7. Proofs for the Theorem

In this section, we show the proof for the proposed equations in the main paper;
First, the following equation holds when $\lambda_f = 1, \forall f \in \mathbb{F}_l \cup \mathbb{F}_h$:

$$\mathcal{L}_2(I_1, I_2) = \sum_{f \in \mathbb{F}_l \cup \mathbb{F}_h} \lambda_f \mathcal{L}_{2,f}(I_1, I_2). \tag{9}$$

Proof. Let $I_1 = (a_{ij}) \in \mathbb{R}^{m \times n}$, and $I_2 = (b_{ij}) \in \mathbb{R}^{m \times n}$.
And for the simplicity of the notation, we define $c_{i,j} = a_{i,j} - b_{i,j}$.
Then,

$$\mathcal{L}_2(I_1, I_2) = \frac{1}{mn} \sum_{i \in \{0,1,\dots m\}} \sum_{j \in \{0,1,\dots n\}} (c_{ij})^2.$$

Let $m' = [\frac{m}{2}], n' = [\frac{n}{2}]$. Then, we can denote $\mathcal{L}_{2, f \in \mathbb{F}_l \cup \mathbb{F}_h}$ as below;

$$\begin{aligned}
\mathcal{L}_{2,LL}(I_1, I_2) &= \frac{1}{m'n'} \sum_{i \in \{0,1,\dots m'\}} \sum_{j \in \{0,1,\dots n'\}} (c_{2i+1,2j+1} + c_{2i+1,2j} + c_{2i,2j+1} + c_{2i,2j})^2, \\
\mathcal{L}_{2,LH}(I_1, I_2) &= \frac{1}{m'n'} \sum_{i \in \{0,1,\dots m'\}} \sum_{j \in \{0,1,\dots n'\}} (-c_{2i+1,2j+1} - c_{2i+1,2j} + c_{2i,2j+1} + c_{2i,2j})^2, \\
\mathcal{L}_{2,HL}(I_1, I_2) &= \frac{1}{m'n'} \sum_{i \in \{0,1,\dots m'\}} \sum_{j \in \{0,1,\dots n'\}} (-c_{2i+1,2j+1} + c_{2i+1,2j} - c_{2i,2j+1} + c_{2i,2j})^2, \\
\mathcal{L}_{2,HH}(I_1, I_2) &= \frac{1}{m'n'} \sum_{i \in \{0,1,\dots m'\}} \sum_{j \in \{0,1,\dots n'\}} (c_{2i+1,2j+1} - c_{2i+1,2j} - c_{2i,2j+1} + c_{2i,2j})^2.
\end{aligned}$$

We can rewrite $\mathcal{L}_2(I_1, I_2)$ as;

$$\mathcal{L}_2(I_1, I_2) = \frac{4}{m'n'} \sum_{i \in \{0,1,\dots m'\}} \sum_{j \in \{0,1,\dots n'\}} c_{2i+1,2j+1}^2 + c_{2i+1,2j}^2 + c_{2i,2j+1}^2 + c_{2i,2j}^2.$$

We use the following identical equation, which holds for $\forall x, y, z, w \in \mathbb{R}$;

$$(x+y+z+w)^2 + (-x-y+z+w)^2 + (-x+y-z+w)^2 + (x-y-z+w)^2 = 4(x^2+y^2+z^2+w^2).$$

We can obtain the following;

$$\sum_{f \in \mathbb{F}_l \cup \mathbb{F}_h} \mathcal{L}_{2,f}(I_1, I_2) = \frac{1}{m'n'} \sum_{i \in \{0,1,\dots m'\}} \sum_{j \in \{0,1,\dots n'\}} 4(c_{2i+1,2j+1}^2 + c_{2i+1,2j}^2 + c_{2i,2j+1}^2 + c_{2i,2j}^2).$$

$$\therefore \mathcal{L}_2(I_1, I_2) = \sum_{f \in \mathbb{F}_l \cup \mathbb{F}_h} 1 \cdot \mathcal{L}_{2,f}(I_1, I_2).$$

□

Second, When the distributions of pixel-wise differences between I_1 and I_2 are i.i.d., and follow $\mathcal{N}(\mu, \sigma^2)$ with $\mu \approx 0$, the following equation holds when $\lambda_f = 1, \forall f \in \mathbb{F}_l \cup \mathbb{F}_h$:

$$4 \log \mathbb{E}[\mathcal{L}_1(I_1, I_2)] + C = \sum_{f \in \mathbb{F}_l \cup \mathbb{F}_h} \lambda_f \log \mathbb{E}[\mathcal{L}_{1,f}(I_1, I_2)], \tag{10}$$

where C is a constant.

