Improving quality of pixel-wise transfer using Attention-based Encoding (SAE) and Abortion method

Wonseok Oh[1,2], Jinkyu Kim[1,2]
[1]**Korea University,** [2]**Vision & AI lab**

abstract

In this study, we present two new methods to enhance image-to-image translation performance. First, we investigate the integration of the Attention Module with the encoder. CBAM aims to improve feature representation in convolutional neural networks, while pSp offers a robust method for encoding input images into their corresponding latent domain. By incorporating CBAM into pSp, we achieve superior feature extraction, leading to more precise image reconstructions, increased control over image translation, and the ability to handle complex tasks, such as multi-modal synthesis or cross-domain translation. Our proposed approach demonstrates notable performance improvement in qualitative analysis compared to existing methods such as pSp, IDInvert and E2Style on the CelebAMask-HQ dataset. Second, we offer a more efficient E2Style model using the abortion method. In the training process, we exclude inefficient iterations by prompting an abortion upon meeting certain conditions. This novel approach has proven effective in addressing the issue of overfitting and improves upon other aspects by comparing previous models.

1 Introduction

In recent times, Generative Adversarial Networks (GAN) [9] have undergone significant advancements. GAN inversion(Figure 1) [10] is a technique aimed at understanding and controlling the latent space of GAN. Its goal is to find the latent vector that best reconstructs a given image using a pre-trained GAN, with applications in image editing, style transfer, and content manipulation.

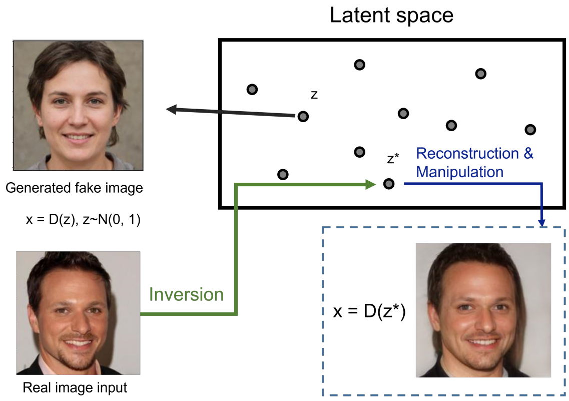

Figure 1: D represents GAN inversion.
Given a real image x as input, D outputs a latent code x^*.
Specifically, x^* is obtained as $x^* = D(z^*)$, where z^* is the optimized noise that reconstructs the input image x.

However, in working with real images, the invertibility requirement poses a challenge. Directly inverting a real image into a latent code often does not result in an accurate reconstruction.

To address this limitation, the pixel2style2pixel (pSp) [1] translation method uses a novel encoder architecture in conjunction with the pretrained StyleGAN [6, 7] generator as a comprehensive image-to-image translation framework. Instead of trying to invert the input image into a latent code, the pSp method directly encodes input images into the desired output latent.

Furthermore, Convolutional Block Attention Module (CBAM) [2] is a neural network architecture component that enhances the feature representation of convolutional neural networks (CNN) [11]. CBAM uses Channel and Spatial Attention to adaptively recalibrate input data's spatial and channel-wise features, aiding in acquiring more discriminative features and improving the network's performance.

To improve the performance of the pSp translation method, we incorporated CBAM into pSp and named this model StyleGAN with Attention-based Encoding (SAE).

2 Access to abortion method

The emergence of deep-learning-based image processing technology has given rise to countless new studies utilizing GAN(Generative Adversarial Networks) which generates new images from input images. GAN structure consists of the generator and the discriminator in an adversarial training process. GAN model learns the latent space, the distribution of the latent vectors of the input image. The encoder converts image to feature vector, while the decoder reconstructs these features - *latent vectors* to new image. GAN inversion is the process of finding the latent vector that allows us to derive an output image most similar to the input. StyleGAN [7] emerged with its novel approach of incorporating a style-based generator. With the use of its new generator based

