $\hat{M}_t^{\text{edit}}$ to $\epsilon_\theta\left(x_t, t, c^{\text{edit}}\right)$
    end if
    $\tilde{\epsilon}_\theta\left(x_t, t\right) = w^{\text{edit}}\epsilon_\theta\left(x_t, t, c^{\text{edit}}\right) + \left(1 - w^{\text{edit}}\right)\epsilon_\theta\left(x_t, t, \varnothing\right)$
  else
    $\tilde{\epsilon}_\theta\left(x_t, t\right) = w^{\text{orig}}\epsilon_\theta\left(x_t, t, c^{\text{orig}}\right) + \left(1 - w^{\text{orig}}\right)\epsilon_\theta\left(x_t, t, \varnothing\right)$
  end if
  Compute $\tilde{\mu}_{\theta,\omega,t,t-1}\left(x_t, t\right)$ from $\tilde{\epsilon}_\theta\left(x_t, t\right)$ via Eq. 2
  $x_{0|t} = \dfrac{x_t - \sigma_t\tilde{\epsilon}_\theta\left(x_t, t\right)}{a_t}$ where $a_t = \sqrt{\bar{\alpha}_t}$ and $\sigma_t = \sqrt{1 - \bar{\alpha}_t}$
  $g_t = -\nabla_{x_t}\mathcal{L}\left(G\left(x_{0|t}\right), G\left(x_0^{\text{ref}}\right)\right)$
  $x_{t-1}^{\text{edit}} = \tilde{\mu}_{\theta,\omega,t,t-1}\left(x_t, t\right) + \rho_t g_t + u_t^{\text{orig}}$
  if text-guided editing **and** combined with P2P **and** local blending **then**
    $x_{t-1}^{\text{edit}} = \text{local\_blend}\left(x_{t-1}^{\text{edit}}, x_{t-1}^{\text{orig}}\right)$
  end if
end for
return x_0^{edit}

C. Additional Discussion on Related Work

C.1. Training-based Editing

Training-based approaches, such as DiffusionCLIP [33] and Asyrp [35], modify the noise network of a pretrained diffusion model through fine-tuning or by incorporating an auxiliary network, resulting in a new noise network that supports generating images with the desired editing attributes. The local directional CLIP loss [17] is commonly used as the training objective. However, these methods require training a new network for each specific editing target, limiting their adaptability to diverse editing scenarios in practice. In contrast, InstructPix2Pix [4] trains an entirely new diffusion model that generates images based on editing instructions. The instruction texts and target edited images for training are generated by GPT-3 [5] and P2P [19], respectively, meaning that the quality of the edits is inherently tied to P2P's performance. Additionally, the high training cost remains a significant drawback of this method.

C.2. Conditional Generation with Diffusion Models

The goal of conditional generation is to sample data from the joint distribution $p(x_0)\,p(y|x_0)$, which can be achieved by learning the score $\nabla \log p(x_t, y)$ of the joint distribution $p(x_t, y)$ via the score matching framework [25, 65]. Class-guided diffusion model [12] learns a noisy classifier $p(y|x_t)$ and combines its gradient with the score $\nabla \log p(x_t)$ learned by another unconditional diffusion model (e.g., DDPM [22]) to obtain $\nabla \log p(x_t, y)$. Meanwhile, classifier-free guidance [21] simultaneously learn both $\nabla \log p(x_t)$ and $\nabla \log p(x_t|y)$ using a single noise network. Energy-guided SDE (EGSDE) [83] extends class-guided diffusion models to solve the image-to-image translation problem. It utilizes a noisy classifier pretrained on both the source and target domains to define a similarity score between noisy samples from the two domains. This score acts as a negative energy guiding the generation of target domain samples toward preserving some properties of the corresponding source domain samples. The energy-based perspective have also been considered in works on generating compositional concepts with diffusion models [40]. FreeDom [79] approximates the time-dependent energy function in EGSDE using Tweedie's formula: $\mathcal{E}(c, x_t, t) = \mathbb{E}_{p(x_0|x_t)}[\mathcal{E}(c, x_0, t)] \approx \mathcal{E}\left(c, x_{0|t}, t\right)$ [9, 16]. This eliminates the reliance on a noisy classifier which is often difficult to obtain in practice and allows FreeDom to leverage any available pretrained model on clean samples x_0 to define the energy function. As a result, FreeDom supports conditional information from segmentation maps, style images, and face IDs. Similarly, UGD [2] utilizes Tweedie's formula but employs a different reparameterization for guidance using external networks.

The EGSDE framework can be considered as a special case of our reverse-time bridge modeling framework, as ours applies to more general Markov processes rather than just diffusion SDEs. Our framework also provides a formula for the bridge's transition distribution, enabling ancestral sampling in a discrete-time setting. Meanwhile, EGSDE usually relies on the Euler-Maruyama method for approximate sampling because it only has access to the instantaneous velocity at time t.

C.3. Diffusion Bridges and Doob's h-Transform

Most diffusion bridge methods [10, 39, 41, 63, 85] focus on the image-to-image translation problem which involves matching two explicit distributions of two domains A, B. They typically assume a diffusion model that generates domain A from Gaussian noise is given, and apply Doob's h-transform [15] to the forward process of this diffusion model to map samples of domain A to those of domain B rather than Gaussian noise. Some approaches like [41, 63] directly learn the h-function, while others [85] utilize an analytical form of the h-function and learn the score of the reverse bridge. Our method, in contrast, applies Doob's h-transform to the backward process to map Gaussian noise to samples with the desired target attributes.

D. Further Details on Experimental Settings

D.1. Text-guided Editing

The P2P hyperparameters for deterministic-inversion-based methods with P2P (including h-Edit-D + P2P) were configured based on the setup in [27]. Specifically, the sampling step proportions for self-attention and cross-attention controls were set to 0.6 and 0.4, respectively. For h-Edit-R and EF with P2P, the proportion of sampling steps for self-attention control was adjusted to 0.35, as 0.6 was found to be excessive for effective editing with these methods. For h-Edit-R and EF without P2P, the first 15 steps were skipped to ensure faithful reconstruction, as recommended in [24]. This skipping was not required for their P2P counterparts. For LEDITS++ [3], we adhered to the hyperparameters specified in the original paper.

D.2. Face Swapping

We utilized the official pretrained models for MegaFS, AFS, and DiffFace, available at MegaFS, AFS, and DiffFace, respectively. Since the official pretrained model for FaceShifter is unavailable, we used an unofficial pretrained model from this repository. For evaluation, we employed a pretrained ArcFace model with the IR-SE-50 backbone ([68, 79]), available through the InsightFace library for evaluation. This model was also used in h-Edit-R, EF, and FaceShifter[1] for generating swapped faces. For DiffFace, the ArcFace model with the ResNet101 backbone from its official code was used for face swapping. MegaFS and AFS relied on the ArcFace model with the IR-SE-50 backbone during training but not during face swapping. Additional evaluations using other face identity representation models are provided in Appendix E.2. CelebA-HQ images were resized to 256×256 and cropped as `x=x[:, :, 35:223, 32:220]` to prepare them for input into the ArcFace model. Following [79], we defined the coefficient ρ_t for the identity similarity reward gradient (Algorithms 2, 1, 3) as $\rho^{\text{face}} \times \sqrt{\bar{\alpha}_t}$, where $\bar{\alpha}_t$ is the Stable Diffusion scheduler coefficient at time step t. For h-Edit-R and EF, ρ^{face} was set to 100.0. For h-Edit-R (3s),ρ^{face} was reduced to 50.0, which provided a better balance between editing effectiveness and

[1] FaceShifter uses the ArcFace model with the IR-SE-50 backbone to extract face identity embeddings during both training and generating swapped faces.

faithfulness when using three optimization steps. To further enhance faithfulness to the original image, we incorporated the negative LPIPS score as an additional reward alongside identity similarity. The LPIPS score, computed using a pretrained VGG network, measures the perceptual similarity between x_0^{edit} and x_0^{orig}. The coefficient for this reward is similar to that of the identity similarity reward. For post-processing, we applied a mask generated by the face parsing model in [78] to preserve the original background while applying edits to the face. This procedure was consistent across all baselines. The face swapping results without using masks are provided in Appdx. F.5.

