ENCOURAGING DISENTANGLED AND CONVEX REPRESENTATION WITH CONTROLLABLE INTERPOLATION REGULARIZATION

Anonymous authors
Paper under double-blind review

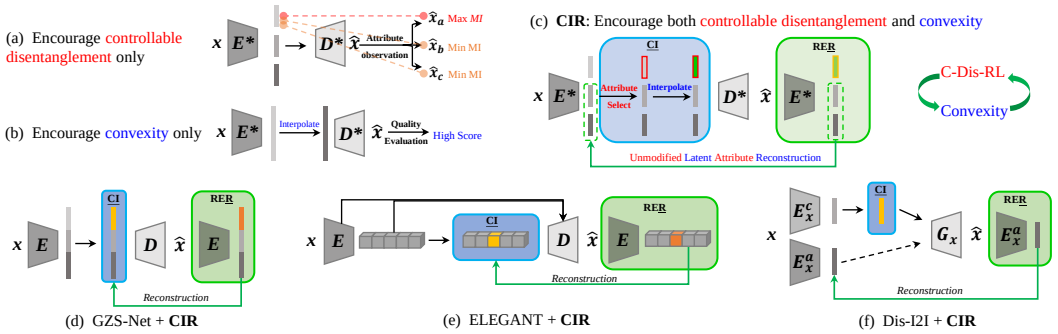
ABSTRACT

We focus on controllable disentangled representation learning (C-Dis-RL), where users can control the partition of the disentangled latent space to factorize dataset attributes (concepts) for downstream tasks. Two general problems remain underexplored in current methods: (1) They lack comprehensive disentanglement constraints, especially missing the minimization of mutual information between different attributes across latent and observation domains. (2) They lack convexity constraints in disentangled latent space, which is important for meaningfully manipulating specific attributes for downstream tasks. To encourage both comprehensive C-Dis-RL and convexity simultaneously, we propose a simple yet efficient method: Controllable Interpolation Regularization (CIR), which creates a positive loop where the disentanglement and convexity can help each other. Specifically, we conduct controlled interpolation in latent space during training and 'reuse' the encoder to help form a 'perfect disentanglement' regularization. In that case, (a) disentanglement loss implicitly enlarges the potential 'understandable' distribution to encourage convexity; (b) convexity can in turn improve robust and precise disentanglement. CIR is a general module and we merge CIR with three different algorithms: ELEGANT, I2I-Dis, and GZS-Net to show the compatibility and effectiveness. Qualitative and quantitative experiments show improvement in C-Dis-RL and latent convexity by CIR. This further improves downstream tasks: controllable image synthesis, cross-modality image translation and zero-shot synthesis. More experiments demonstrate CIR can also improve other downstream tasks, such as new attribute value mining, data augmentation, and eliminating bias for fairness.

1 INTRODUCTION

Disentangled representation learning empowers models to learn an orderly latent representation, in which each separate set of dimensions is responsible for one semantic attribute (Higgins et al., 2016; Chen et al., 2016; Zheng et al., 2019). If we categorize different disentangled representation methods by whether they could *control* the partition of the obtained disentangled latent representation (e.g., explicitly assign first 10 dimensions to be responsible for face attribute), there are two main threads:

(1) **Uncontrollable** disentangled methods, such as Variational Autoencoders (VAEs) (Kingma & Welling, 2014; Higgins et al., 2017; Tran et al., 2017), add prior (e.g., Gaussian distribution) constraints in latent space to implicitly infer a disentangled latent code. Most of them are unsupervised methods that can easily generalize to different datasets and extract latent semantic factors. Yet, they struggle to obtain controllable disentanglement because the unsupervised latent encoding does not map onto user-controllable attributes. (2) **Controllable** disentangled methods, which explicitly control the partition of the disentangled latent space and the corresponding mapping to semantic attributes by utilizing dataset attribute labels or task domain knowledge. Because users can precisely control and design their task-driven disentangled latent representation, controllable disentanglement methods are widely used in various downstream tasks: in cross-modality image-to-image translation, I2I-Dis (Lee et al., 2018) disentangle content and attribute to improve image translation quality; In controllable image synthesis, ELEGANT (Xiao et al., 2018) and DNA-GAN (Xiao et al., 2017) disentangle different face attributes to achieves face attribute transfer by exchanging certain part of their latent encoding across images. In group supervised learning, GZS-Net (Ge et al., 2020a) uses disentangled representation learning to simulate human imagination and achieve zero-shot synthesis.



(d) GZS-Net + **CIR** (e) ELEGANT + **CIR** (f) Dis-I2I + **CIR**

Figure 1: (a-c) Intuitive understanding of Controllable Interpolation Regularization (**CIR**). (a) Only encourage controllable disentangled representation with general Mutual Information (MI) constrain method: maximize the MI between the same attribute across latent and observation domains while minimizing the MI between the different attribute across latent and observation domains. (b) Only encourage convexity with interpolation and image quality evaluation. (c) A simple yet efficient method, CIR, encourages both controllable disentanglement and convexity in latent representation. CIR consists of a Controllable Interpolation (CI) module and a Reuse Encoder Regularization (RER) module. (d-e) **CIR** compatible to different models. (d) GZS-Net (Ge et al., 2020a) + **CIR** (e) ELEGANT (Xiao et al., 2018) + **CIR**. (f) I2I-Dis (Lee et al., 2018) + **CIR**

However, controllable disentangled methods suffer from 2 general problems: 1) The constraints on disentanglement are partial and incomplete, they lack *comprehensive* disentanglement constraints. For example, while ELEGANT enforces that modifying the part of the latent code assigned to an attribute (e.g., hair color) will affect that attribute, it does not explicitly enforce that a given attribute will *not* be affected when the latent dimensions for other attributes are changed. 2) Most of the above-mentioned downstream tasks require manipulating specific attribute-related dimensions in the obtained disentangled representation, for instance, changing only the style while preserving the content in an image-to-image translation task. For such manipulation, the convexity of each disentangled attribute representation (i.e., interpolation within that attribute should give rise to meaningful outputs) is not guaranteed by current methods. Further, convexity demonstrates an ability to "generalize", which implies that the autoencoder structure has not simply memorized the representation of a small collection of data points. Instead, the model uncovered some structure about the data and has captured it in the latent space (Berthelot et al., 2018). How to achieve both comprehensive disentanglement, and convexity in the latent space, is under-explored.

To solve the above problems, we first provide a definition of controllable disentanglement with the final goals of *perfect* controllable disentanglement and of convexity in latent space. Then, we use information theory and interpolation to analyze different ways to achieve disentangled and convex representation learning, respectively. To optimize them together, based on the definition and analysis, we use approximations to create a positive loop where disentanglement and convexity can help each other. We propose Controllable Interpolation Regularization (CIR), a simple yet effective general method that compatible with different algorithms to encourage both controllable disentanglement and convexity in the latent space. Specifically, previous methods use a general autoencoder structure and pre-assign latent code dimensions to specific attributes or concepts to obtain controllable disentangled latent space. CIR first conducts controllable interpolation, i.e., controls which attribute to interpolate and how in the disentangled latent space, then 'reuses' the encoder to 're-obtain' the latent code and add regularization to explicitly encourage *perfect* controllable disentanglement and implicitly boost convexity. We show that this iterative approximation approach converges towards perfect disentanglement and convexity in the limit of infinite interpolated samples.