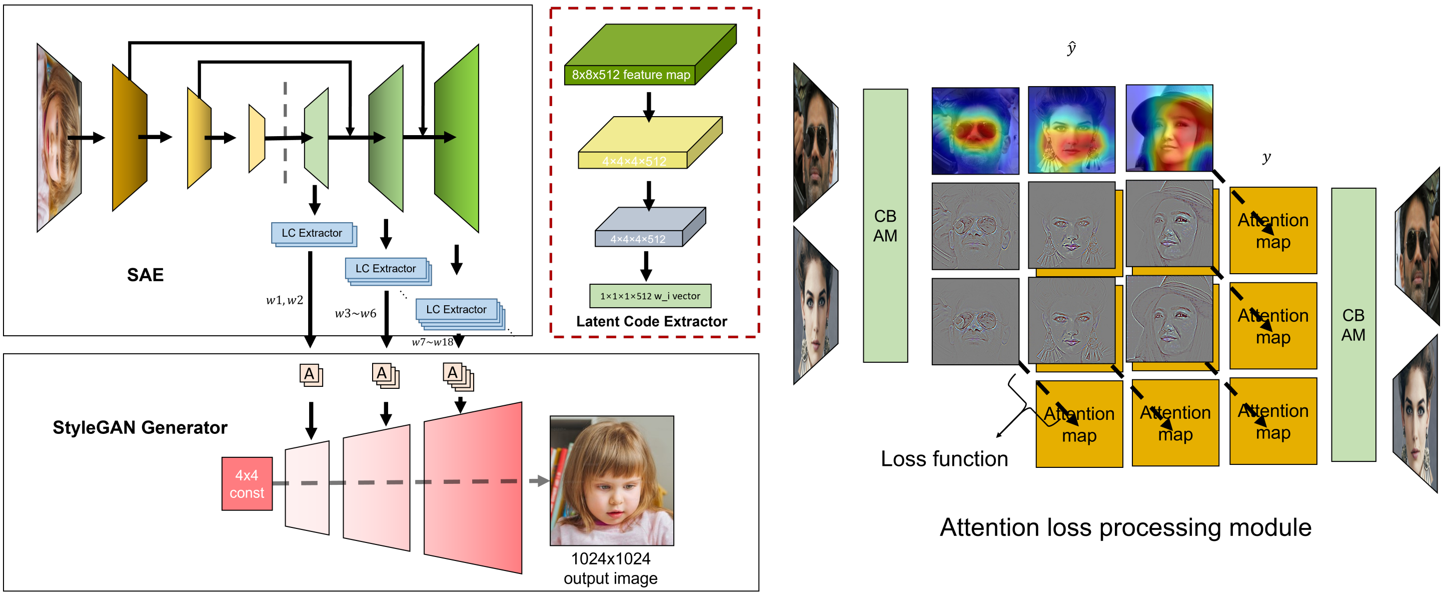

Figure 2: SAE model architecture
The process of generating images use the pSp framework with attention loss processing module layer. Feature maps are extracted from the input image using a standard feature pyramid [14] over a ResNet backbone [12]. Then, feature maps are classified into small or large mapping networks based on their size for each of the 18 target styles. Small mapping networks are used for small feature maps and generate 512-dimensional vectors using the Map2Style network, which are then applied to StyleGAN for image generation. A attention loss processing module layer is added to the pSp module to enhance feature representation and improve image quality.

on style transition, StyleGAN made possible the control over not only overall characteristics but also elaborate details - *skin color, age, gender etc*. E2Style [8] aims to improve the efficiency and effectiveness of StyleGAN inversion. E2Style's process is as follows: First, its encoder network takes into account the hierarchical structure to expect the latent vectors. These are extracted from the encoder's various spatial levels, and with the various details of the pre-trained StyleGAN generator, the learning difficulty is lowered. Second, E2Style accepts shared efficient prediction heads for each level. This includes global average pooling layers of varying size and a full-connected layer, resulting in a more lightweight and efficient network.

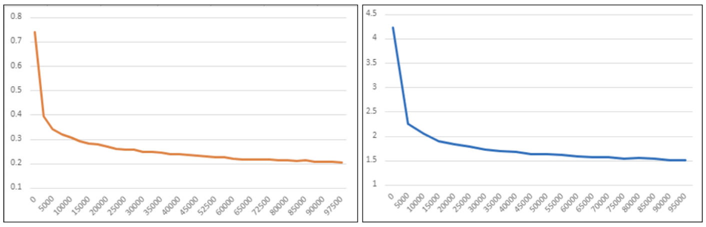

Figure 3: Change of loss values in pSp(*left*) and E2Style(*right*)

Fig.3 shows a gradual but continuous decline in loss function values. We identify points where the change in the loss value has slowed down and become trivial. These suggest the potential occurrence of overfitting in the training set. To address this issue while maintaining the quality of generated images, we propose a novel approach. We highlight the iteration process in E2Style's hierarchical structure, which proposed methods of improving the efficiency and effectiveness of StyleGAN inversion. We introduce the proposal of reducing the fixed number of iterations from the basic E2Style method. This is expected to ease the burden of training calculation as well as lessen the chances of overfitting in training datasets.

