Supplementary Material: Encouraging Disentangled and Convex Representation with Controllable Interpolation Regularization

Yunhao Ge, Zhi Xu, Yao Xiao, Gan Xin, Yunkui Pang, Laurent Itti
University of Southern California, Los Angeles, CA, USA
yunhaoge@usc.edu, itti@usc.edu

Appendix

1. Network Architecture and Training Details

1.1. ELEGANT [8] + CIR

Network Structure

For our ELEGANT + Controllable Interpolation Regularization (CIR), we use the same network architecture as the original ELEGANT paper [8]. We use an autoencoder-structure generator G with an encoder E and a decoder D. The E and D structures are symmetrical with an architecture consisting of five convolutional layers. As for the discriminator D, it adopts multi-scale discriminators $D1$ and $D2$. Both $D1$ and $D2$ use a CNN architecture with four convolutional layers followed by a fully-connected layer. The difference between them is that $D1$ has a larger fully-connected layer while the one of $D2$'s is small.

Training Details

We train the ELEGANT and ELEGANT + CIR models on *CelebA* [4]. The size of input images is 256×256. Both generator and discriminator use Adam with β_1=0.5 and β_2=0.999, batch size 16, learning rate 0.0002 at first and multiply 0.97 every 3000 epochs.

Hyperparameters in the loss function: For reconstruction loss and Adversarial loss, we use $\lambda_{reconstruction} = 5$, $\lambda_{adversarial} = 1$ unmodified. For Controllable Interpolation Regularization loss, we set $\lambda_{CIR} = 1 \times 10^7$ to make the regularization loss has a similar scale as other loss terms and balance the training.

Disentangle details: The encoder of generator G maps an image into a latent code with shape $(512 \times 8 \times 8)$, and ELEGANT will dynamically allocate these spaces to store information of the interesting attributes. For instance, suppose the attributes we want to disentangle are eyeglasses and mustache. Then the input will be [eyeglasses, mustache], and the first half of latent space will store the information of eyeglasses. In other words, we disentangle the latent space along the first dimension and both eyeglasses and mustache get $(256 \times 8 \times 8)$ latent space.

1.2. I2I-Dis [2] + CIR

Network Structure

We use the same network architecture as the original I2I-Dis paper [2]. For all the experiments in this section, we use images from *cat2dog* dataset with size 216×216. There are four modules in I2I-Dis: shared content encoder E^c, domain-specific attribute encoder E^a, generator G, discriminator D. For the shared content encoder E^c, we use an architecture consisting of three convolutional layers followed by four residual blocks. For the domain-specific attribute encoder E^a, we use a CNN architecture with four convolutional layers followed by fully-connected layers. For the generator G, we use an architecture consisting of four residual blocks followed by three fractionally stridden convolutional layers. For the discriminator D, we use an architecture consisting of four convolutional layers followed by fully-connected layers. Our disentangled latent code consists of two-part: shared content attribute latent code z_c with shape $256 \times 54 \times 54$ and domain-specific attribute latent code z_a with shape 8×1.

Training Details

The training of I2I-Dis and I2I-Dis + CIR use Adam optimizer with batch size of 1, learning rate of 0.0001, and exponential decay rates $\beta_1 = 0.5, \beta_2 = 0.999$.

Hyper-parameters in loss function: For reconstruction loss, we use $\lambda_1^{rec} = 10, \lambda_{cc} = 10$. For adversarial loss, we use $\lambda_{adv}^{content} = 1, \lambda_{adv}^{domain} = 1$. For latent regression loss, we use $\lambda_1^{latent} = 10$. For KL divergence loss, we use $\lambda_{KL} = 0.01$. For our controllable interpolation regularization loss, we use $\lambda_{CIR} = 10$.

1.3. GZS-Net [1] + CIR

Network Structure

We use the same network architecture and the same dataset (*Fonts* [1]) as the original GZS-Net paper [1]. The input images are of size 128×128. There are two modules in GZS-Net: an encoder E and a decoder D. The *Fonts* dataset have 5 attributes: content, size, font color, background color and font. Each attribute takes 20 dimensions in the latent space and thereby sums up to a 100-dimensional

vector. The encoder E is composed of two convolutional layers with stride 2, followed by three residual blocks. Then it comes with a convolutional layer with stride 2, followed by a flatten layer that reshapes the response map to a vector. Finally, two fully-connected layers output 100-dimensional vectors as the latent feature. The decoder D mirrors the encoder, composed of two fully-connected layers, followed by a cuboid-reshaping layer. The next is a deconvolutional layer with stride 2, followed by three residual blocks. And finally, two deconvolutional layers with stride 2 produce a synthesized image.