Proof. Similar with proving equation 9, we can derive followings;

$$\mathcal{L}_1(I_1, I_2) = \frac{4}{m'n'} \sum_{i\in\{0,1,\dots m'\}} \sum_{j\in\{0,1,\dots n'\}} |c_{2i+1,2j+1}| + |c_{2i+1,2j}| + |c_{2i,2j+1}| + |c_{2i,2j}|,$$

$$\mathcal{L}_{1,LL}(I_1, I_2) = \frac{1}{m'n'} \sum_{i\in\{0,1,\dots m'\}} \sum_{j\in\{0,1,\dots n'\}} |c_{2i+1,2j+1} + c_{2i+1,2j} + c_{2i,2j+1} + c_{2i,2j}|,$$

$$\mathcal{L}_{1,LH}(I_1, I_2) = \frac{1}{m'n'} \sum_{i\in\{0,1,\dots m'\}} \sum_{j\in\{0,1,\dots n'\}} |-c_{2i+1,2j+1} - c_{2i+1,2j} + c_{2i,2j+1} + c_{2i,2j}|,$$

$$\mathcal{L}_{1,HL}(I_1, I_2) = \frac{1}{m'n'} \sum_{i\in\{0,1,\dots m'\}} \sum_{j\in\{0,1,\dots n'\}} |-c_{2i+1,2j+1} + c_{2i+1,2j} - c_{2i,2j+1} + c_{2i,2j}|,$$

$$\mathcal{L}_{1,HH}(I_1, I_2) = \frac{1}{m'n'} \sum_{i\in\{0,1,\dots m'\}} \sum_{j\in\{0,1,\dots n'\}} |c_{2i+1,2j+1} - c_{2i+1,2j} - c_{2i,2j+1} + c_{2i,2j}|.$$

Using $c_{i,j} \sim \mathcal{N}(\mu, \sigma^2)$, we can obtain followings:

$$(c_{2i+1,2j+1} + c_{2i+1,2j} + c_{2i,2j+1} + c_{2i,2j}) \sim \mathcal{N}(4\mu, 4\sigma^2),$$

$$(-c_{2i+1,2j+1} - c_{2i+1,2j} + c_{2i,2j+1} + c_{2i,2j}) \sim \mathcal{N}(0, 4\sigma^2),$$

$$(-c_{2i+1,2j+1} + c_{2i+1,2j} - c_{2i,2j+1} + c_{2i,2j}) \sim \mathcal{N}(0, 4\sigma^2),$$

$$(c_{2i+1,2j+1} - c_{2i+1,2j} - c_{2i,2j+1} + c_{2i,2j}) \sim \mathcal{N}(0, 4\sigma^2).$$

According to the properties of half-normal distribution, for $p \sim \mathcal{N}(\mu, \sigma^2)$,

$$\mathbb{E}[|p|] = \sigma\sqrt{\frac{2}{\pi}} e^{-\frac{\mu^2}{2\sigma^2}} + \mu \cdot \text{erf}(\frac{\mu}{\sqrt{(2\sigma^2)}}), \text{ where erf}(x) = \int_0^x e^{-t^2} dt.$$

Consequently,

$$\mathbb{E}[|c_{2i+1,2j+1}| + |c_{2i+1,2j}| + |c_{2i,2j+1}| + |c_{2i,2j}|] = 4\sigma\sqrt{\frac{2}{\pi}} e^{-\frac{\mu^2}{2\sigma^2}} + 4\mu \cdot \text{erf}(\frac{\mu}{\sqrt{(2\sigma^2)}}),$$

$$\mathbb{E}[|c_{2i+1,2j+1} + c_{2i+1,2j} + c_{2i,2j+1} + c_{2i,2j}|] = 2\sigma\sqrt{\frac{2}{\pi}} e^{-4\cdot\frac{\mu^2}{2\sigma^2}} + 4\mu \cdot \text{erf}(\frac{2\mu}{\sqrt{(2\sigma^2)}}),$$

$$\mathbb{E}[|-c_{2i+1,2j+1} - c_{2i+1,2j} + c_{2i,2j+1} + c_{2i,2j}|] = 2\sigma\sqrt{\frac{2}{\pi}},$$

$$\mathbb{E}[|-c_{2i+1,2j+1} + c_{2i+1,2j} - c_{2i,2j+1} + c_{2i,2j}|] = 2\sigma\sqrt{\frac{2}{\pi}},$$

$$\mathbb{E}[|c_{2i+1,2j+1} - c_{2i+1,2j} - c_{2i,2j+1} + c_{2i,2j}|] = 2\sigma\sqrt{\frac{2}{\pi}}.$$