3 StyleGAN with Attention-based Encoding

The pSp framework combines a pre-trained StyleGAN generator and the $\mathcal{W}+$ latent space to accurately represent input images. However, directly encoding images into $\mathcal{W}+$ using a single vector from the encoder's last layer has limitations in capturing finer details and reconstructing the image accurately. Therefore, a robust encoder is necessary to map each input image to its corresponding encoding in the latent domain. We propose the SAE model, an enhanced version of the pSp framework that utilizes CBAM in place of its CNN parts. This modification aims to improve the overall performance of the model.

In StyleGAN, it was discovered that style inputs correspond to different levels of detail, divided into three groups - coarse, medium, and fine. Building on this insight, pSp incorporates a feature pyramid to generate three levels of feature maps from which styles are extracted using an intermediate network, map2style, as shown in Figure 2. These categorized styles are then aligned with the hierarchical representation and fed into the generator in correspondence to their scale to generate the output image, completing the translation from input to output pixels through the intermediate style representation. The complete architecture is presented in Figure 2.

4 SAE Loss functions

Our framework utilizes a weighted combination of several objectives to train the encoder. The pixel-wise $\mathcal{L}_2$ loss is utilized as follows:

$$\mathcal{L}_2(\mathbf{x}) = ||\mathbf{x} - D(E(\mathbf{x}))||_2 \tag{1}$$

To learn perceptual similarities, we incorporate the $\mathcal{L}_{LPIPS}$ loss [**?**], which has been shown to better preserve image quality compared to the standard perceptual loss [**?**]:

$$\mathcal{L}_{LPIPS}(\mathbf{x}) = ||F(\mathbf{x}) - F(D(E(\mathbf{x})))||_2 \tag{2}$$

Here, F(·) denotes the perceptual feature extractor. In addition, to encourage the encoder to output latent style vectors closer to the average latent vector, we include the following regularization loss:

$$\mathcal{L}_{reg}(\mathbf{x}) = ||E(\mathbf{x}) - \bar{\mathbf{w}}||_2 \tag{3}$$

To tackle the challenge of preserving the input identity when encoding facial images, we incorporate a dedicated recognition loss measuring the cosine similarity between the output image and its source:

$$\mathcal{L}_{ID}(\mathbf{x}) = 1 - \langle(\mathbf{x}), R(D(E(\mathbf{x})))\rangle \tag{4}$$

The loss function defined up to this point is referred to as the encoder loss, and is defined as:

$$\mathcal{L}_E(\mathbf{x}) = \lambda_1\mathcal{L}(\mathbf{x}) + \lambda_2\mathcal{L}_{LPIPS}(\mathbf{x}) + \lambda_3\mathcal{L}_{ID}(\mathbf{x}) + \lambda_4\mathcal{L}_{reg}(\mathbf{x}) \tag{5}$$

In the Convolutional Block Attention Module (CBAM), used as a sub-module of the pSp encoder during training, there are two essential components: the channel attention module and the spatial attention module. The channel attention module calculates a 1D channel attention map $\mathbf{M_c} \in \mathbb{R}^{C\times1\times1}$, weighting the importance of different channels in the feature map $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$. The spatial attention module computes a 2D spatial attention map $\mathbf{M_s} \in \mathbb{R}^{1\times H\times W}$, emphasizing significant regions in the feature map F.

$$\begin{aligned}\mathbf{F}'(\mathbf{x}) &= \mathbf{M_c}(\mathbf{F}) \otimes \mathbf{F},\\ \mathbf{F}''(\mathbf{x}) &= \mathbf{M_s}(\mathbf{F}') \otimes \mathbf{F},\end{aligned} \tag{6}$$

To generate the channel context descriptors $\mathbf{F^c_{avg}}$ and $\mathbf{F^c_{max}}$ used in the channel attention module, both average-pooling and max-pooling operations are applied to the feature map F to aggregate its spatial information. These operations are used to compute the statistics of the feature map along the channel dimension, resulting in two distinct spatial context descriptors.

