What Decreases Editing Capability? Domain-Specific Hybrid Refinement for Improved GAN Inversion

Pu Cao\textsuperscript{1,2}, Lu Yang\textsuperscript{1}, Dongxv Liu\textsuperscript{1}, Xiaoya Yang\textsuperscript{1}, Tianrui Huang\textsuperscript{1}, Qing Song\textsuperscript{1,*}

\textsuperscript{1}Beijing University of Posts and Telecommunications \textsuperscript{2}Metavatar
\{caopu, soeaver, liudongxv, yangxiaoya, huangtianrui, priv\}@bupt.edu.cn

Abstract

Recently, inversion methods have been exploring the incorporation of additional high-rate information from pre-trained generators (such as weights or intermediate features) to improve the refinement of inversion and editing results from embedded latent codes. While such techniques have shown reasonable improvements in reconstruction, they often lead to a decrease in editing capability, especially when dealing with complex images that contain occlusions, detailed backgrounds, and artifacts. To address this problem, we propose a novel refinement mechanism called Domain-Specific Hybrid Refinement (DHR), which draws on the advantages and disadvantages of two mainstream refinement techniques. We find that the weight modulation can gain favorable editing results but is vulnerable to these complex image areas and feature modulation is efficient at reconstructing. Hence, we divide the image into two domains and process them with these two methods separately. We first propose a Domain-Specific Segmentation module to automatically segment images into in-domain and out-of-domain parts according to their invertibility and editability without additional data annotation, where our hybrid refinement process aims to maintain the editing capability for in-domain areas and improve fidelity for both of them. We achieve this through Hybrid Modulation Refinement, which respectively refines these two domains by weight modulation and feature modulation. Our proposed method is compatible with all latent code embedding methods. Extension experiments demonstrate that our approach achieves state-of-the-art in real image inversion and editing. Code is available at \url{https://github.com/caopulan/Domain-Specific_Hybrid_Refinement_Inversion}.

1. Introduction

Generative Adversarial Networks (GANs) have shown promising results in image generation. Synthetic images are photorealistic with high resolution and are difficult to distinguish from real images \cite{16, 18-20, 42}. Based on their highly semantic latent space, image manipulation and controllable generation are deeply explored. Moreover, GANs can represent a high-quality image prior to improving various tasks, such as face parsing \cite{37-41, 43}, style transfer \cite{21, 44}, and face super-resolution \cite{35}. However, real images are inapplicable for these applications, since most of them require latent codes in GANs' feature space.

Inversion is built to convert real images into GANs’ la-
tent space. The inverted latent codes are required to re-
construct given images by a pretrained generator, which
also embeds semantic information to edit or apply in GAN-
based tasks. Two types of methods generally reach image
embedding. One is training an image encoder to convert
given images to latent codes \cite{25,31}, while another is min-
imizing the discrepancy between given images and recon-
structed images to optimize initial latent codes iteratively
\cite{20}. This process attains the corresponding latent codes
to reconstruct or edit the images. However, latent codes are
low bit-rate \cite{34}, and high-rate details of images may not be
reconstructed faithfully. Hence, many recent works focus
on refining results by modulating additional high-rate infor-
mation, which can be mainly divided into weight modula-
tion and feature modulation. Weight modulation improves
fidelity by iteratively tuning generator’s weight by minimiz-
ing reconstruction error \cite{27} or predicting weight offsets \cite{5}
by Hypernetworks \cite{12}. Feature modulation encodes spatial
details into intermediate features in generator, where the en-
coding process is also reached by iterative optimization \cite{23}
or model prediction \cite{23,34}.

As reconstruction performance increases by refinement
with high-rate information, editing capability is inevitably
decreased, especially on images containing complex parts,
which we show in Figure 3. This phenomenon is due
to the destruction of the generation ability of pretrained
GANs. Supervised by a discriminator, GANs are con-
strained to generate realistic images from sampled latent
codes. However, the additional high-rate information de-
grades the highly semantic latent space of pretrained gen-
erator, making the latent manifold sharp by weight modula-
tion \cite{9} or fixing the spatial distribution by intermediate fea-
tures modulation to overfit the given images to reconstruct.
Particularly, high-rate information needs drastic change to
reconstruct complex parts. Notably, complex images pre-
vail in the natural world. For example, face accessories,
hats, occlusions, and complex backgrounds usually appear
in face photos. Hence, our goal is to design a robust refine-
ment inversion method that can faithfully reconstruct given
images and retain editing capability.