Using the condition that $c_{i,j}$s are *i.i.d.*,

$$\mathbb{E}[\mathcal{L}_1(I_1, I_2)] = \frac{4}{m'n'} \cdot m'n' \cdot (4\sigma\sqrt{\frac{2}{\pi}} e^{-\frac{\mu^2}{2\sigma^2}} + 4\mu \cdot \text{erf}(\frac{2\mu}{\sqrt{2\sigma^2}}))$$

$$= 16\sigma\sqrt{\frac{2}{\pi}} e^{-\frac{\mu^2}{2\sigma^2}} + 16\mu \cdot \text{erf}(\frac{2\mu}{\sqrt{2\sigma^2}}),$$

$$\mathbb{E}[\mathcal{L}_{1,LL}(I_1, I_2)] = \frac{1}{m'n'} \cdot m'n' \cdot (2\sigma\sqrt{\frac{2}{\pi}} e^{-4\cdot\frac{\mu^2}{2\sigma^2}} + 4\mu \cdot \text{erf}(\frac{2\mu}{\sqrt{(2\sigma^2)}}))$$

$$= 2\sigma\sqrt{\frac{2}{\pi}} e^{-4\cdot\frac{\mu^2}{2\sigma^2}} + 4\mu \cdot \text{erf}(\frac{\mu}{\sqrt{(2\sigma^2)}}),$$

$$\mathbb{E}[\mathcal{L}_{1,LH}(I_1, I_2)] = \mathbb{E}[\mathcal{L}_{1,HL}(I_1, I_2)] = \mathbb{E}[\mathcal{L}_{1,HH}(I_1, I_2)] = \frac{1}{m'n'} \cdot m'n' \cdot 2\sigma\sqrt{\frac{2}{\pi}} = 2\sigma\sqrt{\frac{2}{\pi}}.$$

Since $\mu \approx 0$, $\mu \cdot \text{erf}(\frac{\mu}{\sqrt{2\sigma^2}}) \approx 0$. Consequently,

$$\log \mathbb{E}[\mathcal{L}_1(I_1, I_2)] = \log 16 + \log(\sigma\sqrt{\frac{2}{\pi}}) - \frac{\mu^2}{2\sigma^2},$$

$$\log \mathbb{E}[\mathcal{L}_{1,LL}(I_1, I_2)] = \log 2 + \log(\sigma\sqrt{\frac{2}{\pi}}) - 4 \cdot \frac{\mu^2}{2\sigma^2},$$

$$\log \mathbb{E}[\mathcal{L}_{1,LH}(I_1, I_2)] = \log \mathbb{E}[\mathcal{L}_{1,HL}(I_1, I_2)] = \log \mathbb{E}[\mathcal{L}_{1,HH}(I_1, I_2)] = \log 2 + \log(\sigma\sqrt{\frac{2}{\pi}}).$$

$$\log \mathbb{E}[\mathcal{L}_1(I_1, I_2)] = \sum_{f \in \mathbb{F}_l \cup \mathbb{F}_h} \frac{1}{4} \log \mathbb{E}[\mathcal{L}_{1,f}(I_1, I_2)] + C'.$$

$$\therefore 4 \log \mathbb{E}[\mathcal{L}_1(I_1, I_2)] = \sum_{f \in \mathbb{F}_l \cup \mathbb{F}_h} \log \mathbb{E}[\mathcal{L}_{1,f}(I_1, I_2)] + C.$$

□

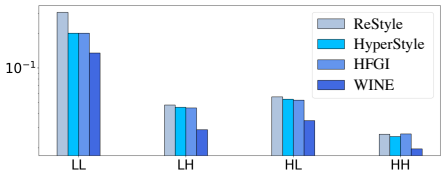


Figure 8. **Comparison of $\mathcal{L}_1$ of the wavelet coefficients.** We plot the average $\mathcal{L}_1$ of each wavelet coefficient between CelebA-HQ test images and corresponding inverted images by various state-of-the-art inversion models. Due to the significant gap between $\mathcal{L}_{1,LL}$ and the rest (about 20 times in linear scale), we display the losses with the logarithmic scale for better visualization. In contrast to other high-rate baseline inversion methods, *e.g.*, HyperStyle and HFGI, WINE notably reduces distortion on high-frequency sub-bands.