Our contributions are: (i) Describe a new abstract framework for *perfect* controllable disentanglement and convexity in the latent space, and use information theory to summarize potential optimization methods. (ii) Propose Controllable Interpolation Regularization (CIR), a general module compatible with different algorithms, to encourage both controllable disentanglement and convex in latent representation by creating a positive loop to make them help each other. CIR is shown to converge towards perfect disentanglement and convexity for infinite interpolated samples. (iii) Demonstrate that better disentanglement and convexity are achieved with CIR on various tasks: controllable image synthesis, cross-domain image-to-image translation and group supervised learning. (iv) Demonstrate how CIR with the encouraged controllable disentangled and convexity representation learning can improve the performance of more downstream tasks: new semantic attribute mining, controllable data augmentation, and eliminating dataset bias for fairness.

2 RELATED WORK

Controllable Disentangled Representation Learning (Controllable-Dis-RL) is different from Uncontrollable Dis-RL (such as VAEs (Kingma & Welling, 2014; Higgins et al., 2017; Chen et al., 2018)), which implicitly achieves disentanglement by incorporating a distance measure into the objective, encouraging the latent factors to be statistically independent. However, these methods and not able to freely control the relationship between attribute and latent dimensions. Controllable Dis-RL learns a partition control of the disentanglement from semantic attribute labels in the latent representation and boosts the performance of various tasks: ELEGANT (Xiao et al., 2018) and DNA-GAN (Xiao et al., 2017) for face attribute transfer; I2I-Dis (Lee et al., 2018) for diverse image-to-image translation; DGNet (Zheng et al., 2019) and IS-GAN (Eom & Ham, 2019) for person re-identification; GZS-Net (Ge et al., 2020a) for controllable zero-shot synthesis with group-supervised learning. However, their constraints on disentanglement are implicit and surrogate by image quality loss, which also misses the constraint between different attributes across latent and observation. As a general module, CIR is compatible and complementary with different Controllable Dis-RL algorithms by directly constraining disentanglement while focusing on minimizing the mutual information between different attributes across latent and observation.

Convexity of Latent Space is defined as a set in which the line segment connecting any pair of points will fall within the rest of the set (Sainburg et al., 2018). Linear interpolations in a low-dimensional latent space often produce comprehensible representations when projected back into high-dimensional space (Engel et al., 2017; Ge et al., 2020b). However, linear interpolations are not necessarily justified in many controllable disentanglement models because latent-space projections are not trained explicitly to form a convex set. VAEs overcome non-convexity by forcing the latent representation into a pre-defined distribution, which may be a suboptimal representation of the high-dimensional dataset. GAIN (Sainburg et al., 2018) adds interpolation in the generator in the middle latent space and uses a discriminative loss on a GAN structure to help optimize convexity. Our method controls the interpolation in a subspace of the disentangled latent space and uses disentanglement regularization to encourage a convex latent space for each semantic attribute.

3 CONTROLLABLE INTERPOLATION REGULARIZATION

3.1 MUTUAL INFORMATION FOR *Perfect* CONTROLLABLE DISENTANGLEMENT

A general autoencoder structure $(D \circ E) : \mathcal{X} \to \mathcal{X}$ is composed of an encoder network $E : \mathcal{X} \to \mathbb{R}^d$, and a decoder network $D : \mathbb{R}^d \to \mathcal{X}$. $\mathbb{R}^d$ is a latent space, compared with the original input space $\mathcal{X}$ (e.g., image space). The disentanglement is a property of latent space $\mathbb{R}^d$ where each separate set of dimensions is responsible for one semantic attribute of given dataset. Formally, a dataset (e.g., face dataset) contains n samples $\mathcal{D} = \{x^{(i)}\}_{i=1}^n$, each accompanied by m attributes $\mathcal{D}_a = \{(a_1^{(i)}, a_2^{(i)}, \dots a_m^{(i)})\}_{i=1}^n$. Each attribute $a_j \in \mathcal{A}_j$ can be either binary (two attribute values, e.g., $\mathcal{A}_1$ may denote wearing glass or not; $\mathcal{A}_1 = \{\text{wear glass}, \text{not wear glass}\}$), or a multi-class attribute, which contains a countable set of attribute values (e.g., $\mathcal{A}_2$ may denote hair-colors $\mathcal{A}_2 = \{\text{black}, \text{gold}, \text{red}, \dots\}$). Controllable disentangled representation learning (Controllable Dis-RL) methods have two properties: (1) Users can explicitly control the partition of the disentangled latent space $\mathbb{R}^d$ and (2) Users can control the semantic attributes mapping between $\mathbb{R}^d$ to input space $\mathcal{X}$. To describe the ideal goal for all Controllable Dis-RL, we define a *perfect* controllable disentanglement property in latent space $\mathbb{R}^d$ and the autoencoder.

Definition 1 *perfect* CONTROLLABLE DISENTANGLEMENT (*perfect*-C-D)$(E, D, \mathcal{D})$*: Given a general encoder $E : \mathcal{X} \to \mathbb{R}^d$, a decoder $D : \mathbb{R}^d \to \mathcal{X}$, and a dataset $\mathcal{D}$ with m independent semantic attributes $\mathcal{A}$, we say the general autoencoder achieve perfect controllable disentanglement for dataset $\mathcal{D}$ if the following property is satisfied: (1) For encoder E, if one attribute $\mathcal{A}_i$ of input x was specifically modified, transforming x into $\hat{x}$, after computing latent codes $z = E(x)$ and $\hat{z} = E(\hat{x})$, the difference between z and $\hat{z}$ should be zero for all latent dimensions except those that represent the modified attribute. (2) Similarly, for decoder D, the latent space change should only influence the corresponding attribute expression in the output (e.g., image) space.*

To encourage a general autoencoder structure model to obtain *perfect* controllable disentanglement property, we propose an information-theoretic regularization with two perspectives (Fig. 1(a)): (1) Maximize the mutual information between the *same* attribute across latent space $\mathbb{R}^d$ and observation input space $\mathcal{X}$; and (2) Minimize the mutual information between the *different* attributes across latent $\mathbb{R}^d$ and observation input space $\mathcal{X}$. Formally:

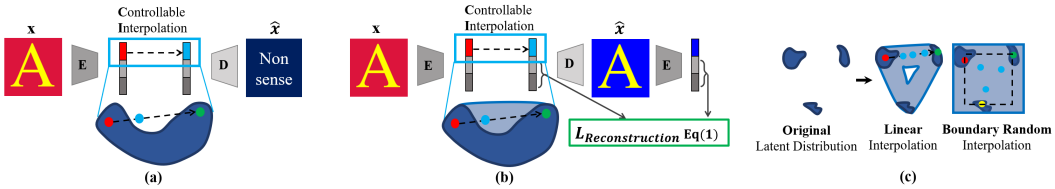


Figure 2: (a) Directly use Controllable Interpolation in GZS-Net (b) Architecture of GZS-Net + **CIR** (c) Convexity optimization with Linear Interpolation (middle) and Boundary Random Interpolation.

$$\begin{aligned} &\max_{E,D}\Big[I(x_{\mathcal{A}_i}, E(x)_{\mathcal{A}_i}) + I(E(x)_{\mathcal{A}_i}, D(E(x))_{\mathcal{A}_i})\Big]; i \in [1..m], \\ &\min_{E,D}\Big[I(x_{\mathcal{A}_i}, E(x)_{\mathcal{A}_j}) + I(E(x)_{\mathcal{A}_i}, D(E(x))_{\mathcal{A}_j})\Big]; i, j \in [1..m], i \neq j. \end{aligned} \tag{1}$$

where $x_{\mathcal{A}_i}$ and $D(E(x))_{\mathcal{A}_i}$ represent the observation of attribute $\mathcal{A}_i$ in $\mathcal{X}$ domain (e.g., hair color in human image); $E(x)_{\mathcal{A}_i}$ represents the dimensions in $\mathbb{R}^d$ that represent attribute $\mathcal{A}_i$.