$$\begin{aligned}\mathbf{M_c}(\mathbf{F}) =& \sigma(MLP(AvgPool(\mathbf{F}) + MLP(MaxPool(\mathbf{F})))\\ =& \sigma(\mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{avg}})) + \mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{max}})))\end{aligned} \tag{7}$$

In contrast to the channel attention module that focuses on the what, the spatial attention module in CBAM is dedicated to identifying the where in the input feature map. To compute the spatial attention, the module applies both average pooling and max pooling operations along the channel axis. These operations generate two spatial context descriptors, which are concatenated to create an efficient feature descriptor. A convolutional layer is subsequently applied to the concatenated descriptor to generate the spatial attention map. This map encodes where to emphasize or suppress in the input feature map, allowing the network to focus on the most relevant spatial regions. By incorporating the spatial attention module into the network, CBAM is able to enhance the performance of various computer vision tasks, such as image classification and object detection.

$$\begin{aligned}\mathbf{M_s}(\mathbf{F}) =& \sigma(f^{7\times7}([AvgPool(\mathbf{F}; MaxPool(F)]))\\ =& \sigma(f^{7\times7}([\mathbf{F^s_{avg}}; \mathbf{F^s_{max}}]))\end{aligned} \tag{8}$$

So eventually the attention map is used to make the attention loss. Therefore the overall loss function is as follows,

$$\mathcal{L}_{total}(\mathbf{x}) = \lambda\mathcal{L}_E + \lambda_4||\mathbf{F}''(\mathbf{x}) - \mathbf{F}''(G(E(\mathbf{x})))||_2 \tag{9}$$

CBAM is renowned for exhibiting superior performance compared to basic CNN. SAE is made by utilizing CBAM as a sub-module of the pSp encoder instead of a standard CNN, there is a noticeable improvement in the output. This enhancement will be demonstrated in the experimental results section of the paper.