Similar to $\mathcal{L}_2$, $\mathcal{L}_1$ seems a fair loss without the frequency bias, which reflects $\mathcal{L}_{1,f \in \mathbb{F}_l \cup \mathbb{F}_h}$ with same weights. However, as shown in Figure 8, we empirically find that $\mathcal{L}_{1,LL}$ is around 20 times larger than $\mathcal{L}_{1,f \in \mathbb{F}_h}$ in case of HyperStyle and HFGI. This leads to the biased training, which results in an apparent decrease of $\mathcal{L}_{1,LL}$, but almost no gain, or even increment of $\mathcal{L}_{1,f \in \mathbb{F}_h}$, compared to state-of-the-art low-rate inversion method, *i.e.*, ReStyle. Consequently, we argue that $\mathcal{L}_1$ contains the low-frequency bias, and needs the wavelet loss to avoid it.

7.1. Information in Sub-band of Images

In Section 3.2, we designed the multi-level wavelet loss to cover broader frequency ranges than $f_{nyq}/2 \sim f_{nyq}$. In Figure 9, we show the results of the inverse wavelet transform by omitting the wavelet coefficients between $f_{nyq}/2 \sim f_{nyq}$ (Config A), $f_{nyq}/2^2 \sim f_{nyq}/2$ (Config B), and $f_{nyq}/2^3 \sim f_{nyq}/2^2$ (Config C). Though A removes the highest frequency sub-bands, *i.e.*, $f_{nyq}/2 \sim f_{nyq}$, among all configs, we cannot find visible degradation of image details. In other words, information in the sub-band $f_{nyq}/2 \sim f_{nyq}$ is mostly higher than the visible image details. Since the firstly proposed wavelet loss (Equation 3 of the main paper) only covers the sub-band $f_{nyq}/2 \sim f_{nyq}$, we should extend the range of sub-bands to effectively preserve the visible details.

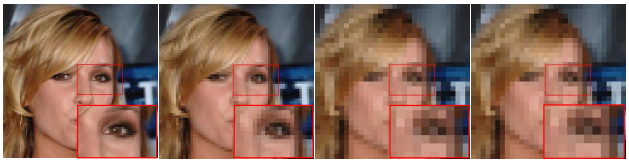


Figure 9. **Inverse wavelet transform results by omitting various wavelet sub-bands.** To check the qualitative image details in each sub-band, we remove the wavelet coefficients between $f_{nyq}/2 \sim f_{nyq}$ (Config A), $f_{nyq}/2^2 \sim f_{nyq}/2$ (Config B), and $f_{nyq}/2^3 \sim f_{nyq}/2^2$ (Config C). From A to B, severe degradation of visible image details does not occur. However, for B to C or C to D, the majority of image details are degraded.

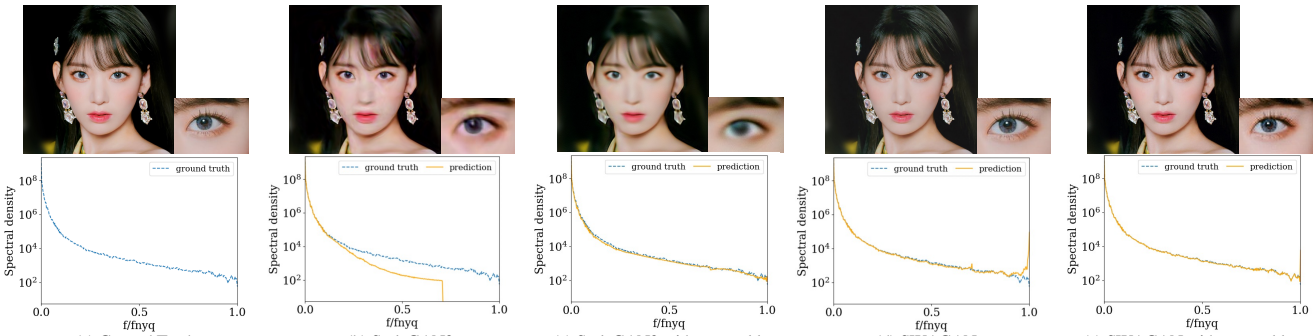
Consequently, we propose a K-level wavelet loss, which enables covering the sub-band $f_{nyq}/2^K \sim f_{nyq}$.