Based on the above illustration, we explore the idea of
“divide and conquer” to address this problem. Specifically,
we divide the image into in-domain and out-of-domain
parts. In-domain parts imply areas close to generators’ out-
put distribution and are desired to perform well on both in-
version and editing. Correspondingly, out-of-domain parts
are areas challenging to inverse or edit and desired to recon-
struct faithfully. Hence, we introduce a hybrid refinement
method to handle them. We refine in-domain parts by tuning
generator weight since it is better to maintain editing ca-
pability. For out-of-domain parts, we straightforwardly invert
them by intermediate features to keep spatial image details.
Notably, our hybrid refinement method is the first work to
analyze and combine feature and weight modulation for im-
proved GAN inversion and achieves extraordinary results as
shown in Figure 1.

Extensive experiments are presented to demonstrate the
effects of our Domain-Specific Hybrid Refinement (DHR).
We achieve state-of-the-art and gain significant improve-
ment in both fidelity and editability. The key contributions
of this work are summarized as follows:

- We analyze the reasons for editing capability degrada-
tion in the refinement process. Based on our analysis,
we introduce in-domain and out-of-domain and pro-
pose Domain-Specific Segmentation to segment im-
ages into these two parts for better inversion.

- We propose Hybrid Modulation Refinement to im-
prove inversion results of in-domain and out-of-
domain parts. We conduct weight modulation on in-
domain part and feature modulation on out-of-domain
part, which can preserve editing capability when refin-
ing the image details.

- We conduct extensive experiments and user studies to
demonstrate the effects of our method. We reach ex-
traordinary performance on real-world image inversion
and editing and achieve the state-of-the-art.

2. Related Work

2.1. GAN Inversion

GAN inversion aims to embed real-world images into a
pretrained generator’s latent space, which can be used to
reconstruct and edit input images. Generally, methods can
be divided into two stages.

The first stage aims to attain low-rate latent codes, usu-
ally in $\mathbb{Z}/W/W^+$ spaces. The latent codes are gained by
an encoder or optimization process. Training an encoder
\cite{6,11,25,26,31,36} to predict latent codes is efficient for in-
fERENCE and is easier to get better trade-offs between fidelity
and manipulation \cite{25,31}. Optimizing initial latent codes
by reconstruction discrepancy gains better fidelity. How-
ever, it may cost several minutes per image \cite{1,2,6,20} and
decreases editability during per-image tuning. Due to low-
rate property, latent codes can only reconstruct coarse in-
formation and drop the details from original images. Mean-
while, there is a trade-off between fidelity and editability,
and many methods introduce additional regularization mod-
ules (e.g., latent code discriminator \cite{31} and latent space
alignment \cite{25}) to address it.

In the second stage, reconstruction and manipulation re-
results from latent codes are refined by high-rate informa-
tion. Refinement methods are mainly divided into weight
modulation and feature modulation. Weight modulation
methods predict or finetune generator weight to improve fidelity. Some methods [5, 8] use hypernetwork [12] to predict weight offsets. The others tune generator by given images, which attain better fidelity but cost much time [9, 27]. Another branch further inverts images to latent feature, which we call feature modulation. HFGI [34] proposes a distortion consultation approach for high-fidelity reconstruction. SAM [23] segments images into various parts and inverts them into different intermediate layers by predicting “invertibility.” All of them only use one of feature and weight modulation to refine results and suffer editing capability degradation.

2.2. GAN-based Manipulation

GANs’ latent spaces encode highly rich semantic information, which develops the GAN-based manipulation task. It aims to edit given images by changing latent codes in certain directions. Many works propose multiple methods to find semantic editing directions in GANs’ latent spaces. Some methods obtain the edit vectors of the corresponding attributes by supervision with the help of attribute-labeled datasets [7, 10, 30]. For example, InterfaceGAN trains SVMs to find attributes’ boundaries in latent space and achieve promising editing results [28]. Without label annotation, other methods explore the latent space by unsupervised [13, 29, 32, 33] or self-supervised ways [15, 24] to find more semantic directions way.

3. Method

3.1. Preliminaries

Inversion is built to bridge real-world images and GANs’ latent space. As latent codes are low-rate, which limits their reconstruction performance, much research has recently focused on additional high-rate information in generation process, which we call refinement methods. They can be mainly divided into two categories: weight modulation and feature modulation. We first formulate them and analyze the causes of editing capacity degradation.