Training Details

We train GZS-Net and GZS-Net + CIR on *Fonts* [1] dataset. We use Adam optimizer with batch size of 8, learning rate of 0.0001, and exponential decay rates $\beta_1 = 0.9, \beta_2 = 0.999$.

Hyper-parameters in loss function: For reconstruction loss, we use $\lambda_1^{rec} = 1, \lambda_{combine} = 1$. For our controllable interpolation regularization loss, we use $\lambda_{CIR} = 0.0001$ at an early stage and $\lambda_{CIR} = 0.01$ after 100000 epochs to balance the training.

2. More Qualitative Results

Fig. 1 shows a larger version of main paper Fig. 1 to show more details.

2.1. ELEGANT [8] + CIR

Fig. 2 shows more results of the task 1 performance on two images face attribute transfer, which is similar to the main paper Fig. 3. We offer three rows for each attribute, including a new attribute (Mouth-Open vs. Mouth-Close).

Fig. 3 shows more results of the task 2 performance on face image generation by exemplars, which is similar to the main paper Fig. 4 but with bangs as our disentangle attribute. The results show that CIR can help to overcome the mode collapse problem in ELEGANT.

2.2. I2I-Dis [2] + CIR

Fig. 4 shows more results of the image-to-image translation, which is similar to the main paper Fig. 5. (a) We generate cat images given fixed identity (domain) attribute latent code and change the 'content' attribute latent code by interpolation. (b) We generate dog images given fixed content attribute latent code and change the 'identity' attribute latent code by sampling.

2.3. GZS-Net [1] + CIR

Fig. 5 shows more results of the interpolation-based controllable synthesis performance on font color, background color, size, and font attributes.

3. Quantitative Experiments Details

3.1. Disentanglement Evaluation by Correlation Coefficient.

We use Spearman's Rank Correlation for latent space correlation computation. It is computed as:

$$r_s = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{1}$$

Here rg_X and rg_Y means the rank variables of X and Y. cov is the covariance function. σ denotes the standard variation.

For ELEGANT + CIR that disentangles eyeglasses and mustache, we collect 10,000 images from *CelebA* [4] and obtain the same number of $(512 \times 8 \times 8)$ latent matrices from encoder. Then we average the vectors along the 2^{nd} and 3^{rd} dimensions and produce squeezed matrices of size 512. This preprocessing step is following the interpolation strategy, which helps to display the intra correlation more clearly.

For GZS-Net + CIR, 10,000 images are fetched from *Fonts* and corresponding latent vectors with size 100 are computed. No preprocessing is applied.

All the latent matrices (or vectors) are normalized before putting into Spearman's Rank Correlation calculation. The normalization is calculated as:

$$norm(v_i) = (v_i - \overline{v_i}) \;/\; \sigma_{v_i},\ \forall\, i \in \{1, 2, ..., |v|\} \tag{2}$$

v_i is the value of each dimension i in v. $\overline{v_i}$ is the average of v_i and σ_{v_i} is the standard variance.

4. More Downstream Tasks and Applications

We conduct more experiments to demonstrate 3 potential applications with the more convex and robust disentangled latent space by CIR.