8. Generator Training with Spectral Loss

Previous works [9,15,34] propose an objective function to precisely learn the frequency distribution of the training data, which we comprehensively named as *spectral loss*. [15] designed a spectral loss function that measures the distance between fake and real images in the frequency domain that captures both amplitude and phase information. [9] proposed a spectral loss that measures the binary cross entropy between the azimuthal integration over the power spectrum of fake and real images. [34] used a simple $\mathcal{L}_2$ loss between the logarithm of the azimuthal average over power spectrum in normalized polar coordinates, *i.e.* reduced spectrum, of fake and real images. We adopted the spectral loss term of [34] for our experiment :

$$\mathcal{L}_S = \frac{1}{H/\sqrt{2}} \sum_{k=0}^{H/\sqrt{2}-1} \| \log(\tilde{S}(G(z)))[k] - \log(\tilde{S}(\mathbf{I}))[k] \|_2^2, \tag{11}$$

where $\tilde{S}$ is the reduced spectrum, $G(z)$ is the generated image, and $\mathbf{I}$ is the ground truth real image.



(a) Ground Truth (b) StyleGAN2 (c) StyleGAN2 with spectral loss (d) SWAGAN (e) SWAGAN with spectral loss

Figure 10. **Regression (top row) and spectral density plot (bottom row) of ground truth image and generated images trained with/without additional spectral loss.** Here, we used the spectral loss introduced in [34]. For both StyleGAN2 and SWAGAN generators, the additional spectral loss induced artifacts to coercively match the frequency distribution. We recommend you zoom in to carefully observe the reconstructed details.

Here, we conducted a single-image reconstruction task, which is widely done [11, 34] to investigate the effectiveness of explicit frequency matching in refining high-fidelity details. For StyleGAN2 [19] and SWAGAN [11] generator, we used the latent optimization [18] method to reconstruct a single image, each with and without the spectral loss. All images are generated to resolution 512×512, with the weight of spectrum loss $\times 0.1$ of the original $\mathcal{L}_2$ loss.

Figure 10 shows the reconstructed images and spectral density plots for each case. As seen in Figure 10(a), the spectrum of a natural image follows an exponential decay. Using $\mathcal{L}_2$ singularly made both StyleGAN2 and SWAGAN generators overfit to the mostly existing low-frequency distribution. (b) StyleGAN2 struggled to learn the high-fidelity details, creating an unrealistic image. (d) SWAGAN was capable of fitting most of the high-frequency parts, except created some excessive high-frequency noise due to checkerboard patterns. Though utilizing the spectral loss for both generators (c,e) exquisitely matched all frequency distributions, qualitative results were degraded. Matching the frequency induced unwanted artifacts to the images, and caused the degradation. Due to the absence of the spatial information, the loss based on the spectral density inherently cannot reconstruct high-frequency details. Comparably, our wavelet loss minimizes the $\mathcal{L}_1$ distance of high-frequency bands in the spatial frequency domain, restoring meaningful high-fidelity features.

9. Experimental Details

9.1. Training Details

In our experiments, we implement our experiments based on the pytorch-version code [2] for SWAGAN [11]. We converted the weights of pre-trained SWAGAN generator checkpoint from the official TensorFlow code[3] to pytorch version. We trained our model on a single GPU and took only 6 hours for the validation loss to saturate, whereas other StyleGAN2-based baselines required more than 2 days of training time.

[2]https://github.com/rosinality/stylegan2-pytorch
[3]https://github.com/rinongal/swagan

$\lambda_{wave,ADA}$	0	0.01	0.05	**0.1**	0.5
$\mathcal{L}_2 \downarrow$	0.024	0.017	0.15	**0.011**	0.021
$\mathcal{L}_1, wave \downarrow$	0.274	0.249	0.243	**0.230**	0.248
SSIM $\uparrow$	0.717	0.724	0.730	**0.753**	0.719

Table 3. Quantitative comparison with various wavelet loss ratio.

Here, we explain the details of our reconstruction loss terms: $\mathcal{L}_2$, $\mathcal{L}_{id}$, and $\mathcal{L}_{LPIPS}$. We leverage $\mathcal{L}_2$, as it is most effective in keeping the generated image similar to the original image pixel-wise. $\mathcal{L}_{id}$ is an identity loss defined as:

$$\mathcal{L}_{id} = 1 - < R(G_0(w)), R(\mathbf{I}) >, \tag{12}$$

where R is the pre-trained ArcFace [8] model, and $\mathbf{I}$ is the ground truth image. $\mathcal{L}_{id}$ minimizes the cosine distance between two face images to preserve the identity. LPIPS [42] enhances the perceptual quality of the image by minimizing the distance on the feature space of ImageNet [7] pre-trained network. For training, we used weights $\lambda_{\mathcal{L}_2}$=1, λ_{id}=0.1, λ_{LPIPS}=0.8, respectively, which follows the widely adopted experimental setups in previous GAN inversion methods.