D.3. Combined Text-guided and Style Editing

In combined text-guided and style editing, we disabled local blending in P2P as our experiments indicated that it negatively impacts style editing performance. For EF + P2P, following [79], we scaled the gradient norm of the style loss reward at each time t by the norm of $\left[\epsilon\left(x_t, t, c^{\text{edit}}\right) - \epsilon\left(x_t, t, \varnothing\right)\right]$. This corresponds to defining the coefficient ρ_t for style editing in EF + P2P as:

$$\rho_t := \rho^{\text{sty}} * \frac{\left\|\left(\epsilon\left(x_t, t, c^{\text{edit}}\right) - \epsilon\left(x_t, t, \varnothing\right)\right)\right\|_2}{\|g_t\|_2} \tag{67}$$

For h-Edit-R + P2P, we scaled the gradient norm of the style reward to match the norm of the text-guided editing function $f\left(\cdot\right)$ in Eq. 24. This approach leverages the disentangled update mechanism unique to our method (Sections 3 and A.3). Accordingly, the coefficient ρ_t for the style editing term in h-Edit-R + P2P is defined as:

$$\rho_t := \rho^{\text{sty}} * \frac{\|f\left(x_t, t\right)\|_2}{\|g_t\|_2} \tag{68}$$

E. Additional Experimental Results

E.1. Text-guided Editing

E.1.1 Deterministic-inversion-based methods

Fig. 5 shows additional edited images produced by h-Edit-D + P2P alongside other deterministic-inversion-based editing methods with P2P [7, 27, 38, 45, 46]. h-Edit-D + P2P consistently outperforms the baselines in handling difficult edits, while maintaining faithful reconstruction, as reflected in the quantitative results in Table 1. For instance, our method successfully removes the boy's tie (first row, right) and transforms the car into a motorcycle (seventh row, right), tasks where most other methods struggle. Although NP + P2P and NT + P2P demonstrate strong editing capabilities, as evidenced by their high local CLIP similarity scores in Table 1, they are not good at preserving non-edited content compared to other methods. Conversely, NMG + P2P, StyleD + P2P, and PnP Inv + P2P achieve high fidelity to the original image, but fail to deliver effective edits in many cases.

E.1.2 Random-inversion-based methods

In Fig. 6, we present additional visual comparisons of h-Edit-R + P2P against EF + P2P and LEDITS++. These visualizations are consistent with the quantitative results in Table 1, confirming that our method surpasses both EF + P2P and LEDITS++ in editing effectiveness and faithfulness. Further qualitative results of h-Edit-R and EF without P2P are shown in Fig. 7, where our method once again demonstrates superior performance.

E.2. Face Swapping

Since h-Edit-R, EF, and FaceShifter utilize the same ArcFace model for both face swapping and evaluation, this may lead to more favorable identity matching results for these methods compared to other baselines. To ensure a fair comparison, we reassessed the identity transfer quality of all methods using alternative face identity representation models. Specifically, we used VGG-Face [49], FaceNet128, FaceNet512 [59] and ArcFace with the ResNet34 backbone. These models were implemented in TensorFlow with pretrained weights available through the DeepFace repository [60, 61]. Quantitative results of this evaluation are provided in Table 2.

Interestingly, DiffFace achieves the best performance across all face identity representation models used for evaluation. h-Edit-R (3s) and h-Edit-R rank second and third, respectively, outperforming EF and FaceShifter but falling slightly short of DiffFace. This demonstrates that our method is capable of effective face swapping, even without being explicitly trained for this task like DiffFace, as further illustrated by the qualitative results in Fig. 8. We hypothesize that DiffFace's good

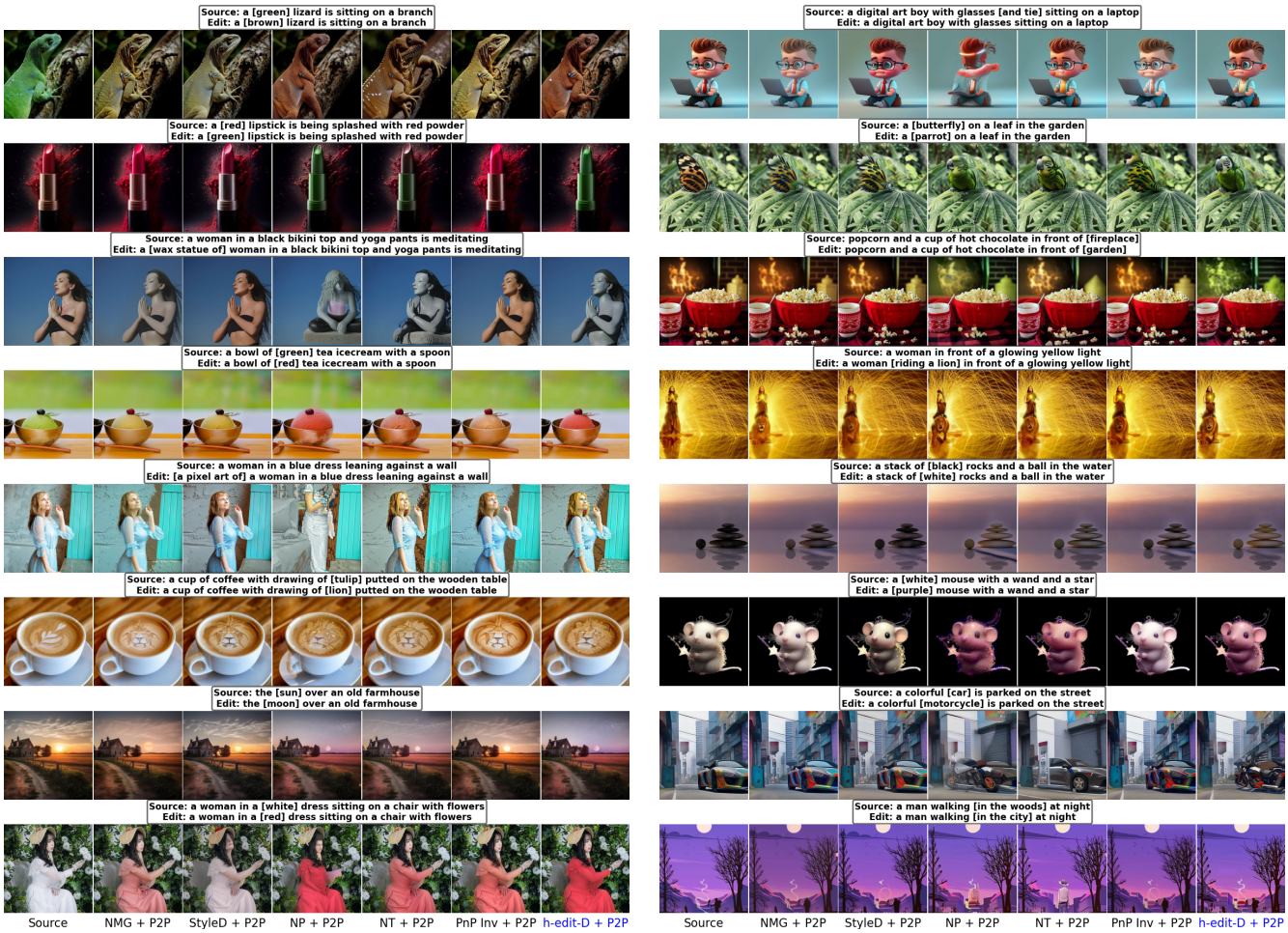


Figure 5. Additional visual comparisons between h-Edit-D + P2P and other deterministic-inversion-based methods with P2P.