3.2 Convexity Constraint with Interpolation

A convex latent space has the property that the line segment connecting any pair of points will fall within the rest of the space (Sainburg et al., 2018). As shown in Fig. 2(a), the dark blue region represents the 2D projection of the latent representation of one attribute for a dataset. This distribution would be non-convex, because the light blue point, though between two points in the distribution (the red and green points), falls in the space that does not correspond to the data distribution. This non-convexity may cause that the projection back into the image space does not correspond to a proper semantically meaningful realistic image. This limitation makes disentanglement vulnerable and hinders potential latent manipulation in downstream tasks. The result of Fig. 4 and 5in experiments illustrate this problem.

To encourage a convex data manifold, the usefulness of interpolation has been explored in the context of representation learning (Bengio et al., 2013) and regularization (Verma et al., 2018). As is shown in Fig. 1(b), we summarize the constraint of convexity in the latent space: we use a dataset-related quality evaluation function $Q()$ to evaluate the "semantic meaningfulness" of input domain samples; a higher value means high quality and more semantic meaning. After interpolation in latent space $\mathbb{R}^d$, we want the projection back into the original space to have a high $Q()$ score. Formally:

$$\max_{E,D}\Big\{\mathbb{E}_{x_1,x_2\in\mathcal{D}}\Big[Q(D(\alpha E(x_1) + (1-\alpha)E(x_2)))\Big]\Big\} \tag{2}$$

where x_1 and x_2 are two data samples and $\alpha \in [0..1]$ controls the latent code interpolation in $\mathbb{R}^d$. Fig. 2 (c) shows two kinds of interpolation: Linear Interpolation (LI) and Boundary Random Interpolation (BRI). (More discussion in Appendix Sec.C.1)

The dataset-related quality evaluation function $Q()$ also has different implementations: (Sainburg et al., 2018) utilizes additional discriminator and training adversarially on latent interpolations; (Berthelot et al., 2018) uses a critic network as a surrogate which tries to recover the mixing coefficient from interpolated data.

3.3 CIR: encourage both Controllable Dis-RL and Convexity

Our goal is to encourage a controllable disentangled representation, and, for each semantic attribute-related latent dimension, the created space should be as convex as possible. Specifically, we want to optimize both controllable disentanglement (Eq. 1) and convexity (Eq. 2) for each semantic attribute. In practice, each mutual information term in Eq. 1 is hard to optimize directly as it requires access to the posterior. Most of the current methods use approximation to obtain the lower bound for optimizing the maximum (Chen et al., 2016; Belghazi et al., 2018) or upper bound for optimizing minimum (Kingma & Welling, 2014). However, it is hard to approximate so many $(2m(m-1)+2m)$ different mutual information terms in Eq. 1) simultaneously, not to mention considering the convexity of m latent space (Eq. 2) as well. To optimize them together, we propose to use a controllable disentanglement constraint to help the optimization of convexity and in turn, use convexity constraint to help a more robust optimization of the controllable disentanglement. In other words, we create a positive loop between controllable disentanglement and convexity, to help each other. Specifically, as shown in Fig. 1(c), we propose a simple yet efficient regularization method, Controllable Interpolation Regularization (CIR), which consists of two main modules: a Controllable Interpolation (CI) module and a Reuse Encoder Regularization (RER) module. It works as follows: an input sample x goes

through E to obtain latent code $z = E(x)$. Because our goal is controllable disentanglement, on each iteration we only focus on one attribute. CI module first selects one attribute $\mathcal{A}_i$ among all m attributes, and then interpolates along the $\mathcal{A}_i$ related latent space in z while preserving the other unselected attributes, yielding $z_{\mathcal{A}_i}$. After D translates the interpolated latent $z_{\mathcal{A}_i}$ back to image space, the RER module takes $D(z_{\mathcal{A}_i})$ as input and reuses the encoder to get the latent representation $z^{re}_{\mathcal{A}_i} = E(D(z_{\mathcal{A}_i}))$. RER then adds a reconstruction loss on the *unmodified latent space* as a regularization:

$$L_{\text{reg}} = ||z_{-\mathcal{A}_i} - z^{re}_{-\mathcal{A}_i}||_{l1} \tag{3}$$

where $z_{-\mathcal{A}_i}$ and $z^{re}_{-\mathcal{A}_i}$ denote the all latent dimensions of $z_{\mathcal{A}_i}$ and $z^{re}_{\mathcal{A}_i}$ respectively, except those that represent the modified attribute $\mathcal{A}_i$. Eq. 3 explicitly optimizes Eq. 1: in each iteration, if the modified latent region $z_{\mathcal{A}_i}$ only influences the expression of $x_{\mathcal{A}_i}$, then, after reusing E, the unmodified region in $E(D(z_{\mathcal{A}_i}))$ should remain as is (min E, D in Eq. 1). On the one hand, for those unselected attributes, their information should be preserved in the whole process (max E, D in Eq. 1). Eq. 3 also implicitly optimizes Eq. 2: if the interpolated latent code is not 'understandable' by E and D, the RER module does not work and the L_{reg} would be large. Fig. 2 (a) and (b) abstractly demonstrate the latent space convexity difference before and after adding CIR to GZS-Net (Ge et al., 2020a). Convexity and disentanglement are dual tasks in the sense that one can help enhance the other's performance. On the other hand, the reconstruction loss towards *perfect* controllable disentanglement implicitly encourages a convex attribute latent space; The more convex the latent space, the more semantically meaningful samples synthesized by interpolation will help the optimization of controllable disentanglement, which encourages a more robust C-Dis-RL. From the perspectives of loss function and optimization, if the reconstruction loss could decrease to zero for a given dataset augmented by many interpolated samples, then perfect disentanglement and convexification are achieved. That is, CIR forces, in the limit of infinite interpolated samples, the disentangled latent representation of every attribute to be $convex$, where every interpolation along every attribute is guaranteed to be meaningful.

4 Qualitative Experiments

We qualitatively evaluated the effectiveness of our CIR as a general module to encouraging both C-Dis-RL and the convexity in latent space. We merged it into three baseline models on three different tasks: Sec. 4.1 for multiple face attributes transfer with ELEGANT (Xiao et al., 2018), Sec. 4.2 for cross-modality image-to-image translation with I2I-Dis (Lee et al., 2018) and Sec. 4.3 for zero-shot synthesis through group-supervised learning with GZS-Net (Ge et al., 2020a). They all contain a general autoencoder structure and their performance highly depends on the C-Dis-RL latent space.