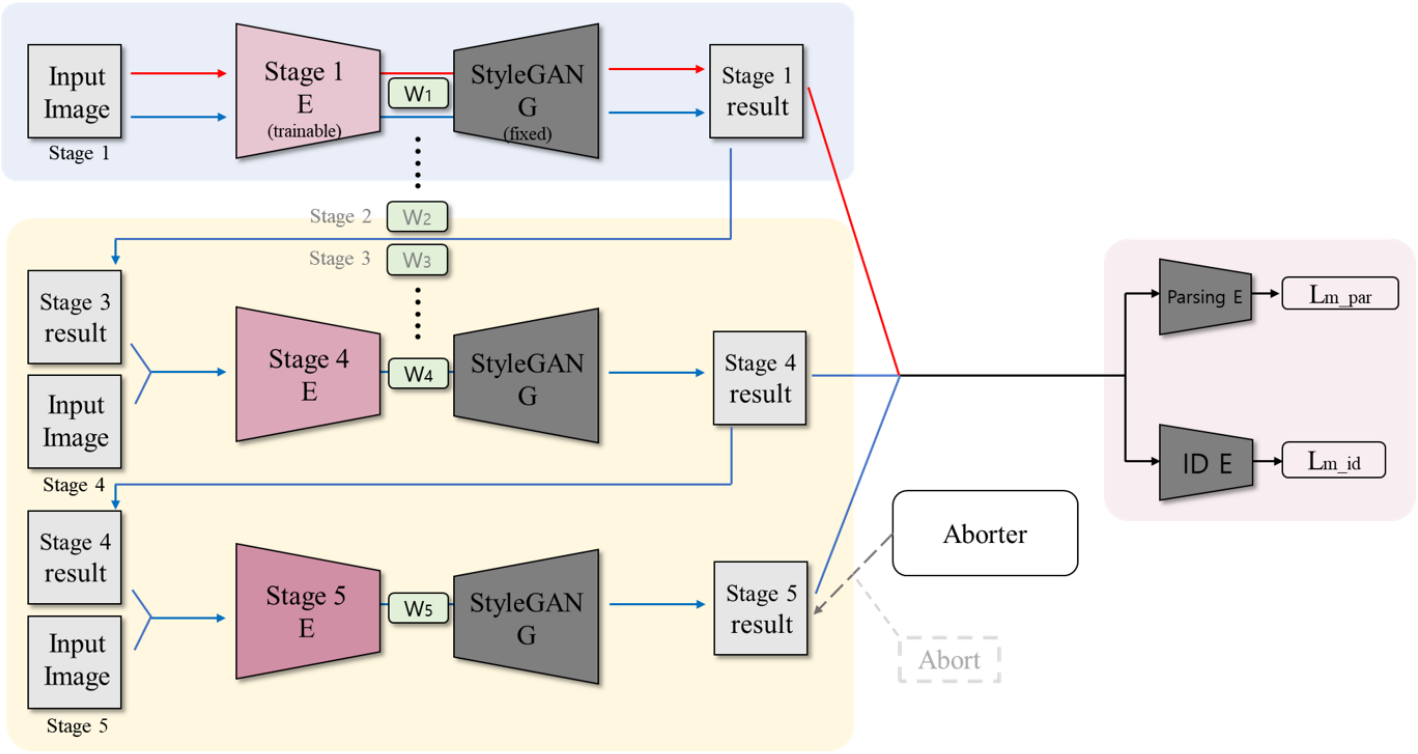

Figure 4: E2Style with Abortion. Abortor prevents overffitng by truncating the last stage under certain conditions.

5 Abortion method Loss functions

5.1 Functional Equation

We adopt the same functions introduced by E2Style. E2Style's loss function can be largely divided into two parts.

Common losses. Common losses consist of $\mathcal{L}_2$ loss and $\mathcal{L}_{LPIPS}$ [3] loss . $\mathcal{L}_2$ refers to the difference between the input image and the output image, which has been reconstructed through the encoder and the StyleGAN generator, acts as decoder, on a pixel level.

$$\mathcal{L}_2 = \|\mathbf{x} - D(E(\mathbf{x}))\|_2 \tag{1}$$

Using $\mathcal{L}_2$ solely is insufficient to discern the features of the reconstruction result. Therefore, we derive the feature-level loss through the additional use of $\mathcal{L}_{LPIPS}$ loss.

$$\mathcal{L}_{LPIPS} = \|F(\mathbf{x}) - F(D(E(\mathbf{x})))\|_2 \tag{2}$$

Multi-Layer Loss. Multi-Layer Loss consists of Identity Loss and Parsing Loss between multi layer outputs. In GAN inversion, it is crucial to conserve the identity information of the original image's attributes. Multi-Layer Identity Loss refers to maintained consistency between the input image and the inverted output image

$$\mathcal{L}_{ID} = \sum_{k=1}^{5}(1 - cos(N_f(\mathbf{x}), (N_f(D(E(\mathbf{x}))))) \tag{3}$$

cos refers to cosine similarity. N_f(x) is the feature that corresponds to semantic level k in the facial recognition network N [4] for image $\mathbf{x}$. The Parsing Loss function in the Multi-Layer works in the opposite way from Identification Loss by separating different features.

$$\mathcal{L}_{PAR} = \sum_{k=1}^{5}(1 - cos(P_k(\mathbf{x}), (P_k(D(E(\mathbf{x}))))) \tag{4}$$

Likewise, P_k(x) refers to the feature that corresponds to the kth semantic level in the pre-trained facial parsing network P [5] for input image $\mathbf{x}$. Parsing loss operates in tandem with Identity loss in a complementary way, enabling the two loss functions to operate as each feature in the multilayer forms clusters.

Summary. To sum up, the value of the total loss function can be expressed as below:

$$\mathcal{L}_{total} = \lambda_1\mathcal{L}_2 + \lambda_2\mathcal{L}_{LPIPS} + \lambda_3\mathcal{L}_{ID} + \lambda_4\mathcal{L}_{PAR} \tag{5}$$

5.2 Abortion method description

In this paper, we propose the abortion technique in the E2Style training process. Previous E2Style does not allow for flexible adjustment of the number of iterations. So it is expected a vast load of data in order to extract the images, and also lies the risk of overfitting in the training set. Our approach proposes two type of abortion methods to counter such drawbacks, through which we expect more efficient results.

The first method is **relative method**. We determine a relative call condition for the abortion function. In this method we begin by designating previous loss = ∞. After the 1st iteration, we calculate the value of loss and subtract it from the previous loss. Here, if the result is positive, we reset the value of $abort\_count$ to 0, because it has yielded a result image of higher quality. Otherwise, we continuously add 1 to the value of $abort\_count$. The previous loss, which we denoted as ∞ prior to the 1st iteration will now be updated to the newly derived loss. Once count reaches 10 after a series of reiterations, the abortion function is called and $abort\_count$ is reset. The abortion function then indefinitely reduces the number of repetitions the input images will undergo.