**Formulation.** We denote the original generation process as $X = G(w)$, where $G$ is the generator, and $w$ is latent code which can represent each latent space (e.g., $Z/W/W^+$). In the refinement process, we use encoded latent codes which can be attained by off-the-shelf encoders (e.g., pSp [26] and e4e [31]).

Weight modulation methods predict [5] or optimize [20] weight $\theta$ by minimizing reconstruction error to get $\theta^*$, and the inference process is denoted as $X = G(w, \theta^*)$. Feature modulation methods invert images into the intermediate feature, which follows $X = G(w, f)$. Defining $\mathcal{L}$ as the distance of images, we can illustrate these two refinement processes as follow:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(x, G(w; \theta)) \quad (1)$$

$$f^* = \arg \min_{f} \mathcal{L}(x, G(w, f)) \quad (2)$$

**Impacts on editing capability.** Weight and feature modulation impact image manipulation in different aspects. The schematics are shown in Figure 2. Since the feature modulation mechanism fixes the intermediate feature distribution at one of the layers, the effects of edit vectors applied to previous layers cannot edit the features of the latter layers. Although many existing works make efforts to maintain the editing effects, including training with adaptive distortion alignment [17, 34], their solutions still sacrifice fidelity or
Figure 3. **Impacts on editing capability of weight modulation.** We show the input images, inversion results, and two editing results (smile and age) from PTI [27]. For those easy samples, editing results are reasonable. However, editing capability degrades significantly on hard samples.

3.2. **Domain-Specific Hybrid Refinement**

Based on the above analysis, weight modulation gains better editing capability but is easily affected by complex image areas, while feature modulation has better reconstruction ability. Hence, we explore the idea of “divide and conquer” and use both of them to improve GAN inversion. In this work, we conduct Domain-Specific Hybrid Refinement (DHR) to deal with real-world image inversion, and the pipeline is shown in Figure 4.

We first propose the concepts of *in-domain* and *out-of-domain*. *In-domain* refers to areas that have a similar distribution to the generator’s output space, which makes them easy to invert. Instead, *out-of-domain* areas misalign with output space, making them challenging to invert. For instance, in the face domain, *in-domain* areas mainly include the face and hair, while *out-of-domain* areas encompass occlusions, backgrounds, and artifacts. Additionally, *in-domain* areas are more editable, such as smiling, lipstick, and eye openness.

Hence, we propose a hybrid refinement method, which segments images into *in-domain* and *out-of-domain* and applies weight and feature modulation to them respectively. Our framework is shown in Figure 4, which consists of three components.

The image Embedding module aims to embed images into latent codes, which we use an off-the-shelf encoder (e.g., e4e [31] and LSAP [25]). Given input images $X$, the encoder predicts its $W^+$ space latent codes, which we denote as $w = E(X)$, where $E$ is an encoder.

**Domain-Specific Segmentation** predicts a binary mask which indicates *in-domain* and *out-of-domain* areas:

$$m = S(X)$$

where $m \in \{0, 1\}^{H \times W}$. It segments images into two domains, which will be used for refinement.

In Hybrid Modulation Refinement, weight modulation is employed to refine *in-domain* areas and restore image details for both inversion and editing results. Since the reconstruction discrepancy of *in-domain* areas is low, weight deviation is limited, thereby preserving the editing capacity. Conversely, for *out-of-domain* parts, we use feature modulation to refine them spatially and fix feature distribution during editing. Hence, those hard-to-invert parts do not affect editing ability. Given domain segmentation results $m$, we modulate weight $\theta$ and feature $f$ by minimizing reconstruction error in *in-domain* and *out-of-domain* part, respectively:

$$\theta^*, f^* = \arg \min_{\theta, f} \mathcal{L}(X, G(w, f, m; \theta))$$  \hspace{1cm} (3)

Figure 2 illustrates the difference in the updated generator manifold between our hybrid technique and the previous refinement mechanisms. By adopting the hybrid approach, the generator manifold undergoes minimal changes and does not become too sharp at the given data point, which effectively maintains the editing capability. Notably, we conduct two refinements simultaneously and visualize $G(w, f^*; \theta^*)$ and $G(w; \theta^*)$ separately to demonstrate the effect of our method.
We propose a Domain-Specific Segmentation module to segment images into two domains: in-domain and out-of-domain. However, training an end-to-end learning-based domain segmentation model requires a large and annotated dataset, which can be costly. Although previous work [23] trains an invertibility prediction model by self-supervision, results are inaccurate in some complex areas, as we demonstrate in the appendix. To address this issue and achieve robust results on real-world images without requiring data annotation, we design an automatic segmentation pipeline consisting of two steps: “partition” and “binarize”, which are combined with a parsing model. The pipeline is shown in Figure 5.