4.1 **CIR** + ELEGANT (Xiao et al., 2018) for multiple face attributes transfer

We conduct the same face attribute transfer tasks as in ELEGANT (Xiao et al., 2018) paper with *CelebA* (Liu et al., 2015). **Task 1**: taking two face images with the opposite attribute as input and generate new face images which exactly transfer the opposite attribute between each other. **Task 2**: generate different face images with the same style of the attribute in the reference images. Both of the two tasks require a robust controllable disentangled latent space to swap the attributes of interest to synthesize new images and the convexity of latent space influences image quality.

For training, we inherit the network structure of ELEGANT (Xiao et al., 2018), which adopts the U-Net (Ronneberger et al., 2015) structure to generate high-resolution images with exemplars. In this way, the output of the encoder is the latent code of disentangled attributes and the context information is contained in the output of the intermediary layer of the encoder. ELEGANT adopts an iterative training strategy: training the model with respect to a particular attribute each time. We use the same training strategy except for adding our regularization loss term Eq. 3. As described in Sec. 3.3 and Fig. 1 (e), to encourage the disentanglement and convexity of attribute $\mathcal{A}_i$, CIR interpolates $\mathcal{A}_i$-related dimensions in latent code and constrains the other latent dimensions to remain unchanged after D and reused E. Specifically, when training ELEGANT about the $\mathcal{A}_i$ attribute **Eyeglasses** at a given iteration, we obtain the latent code $zA = E(A)$ and $zB = E(B)$ with E for each pair of images A and B with opposite $\mathcal{A}_i$ attribute value. The disentangled latent code is partitioned into $z_{+\mathcal{A}_i}$ for latent dimensions related to $\mathcal{A}_i$, and $z_{-\mathcal{A}_i}$ for unrelated dimensions. We interpolate in $z_{+\mathcal{A}_i}$ with zA and zB while keeping the other dimensions $z_{-\mathcal{A}_i}$ as is to obtain interpolated latent code $zA_{\mathcal{A}_i}$ and $zB_{\mathcal{A}_i}$. After D and reuse E, we get the reconstructed latent representation $zA^{re}_{\mathcal{A}_i} = E(D(zA_{\mathcal{A}_i}, zA))$ and $zB^{re}_{\mathcal{A}_i} = E(D(zB_{\mathcal{A}_i}, zB))$. The reconstruction loss as a regularization is:

$$L_{\text{reg}} = ||zA_{-\mathcal{A}_i} - zA^{re}_{-\mathcal{A}_i}||_{l2} + ||zB_{-\mathcal{A}_i} - zB^{re}_{-\mathcal{A}_i}||_{l2} \tag{4}$$

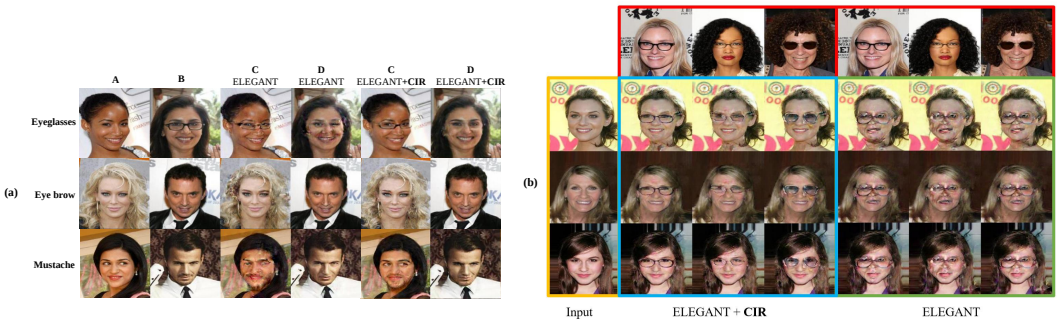


Figure 3: ELEGANT + **CIR** performance of (a) task 1 for two images face attribute transfer and (b) task 2 for face image generation by exemplars. (More results in Appendix Sec. B.1)

The overall generative loss of ELEGANT + CIR is:

$$\mathcal{L}(G) = L_{\text{reconstruction}} + L_{\text{adv}} + \boldsymbol{\lambda}_{\text{CIR}} \boldsymbol{L}_{\text{reg}} \tag{5}$$

where $L_{\text{reconstruction}}$ and L_{adv} are ELEGANT original loss terms, $\lambda_{\text{CIR}} > 0$ control the relative importance of the loss terms. we keep the discriminative loss. (More details in Appendix Sec. A.1)

Fig. 3(a) shows the task 1 performance on two images face attribute transfer. Take Eyeglasses as an example attribute to swap: C and D should keep all other attributes unmodified except for Eyeglasses. ELEGANT generated C and D have artifacts in Eyeglasses-unrelated regions, which means ELEGANT cannot disentangle well in latent space. After adding CIR, the generated C and D better preserve the irrelevant regions during face attribute transfer, which demonstrates that CIR helps encourage a more convex and disentangled latent space. The Eyebrow and Beard attribute results also show the improvement from CIR. Fig. 3(b) shows the task 2 performance on face image generation by exemplars. Similarly, ELEGANT generated new images with artifacts in Eyeglasses-unrelated regions that cannot disentangle well. Synthesis is also inferior in the glasses region, which we posit is due to non-convexity in the eyeglass-related latent space. With the help of CIR, the generated images improve both Eyeglass quality and irrelevant region preservation.

4.2 **CIR** + I2I-DIS (LEE ET AL., 2018) FOR CROSS MODALITY IMAGE TRANSLATION

We conduct the same image-to-image translation task as in I2I-Dis (Lee et al., 2018) paper with *cat2dog* dataset (Lee et al., 2018). There are two image domains $\mathcal{X}$ (cat) and $\mathcal{Y}$ (dog), I2I-Dis embeds input images onto a shared content space $\mathcal{C}$ with specific encoders ($E^c_{\mathcal{X}}$ and $E^c_{\mathcal{Y}}$), and domain-specific attribute spaces $\mathcal{A}_{\mathcal{X}}$ and $\mathcal{A}_{\mathcal{Y}}$ with specific encoders ($E^a_{\mathcal{X}}$ and $E^a_{\mathcal{Y}}$) respectively. After that, new images can be synthesized by transferring the shared content attribute cross-domain (between cat and dog), such as generating unseen dogs with the same content attribute value (pose and outline) as the reference cat. Domain-specific attribute $\mathcal{A}_{\mathcal{X}}$ and $\mathcal{A}_{\mathcal{Y}}$ already been constraint by adding a KL-Divergence loss with Gaussian distribution; thus, we can freely sample in Gaussian for synthesis. The shared content space $\mathcal{C}$ could be encouraged as a more convex and disentangled space by CIR.