Data Augmentation. We design a letter image classification experiment with *Fonts* [1] to evaluate how interpolation-based controllable synthesis ability, empowered by CIR, as a data augmentation method, improves the downstream classification task. We tailored three datasets from *Fonts*, each of them has ten letters as labels. The large training set (D_L) and testing set (D_{test}) have the same number of images with the same attribute values. We take a subset of D_L to form a small training set D_S with fewer attribute values. For data augmentation, we first train the GZS-Net and GZS-Net + CIR on D_S, and then we use the trained models to generate 1000 new images by interpolation-based attribute controllable synthesis. We combine the synthesized images with D_S and form two augmented training sets D_{S+G} (GZS-Net) and D_{S+G+C} (GZS-Net + CIR), respectively. All test accuracy shown in (Table 1), which shows an improved data augmentation performance on downstream tasks with the help of CIR. (more details in Supplementary)

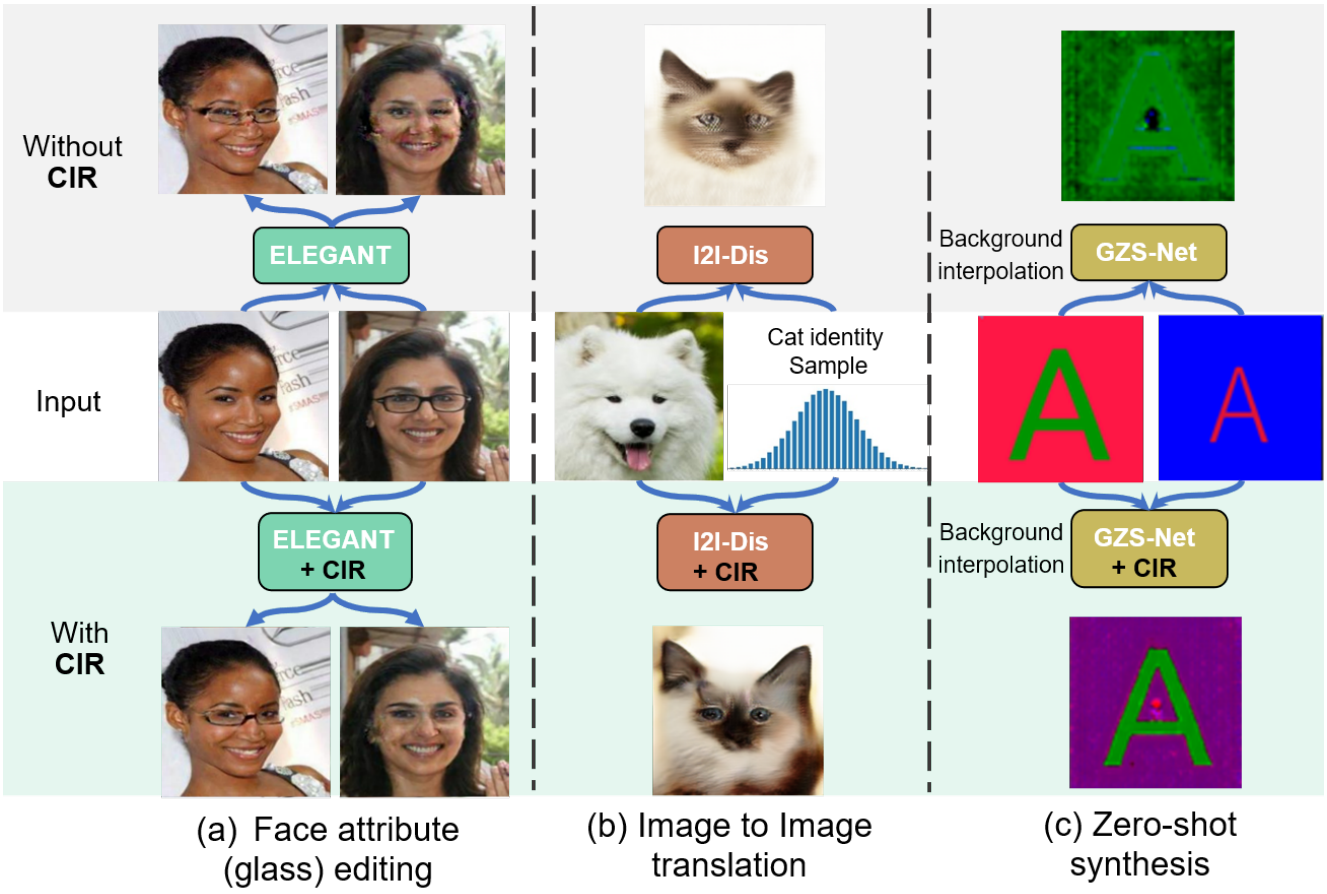


Figure 1. Our proposed approach CIR improves the result quality of 3 tasks by encouraging both disentanglement and convexity in the latent space: (a) Face attribute editing with ELEGANT (add/remove glasses on face); CIR is better able to transfer glasses with less disturbance on other face parts. (b) Image to image translation transfer from a dog image to a cat image with same pose (content); CIR better matches the desired pose with fewer artifacts. (c) Zero-shot synthesis with GZS-Net to synthesize an image with a new background by interpolating in the corresponding latent space; CIR better interpolates the background only without changing letter size, color or font style.