We utilize a superpixel algorithm [3] to partition the input image into multiple areas. Each partition is represented as \( \{m_s^i\}_{i=1}^N \). Categorizing each partition into in-domain and out-of-domain without manual annotation is a crucial challenge. However, categorizing each partition into in-domain and out-of-domain without any manual annotation poses a significant challenge. To address this issue, we employ a coarse optimization in the \( W \) space. Here, we initialize the latent codes with mean values and optimize them for a few steps. Since in-domain are those easy-to-invert areas, the coarse inversion result \( X_{\text{coarse}} \) could reconstruct in-domain areas effectively. We compute the perceptual loss \( \mathcal{L}(X, X_{\text{coarse}}) \) between the coarse reconstruction image and the input image, as shown at the bottom of Figure 5. The white area indicates a higher loss value, while the black area represents a lower loss value. As can be observed, the loss value of the occluded area is significantly higher than that of the face area. We then calculate the average loss of each partition as follows:

\[
v_i = \frac{\mathcal{L} \odot m_s^i}{\|m_s^i\|}
\]

and binarize them by threshold \( \tau \) and attain \( m_s \).

To compensate for the missed segmentation of small areas by superpixel, we also utilize a parsing model. The parsing model categorizes face components like eye, mouth, and background [46]. We obtain parsing results \( m_p \), which we manually categorize as in-domain or out-of-domain based on the face components. Finally, domain-specific segmentation results are combined by parsing results and superpixel results: \( m = m_p \times m_s \). Our Domain-Specific Segmentation module gains fine segmentation results without data annotation and is more effective for real-world images.

### 3.4. Hybrid Modulation Refinement

To maintain the editing capability of a pre-trained GAN and faithfully recover image details, we introduce a Hybrid Modulation Refinement module. This module includes two refinement aspects: weight modulation and feature modulation. Weight modulation aims to minimize in-domain reconstruction error by tuning the generator’s parameters, while feature modulation is applied to out-of-domain areas.
Figure 6. **Illustration of Hybrid Modulation Refinement module.** We refine *in-domain* areas and *out-of-domain* areas by weight and feature modulation, respectively. Black lines indicate forward flows, and red arrows represent gradient flows.

by optimizing the intermediate feature. The forward and backward processes are shown in Figure 6.

For the *l*th layer of the total *k* stages in the generator, we denote the original generator’s feature as \( f_l = G_l(w; \theta) \). An additional modulated feature is marked as \( f \), which is initialized by \( f_l \). When the latent codes are fixed, the original feature \( f_l \) is only relevant to \( \theta \). We formulate the forward process given the segmentation result \( m \) as follows:

\[
f' = f_l \odot m + f \odot (1 - m)
\]

Then \( f' \) represents the output of the first \( l \) layers and generates the final images.

For the backward process, we update weight and feature in a parallel optimization process to make them focus on the corresponding domains. We set *in-domain* and *out-of-domain* areas as targets for them, respectively. We use mean square error \( L_2 \) and perceptual loss \( L_{lpips} \) as objectives. Calculating the reconstruction errors, we backward loss with segmentation result \( m \):

\[
\mathcal{L} = L_2 + \lambda L_{lpips}
\]

\[
\nabla f = \frac{\partial}{\partial f} \left( \frac{\mathcal{L} \odot (1 - m)}{||1 - m||} \right)
\]

\[
\nabla \theta = \frac{\partial}{\partial \theta} \left( \frac{\mathcal{L} \odot m}{||m||} \right)
\]

where \( \lambda \) is a hyper-parameter. The parallel optimization mechanism constrains the impact from different domains. As *in-domain* areas suffer small deviation, \( \nabla \theta \) is lower, resulting in less effect on the generator manifold. This significantly retains its highly semantic property and preserves editing capability.

### 4. Experiments

#### 4.1. Experimental Settings

**Datasets.** We evaluate all methods on the CelebA-HQ [16, 22] test set (2,824 images). Encoders and the generator are trained on FFHQ [19] (70,000 images).