The second is the **absolute method**, in which we set a random parameter a. This parameter acts as an absolute criterion that determines whether to reset $abort\_count$ to 0 or proceed to add increments of 1. Then, the value of loss is derived by continuously adding one iteration until abortion occurs. If the value of (a – loss) is positive, this, as mentioned above results in a higher quality image and therefore resets $abort\_count$ to 0. Otherwise, $abort\_count$ will increase by increments of 1. Likewise, $abort\_count$ exceeding 10 will call the abortion function and result in $abort\_count$ being reset. This process is as shown in Fig. 4.

By applying the new methods proposed in this paper, we can reduce the amount of calculations by excluding inefficient iterations, and expect positive results by avoiding overfitting issues with our training dataset.

In terms of implementation, the method of reducing iteration should be approached thoughtfully. In particular, when using absolute abortion method, setting a high threshold may result in inappropriate abortion which occurs before the sufficient training. It leads to sub-optimal performance due to an insufficiently trained encoder. Therefore, careful consideration and optimization of the threshold value are necessary to ensure the training process which achieves the desired performance while avoiding potential negative impacts.

6 Experimental Results of SAE

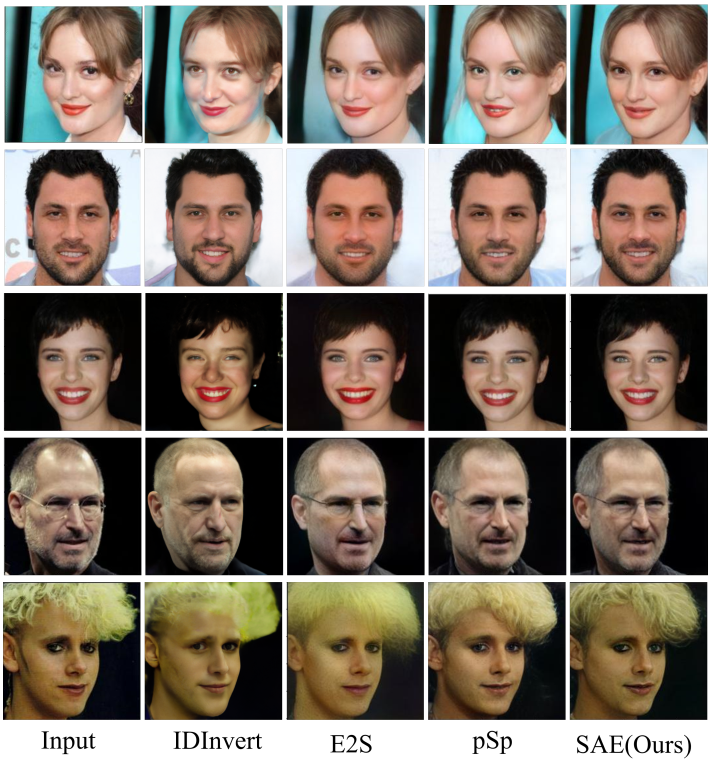

Figure 5: SAE with other GAN inversion models

As seen in the Figure 5, our SAE module shows slightly better performance in finer details compared to the results using pSp, IDInvert [13] and E2Style models. This indicates that our proposed CBAM loss function enhances the performance of GAN.

7 Experimental Results of Abortion method

To demonstrate the benefits of abortion in the training, we compared our module with the existing modules using multiple images.

Implementation detail. For absolute abortion method to be used, prior knowledge of the loss function in E2Style is necessary. Therefore, in our experiments, we used the relative abortion method instead. The initial weights for the weighting factor of each loss function were set as 1, 0.8, 0.5, and 1, respectively. We trained our model using a dataset of 25,000 images from CelebAMask-HQ [5] and evaluated it on a randomly selected set of 5,000 images that were not used in the training set.

Results. Examples of the training results for our model and other models are shown below.

