A. Algorithm

We apply our method on a pre-trained Stable Diffusion (SD) model, which contains an encoder $\mathcal{E}$, a decoder $\mathcal{D}$, and a noise predictor ϵ_θ. The full pipeline of our method is depicted by Algorithms 1 to 3. Note the functions Invert(*) and Sample (*) refer to a DDIM inversion step and a DDIM sampling step, respectively, and Att(*) denotes the self-attention mechanism in Stable Diffusion.

Algorithm 1 Overall Framework

Input: Reference texture I^{R}
Output: Output texture I^*

$I^{\mathrm{tar}} \leftarrow \mathrm{USER\_EDIT}(I^{\mathrm{R}})$
$I^{\mathrm{IR}} \leftarrow I^{\mathrm{tar}}$
$z_{\mathrm{T}}^{\mathrm{tar}} \leftarrow \mathrm{StruPreserving\_Inversion}(I^{\mathrm{tar}}, I^{\mathrm{IR}})$
$I^*_{\mathrm{coarse}} \leftarrow \mathrm{FineTexture\_Sampling}(z_{\mathrm{T}}^{\mathrm{tar}}, I^{\mathrm{R}})$
$z_{\mathrm{T}}^* \leftarrow \mathrm{StruPreserving\_Inversion}(I^*_{\mathrm{coarse}}, I^{\mathrm{IR}})$
$I^* \leftarrow \mathrm{FineTexture\_Sampling}(z_{\mathrm{T}}^*, I^{\mathrm{R}})$
Return I^*

Algorithm 2 Structure-preserving Inversion

Input: A target image I^{tar}, an inversion reference I^{IR}
Output: Inversion code $z_{\mathrm{T}}^{\mathrm{tar}}$

$z_0^{\mathrm{IR}} \leftarrow \mathcal{E}(I^{\mathrm{IR}})$
$\{z_0^{\mathrm{IR}}, z_1^{\mathrm{IR}}, \dots, z_{\mathrm{T}}^{\mathrm{IR}}\} \leftarrow \mathrm{DDIM\_INVERSION}(z_0^{\mathrm{IR}})$
$z_0^{\mathrm{tar}} \leftarrow \mathcal{E}(I^{\mathrm{tar}})$
for $t = 0, 1, \dots \mathrm{P} - 1$ do
    $\{Q_{\mathrm{T}-t}^{\mathrm{IR}}, K_{\mathrm{T}-t}^{\mathrm{IR}}, V_{\mathrm{T}-t}^{\mathrm{IR}}\} \leftarrow \epsilon_\theta(z_{\mathrm{T}-t}^{\mathrm{IR}}, t)$
    $\{Q_t^{\mathrm{tar}}, K_t^{\mathrm{tar}}, V_t^{\mathrm{tar}}\} \leftarrow \epsilon_\theta(z_t^{\mathrm{tar}}, t)$
    $\epsilon = \epsilon_\theta(z_t^{\mathrm{tar}}, t) \sim \mathrm{Att}(Q_t^{\mathrm{tar}}, K_{\mathrm{T}-t}^{\mathrm{IR}}, V_{\mathrm{T}-t}^{\mathrm{IR}}))$
    $z_{t+1}^{\mathrm{tar}} \leftarrow \mathrm{Invert}(z_t^{\mathrm{tar}}, \epsilon, t)$
end for
for $t = \mathrm{P}, \mathrm{P}+1, \dots, \mathrm{T}-1$ do
    $\{Q_t^{\mathrm{tar}}, K_t^{\mathrm{tar}}, V_t^{\mathrm{tar}}\} \leftarrow \epsilon_\theta(z_t^{\mathrm{tar}}, t)$
    $\epsilon = \epsilon_\theta(z_t^{\mathrm{tar}}, t) \sim \mathrm{Att}(Q_t^{\mathrm{tar}}, K_t^{\mathrm{tar}}, V_t^{\mathrm{tar}})$
    $z_{t+1}^{\mathrm{tar}} \leftarrow \mathrm{Invert}(z_t^{\mathrm{tar}}, \epsilon, t)$
end for
Return $z_{\mathrm{T}}^{\mathrm{tar}}$

Algorithm 3 Fine-texture Sampling

Input: A start code z_{T}^*, a reference texture I^{R}
Output: Output texture I^*

$z_0^{\mathrm{R}} \leftarrow \mathcal{E}(I^{\mathrm{R}})$
$\{z_0^{\mathrm{R}}, z_1^{\mathrm{R}}, \dots, z_{\mathrm{T}}^{\mathrm{R}}\} \leftarrow \mathrm{DDIM\_INVERSION}(z_0^{\mathrm{R}})$
for $t = \mathrm{T}, \mathrm{T}-1, \dots, \mathrm{T}-\mathrm{S}-1$ do
    $\{Q_t^*, K_t^*, V_t^*\} \leftarrow \epsilon_\theta(z_t^*, t)$
    $\epsilon = \epsilon_\theta(z_t^*, t) \sim \mathrm{Att}(Q_t^*, K_t^*, V_t^*)$
    $z_{t-1}^* \leftarrow \mathrm{Sample}(z_t^*, \epsilon, t)$
end for
for $t = \mathrm{T}-\mathrm{S}, \mathrm{T}-\mathrm{S}+1, \dots, 1$ do
    $\{Q_t^{\mathrm{R}}, K_t^{\mathrm{R}}, V_t^{\mathrm{R}}\} \leftarrow \epsilon_\theta(z_t^{\mathrm{R}}, t)$
    $\{Q_t^*, K_t^*, V_t^*\} \leftarrow \epsilon_\theta(z_t^*, t)$
    $\epsilon = \epsilon_\theta(z_t^*, t) \sim \mathrm{Att}(Q_t^*, K_t^{\mathrm{R}}, V_t^{\mathrm{R}})$
    $z_{t-1}^* \leftarrow \mathrm{Sample}(z_t^*, \epsilon, t)$
end for
$I^* \leftarrow \mathcal{D}(z_0^*)$
Return I^*