We use the same network architecture and training strategy as I2I-Dis (Lee et al., 2018). except for adding our regularization loss term Eq. 1. As described in Sec. 3.2 and Fig. 1 (f), during each training iteration, a cat image x and a dog image y go through corresponding encoders and each of them produce latent codes of domain ($zx_a = E^a_{\mathcal{X}}(x)$, $zy_a = E^a_{\mathcal{Y}}(y)$) and content ($zx_c = E^c_{\mathcal{X}}(x)$, $zy_c = E^c_{\mathcal{Y}}(y)$). Then a interpolated content attribute latent code zxy_c (between zx_c and zy_c) concatenates with the domain attribute latent code of cat image zx_a and dog image zy_a respectively and forms two new latent codes, and decoders turns them into new images $u = G_{\mathcal{X}}(zx_a, zxy_c)$, $v = G_{\mathcal{Y}}(zy_a, zxy_c)$. To encourage the disentanglement and convexity of the content attribute, we reuse $E^a_{\mathcal{X}}$ and $E^a_{\mathcal{Y}}$ to get the reconstructed domain attribute latent representations $zx^{re}_a = E^a_{\mathcal{X}}(u)$, $zy^{re}_a = E^a_{\mathcal{Y}}(v)$ and add the reconstruction loss as a regularization:

$$L_{\text{reg}} = ||zx^{re}_a - zx_a||_{l1} + ||zy^{re}_a - zy_a||_{l1} \tag{6}$$

The overall loss of I2I-Dis + CIR is

$$\begin{aligned} \mathcal{L} =& \lambda^{content}_{adv} L^c_{adv} + \lambda^{cc}_1 L^{cc}_1 + \lambda^{domain}_{adv} L^{domain}_{adv} + \\ & \lambda^{recon}_1 L^{recon}_1 + \lambda^{latent}_1 L^{latent}_1 + \lambda_{KL} L_{KL} + \boldsymbol{\lambda}_{\text{CIR}} \boldsymbol{L}_{\text{reg}} \end{aligned} \tag{7}$$

where content and domain adversarial loss L^c_{adv} L^{domain}_{adv}, cross-cycle consistency loss L^{cc}_1, self-reconstruction loss L^{recon}_1, latent regression loss L^{latent}_1 and KL loss L_{KL} are I2I-Dis original loss terms, $\lambda > 0$ control the relative importance of the loss terms. (More details in Appendix Sec. A.2).

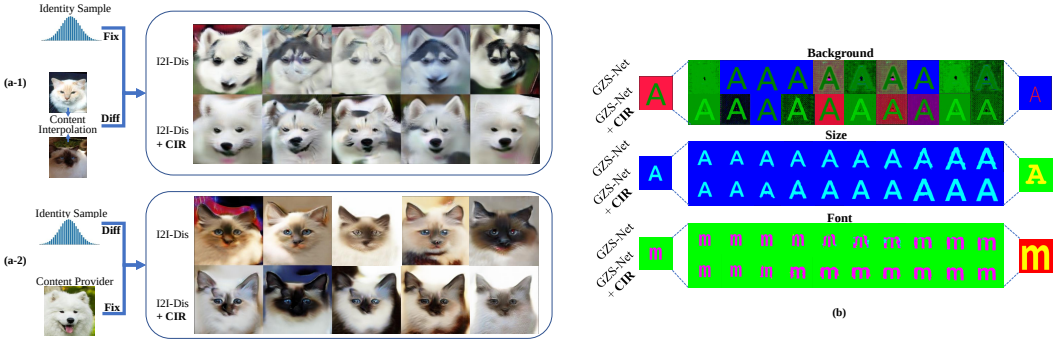


Figure 4: (a) I2I-Dis + **CIR** performance of diverse image-to-image translation; (b) GZS-Net + **CIR** performance of interpolation-based attribute controllable synthesis

Fig. 4 (a) shows the image-to-image translation performance. (a-1) We fix the identity (domain) latent code and change the content latent code by interpolation; generated images should keep the domain attribute (belong to the same dog). I2I-Dis generated dog images have artifacts, which means the non-convex latent space cannot 'understand' the interpolated content code. After adding CIR, the generated images have both better image quality and consistency of the same identity. (a-2) We fix the content latent code and change the identity by sampling; generated images should keep the same content attribute (pose and outline). Cat images generated by I2I-Dis have large pose variance (contain both left and right pose), and large face outline variance (ear positions/sizes). After adding CIR, the generated images have smaller pose and outline variance. (More results in Appen. Sec. B.2)

4.3 **CIR** + GZS-NET (GE ET AL., 2020A) FOR ZERO-SHOT SYNTHESIS

We use the same architecture of autoencoders as GZS-Net (Ge et al., 2020a) and *Fonts* dataset (Ge et al., 2020a). The latent feature after encoder E is a 100-dim vector, and each of the five *Fonts* attributes (content, size, font color, background color, font) covers 20-dim. The decoder D, symmetric to E, takes the 100-dim vector as input and outputs a synthesized sample. We use the same Group-Supervised learning training strategy as GZS-Net except for adding our regularization loss term Eq. 1, which is exactly the same as the one described in Sec. 3.3 and Fig. 1 (d). Besides the reconstruction loss L_{r}, swap reconstruction loss L_{sr} and cycle swap reconstruction loss L_{csr} which are same as GSL, we add a regularization reconstruction loss L_{reg}. The total loss function is:

$$\mathcal{L}(E, D) = L_{\text{r}} + \lambda_{\text{sr}} L_{\text{sr}} + \lambda_{\text{csr}} L_{\text{csr}} + \boldsymbol{\lambda}_{\text{CIR}} \boldsymbol{L}_{\text{reg}} \tag{8}$$

where $\lambda_{\text{sr}}, \lambda_{\text{csr}}, \lambda_{\text{CIR}} > 0$ control the relative importance of the loss terms.

Fig. 4(b) shows the interpolation-based controllable synthesis performance on background, size, and font attributes. Take background interpolation synthesis as an example: we obtain background latent codes by interpolating between the left and right images, and each of them concatenates with the unselected 80-dim latent code from the left image. Generated images should keep all other attributes unmodified except for the background. GZS-Net generated images have artifacts in background-unrelated regions, i.e., GZS-Net cannot disentangle well in latent space. After adding our CIR, the generated images better preserve the irrelevant areas during synthesis. The size and font attribute results also show improvement from CIR. (More results in Appendix. Sec. B.3)

5 QUANTITATIVE EXPERIMENTS

We conduct five quantitative experiments to evaluate the performance of CIR on controllable disentanglement and convexity.

Controllable Disentanglement Evaluation by Attribute Co-prediction. Can latent features of one attribute predict the attribute value? Can it also predict values for other attributes? Under *perfect* controllable disentanglement, we should answer *always* for the first and *never* for the second. We quantitatively assess disentanglement by calculating a model-based confusion matrix between attributes. We evaluate GZS-Net (Ge et al., 2020a) + **CIR** with the *Fonts* (Ge et al., 2020a) dataset (latent of ELEGANT and I2I-Dis are not suitable). Each image in *Fonts* contains an alphabet letter rendered using 5 independent attributes: content (52 classes), size (3), font color (10), background color (10), and font (100). We take the test examples and split them 80:20 for train_{DR}:test_{DR}. For each attribute pair $j, r \in [1..m] \times [1..m]$, we train a classifier (3 layer MLP) from g_j of train_{DR} to the attribute values of r, then obtain the accuracy of each attribute by testing with g_j of test_{DR}. Fig. 5(a) compares how well features of each attribute (row) can predict an attribute value (column): perfect

(a)

	GZS-Net					GZS-Net + **CIR**				
$\mathcal{A}$ ($\lvert\mathcal{A}\rvert$)	**C**	**S**	**FC**	**BC**	**St**	**C**	**S**	**FC**	**BC**	**St**
Content (52)	**.99**	.92	.11	.13	.30	**1**	.57	.11	.12	.01
Size (3)	.78	**1.0**	.11	.15	.36	.02	**1.0**	.11	.12	.01
FColor (10)	.70	.88	**1.0**	.16	.23	.02	.74	**1.0**	.41	.01
BColor (10)	.53	.78	.21	**1.0**	.15	.02	.57	.25	**1.0**	.02
Style (100)	.70	.93	.12	.12	**.63**	.02	.58	.11	.11	**.69**