**Baselines.** We compare our model to previous state-of-the-art refinement methods, i.e., ReStyle [4], HFGI [34], SAM [23], and PTI [27]. We use pSp [26], e4e [31] and LSAP [25] as encoders. Moreover, the performance of encoder-based methods is also reported. All of weights of encoders come from their official release.

**Metrics.** We evaluate all methods at two respects: inversion and editing. For inversion ability, we conduct MSE, LPIPS [45], and identity similarity calculated by a face recognition model [14]. MSE straightforwardly measures the image distortion, and LPIPS evaluates the visual discrepancy. Identity similarity further compares the identity consistency during inversion. Moreover, we perform user studies to evaluate the perceptual performance of inversion and editing.

### 4.2. Main Results

**Quantitative results.** We first evaluate the reconstruction ability quantitatively. The results are reported in Table 1. We compare our method with four previous refinement methods and employ two encoders. For PTI, we conduct experiments with encoders and their proposed \( W_{pivot} \).

As one can see, DHR achieves the best performance on all metrics. Employed with e4e, it gains 0.0036 MSE, which is 7.5% of the vanilla e4e, 8.3% of ReStyle, 17% of HFGI, 25% of SAM, and 48% of PTI. For LPIPS and identity similarity, it demonstrates similar superiority and surpasses other methods by a large margin.

This is because our hybrid refinement approach attains an extraordinary trade-off between fidelity and editability. The previous feature modulation methods (i.e., HFGI and SAM) constrain the effects of the additional feature information by elaborate fusion module or feature regularization to preserve editing capability, and Restyle and PTI encode spatial details in low-rate latent codes or global convolutional weights (hard to recover spatial visual details). All of them employ stronger limitations on high-rate information to maintain editing capability. We will then demonstrate that our DHR not only gains better quantitative reconstruction results but also attain better editing results without any...
Figure 7. **Inversion and editing results.** The second column is the segmentation results of *in-domain* and *out-of-domain* areas. Our method restores almost all image details.

**Qualitative results.** We illustrate the inversion and editing results of DHR in Figure 7. The second column is the results from the Domain-Specific Segmentation module, and the third is our inversion results. Some dedicated areas are categorized into *out-of-domain* domain (black area), such as the braid area in the first image. We edit them in four directions, i.e., smile, young, exposure, and lipstick. All image details are preserved in both inversion and editing results, such as hairstyle (the first row), earrings (the fifth row), and hats.

We further conduct a qualitative comparison with other methods, shown in Figure 8. Although inversion results are reasonable, editing capability degradation occurs in baselines, e.g., decorations and occlusion blur. Our editing results are more faithful and photorealistic.
Figure 8. **Comparisons to previous methods.** We compare the inversion and editing results with PTI [4], HFGI [34], and SAM [23]. Although HFGI and SAM reach reasonable inversion results, image distortion and details loss occur in editing results. Our method attains the best fidelity and editing performance. Image details are reserved in both phases, and our results are the most natural.