In Table 3, we show the effect of our proposed wavelet loss via adjusting the weight $\lambda_{wave,ADA}$ respective to the weight $\lambda_{\mathcal{L}_1}$ in Eq. 6 of the main paper. Reminder that the ADA loss aims to minimize the discrepancy in residual wavelet features. Increasing the weight $\lambda_{wave,ADA}$ up to 0.1 shows that incorporating the wavelet loss effectively enhances the reconstruction of image-wise details, particularly in high-frequency regions. However, exceeding a weight of 0.1 leads to a decline in performance, as most image information resides in the low-frequency sub-band. In general, we applied balanced weights that effectively reconstruct high-frequency sub-bands without compromising the generation of low-frequency sub-bands.

9.2. Dataset Description

In this section, we describe the datasets used for experiments in the main paper.

Flickr-Faces-HQ (FFHQ) dataset. Our model and all baselines are trained with FFHQ [18], a well-aligned human face

dataset with 70,000 images of resolution 1024 × 1024. FFHQ dataset is widely used for training various unconditional generators [17–19], and GAN inversion models [3, 4, 27, 32, 37, 39]. All of the baselines we used in the paper use the FFHQ dataset for training, which enables a fair comparison.

CelebA-HQ dataset. CelebA-HQ dataset contains 30,000 human facial images of resolution 1024 × 1024, together with the segmentation masks. Among 30,000 images, around 2,800 images are denoted as the test dataset. We use the official split for the test dataset, and evaluate every baseline and our model with all images in the test dataset.

Animal-Faces-HQ (AFHQ) dataset. AFHQ [5] dataset contains 15,000 high-quality images of cats, dogs, and wildlife animals at 512 × 512 resolution. We used 5000 images of wild animals for training WINE, and the test-set for evaluation.

9.3. Baseline Descriptions

In this section, we describe the existing GAN inversion baselines, which we used for comparison in Section 4. We exclude the model which needs image-wise optimization, such as Image2StyleGAN [1] or Pivotal Tuning [33].

pSp pixel2Style2pixel (pSp) adopts pyramid [22] network for the encoder-based GAN inversion. pSp achieves the state-of-the-art performance among encoder-based inversion models at the time. Moreover, pSp shows the various adaptation of the encoder model to the various tasks using StyleGAN, such as image inpainting, face frontalization, or super-resolution.

e4e encoder4editing ($e4e$) proposes the existence of the trade-off between distortion and the perception-editability of the image inversion. In the other words, $e4e$ proposes that the existing GAN inversion models which focus on lowering distortion, sacrifice the perceptual quality of inverted images, and the robustness on the editing scenario. $e4e$ suggests that maintaining the latent close to the original StyleGAN latent space, *i.e.*, W, enables the inverted image to have high perceptual quality and editability. To this end, $e4e$ proposes additional training loss terms to keep the latent close to W space. Though distortion of pSp is lower than $e4e$, $e4e$ shows apparently higher perceptual quality and editabilty than pSp.

ReStyle ReStyle suggests that a single feed-forward operation of existing encoder-based GAN inversion models, *i.e.*, pSp and $e4e$, is not enough to utilize every detail in the image. To overcome this, ReStyle proposes an iterative refinement scheme, which infers the latent with feed-forward-based iterative calculation. The lowest distortion that Restyle achieves among encoder-based GAN inversion models shows the effectiveness of the iterative refinement scheme. Moreover, the iterative refinement scheme can be adapted to both pSp and $e4e$, which enables constructing models that have strengths in lowering distortion, or high perceptual quality-editability, respectively. To the best of our knowledge, ReStyle$_{pSp}$ achieves the lowest distortion among encoder-based models which do not use generator-tuning method[4]. Since we utilize baselines that achieve lower distortion than ReStyle$_{pSp}$, *i.e.*, HyperStyle and HFGI, we only use ReStyle$_{e4e}$ to evaluate its high editability.

[4]IntereStyle [27] achieves lower distortion on the *interest region* than ReStyle$_{pSp}$, but not for the whole image region.

HyperStyle To make a further improvement from ReStyle, Pivotal Tuning [33] uses the input-wise generator tuning. However, this is extremely time-consuming, and inconvenient in that it requires separate generators per every input image. To overcome this, HyperStyle adopts HyperNetwork [13], which enables tuning the convolutional weights of pre-trained StyleGAN only with the feed-forward calculation. Starting from the latent obtained by $e4e$, HyperStyle iteratively refines the generator to reconstruct the original image with the fixed latent. HyperStyle achieves the lowest distortion among encoder-based GAN inversion models at the time.