Table 1. Controllable augmentation performance (the $\star$ means that synthesized images with new attributes are added into training set)

Attribute \ Dataset	D_L	D_S	D_{S+G}	D_{S+G+C}	D_{test}
Size	3	2	2$\star$	2$\star$	3
Font Color	6	3	3$\star$	3$\star$	6
Back Color	3	3	3$\star$	3$\star$	3
Fonts	10	3	3$\star$	3$\star$	10
Dataset Size	5400	540	540+1000	540+1000	5400
Train Accuracy	98%	99%	99%	99%	N/A
Test Accuracy	94%	71%	74%	**76%**	N/A

Bias Elimination for Fairness. Dataset bias may influence the model performance significantly. [5] listed lots of bias resources and proved that eliminating bias is significant. A more convex and disentangled representation with CIR could be a solution to the bias problem by first disentangle the bias attribute and then remove them in the final decision. We use the *Fonts* dataset to simulate the bias problem. We tailored three datasets, a biased training dataset $\mathcal{D}^{\mathrm{B}}$, two unbiased dataset: $\mathcal{D}^{\mathrm{UB}}$ for training and $\mathcal{D}^{\mathrm{T}}$ for test. In $\mathcal{D}^{\mathrm{B}}$, we entangle the two attributes, letter and background color, as dataset bias. $\mathcal{D}^{\mathrm{B}}$ consists of three-part: G1, G2, and G3, where each letter has 1, 3, and 6 background colors, respectively. (more details in Supplementary) Then, we use $\mathcal{D}^{\mathrm{B}}$ and $\mathcal{D}^{\mathrm{UB}}$ to train letter classifier with resnet-18 respectively and test on $\mathcal{D}^{\mathrm{T}}$ as the control groups. As is shown in Table. 2, the classifier trained on $\mathcal{D}^{\mathrm{B}}$, only gets 81% test accuracy while classifier trained on $\mathcal{D}^{\mathrm{UB}}$ obtains 99% test accuracy. As shown in Fig. 8, Grad-Cam's [6] results proved that the classifier would regard background color as essential information if it entangled with letters. We use the more convex and disentangled representation of CIR to solve the entangled bias in $\mathcal{D}^{\mathrm{B}}$. We first train a GZS-Net + CIR use $\mathcal{D}^{\mathrm{B}}$. Then we train a letter classifier on the latent representation instead of image space, where we explicitly drop the background color-related dimensions (bias attribute) and use the rest of the latent code as input. After training, the accuracy rises to 98%. Hence, we eliminate the dataset bias with the help of robust disentangled latent by CIR.