Fig.6 shows that our model demonstrates excellent detail reproduction of the original images. Our model excels at implementing small details such as visible teeth in an open mouth and direction-

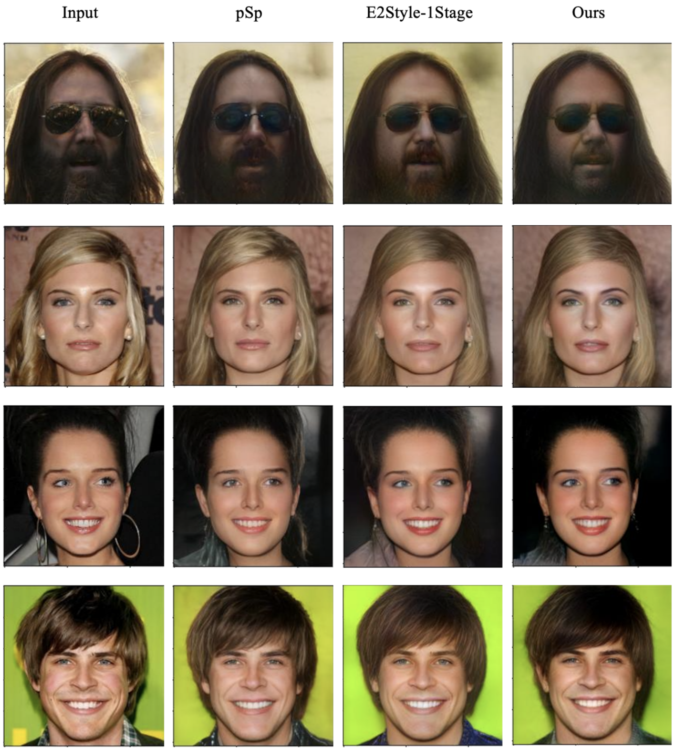

Figure 6: Visual comparison of the GAN inversion models

ally oriented pupils, which set our model apart from others. This result can be thought to have occurred by preventing overfitting to the training set through our abortion method.

In terms of the amount of computation for training, pSp model has only 1 fixed stage, and the E2Style model has N fixed stages. If solely one stage is passed to be trained, the performance will be lower than a model that passes through N-stages. However, a model with N fixed stages carries a greater risk of overfitting than a model that completes only one stage. Moreover, N-stages require N times more operations on all training sets, resulting in a slower learning speed. Our model is implemented in a direction that overcomes these two drawbacks, starting from the N-stages and gradually reducing the number of stages, maintaining a performance level similar to that of a model that completes the N-stages while dealing with overfitting and reducing the computational load.

8 Conclusion

We suggest two novel approaches to improve image-to-image generation quality between state-of-art models. These two modules presented both showed better results compared to the most recent results. SAE which includes the combination of the attention module and encoder, may result in better reconstruction of images, finer control of image translation, and improved handling of complex tasks. After adding the attention module and creating an attention loss for training, the qualitative performance improved compared to the pSp, IDInvert, and E2Style methods. Abortion method allows the model to process dynamic stages to prevent overfitting by introducing the abortion technique to the E2Style model. Our model demonstrates three advantages over the previous model, namely, **i**. reducing overfitting, **ii**. decreasing computational complexity compared to the fixed N-stage E2Style model, and **iii**. preserving fine-grained details of the original images.

References

[1] Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D. (2021). Encoding in Style: A StyleGAN Encoder for Image-to-Image Translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[2] Woo, S., Park, J., Lee, J. Y., Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV).

[3] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 586–595, 2018.

[4] J. Deng, J. Guo, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4685–4694, 2019.

[5] Z. Liu, https://github.com/switchablenorms/CelebAMask-HQ/tree/ master/face parsing, accessed: Mar. 2021. [Online].

[6] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” ArXiv, vol. abs/1710.10196, 2018.

[7] Tero Karras, Samuli Laine, Timo Aila. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. eprint arXiv:1812.04948, ()

[8] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Weiming Zhang, Lu Yuan, Gang Hua, Fellow, IEEE, NEnghai Yu. (2022). E2Style: Improve the Efficiency and Effectiveness of StyleGAN Inversion. Wei-2022, 3267-3280

[9] Goodfellow, Ian, et al. ”Generative adversarial networks.” Communications of the ACM 63.11 (2020): 139-144.

[10] Xia, Weihao, et al. ”Gan inversion: A survey.” IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).

[11] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Communications of the ACM 60.6 (2017): 84-90.

[12] He, Kaiming, et al. ”Deep residual learning for image recognition.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[13] Zhu, Jiapeng, et al. ”In-domain gan inversion for real image editing.” Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16. Springer International Publishing, 2020.

[14] Lin, Tsung-Yi, et al. ”Feature pyramid networks for object detection.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.