Model	Metric	FaceShifter	MegaFS	AFS	DiffFace	EF	h-edit-R	h-edit-R (3s)
ArcFace (ResNet34)	Cosine Sim. ↑	0.54	0.33	0.44	**0.56**	0.50	0.52	<u>0.55</u>
VGG-Face	L2 Dist. ↓	0.99	1.12	1.03	**0.96**	1.02	1.00	<u>0.97</u>
FaceNet128	L2 Dist. ↓	0.83	1.02	0.86	**0.77**	0.83	<u>0.80</u>	**0.77**
FaceNet512	L2 Dist. ↓	<u>0.81</u>	1.01	0.87	**0.77**	0.83	<u>0.81</u>	**0.77**

Table 2. Face identity transfer results evaluated using face identity representation models different from the ArcFace model with the IR-SE-50 backbone.

performance may be attributed to (i) its use of an ArcFace model with a larger backbone (ResNet101) for face swapping and (ii) training on a larger dataset compared to the pretrained diffusion model employed by our method.

E.3. Combined Text-guided and Style Editing

Fig. 9 illustrates the changes in style loss, local CLIP similarity, and LPIPS score as the style editing coefficient ρ^{sty} is varied from 0.1 to 1.0 for h-Edit-R + P2P and from 1.1 to 2.0 for EF + P2P. While the ranges of ρ^{sty} differ, the resulting style loss, local CLIP similarity, and LPIPS score ranges are comparable, validating the appropriateness of our parameter selection. Increasing ρ^{sty} improves style transfer (lower style loss) but compromises text-guided editing quality in terms of both effectiveness and faithfulness (lower local CLIP similarity and higher LPIPS respectively). Since determining the

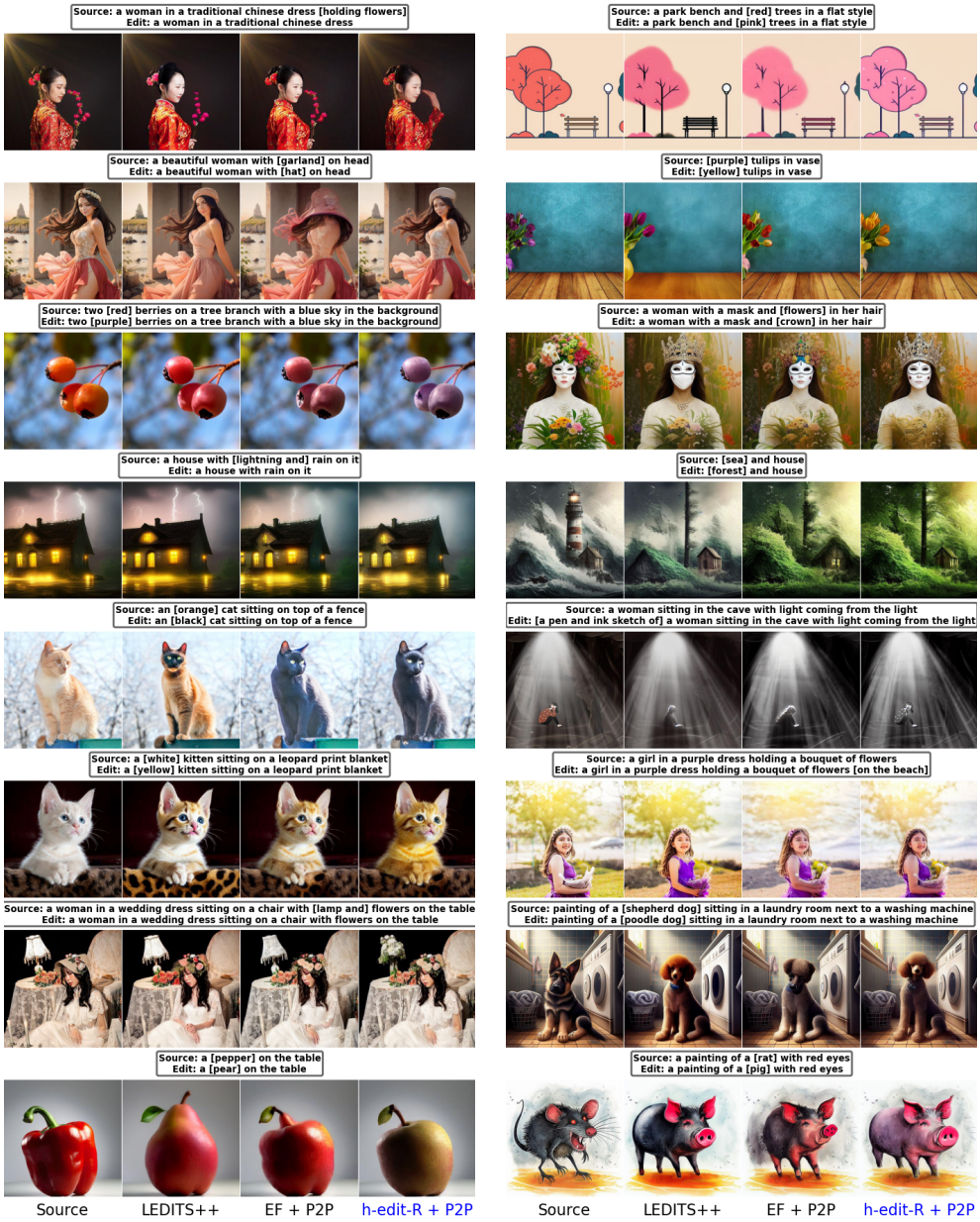


Figure 6. Additional qualitative results of h-Edit-R, EF, and LEDITS+++ with P2P.

optimal value of ρ^{sty} for achieving a balance between style and text-guided editing is nontrivial, we identified candidate values near the intersection of the style loss and LPIPS curves. Combining this with visual inspection, we selected ρ^{sty} value of 0.6 for h-Edit-R and 1.5 for EF.

Although EF exhibits similar quantitative trends to our method when ρ^{sty} is varied, its qualitative behavior is notably different. As shown in Fig. 10, our method smoothly incorporates more style information into the edited images while preserving their global structure as ρ^{sty} increases. In contrast, EF often modifies the global structure of the images to accommodate the increased ρ^{sty}. This advantage of our approach likely stems from the natural decomposition of the update into reconstruction and editing terms (Eq. 18), enabling style edits to be added to the text-guided editing term with minimal impact on the reconstruction term. EF, on the other hand, lacks such a decomposition, meaning the introduction of the style editing term directly affects reconstruction. These findings highlight the limitations of relying solely on quantitative metrics to compare our method with EF, as they may fail to capture important qualitative differences.