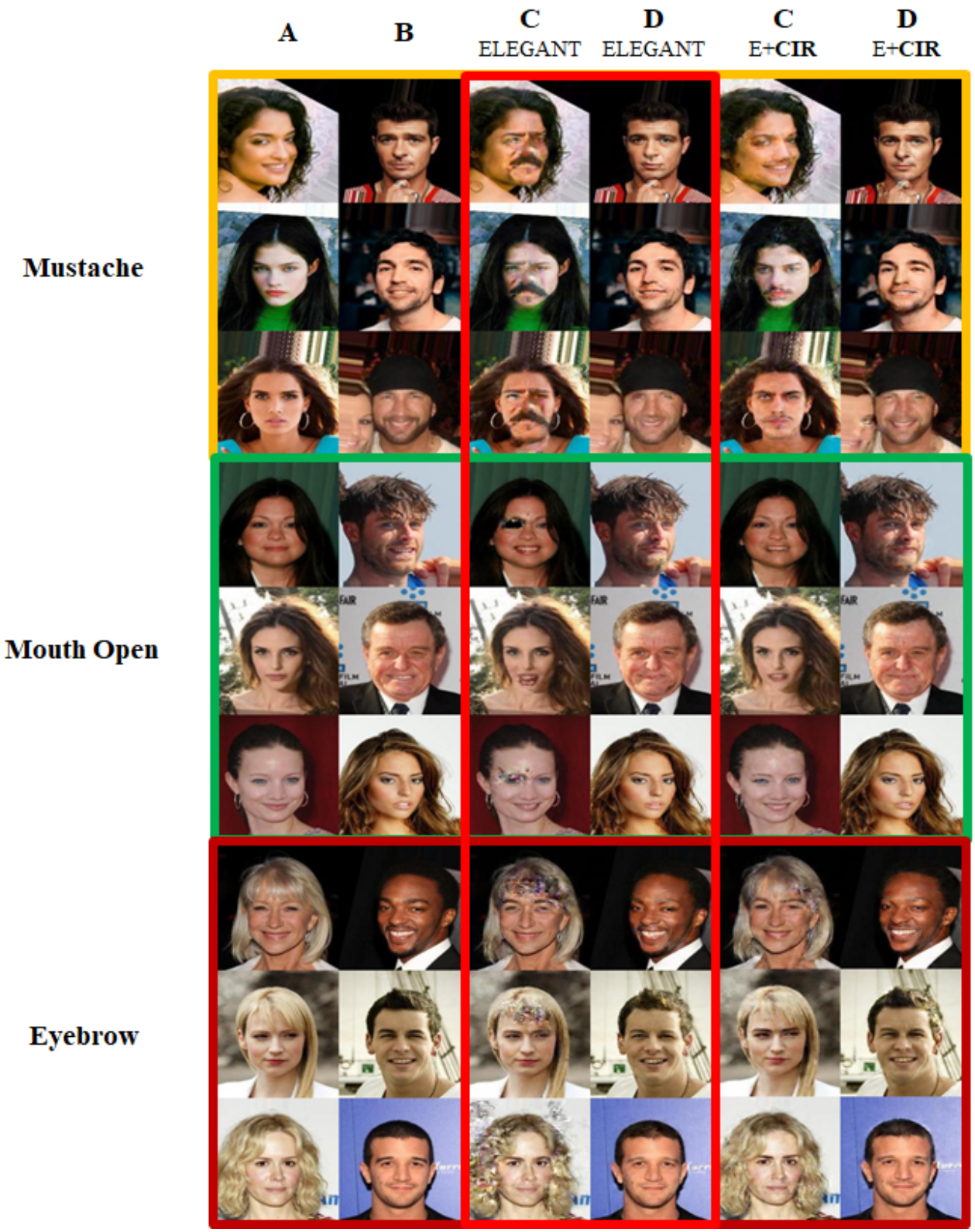


Figure 2. More examples of ELEGANT+**CIR** (E+**CIR**) performance of task 1 for two images face attribute transfer

Figure 3. ELEGANT + **CIR** Performance of task 2 for face image generation by exemplars

Table 2. Bias elimination experiment results

Dataset \ Model	resnet18 $\mathcal{D}^{\mathrm{B}}$	resnet18 $\mathcal{D}^{\mathrm{UB}}$	GZS-Net + CIR $\mathcal{D}^{\mathrm{B}}$
Test(Letters in G1)	52.73%	99.17%	96.77%
Test(Letters in G2)	82.63%	98.67%	98.97%
Test(Letters in G3)	99.13%	98.30%	98.46%
Train	99.44%	98.82%	99.98%
Test	**81.32%**	98.63%	**98.11%**

Table 3. Bias elimination dataset setting

Dataset		Number of letters	Number of colors
$\mathcal{D}^{\mathrm{B}}$	G1	15	1
	G2	15	3
	G3	22	6
$\mathcal{D}^{\mathrm{UB}}$		52	6
$\mathcal{D}^{\mathrm{T}}$		52	6

Mining New Attribute Value. Fig.6 shows our results of mining new attribute value. To find a good exploration direction and mine new attribute values, we explore the distribution of each attribute value in the corresponding attribute-convex latent space (e.g., the distribution of different background colors in a convex background color latent space: $\mathcal{A}_{back} = \{\text{blue}, \text{red}, \text{green}, \text{yellow}, \dots\}$).