(b)

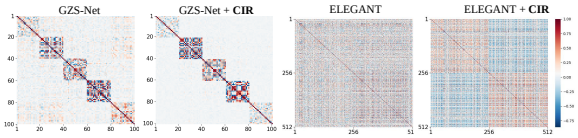


Figure 5: (a) c-Dis-RL analysis. Diagonals are bolded. (b) c-Dis-RL Evaluation by Correlation Coefficient. Intra-attribute correlation increases with CIR (GZS-Net (top): 7.2%, ELEGANT (bottom): 3.2%) while inter-attribute decreases (GZS-Net: 60.9%, ELEGANT: 3.1%).

should be as close as possible to Identity matrix, with off-diagonal entries close to random (i.e., 1 / $|\mathcal{A}_r|$). The off-diagonal values of GZS-Net show the limitation of disentanglement performance; with CIR's help, the co-prediction value shows a nearly perfect disentanglement.

Controllable Disentanglement Evaluation by Correlation Coefficient. For each method, we collect 10,000 images from the corresponding dataset (ELEGANT (Xiao et al., 2018) with *CelebA* (Liu et al., 2015), GZS-Net with *Fonts* (Ge et al., 2020a)) and obtain 10,000 latent codes by Es, we calculate the correlation coefficient matrix between dimensions in latent space. A perfect disentanglement should yield high intra-attribute correlation but low inter-attribute correlation. ELEGANT disentangles two attributes: eyeglasses and mustache, each of which covers 256-dimensions. GZS-Net disentangles five attributes: content, size, font color, background color, and font; each covers 20-dimensions. Fig. 5 shows that CIR improves the disentanglement in latent space, as demonstrated by higher intra-attribute and lower inter-attribute correlations (More details in Appendix Sec. C.2).

Convexity Evaluation with Image Quality Score. To evaluate the overall convexity in latent space, we use an image quality classifier to evaluate the quality of images generated by interpolating in latent space. We train a specific image quality classifier for each baseline algorithm and corresponding dataset. Take ELEGANT as an instance: To train a classifier for ELEGANT and ELEGANT + CIR, we use 3000 *CelebA* original images as positive, high-quality images. To collect negative images, we first randomly interpolate the latent space of both ELEGANT and ELEGANT + CIR and generate interpolated images for negative low-quality images; then, we manually select 3000 low-quality images (artifact, non-sense, fuzzy ...) and form a 6000 images training set. After training an image quality classifier, we test it on 1500 images generated by interpolation-based attribute controllable synthesis as Exp. 4.1. Table 2 shows the average probability of high-quality images (higher is better). The training and testing for I2I-Dis (+ CIR) and GZS-Net (+ CIR) are similar.

Controllable Disentanglement Evaluation with Perceptual Path Length Metric. StyleGAN (Karras et al., 2019) proposes the perceptual path length metric to quantify the latent space entanglement through interpolation. We conduct these experiments and the results are shown in Table. 2 (Lower value represents better disentanglement latent space. More details in Appendix Sec. C.3).

Algorithms	Train images	Test images	High quality probability
ElEGANT	6000	1500	12%
ElEGANT + **CIR**			**60%**
I2I-Dis	1500	1500	18%
I2I-Dis + **CIR**			**33%**
GZS-Net	6000	1000	13%
GZS-Net + **CIR**			**40%**

Table 1: Convexity Evaluation with Image Quality Score

Algorithms	I2I-Dis	I2I-Dis + **CIR**	ElEGANT	ElEGANT + **CIR**
MSE	29	21	1.23	0.68

Table 2: Disentanglement Evaluation with StyleGAN Perceptual Path Length Metric

***Perfect* Disentanglement Property Evaluation.** As we defined in Sec. 3.3, *Perfect* disentanglement property can be evaluated by the difference of the unmodified attribute related dimensions in $\mathbb{R}^d$ after modifying a specific attribute $\mathcal{A}_i$ in image space. For the two methods in each column (Table 3) and corresponding datasets, we modify one attribute value $\mathcal{A}_i$ of each input x and get $\hat{x}$, then obtain latent codes ($z = E(x)$, $\hat{z} = E(\hat{x})$) with two methods' encoders respectively. After we normalized the latent codes from two methods into the same scale, we calculate the Mean Square Error (MSE) of the unmodified region $MSE(z_{-\mathcal{A}_i}, \hat{z}_{-\mathcal{A}_i})$ between z and $\hat{z}$ (lower is better). Table 3 shows that after adding CIR, we obtain a lower MSE, which means CIR encourages a better disentangled latent space.

Table 3: *Perfect* Disentanglement Property Evaluation

Algorithms	ElEGANT	I2I-Dis	GZS-Net	ElEGANT + **CIR**	I2I-Dis + **CIR**	GZS-Net + **CIR**
MSE	1.9	1.8	3.42	**0.38**	**0.1**	**0.27**

6 More Downstream Tasks and Applications

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

(a)

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%**

(b)

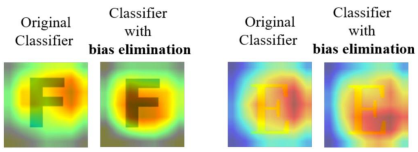


Figure 6: (a) Bias elimination experiment results. (b) The influence of bias shown by Grad-Cam.

Mining New Attribute Value. We show the performance of novel attribute mining with encouraged C-Dis-RL and convex latent space. Appendix Fig. 11 shows the new background color and font color generated through interpolation given only six original colors during training, which shows a better consistency and disentanglement with the help of CIR. (More details in Appendix Sec. D.1).

Data Augmentation. We design a letter image classification experiment with *Fonts* (Ge et al., 2020a) 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 5), which shows an improved data augmentation performance on downstream tasks with the help of CIR. (more details in Appendix Sec. D.2)

Table 4: Controllable augmentation performance

Attribute \ Dataset	D_L	D_S	D_{S+G}	D_{S+G+C}	D_{test}
Dataset Size	5400	540	540+1000	540+1000	5400
Test Accuracy	94%	71%	74%	**77%**	N/A

Bias Elimination for Fairness. Dataset bias may influence the model performance significantly. (Mehrabi et al., 2019) listed lots of bias resources and proved that eliminate 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 Appendix Sec. D.3) 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 Fig. 6(a), 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. 6(b), Grad-Cam's (Selvaraju et al., 2017) 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 rose to 98%. Hence, we eliminate the dataset bias with the help of robust disentangled latent by CIR.

7 Conclusion

We proposed a general disentanglement module, Controllable Interpolation Regularization (CIR), compatible with different algorithms to encourage more convex and robust disentangled representation learning. We show the performance of CIR with three baseline methods ELEGANT, I2I-Dis, and GZE-Net. CIR first conducts controllable interpolation in latent space and then 'reuses' the encoder to form an explicit disentanglement constraint. Qualitative and quantitative experiments show that CIR improves baseline methods performance on different controllable synthesis tasks: face attribute transfer, diverse image-to-image transfer, and zero-shot image synthesis with different datasets: CelebA, cat2dog and Fonts respectively. We also prove that CIR can improve additional downstream tasks, such as new attribute value mining, data augmentation, and eliminating dataset bias for fairness.