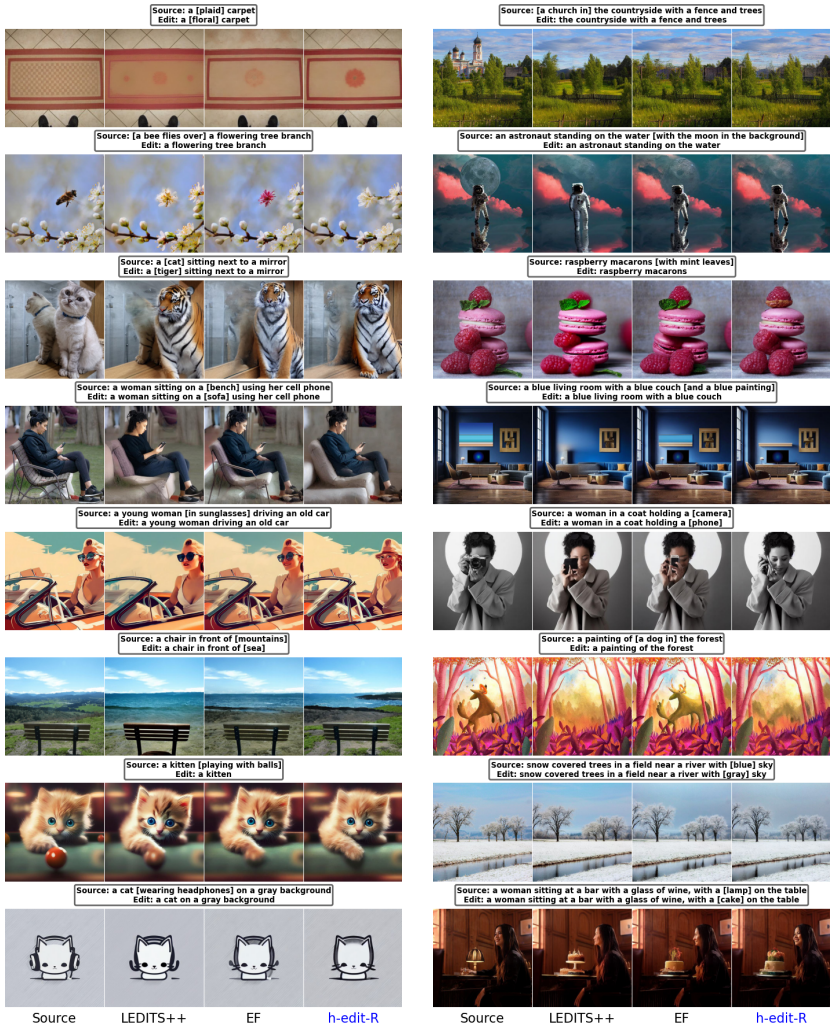


Figure 7. Additional qualitative results of h-Edit-R, EF (without P2P), and LEDITS++.

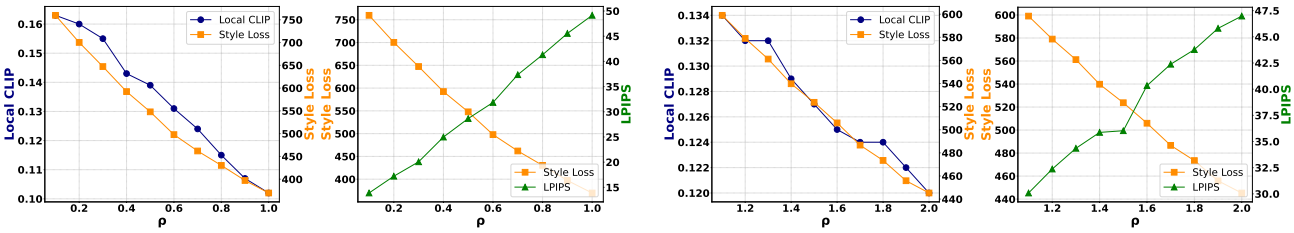
In Fig. 11, we present addition visualizations comparing h-Edit-R + P2P and EF + P2P, with ρ^{sty} set to the optimal values for each method. The results clearly demonstrate that our method combined with P2P surpasses EF + P2P in both style transfer and text-guided editing, achieving superior quality and consistency.

E.4. Results when Combining with MasaCtrl and Plug-and-Play

In this section, we compare the performance of h-Edit with other baselines when combined with MasaCtrl [6] and Plug-and-Play (PnP) [70]. For MasaCtrl, we adopted the implementation from the PnP Inversion paper [27] which omits the source prompt during editing. We observed that this approach yields more stable results compared to using the source prompt. Since editing methods like NT, NP and NMG are incompatible with this setting, they were excluded in our experiments with MasaCtrl.

As shown in Table 3, both h-Edit-R and h-Edit-D significantly outperform EF and deterministic-inversion-based baselines when combined with either MasaCtrl or PnP. For example, with PnP, h-Edit-D and h-Edit-R surpass NT and EF by 0.014 and 0.029 on the local directional CLIP metric, while achieving about 0.70 and 0.90 lower LPIPS scores, respectively. It is also evident that PnP is a more effective attention control method than MasaCtrl on the PIE-Bench dataset. However, both PnP and MasaCtrl are less effective and stable than P2P [19], as indicated by our quantitative results in Tables 1 and 3, and through our observations.

Figure 8. Additional qualitative comparisons between our method and other face swapping baselines. Identity similarity scores (higher is better) computed using ArcFace with the ResNet34 backbone are displayed below each output.



(a) h-Edit-R + P2P

(b) EF + P2P

Figure 9. Changes in style loss, local CLIP similarity, and LPIPS score of h-Edit-R + P2P and EF + P2P when ρ^{sty} is varied from 0.1 to 1.0 for h-Edit-R + P2P and from 1.1 to 2.0 for EF + P2P.

Attn.	Inv.	Method	CLIP Sim.↑	Local CLIP↑	DINO Dist.$_{\times 10^2}$↓	LPIPS$_{\times 10^2}$↓	SSIM$_{\times 10}$↑	PSNR↑
MasaCtrl	Deter.	PnP Inv	**0.243**	0.068	2.47	8.79	8.13	22.64
		h-Edit-D	**0.243**	**0.071**	**2.38**	**8.62**	**8.16**	**22.85**
	Random	EF	0.241	0.059	2.75	8.57	8.15	22.49
		h-Edit-R	**0.242**	**0.065**	**2.46**	**8.42**	**8.18**	**22.68**
PnP	Deter.	NP	0.250	0.152	1.84	8.55	8.19	25.05
		NT	0.251	0.144	1.58	7.94	8.24	25.53
		NMG	0.253	0.101	2.08	9.96	8.02	23.20
		PnP Inv	0.253	0.109	1.75	9.29	8.15	24.18
		h-Edit-D	**0.254**	**0.158**	**1.51**	**7.28**	**8.33**	**25.68**
	Random	EF	0.253	0.118	1.48	6.87	8.32	24.77
		h-Edit-R	**0.255**	**0.147**	**1.39**	**5.97**	**8.43**	**25.75**

Table 3. Text-guided editing results with MasaCtrl [6] and Plug-n-Play [70] on PIE-Bench. h-Edit significantly outperforms other baselines in all metrics.

Figure 10. Visualizations of edited images with ρ^{sty} values ranging from $\{1.4, 1.5, 1.6, 1.7, 1.8\}$ for EF + P2P and $\{0.4, 0.5, 0.6, 0.7, 0.8\}$ for h-Edit-R + P2P.

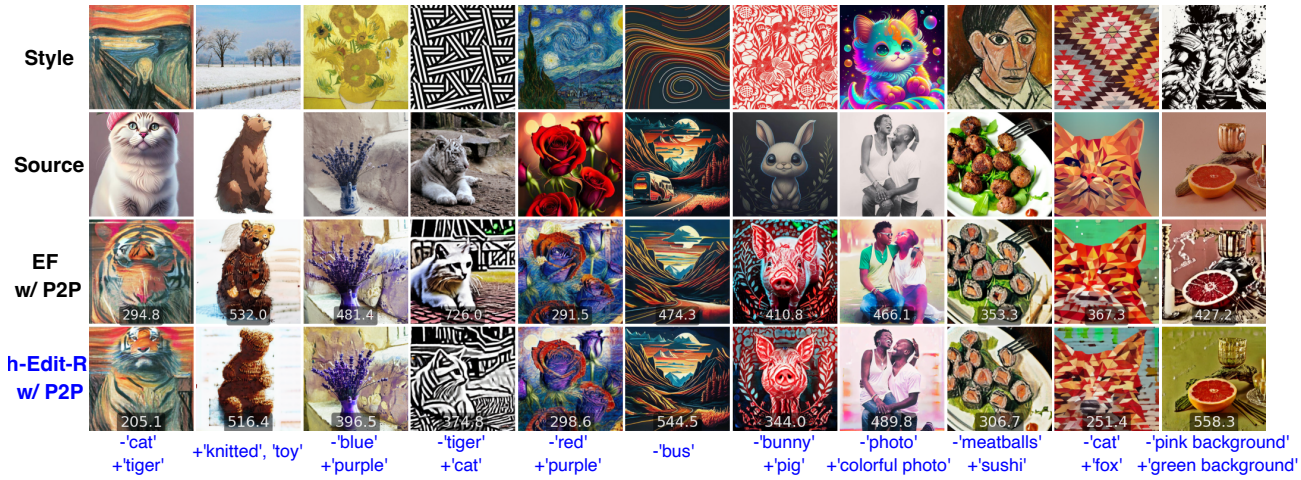


Figure 11. Additional qualitative results of h-Edit-R + P2P and EF + P2P for the combined style and text-guided editing task. Style loss values (lower is better) are displayed below each output image.