Two common kinds of distribution are considered:
1) **Gaussian**. For those attributes (object color) whose attribute value (blue color) has slight intra-class variance (all blues look similar), their distribution can be seen as a Gaussian distribution. We can use K-means [3] to find the center of each object color and guide the interpolation and synthesis.
2) **Non-Gaussian**. We treat each attribute value as a binary semantic label (e.g., wear glasses or not wear glasses). We assume a hyperplane in the latent space serving as the separation boundary [7], and the distance from a sample to this hyperplane is in direct proportion to its semantic score. We can train an SVM to find this boundary and use the vector orthogonal to the border and the positive side to represent a Unit Direction Vector (UDV). We can then use the UDVs or a combination to achieve precise attribute synthesis and find new attribute values. As shown in Fig. 9 (a), we can find the boundaries and UDVs by SVM for each attribute value. To solve the precision problem in attribute synthesis, Fig. 9 (c) shows moving towards the z value of the cluster center directly for Gaussian; Fig. 9 (d) shows moving from the start point, across the boundary, to the target attribute value, by adding the UDV of the target attribute for non-Gaussian. Fig. 9 (b) shows that we can combine the UDVs to discover new attribute values.

Here we explore the distribution of disentangled representation and mining the relationship between movement in high dimension x space and low dimension z space to answer the question: Which direction of movement can help