REFERENCES

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pp. 531–540. PMLR, 2018.

Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In *International conference on machine learning*, pp. 552–560, 2013.

David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018.

Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016.

Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *International Conference on Machine Learning*, pp. 1068–1077. PMLR, 2017.

Chanho Eom and Bumsub Ham. Learning disentangled representation for robust person re-identification. *arXiv preprint arXiv:1910.12003*, 2019.

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

Yunhao Ge, Jiaping Zhao, and Laurent Itti. Pose augmentation: Class-agnostic object pose transformation for object recognition. In *European Conference on Computer Vision*, pp. 138–155. Springer, 2020b.

I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

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)*, pp. 35–51, 2018.

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

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*, pp. 3730–3738, 2015.

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

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Tim Sainburg, Marvin Thielk, Brad Theilman, Benjamin Migliori, and Timothy Gentner. Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions. *arXiv preprint arXiv:1807.06650*, 2018.

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*, pp. 618–626, 2017.

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

Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1415–1424, 2017.

Vikas Verma, Alex Lamb, Christopher Beckham, Aaron Courville, Ioannis Mitliagkis, and Yoshua Bengio. Manifold mixup: Encouraging meaningful on-manifold interpolation as a regularizer. *arXiv preprint arXiv:1806.05236*, 7, 2018.

Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Dna-gan: Learning disentangled representations from multi-attribute images. *arXiv preprint arXiv:1711.05415*, 2017.

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)*, pp. 172–187, September 2018.

Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2138–2147, 2019.

A NETWORK ARCHITECTURE AND TRAINING DETAILS

A.1 ELEGANT XIAO ET AL. (2018) + CIR

Network Structure

For our ELEGANT + Controllable Interpolation Regularization (CIR), we use the same network architecture as the original ELEGANT paper Xiao et al. (2018). In addition, 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* Liu et al. (2015). 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.

A.2 I2I-DIS LEE ET AL. (2018) + CIR

Network Structure

We use the same network architecture as the original I2I-Dis paper Lee et al. (2018). For all the experiments in this section, we use images from the *cat2dog* dataset with the size of 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 a batch size of 1, the 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 $\lambda_{CIR} = 10$.

Different interpolation methods Fig. 2 (c) shows two kinds of interpolation: Linear Interpolation (LI) and Boundary Random Interpolation (BRI). Linear interpolation is easy to implement while facing two problems: (1) low efficiency, as it only explores along lines while convexification may require filling whole subspaces. (2) may not fill the whole space (e.g., leave a hole in the middle; see Fig. 2 (c)). BRI can solve these problems. BRI first collects an image set S (contains $s > 2$

images) and obtains the corresponding latent code $\mathcal{S}_z = \{z^{(i)}\}_{i=1}^{s}$. Then it calculates the maximum and minimum values as the upper and lower bounds of each dimension. After that, we randomly sample k latent codes in the region created by the boundary. Fig. 2 (c) shows the situation in 2D space. BRI uses 'small subspaces' to cover 'big subspaces'.

A.3 GZS-NET GE ET AL. (2020A) + **CIR**

Network Structure

We use the same network architecture and the same dataset (*Fonts* Ge et al. (2020a)) as the original GZS-Net paper Ge et al. (2020a). 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 has 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 latest 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* Ge et al. (2020a) 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.

B MORE QUALITATIVE RESULTS

B.1 ELEGANT XIAO ET AL. (2018) + CIR

Fig. 7 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. 8 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.

B.2 I2I-DIS LEE ET AL. (2018) + CIR

Fig. 9 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.

B.3 GZS-NET GE ET AL. (2020A) + CIR

Fig. 10 shows more results of the interpolation-based controllable synthesis performance on font color, background color, size, and font attributes.

C QUANTITATIVE EXPERIMENTS DETAILS

C.1 DIFFERENT INTERPOLATION METHODS

Linear interpolation is easy to implement while facing two problems: (1) low efficiency, (2) may not fill the whole space (e.g., leave a hole in the middle; see Fig. 2 (c)). BRI can solve these problems.

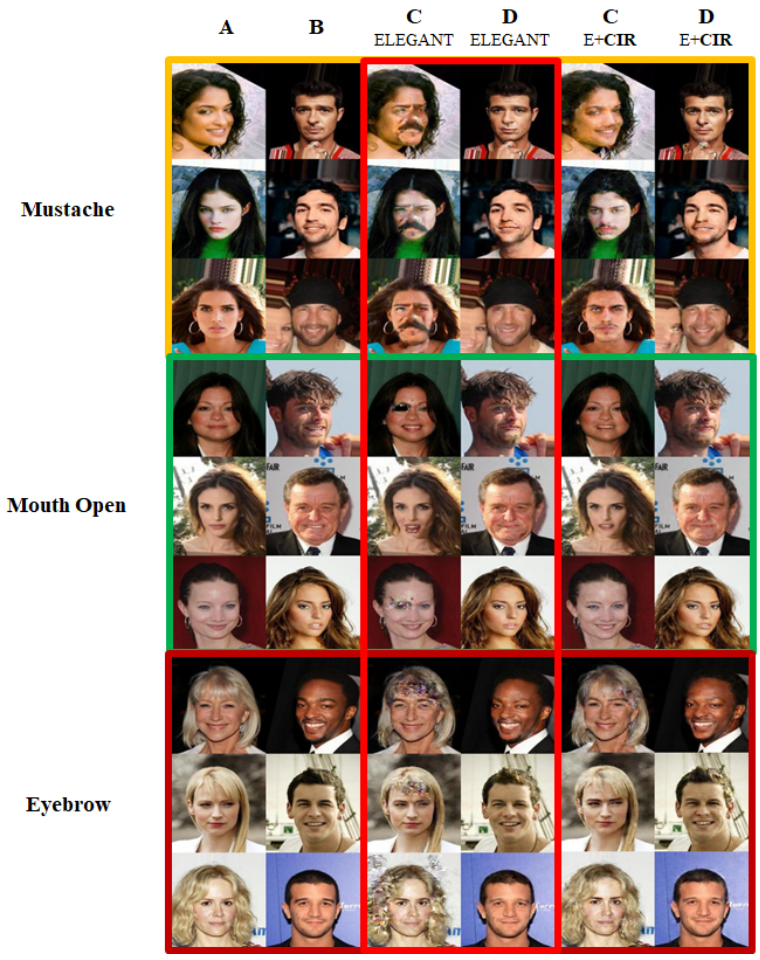


Figure 7: More examples of ELEGANT+**CIR** (E+**CIR**) performance of task 1 for two images face attribute transfer

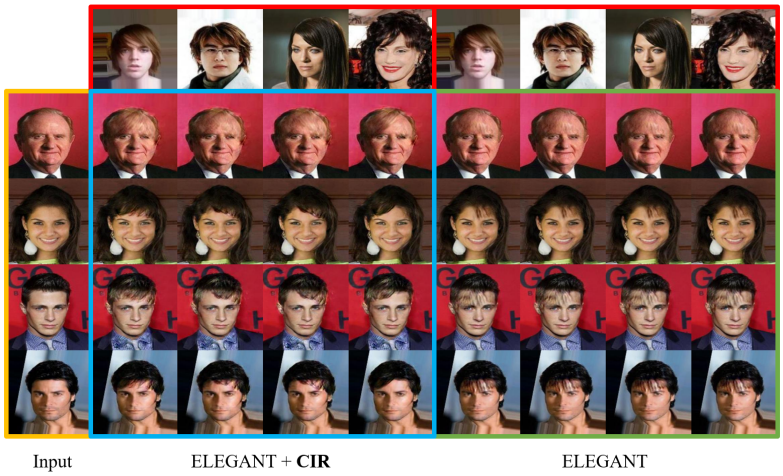


Figure 8: ELEGANT + **CIR** Performance of task 2 for face image generation by exemplars