F. Ablation Studies

F.1. Impact of $\hat{w}^{\text{orig}}$

$\hat{w}^{\text{orig}}$	CLIP Sim.↑	Local CLIP↑	DINO Dist.$_{\times 10^2}$↓	LPIPS$_{\times 10^2}$↓	SSIM$_{\times 10}$↑	PNSR↑
1.0	**0.256**	0.118	1.64	6.00	8.38	25.75
3.0	0.255	0.137	1.52	5.49	8.44	26.36
5.0	**0.256**	0.159	**1.45**	**5.08**	8.50	**26.97**
7.0	0.254	**0.173**	1.60	5.22	**8.51**	26.94
9.0	0.241	0.172	2.30	6.44	8.40	26.03

Table 4. Quantitative results of h-Edit-R + P2P when varying $\hat{w}^{\text{orig}}$ from 1.0 to 9.0 while keeping w^{edit} and w^{orig} fixed at 7.5 and 1.0, respectively. The best value for each metric is highlighted in bold.

In this section, we study the impact of $\hat{w}^{\text{orig}}$ in Eq. 24 by varying its value within $\{1.0, 3.0, 5.0, 7.0, 9.0\}$ while keeping $w^{\text{orig}} = 1$ and $w^{\text{edit}} = 7.5$ fixed for h-Edit-R + P2P. Quantitative and qualitative results are shown in Table 4 and Fig. 12, respectively. The results indicate that increasing $\hat{w}^{\text{orig}}$ to a suitable value enhances both editing accuracy and fidelity. For example, raising $\hat{w}^{\text{orig}}$ from 1.0 to 7.0 restores the woman's armor suit in the first row on the left of Fig. 12 while also straightening her hair. Similarly, it effectively removes the balloons in the background while preserving the original appearance of the girl in a red dress in the twelfth row on the left. As discussed in Section A.3, $\hat{w}^{\text{orig}}$ controls how much of the original image's information is excluded during editing. Larger values of $\hat{w}^{\text{orig}}$ helps isolate essential information, enabling precise localization of edits. However, excessively high values (i.e., exceeding w^{edit}) may degrade reconstruction quality by removing too much original

information. This is evident in the case of changing colorful paint to drab paint in the last row on the right. These observations suggest that the optimal value of $\hat{w}^{\text{orig}}$ is case-dependent, for $w^{\text{edit}} = 7.5$, we found $\hat{w}^{\text{orig}} = 5.0$ achieves the best balance between editing effectiveness and faithfulness.

F.2. Impact of w^{edit}

We investigate the influence of w^{edit} in Eq. 24 for h-Edit-R + P2P by analyzing edited images across different $\left(w^{\text{edit}}, \hat{w}^{\text{orig}}\right)$ pairs: $\{(7.5, 3.0), (7.5, 5.0), (10.0, 7.0), (10.0, 9.0), (12.5, 9.0), (12.5, 11.0)\}$. Qualitative results are provided in Fig. 13. In general, higher w^{edit} values enhance editing effectiveness, allowing to handle difficult edits. For example, increasing w^{edit} from 7.5 to 12.5 successfully introduces dragons to the images in the final row of Fig. 13. However, higher w^{edit} can degrade reconstruction quality in non-edited regions, requiring a proportional increase in $\hat{w}^{\text{orig}}$ to mitigate this effect. Even so, this approach may not succeed in all scenarios. We can overcome this issue by using multiple optimization steps (available for implicit h-Edit). This technique progressively refines edits via applying the score function iteratively, effectively handling challenging cases while maintaining good reconstruction.

F.3. Impact of multiple optimization steps in implicit h-Edit

Fig. 14 highlights the advantage of the implicit version of h-Edit when utilizing multiple optimization steps. Increasing the number of optimization steps significantly enhances editing accuracy while maintaining minimal degradation in reconstruction quality. This capability is unique to the implicit version and cannot be replicated by simply increasing the number of sampling steps. For instance, the explicit version, even with 200 sampling steps, performs only comparably or slightly better than the default implicit version with 50 sampling steps and one optimization step, yet it falls notably short compared to the implicit version with three optimization steps.

Additionally, the effectiveness of multiple optimization steps is evident in the face swapping task, where h-Edit-R with three optimization steps outperforms its one-step counterpart, as presented in Section E.2.

F.4. Comparison between explicit and implicit versions

Attn.	Steps	Method	CLIP Sim.	Local CLIP↑	DINO Dist.$_{\times 10^2}$↓	LPIPS$_{\times 10^2}$↓	SSIM$_{\times 10}$↑	PSNR↑
None	25	h-Edit-R (ex)	0.252	0.139	1.10	5.10	8.49	26.79
		h-Edit-R (im)	0.255	0.148	1.39	5.98	8.41	25.77
	50	h-Edit-R (ex)	0.253	0.141	1.10	5.07	8.51	27.00
		h-Edit-R (im)	0.255	0.148	1.28	5.55	8.46	26.43
P2P	25	h-Edit-R (ex)	0.254	0.153	1.38	5.04	8.50	26.81
		h-Edit-R (im)	0.255	0.150	1.38	5.03	8.50	26.88
	50	h-Edit-R (ex)	0.256	0.158	1.47	5.10	8.50	26.85
		h-Edit-R (im)	0.256	0.159	1.45	5.08	8.50	26.97

Table 5. Quantitative comparison of h-Edit-R implicit and explicit forms, with and without P2P, evaluated over 25 and 50 sampling steps.

In this section, we compare the explicit and implicit versions of h-Edit-R with and without P2P, using either 25 or 50 sampling steps. Without P2P, the implicit version generally performs more accurate edits than the explicit counterpart, though the results vary by case, as shown in Table 5 and Fig. 15. However, when combined with P2P, the two versions perform comparably. Instances where implicit h-Edit-R outperforms the explicit version, and vice versa, are illustrated in Fig. 16. Our preference for the implicit version as the default is not primarily due to its performance relative to the explicit version but rather its ability to support multiple optimization steps, which offers greater flexibility.

F.5. Face swapping without masks

We demonstrate that our h-Edit-R method can perform face swapping without relying on mask postprocessing techniques for reconstruction, with qualitative results of h-Edit-R (3s) shown in Fig. 17. h-Edit-R without masks achieves near-perfect faithful reconstruction, with minor background changes. For instance, in the third row (left), it preserves background text, while in more complex backgrounds, such as dense text (last row, right) or intricate shirt patterns (last row, left), it maintains individual features with slight background blurring. This capability is unique to our method, as state-of-the-art approaches like DiffFace and FaceShifter rely on masks for faithful reconstruction. These findings suggest that in scenarios where masks are unavailable, our method is a robust choice for face editing with minimal reconstruction error.

Inv.	Attn.	Method	Time (s)↓
Deter.	P2P	NP	**21.68**
		NT	186.84
		StyleD	467.16
		NMG	35.67
		PnP Inv	37.65
		h-Edit-D	48.63
Random	None	EF	23.20
		LEDITS++	**18.31**
		h-Edit-R	33.07
	P2P	EF	**32.95**
		h-Edit-R	50.21

(a) Editing time for text-guided editing methods

Method	Time (s)↓
FaceShifter	1.31
MegaFS	2.29
AFS	**1.03**
DiffFace	46.42
EF	26.11
h-edit-R	26.34
h-edit-R (3s)	51.36

(b) Editing time for face swapping methods

Inv.	Attn.	Method	Time (s)↓
Random	P2P	EF	**44.32**
		h-Edit-R	50.68

(c) Editing time for combined text-guided and style editing methods

Table 6. Editing times per image (in seconds) of our method and baselines across three tasks: text-guided editing (left), face swapping (top right), and combined text-guided and style-based editing (bottom right). Experiments were conducted on an NVIDIA V100 GPU 32GB.