<table>
<thead>
<tr>
<th>Method</th>
<th>Encoder</th>
<th>MSE ↓</th>
<th>LPIPS ↓</th>
<th>Similarity ↑</th>
<th>MSE ↓</th>
<th>LPIPS ↓</th>
<th>Similarity ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReStyle [4]</td>
<td>pSp</td>
<td>0.0276</td>
<td>0.1298</td>
<td>0.5816</td>
<td>0.0276</td>
<td>0.1298</td>
<td>0.5816</td>
</tr>
<tr>
<td>e4e</td>
<td></td>
<td>0.0429</td>
<td>0.1904</td>
<td>0.5062</td>
<td>0.0429</td>
<td>0.1904</td>
<td>0.5062</td>
</tr>
<tr>
<td>HFGI [34]</td>
<td>e4e</td>
<td>0.0210</td>
<td>0.1727</td>
<td>0.6816</td>
<td>0.0210</td>
<td>0.1727</td>
<td>0.6816</td>
</tr>
<tr>
<td>LSAP</td>
<td></td>
<td>0.0210</td>
<td>0.0945</td>
<td>0.7405</td>
<td>0.0210</td>
<td>0.0945</td>
<td>0.7405</td>
</tr>
<tr>
<td>SAM [23]</td>
<td>e4e</td>
<td>0.0948</td>
<td>0.1704</td>
<td>0.5586</td>
<td>0.0948</td>
<td>0.1704</td>
<td>0.5586</td>
</tr>
<tr>
<td>LSAP</td>
<td></td>
<td>0.1117</td>
<td>0.0939</td>
<td>0.6184</td>
<td>0.1117</td>
<td>0.0939</td>
<td>0.6184</td>
</tr>
<tr>
<td>PTI [27]</td>
<td>e4e</td>
<td>0.0079</td>
<td>0.0750</td>
<td>0.8533</td>
<td>0.0079</td>
<td>0.0750</td>
<td>0.8533</td>
</tr>
<tr>
<td>Wpivot</td>
<td></td>
<td>0.0067</td>
<td>0.0666</td>
<td>0.8696</td>
<td>0.0067</td>
<td>0.0666</td>
<td>0.8696</td>
</tr>
<tr>
<td>DHR (ours)</td>
<td>e4e</td>
<td>0.0015</td>
<td>0.0435</td>
<td>0.8704</td>
<td>0.0015</td>
<td>0.0435</td>
<td>0.8704</td>
</tr>
<tr>
<td>LSAP</td>
<td></td>
<td>0.0035</td>
<td>0.0436</td>
<td>0.8780</td>
<td>0.0035</td>
<td>0.0436</td>
<td>0.8780</td>
</tr>
<tr>
<td>Encoder-Only</td>
<td>pSp [26]</td>
<td>0.0351</td>
<td>0.1628</td>
<td>0.5591</td>
<td>0.0351</td>
<td>0.1628</td>
<td>0.5591</td>
</tr>
<tr>
<td>e4e [31]</td>
<td></td>
<td>0.0475</td>
<td>0.1991</td>
<td>0.4966</td>
<td>0.0475</td>
<td>0.1991</td>
<td>0.4966</td>
</tr>
<tr>
<td>LSAP [25]</td>
<td></td>
<td>0.0397</td>
<td>0.1766</td>
<td>0.5305</td>
<td>0.0397</td>
<td>0.1766</td>
<td>0.5305</td>
</tr>
</tbody>
</table>

Table 1. **Fidelity results on face domain.** We compare DHR to three previous refinement methods with two powerful encoders. The results of these encoders are also presented at the bottom.

<table>
<thead>
<tr>
<th>Method</th>
<th>Inversion</th>
<th>Editing</th>
<th>Smile</th>
<th>Young</th>
<th>Exposure</th>
<th>Lipstick</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReStyle [4]</td>
<td>94%</td>
<td></td>
<td>100%</td>
<td>100%</td>
<td>90%</td>
<td>94%</td>
</tr>
<tr>
<td>HFGI [34]</td>
<td>94%</td>
<td></td>
<td>84%</td>
<td>86%</td>
<td>100%</td>
<td>96%</td>
</tr>
<tr>
<td>SAM [23]</td>
<td>100%</td>
<td></td>
<td>92%</td>
<td>94%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>PTI [27]</td>
<td>96%</td>
<td></td>
<td>100%</td>
<td>84%</td>
<td>88%</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. **User study.** We conduct user studies on inversion and editing tasks. The values in the table indicate the percentage of images where users prefer our results. Results show our method is more faithful and photorealistic.

### 4.3. User Study

We conduct user studies to demonstrate the performance of inversion and editing. Results are shown in Table 2. We randomly select 50 different images and invert and edit them by HFGI [34], SAM [23], PTI [27], and our method. We then ask three users to make a preference for each pair of images. A higher value implies users prefer our results. As can be seen, our results are highly preferred by users, which are most all above 90%, compared to previous state-of-the-art methods. It illustrates that our method decreases image distortion and attains better photorealism of reconstruction and manipulation results.

### 5. Conclusion

In this work, we present a novel inversion approach, Domain-Specific Hybrid Refinement, aimed at improving GAN inversion and editing capability. We investigate the causes of editing ability degradation in the refinement process and introduce a “divide and conquer” strategy to address this issue. Our method consists of two main components: Domain-Specific Segmentation and Hybrid Modulation Refinement. The former segments images into in-domain and out-of-domain parts without data annotation, while the latter refines them by weight and feature modulation, respectively. Our method achieves promising results in both inversion and editing tasks, with significant improvements over existing approaches.

**Acknowledgements** This work was supported by the National Key Research and Development Program of China (Grant No. 2022YFC3302200), China Postdoctoral Science Foundation (2022M710467), National Key Research and Development Program of China (Grant No. 2021YFF0500900), and Intelligent Logistics Interdisciplinary Team Project of BUPT.
References


[27] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. ACM Trans. Graph., 2021. 2, 3, 4, 6, 8