HFGI HFGI points out the limitation of the low-rate inversion methods and argues that encoders should adopt larger dimensions of tensors to transfer high-fidelity image-wise details. To achieve this, HFGI adapts feature fusion, which enables mixing the original StyleGAN feature with the feature obtained by the image-wise details.

StyleRes StyleRes handles the trade-off between the reconstruction and editing quality of real images. In order to obtain high-quality editing in high-rate latent spaces, StyleRes learns residual features in higher latent codes and how to transform these residual features to adapt to latent code manipulations. StyleRes achieves the lowest distortion among every GAN inversion method, except our model.

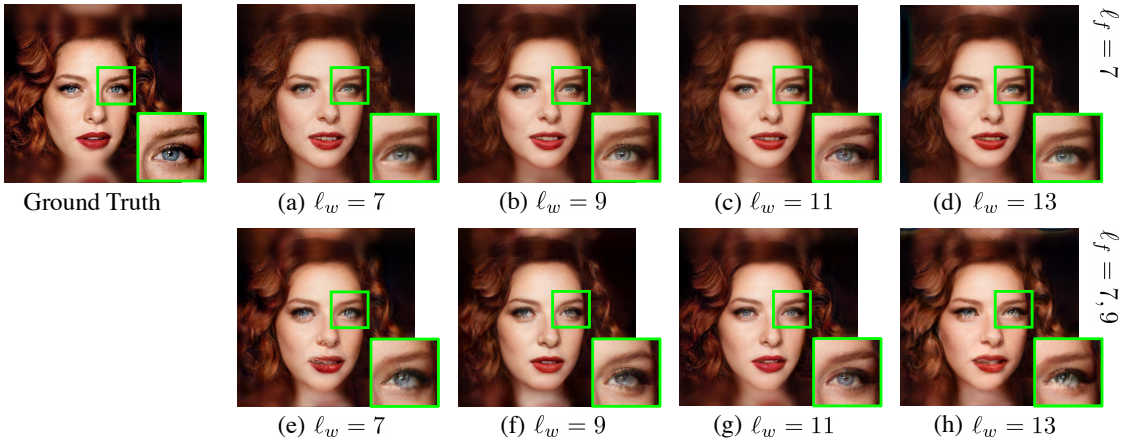


Figure 11. **Qualitative Comparison of WINE Inversion with Fusion in Different Layers.** Each image represents the inversion results for each scenario in Table 5. The first row (a)-(d) displays inverted images with feature fusion in a single layer $\ell_f = 7$, with wavelet fusion in layer $\ell_w = 7$, $\ell_w = 9$, $\ell_w = 11$, and $\ell_w = 13$, respectively. The second row (e)-(h) displays inverted images with feature fusion in multi-layers $\ell_f = 7$ and 9, with wavelet fusion in layer $\ell_w = 7$, $\ell_w = 9$, $\ell_w = 11$, and $\ell_w = 13$, respectively. We recommend you zoom in for a careful look into the details.

	pSp	e4e	ReStyle	HS	HFGI	StyleRes	**Ours**
Smile	21.66	21.51	13.87	15.65	15.17	17.03	**14.50**
Gender	26.45	29.31	26.14	19.92	19.24	19.90	**19.46**
Lipstick	40.88	40.27	34.27	33.45	31.30	**28.53**	31.21
Average ↓	29.66	30.36	24.76	23.01	21.91	21.82	**21.72**

Table 4. Quantitative comparison on editability.

9.4. Quantitative Comparison on Editability

Recently, StyleRes [30] proposed a method to quantitatively measure editability using FID (Frechet Inception Distance), based on image distributions from the CelebA-HQ annotation dataset. Specifically, after selecting a feature to edit, the Inception network output distribution of real images that possess the desired feature (positive) is computed. Then, real images that do not have the feature (negative) are edited, and the Inception network output distribution of the resulting fake images is obtained. The underlying idea is that the more realistic the edited images are and the better the feature is reflected, the smaller the distance between these two distributions will be. In Table 4, we compared FID editability related to three features, *e.g.*, smile, gender, and lipstick. In two features out of three, ours showed the lowest FID and the second lowest in the rest feature. Overall, we compared the average FID, where ours showed the lowest score out of every baseline.

10. Ablation Studies

10.1. Choice of Fusion Layer

We additionally provide both quantitative and qualitative ablation results for the inversion performance of WINE with fusion in different layers. Note that in our main experiment, we apply feature fusion in layers $\ell_f = 7$ and 9, and wavelet fusion in layer $\ell_w = 11$. Each layer corresponds to a fusion of spatial features with resolution 64×64 and 128×128, and wavelet coefficients of dimension $w \in \mathbb{R}^{12\times128\times128}$.