F.6. Running time

Table 6 shows the editing times per image of our method and baselines for three editing tasks: text-guided editing, face swapping, and combined text-guided and style editing.

In the text-guided setting, among deterministic-inversion-based methods, h-Edit-D + P2P requires longer computation time (48.63s) than NP + P2P (21.68s), PnP Inv + P2P (37.65s), and NMG + P2P (35.67s) due to additional U-Net calls for reconstruction and editing term computation. However, this additional 12-second overhead compared to PnP Inv + P2P yields significantly improved performance, with a 0.05 increase in local CLIP Similarity and 0.6×10^{-2} better LPIPS (Table 1). While NP + P2P achieves the fastest processing time by simply substituting source embedding for null embedding during editing, it suffers from substantially lower reconstruction quality. Our favorable trade-off between computation time and editing quality extends to comparisons with random-inversion-based methods. LEDITS++ is the fastest as they leverage high-order solvers [13, 43, 84] - a feature that could also be incorporated into our method.

In the face swapping task, diffusion-based methods generally require longer processing time per image compared to GAN-based methods (FaceShifter [37]: 1.31s) or StyleGAN-based approaches (MegaFS [86]: 2.29s, AFS [71]: 1.03s) due to their iterative sampling nature. Among diffusion-based methods, h-Edit-R (26.34s) and EF (26.11s) achieve the fastest processing times. Despite sharing the same sampling steps, h-Edit-R outperforms DiffFace (46.42s) in efficiency as DiffFace requires additional gaze detection and face parsing models at each step, beyond the common ArcFace computation. While our h-Edit-R with 3 optimization steps variant shows slightly increased computation time (51.36s), it achieves better ArcFace ID similarity compared to DiffFace with comparable reconstruction quality. Notably, as training-free approaches, our method and EF offer immediate deployment advantages over DiffFace and GAN-based methods that require task-specific training.

In the combined text-guided and style editing task, h-Edit-R + P2P (50.68s) shows only a moderate increase from its text-guided variant (50.21s) by avoiding U-Net backpropagation for style editing. In contrast, EF + P2P with FreeDom [79]'s technique requires additional backpropagation computation, resulting in a larger time increase from its text-guided counterpart (32.95s to 44.32s).

G. Analysis on Metrics

During our text-guided editing experiments, we observed that CLIP similarity and DINO distance metrics could yield inconsistencies between quantitative and qualitative results. For CLIP similarity, we hypothesize that this occurs because the attribute being edited often constitutes only a small portion of the target prompt. In such cases, even accurate edits may result in minor improvements in CLIP similarity, whereas unintended changes to other attributes can lead to significant drops. Consequently, methods that make no edits and simply preserve the original image may achieve comparable or better CLIP similarity scores than methods that successfully perform challenging edits. This phenomenon is evident with NP and NT - the two strong editing methods capable of handling challenging edits more effectively than PnP Inv, as shown in Fig. 5.

However, their CLIP similarity scores are lower than that of PnP Inv, as illustrated in Table 1.

In the case of DINO distance, since this metric is computed on the entire image rather than the non-editing region, it can yield poor results in significant editing scenarios like changing background color or removing objects even when original non-editing content is perfectly preserved.

H. Ethical Considerations

Our work aims to advance the development of effective and efficient diffusion-based image editing methods, fostering contributions to both academic research and real-world applications. However, we recognize that these advancements could be misused for harmful purposes, such as generating misinformation or damaging individuals' reputations. To address these risks, it is crucial to implement safeguards that detect and prevent unethical applications. One potential approach is to employ a detection framework that analyzes edited images and flags or discards outputs that violate ethical guidelines or pose potential harm to society. Such proactive measures can help ensure that this technology is used responsibly and ethically.

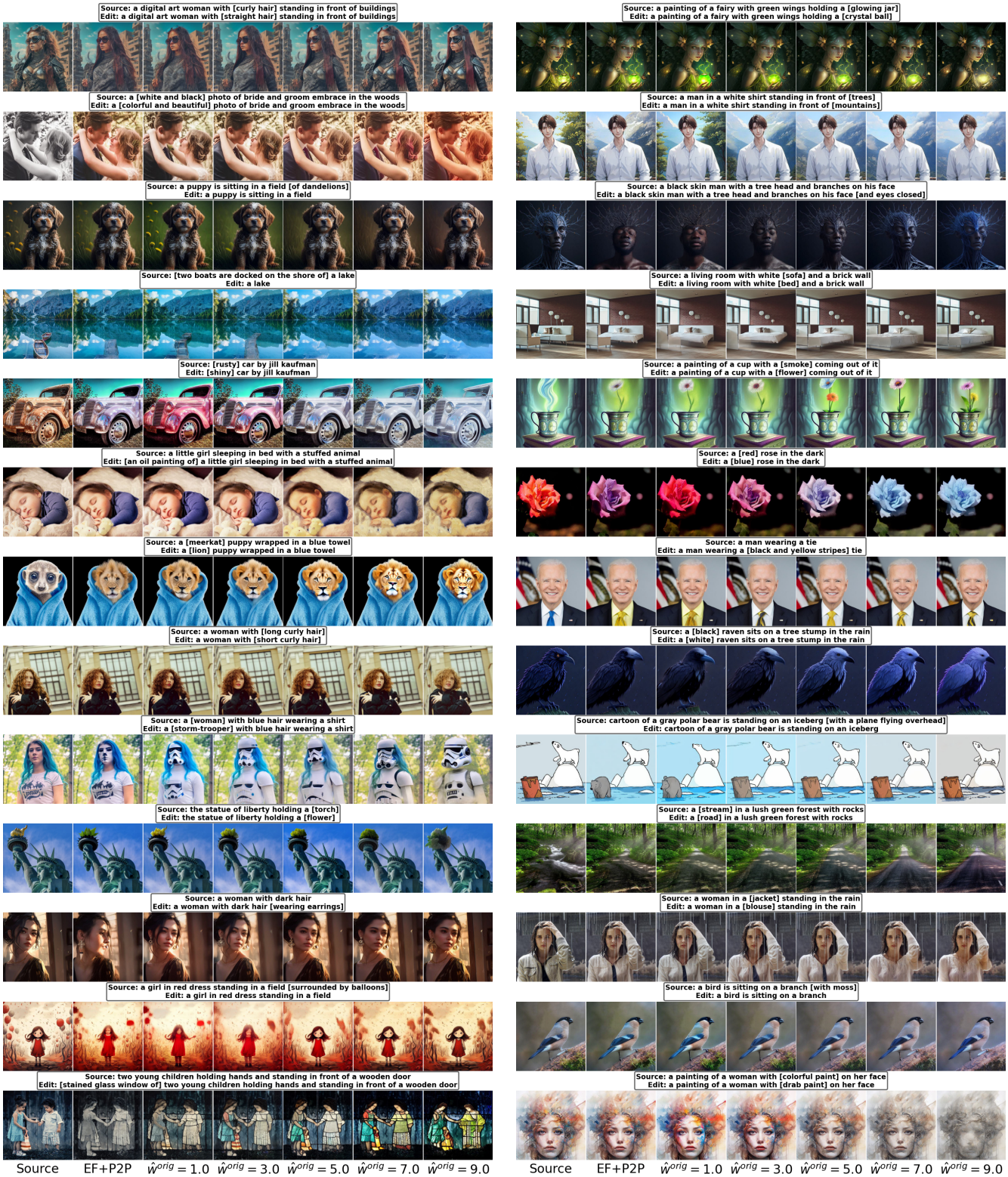


Figure 12. Qualitative results of h-Edit-R + P2P when varying $\hat{w}^{\text{orig}}$ from 1.0 to 9.0 while keeping w^{edit} and w^{orig} fixed at 7.5 and 1.0, respectively. Increasing $\hat{w}^{\text{orig}}$ to an appropriate value improves both editing accuracy and fidelity.

