A. Limitations

We now discuss limitations, which we have already realized, for our work. First, for the inversion task, although our proposed method achieves improved reconstruction quality, there are still some differences between the input and reconstructed images, especially for the out-of-domain input. We think it is mainly caused by the finite discriminative ability of \( W^+ \) space. As is described in [8], the distortion can be significantly reduced by adding more information from the source. Moreover, since we apply the multi-head attention, the training speed is slower due to the complex matrix multiplication. Second, for the reference-based editing task, we adopt a transformer-based module in the latent space, resulting in less diversity for some attributes compared with direct editing on the images, in which the mode seeking loss [4] can encourage the diversity in the pixel domain. But our method is lightweight and more flexible.

B. Ablations and Analysis

We further validate the benefit of transformer by comparing among pSp [5], our full model with both self- and cross-attention, and ours w/o self-attention in Tab. 1. [5] maps image features to \( W^+ \) by individual mapping networks, though \( W^+ \) obtain the image features directly and completely, the relation between each \( w \) is not tightly enough. In our model, cross-attention is necessary to update queries by fusing image features, and self-attention is also important in constructing the potential relation between queries.

C. Label-based Editing Methods

We propose first- and second-order label-based editing methods in the main text. To give a detailed explanation, we provide the pseudo codes in PyTorch style. Algorithm 1 and Algorithm 2 illustrate the first- and second-order methods, respectively. Moreover, we measure the disentanglement of five attributes by Re-scoring [6] in Fig. 1. The top row lists edited attributes, and the scores are the classification logits changes between original and edited images. Considering human judgements, we further conduct a user study. We ask 60 volunteers to evaluate the methods in two aspects: image quality and disentanglement. Results are shown in Tab. 2.

D. Training Details

We adopt a pretrained StyleGAN2 [1] generator in our experiments, in which the synthesis network is fixed and...
the mapping network (MLP) is trained. In the multi-head attention of the transformer block, the number of heads is set to 4, and the dimension of each head is 512. For inversion task, the Ranger optimizer is used in training, which is a combination of Rectified Adam [3] with the Lookahead technique [10]. We train the model for $6 \times 10^5$ iterations with a batch size of 8, the learning rate is set to $1 \times 10^{-4}$. For the reference-based editing task, we use the Adam [2] optimizer to train the model for $1 \times 10^4$ iterations with a batch size of 8, the learning rate is set to $1 \times 10^{-3}$. All experiments are implemented on 2 NVIDIA RTX 2080Ti GPUs.

E. More Results

In this section, we provide more results of inversion, label-based editing and reference-based editing in Fig. 2, Fig. 3, Fig. 4.

Algorithm 1 First-order Label-based Editing

```python
1 # w: input latent code (18, 512), C: latent classifier, y_t: target label
2 predicted = C(w)
3 loss = torch.nn.BCELoss(predicted, y_t)
4 loss.backward()
5 direct = w.grad
6 direct = direct / torch.norm(direct, dim=1)
7 w_edit = w - alpha * direct # alpha is a scaling factor.
```

Algorithm 2 Second-order Label-based Editing

```python
1 # w: input latent code (18, 512), C: latent classifier, y_t: target label
2 r_d = torch.randn(18, 512)
3 r_0 = torch.zeros(18, 512)
4 w_d = w + kasi * r_d # kasi is a small number, we set it to 10e-4.
5 w_0 = w + r_0
6 predicted_d = C(w_d)
7 loss = torch.nn.BCELoss(predicted_d, y_t)
8 loss.backward()
9 direct_d = r_d.grad
10 C.zero_grad()
11 predicted_0 = C(w_0)
12 loss = torch.nn.BCELoss(predicted_0, y_t)
13 loss.backward
14 direct_0 = r_0.grad
15 direct = direct_d - direct_0
16 direct = direct_d / torch.norm(direct, dim=1)
17 w_edit = w - alpha * direct # alpha is a scaling factor.
```

References


Figure 2. More results of inversion compared with [5] and [7].
Figure 3. More results of label-based editing on five attributes.
Figure 4. More results of reference-based editing on three attributes. The edited images take the style of Bangs, Mouth and Gender from different reference images.