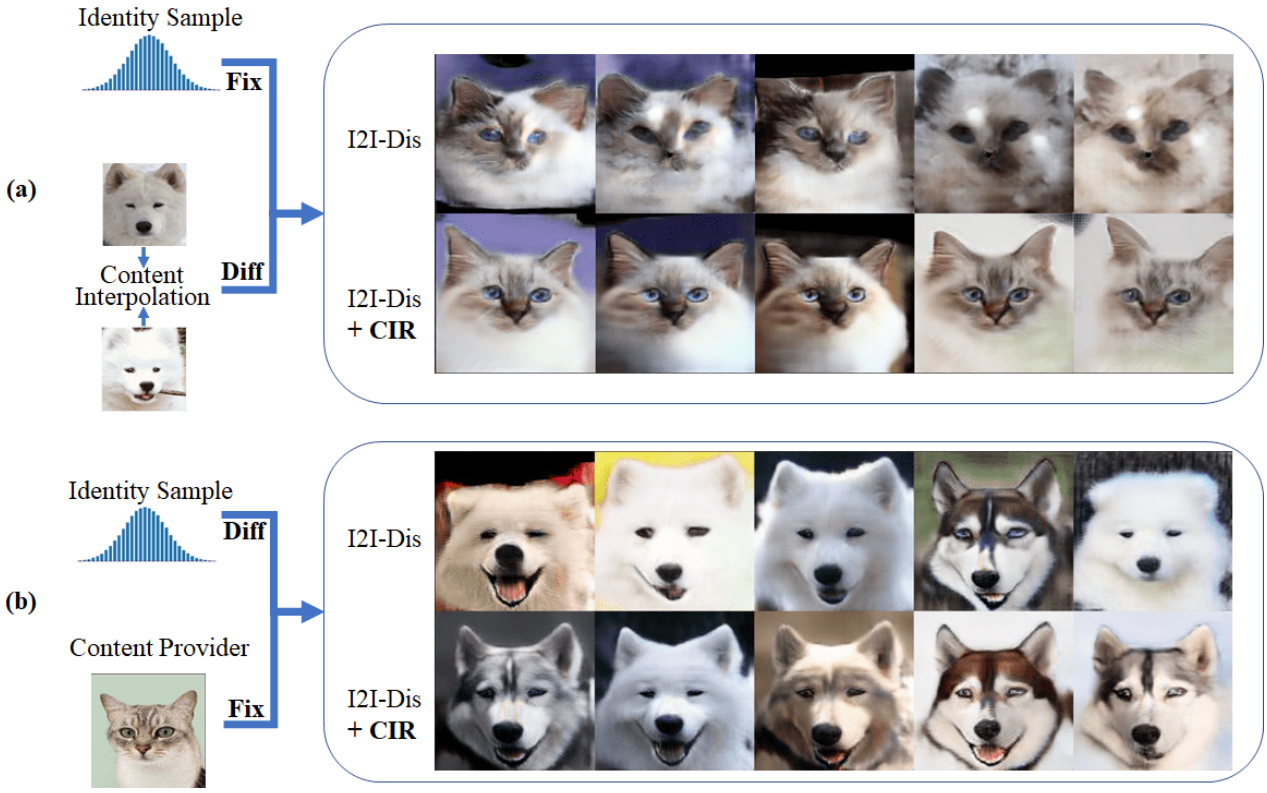


Figure 4. I2I-Dis + **CIR** performance of diverse image-to-image translation

us to find new attributes?

For each background color, we train a binary color classifier to label interpolated points in the z space and assign a color score for each of them. Then we use SVM to find the boundary and obtain UDV for this attribute value. Since the UDV is the most effective direction to change the semantic score of samples, if we move z value of the given image towards UDV, its related semantic score would increase fast. To explore more new attributes, the combination of UDVs may be a good choice. For instance, if the given picture is green, the new colors may fall in the path from green to blue and the path from green to red. Thus, it is reasonable to set our move direction as $v = v_{blue} + v_{red} - v_{green}$ (v represents UDV). The 1^{st} row of Fig. 7 shows the results of changing z value with the combine vector $v_{blue} + v_{red} - v_{green}$. On the contrast, the 2^{nd} row only use v_{blue} and the 3^{rd} row only use v_{red}. We can find that both the 2^{nd} and the 3^{rd} row only find one color while the 1^{st} row finds more.