Figure 13. Qualitative results of h-Edit-R + P2P when varying $\left(w^{\text{edit}}, \hat{w}^{\text{orig}}\right)$ within $\{(7.5, 3.0),\ (7.5, 5.0),\ (10.0, 7.0),\ (10.0, 9.0),\ (12.5, 9.0)\ (12.5, 11.0)\}$. Higher w^{edit} values effectively handle challenging edits but may compromise reconstruction quality.

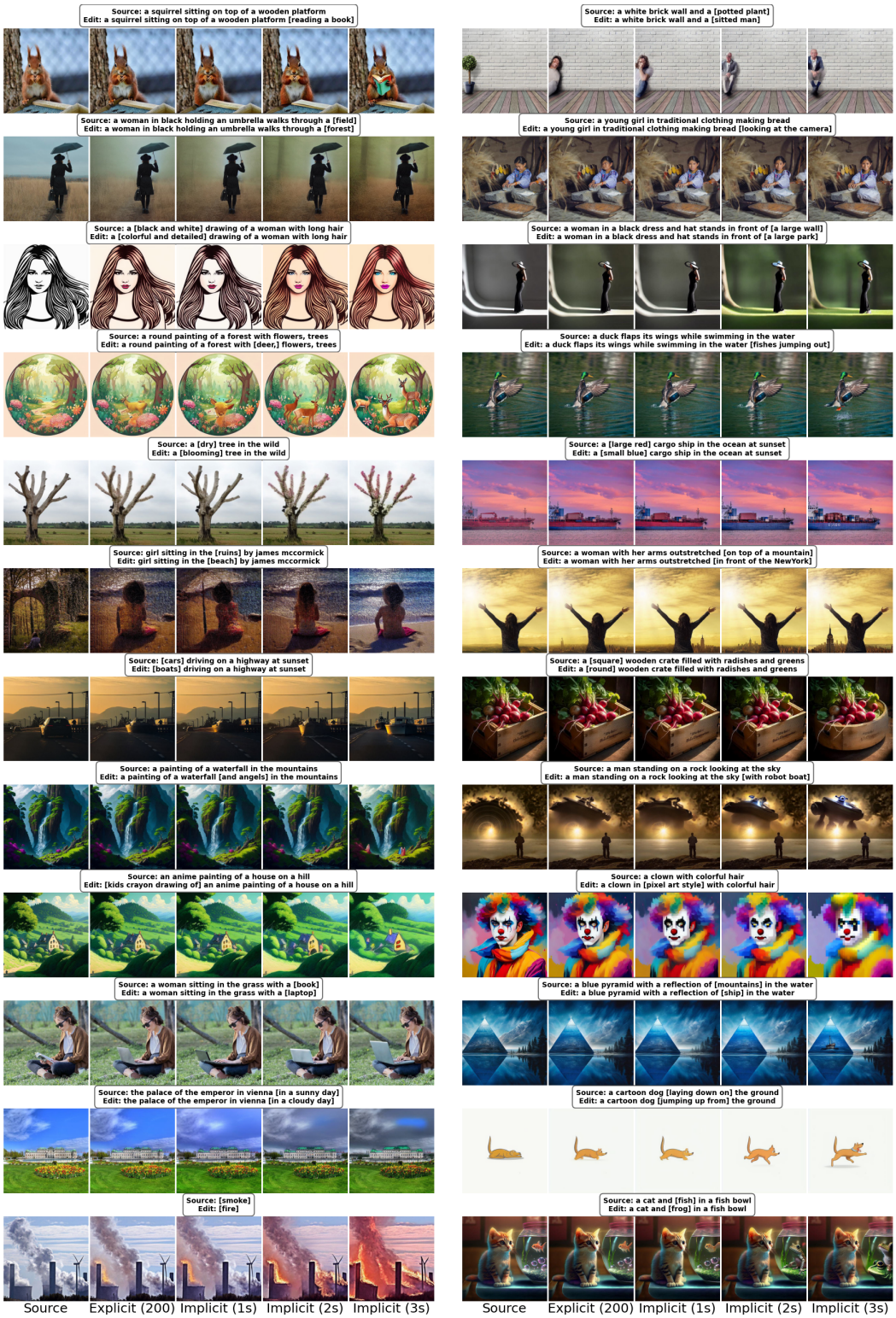


Figure 14. Qualitative examples of implicit *h*-Edit-R + P2P with 50 sampling steps using one, two and three optimization steps (1s/2s/3s), compared to its explicit counterpart with 200 sampling steps. More optimization steps effectively handle challenging cases, outperforming increased sampling steps in the explicit form.

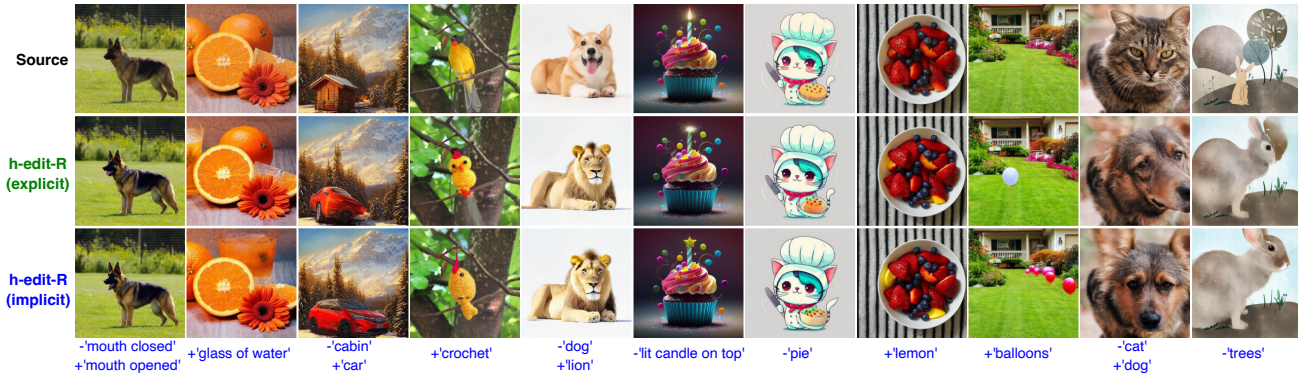


Figure 15. Qualitative visualizations comparing the explicit and implicit versions of h-Edit-R with 25 sampling steps.

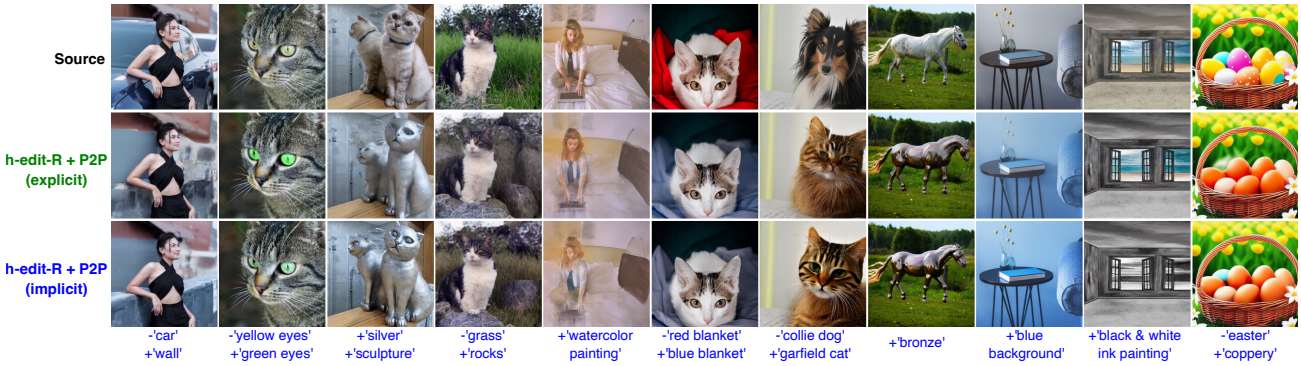


Figure 16. Qualitative visualizations comparing the explicit and implicit versions of h-Edit-R + P2P with 25 sampling steps.