From the quantitative results in Table 5, we observed that the feature fusion on two layers $\ell_f = 7$ and 9 showed better reconstruction accuracy than on a single layer $\ell_f = 7$. Additionally, wavelet fusion in lower layers ($\ell_w < 11$) was not sufficient enough to preserve the high-fidelity details, especially in the high-frequency region *i.e.* $\mathcal{L}_{wave}$. Wavelet fusion in the higher layer ($\ell_w = 13$) also degraded the inversion performance, which can be more carefully observed in Figure 11.

Figure 11 shows the inverted images for each scenario in Table 5. It is noticeable that fusion in a single layer (a)-(d) failed to retain high-frequency details like the hand and hair texture. Comparably, in the case of multi-layer feature fusion (e)-(h), inverted images reconstructed more high-frequency details. Yet, wavelet fusion in the lower layers (e), (g), and higher layers (h) generated unwanted distortions, which eventually degraded the image fidelity. Overall, our scenario (g) empirically showed the most promising reconstruction quality, generating realistic images with the least distortion.

10.2. Design of Fusion Methods

To prove the effectiveness of the wavelet fusion, we compared the performance of WINE with the model which uses the feature fusion, proposed in HFGI [39], instead of the wavelet fusion in the same resolution layer. In Table 6, we compared the performance of models with the following four settings: The original HFGI which uses feature fusion at $\ell_f = 7$, HFGI with additional feature fusion at $\ell_f = 9$ and 11, WINE with the feature fusion at $\ell_f = 7$, 9, and 11, and the original WINE which uses the feature fusion at $\ell_f = 7$ and 9, and the wavelet fusion at $\ell_w = 11$. First, simply adding the feature fusion to the higher layer is not helpful for improving the model. If we change it to the WINE method, *i.e.*, change the generator and add the wavelet loss, the performance

Table 5. **Ablation of the Fusion Layers for WINE.** We compared the inversion performance of WINE with feature and wavelet fusion in different layers. Feature fusion on layers $\ell_f = 7$ and 9, and wavelet fusion on layer $\ell_w = 11$ consistently showed the highest fidelity and reconstruction quality among all scenarios.

Feature Fusion	Wavelet Fusion	$L_2 \downarrow$	$L_{wave} \downarrow$	LPIPS $\downarrow$	SSIM $\uparrow$	ID sim $\uparrow$
$\ell_f = 7$	$\ell_w = 7$	0.028	0.359	0.365	0.667	0.796
	$\ell_w = 9$	0.026	0.356	0.362	0.701	0.830
	$\ell_w = 11$	0.026	0.325	0.364	0.727	0.847
	$\ell_w = 13$	0.024	0.314	0.366	0.727	0.845
$\ell_f = 7$ and 9	$\ell_w = 7$	0.020	0.327	0.346	0.711	0.849
	$\ell_w = 9$	0.016	0.289	0.330	0.724	0.880
	$\ell_w = 11$	**0.011**	**0.230**	**0.277**	**0.753**	**0.906**
	$\ell_w = 13$	0.020	0.307	0.342	0.722	0.861

Table 6. **Ablation of the Fusion Methods for WINE.** We compared the inversion performance of WINE with the model which uses wavelet fusion instead of feature fusion. Though changing all the fusion methods with the feature fusion achieves better results than HFGI, still it shows a big performance degradation compared to WINE.

Model	Fusion Layers	$L_2 \downarrow$	$L_{wave} \downarrow$	SSIM $\uparrow$	ID sim $\uparrow$
HFGI	$\ell_f = 7$	0.023	0.351	0.661	0.864
	$\ell_f = 7, 9, 11$	0.036	0.377	0.704	0.795
WINE	$\ell_f = 7, 9, 11$	0.017	0.302	0.699	0.873
	$\ell_f = 7, 9$ and $\ell_w = 11$	**0.011**	**0.230**	**0.753**	**0.906**

significantly improves. After changing the feature fusion at the 11^{th} layer, the performance remarkably improved and achieved state-of-the-art results on various metrics.

11. Limitation and Future Work

Our proposed WINE excels in producing high-quality images by efficiently transferring the residual high-frequency information to the generator. However, we only provided empirical results with specifically SWAGAN, a wavelet-based StyleGAN as the generator. As our proposed wavelet fusion pertains to generators with intermediate wavelet coefficients, we can potentially generalize our approach to other wavelet-based generators that provides inversion and editing abilities. We will work on applying our method to the inversion of wavelet-based diffusion models and other wavelet expanded generators in the near future.