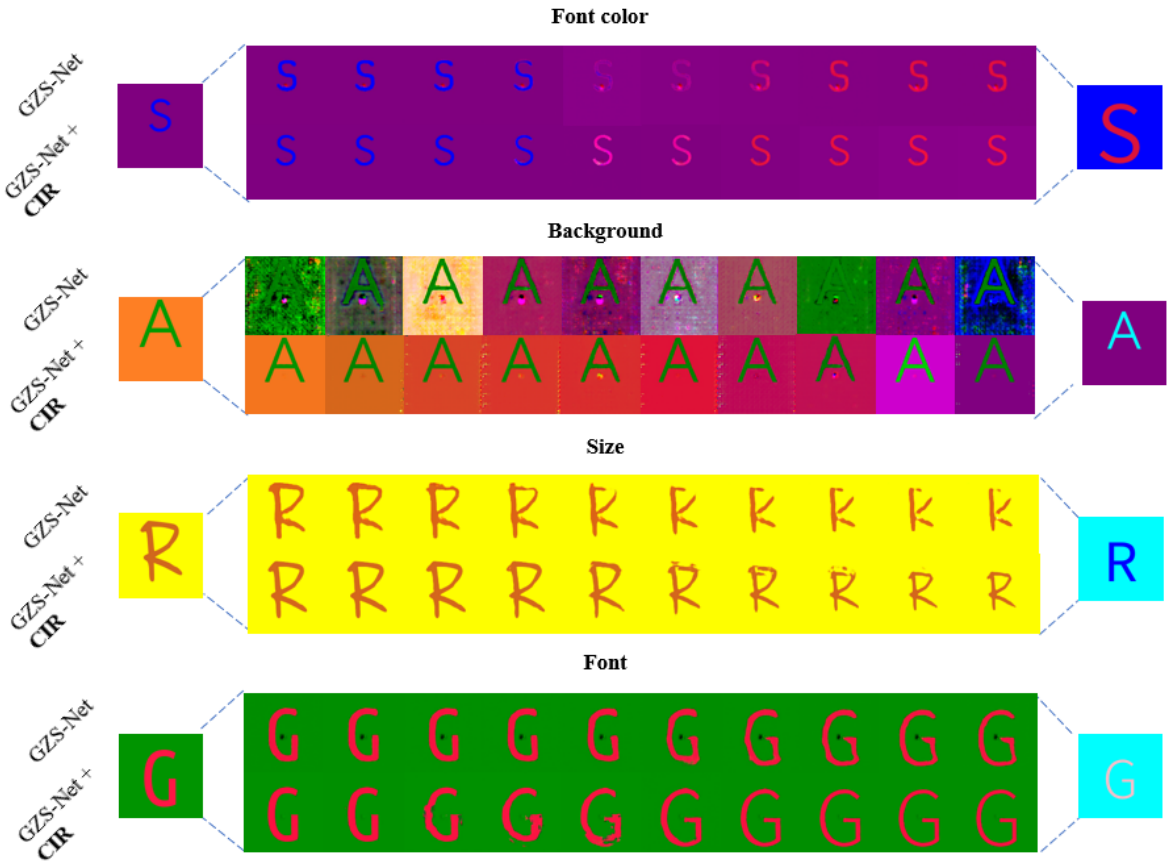


Figure 5. More results of GZS-Net + **CIR** performance of interpolation-based attribute controllable synthesis

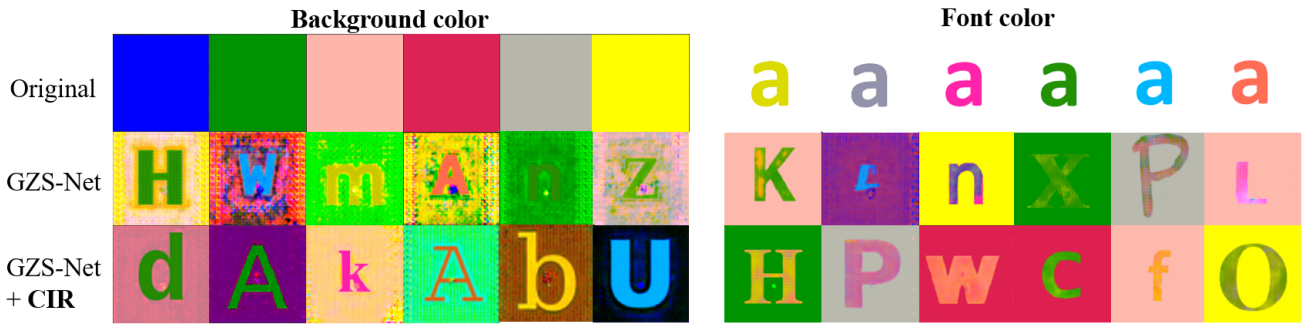


Figure 6. Controllable mining novel background and font color by interpolation in latent space.

Figure 7. Mining new attribute values with UDV

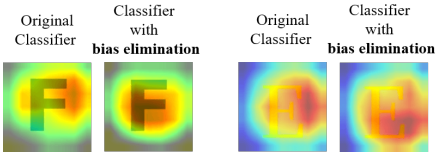
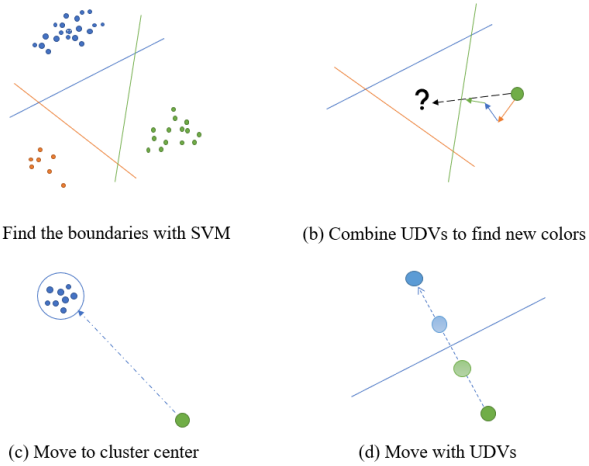


Figure 8. The influence of bias shown by Grad-Cam



(a) Find the boundaries with SVM

(b) Combine UDVs to find new colors

(c) Move to cluster center

(d) Move with UDVs

Figure 9. Towards controllable exploration direction

References

[1] Yunhao Ge, Sami Abu-El-Haija, Gan Xin, and Laurent Itti. Zero-shot synthesis with group-supervised learning. *arXiv preprint arXiv:2009.06586*, 2020.

[2] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018.

[3] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[4] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[5] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning, 2019.

[6] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[7] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *arXiv preprint arXiv:2005.09635*, 2020.

[8] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–187, September 2018.