Figure 17. Swapped faces generated by h-Edit-R (3s) with and without masks.

References

[1] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3): 313–326, 1982. 4

[2] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *ICLR*, 2024. 5, 17

[3] Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *CVPR*, pages 8861–8870, 2024. 6, 17

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pages 18392–18402, 2023. 16

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, pages 1877–1901. Curran Associates, Inc., 2020. 16

[6] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *ICCV*, pages 22560–22570, 2023. 6, 21, 22

[7] Hansam Cho, Jonghyun Lee, Seoung Bum Kim, Tae-Hyun Oh, and Yonghyun Jeong. Noise map guidance: Inversion with spatial context for real image editing. In *ICLR*, 2024. 1, 6, 18

[8] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *ICCV*, pages 14367–14376, 2021. 1

[9] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *ICLR*. The International Conference on Learning Representations, 2023. 5, 17

[10] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *NeurIPS*, 34:17695–17709, 2021. 3, 17

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. 7

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 1, 2, 4, 17

[13] Kien Do, Duc Kieu, Toan Nguyen, Dang Nguyen, Hung Le, Dung Nguyen, and Thin Nguyen. Variational flow models: Flowing in your style. *arXiv preprint arXiv:2402.02977*, 2024. 25

[14] Wenkai Dong, Song Xue, Xiaoyue Duan, and Shumin Han. Prompt tuning inversion for text-driven image editing using diffusion models. In *ICCV*, pages 7430–7440, 2023. 3, 6

[15] Joseph L Doob and JI Doob. *Classical potential theory and its probabilistic counterpart*. Springer, 1984. 2, 3, 17

[16] Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. 5, 17

[17] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 16

[18] Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Anastasis Stathopoulos, Xiaoxiao He, Yuxiao Chen, et al. Proxedit: Improving tuning-free real image editing with proximal guidance. In *WACV*, pages 4291–4301, 2024. 6

[19] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2023. 1, 3, 6, 14, 15, 16, 21

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NIPS*, 30, 2017. 7

[21] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1, 2, 17

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 1, 2, 17

[23] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Shifeng Chen, and Liangliang Cao. Diffusion model-based image editing: A survey. *arXiv preprint arXiv:2402.17525*, 2024. 1

[24] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *CVPR*, pages 12469–12478, 2024. 1, 2, 3, 4, 5, 6, 16, 17

[25] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 17

[26] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016. 8

[27] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. *ICLR*, 2024. 1, 2, 3, 4, 5, 6, 17, 18, 21

[28] Tero Karras. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 7

[29] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 7

[30] Jack Karush. On the chapman-kolmogorov equation. *The Annals of Mathematical Statistics*, 32(4):1333–1337, 1961. 11

[31] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *CVPR*, pages 6007–6017, 2023. 5

[32] Duc Kieu, Kien Do, Toan Nguyen, Dang Nguyen, and Thin Nguyen. Bidirectional diffusion bridge models. *arXiv preprint arXiv:2502.09655*, 2025. 3, 11

[33] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, pages 2426–2435, 2022. 5, 6, 16

[34] Kihong Kim, Yunho Kim, Seokju Cho, Junyoung Seo, Jisu Nam, Kychul Lee, Seungryong Kim, and KwangHee Lee. Diffface: Diffusion-based face swapping with facial guidance. *arXiv preprint arXiv:2212.13344*, 2022. 7

[35] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *ICLR*, 2023. 5, 16

[36] Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *CVPR*, pages 1952–1961, 2023. 3

[37] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. Advancing high fidelity identity swapping for forgery detection. In *CVPR*, pages 5074–5083, 2020. 7, 25

[38] Senmao Li, Joost van de Weijer, Taihang Hu, Fahad Shahbaz Khan, Qibin Hou, Yaxing Wang, and Jian Yang. Stylediffusion: Prompt-embedding inversion for text-based editing. *arXiv preprint arXiv:2303.15649*, 2023. 3, 5, 6, 18

[39] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I2sb: image-to-image schrödinger bridge. In *ICML*, pages 22042–22062, 2023. 3, 17

[40] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *ECCV*, pages 423–439. Springer, 2022. 17

[41] Xingchao Liu and Lemeng Wu. Learning diffusion bridges on constrained domains. In *ICLR*, 2023. 3, 17

[42] Xingchao Liu, Lemeng Wu, Mao Ye, and Qiang Liu. Let us build bridges: Understanding and extending diffusion generative models. *arXiv preprint arXiv:2208.14699*, 2022. 3

[43] 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 preprint arXiv:2211.01095*, 2022. 25

[44] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022. 1, 5, 7

[45] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807*, 2023. 6, 18

[46] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, pages 6038–6047, 2023. 1, 2, 3, 5, 6, 18

[47] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, pages 16784–16804. PMLR, 2022. 1

[48] Zhihong Pan, Riccardo Gherardi, Xiufeng Xie, and Stephen Huang. Effective real image editing with accelerated iterative diffusion inversion. In *ICCV*, pages 15912–15921, 2023. 6

[49] Omkar Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*. British Machine Vision Association, 2015. 18

[50] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH*, pages 1–11, 2023. 6

[51] Eckhard Platen Peter E. Kloeden. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, 1992. 4

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 6

[53] Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996. 2, 4

[54] L Chris G Rogers and David Williams. *Diffusions, Markov processes and martingales: Volume 2, Itô calculus*. Cambridge university press, 2000. 2, 3

[55] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2, 4, 5, 6

[56] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 1

[57] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 1

[58] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*. Cambridge University Press, 2019. 2, 3

[59] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 18

[60] Sefik Serengil and Alper Ozpinar. A benchmark of facial recognition pipelines and co-usability performances of modules. *Journal of Information Technologies*, 17(2):95–107, 2024. 18

[61] Sefik Ilkin Serengil and Alper Ozpinar. Lightface: A hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 23–27. IEEE, 2020. 18

[62] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265. PMLR, 2015. 1, 2

[63] Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion schrödinger bridges. In *UAI*, pages 1985–1995. PMLR, 2023. 3, 17

[64] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1, 2, 3

[65] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32, 2019. 1, 2, 17

[66] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 4

[67] Alexander Y Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In *AISTATS*, pages 1279–1287. PMLR, 2024. 3

[68] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. 17

[69] Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *CVPR*, pages 10748–10757, 2022. 6

[70] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, pages 1921–1930, 2023. 1, 6, 21, 22

[71] Truong Vu, Kien Do, Khang Nguyen, and Khoat Than. Face swapping as a simple arithmetic operation. *arXiv preprint arXiv:2211.10812*, 2022. 7, 25

[72] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *CVPR*, pages 22532–22541, 2023. 6

[73] 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. 6

[74] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*, pages 681–688. Citeseer, 2011. 2, 4

[75] Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *ICCV*, pages 7378–7387, 2023. 3

[76] Qiucheng Wu, Yujian Liu, Handong Zhao, Ajinkya Kale, Trung Bui, Tong Yu, Zhe Lin, Yang Zhang, and Shiyu Chang. Uncovering the disentanglement capability in text-to-image diffusion models. In *CVPR*, pages 1900–1910, 2023. 5

[77] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with language-guided diffusion models. In *CVPR*, pages 9452–9461, 2024. 5

[78] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, pages 325–341, 2018. 18

[79] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *ICCV*, pages 23174–23184, 2023. 1, 5, 8, 16, 17, 18, 25

[80] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 1

[81] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6

[82] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *CVPR*, pages 6027–6037, 2023. 5

[83] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. *NeurIPS*, 35:3609–3623, 2022. 6, 17

[84] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023. 25

[85] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *ICLR*, 2024. 3, 17

[86] Yuhao Zhu, Qi Li, Jian Wang, Cheng-Zhong Xu, and Zhenan Sun. One shot face swapping on megapixels. In *CVPR*, pages 4834–4844, 2021. 7, 25