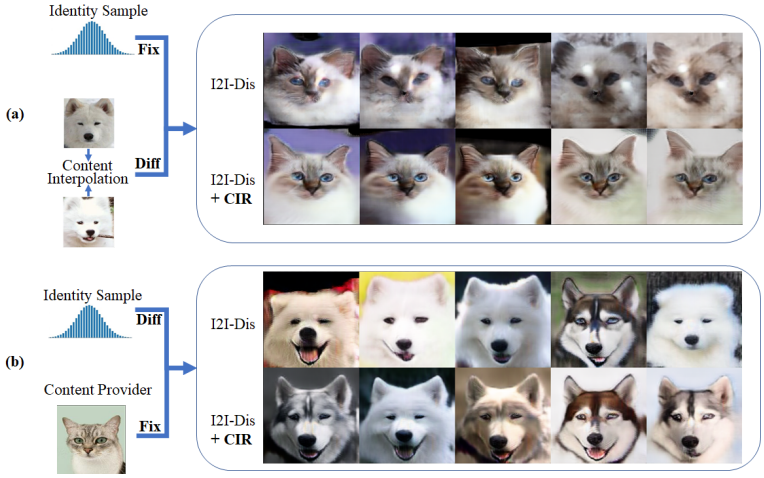


Figure 9: I2I-Dis + **CIR** performance of diverse image-to-image translation

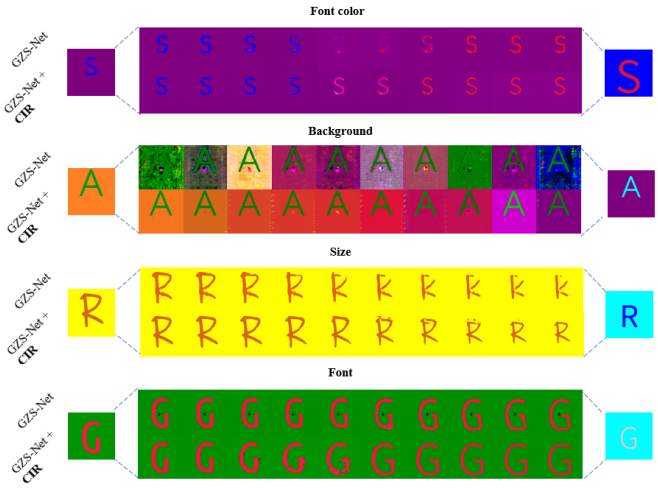


Figure 10: More results of GZS-Net + **CIR** performance of interpolation-based attribute controllable synthesis

BRI first collects an image set S (contains $s > 2$ images) and obtains the corresponding latent code $\mathcal{S}_z = \{z^{(i)}\}_{i=1}^{s}$. Then it calculates the maximum and minimum values as the upper and lower bounds of each dimension. After that, we randomly sample k latent codes in the region created by the boundary. Fig. 2 (c) shows the situation in 2D space. BRI uses 'small subspaces' to cover 'big subspaces'.

C.2 Disentanglement Evaluation by Correlation Coefficient.

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

$$r_s = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{9}$$

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* (Liu et al., 2015) 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:

$$norm(v_i) = (v_i - \overline{v_i}) \,/\, \sigma_{v_i},\ \forall\, i \in \{1, 2, ..., |v|\} \tag{10}$$

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.

C.3 Controllable Disentanglement Evaluation with perceptual path length metric

We use the similar method with the perceptual path length metric proposed by StyleGAN (Karras et al., 2019). We subdivide a latent space interpolation path into linear segments, the definition of total perceptual length of this segmented path is the sum of perceptual differences over each segment. In our experiment, we use a small subdivision epsilon $\epsilon = 10^{-4}$. We use linear interpolation(lerp) in our experiment. Thus, The average perceptual path length in latent space $\mathcal{Z}$ is

$$l_{\mathcal{Z}} = \mathbb{E}\left[\frac{1}{\epsilon^2} d\left(G\left(\text{lerp}\left(\mathbf{z}_1, \mathbf{z}_2; t\right)\right), G\left(\text{lerp}\left(\mathbf{z}_1, \mathbf{z}_2; t+\epsilon\right)\right)\right)\right]$$

Where $\mathbf{z}_1, \mathbf{z}_2$ is the start point and the end point in latent space. G is the generator in the model, it can be a decoder in Auto-encoder or generator in a GAN-based model. $t \sim U(0, 1)$.

D Downstream Tasks and Applications Details

D.1 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 Likas et al. (2003) 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

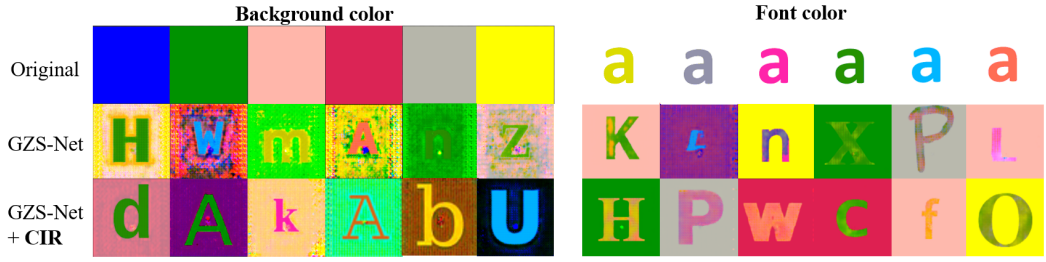
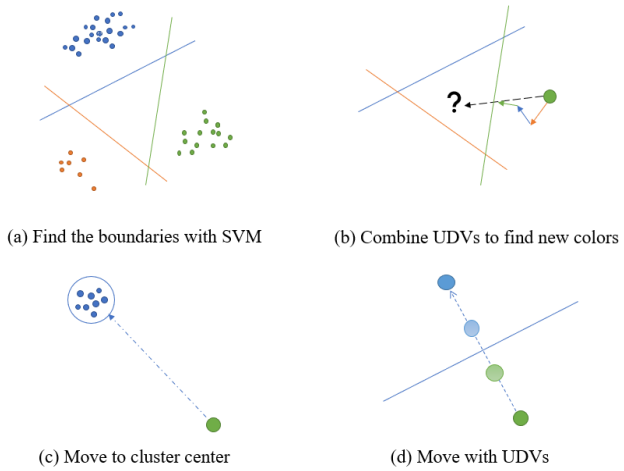


Figure 11: Controllable mining novel background and font color by interpolation in latent space.

Figure 12: Mining new attribute values with UDV

wear glasses). We assume a hyperplane in the latent space serving as the separation boundary Shen et al. (2020), 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. 13 (a), we can find the boundaries and UDVs by SVM for each attribute value. To solve the precision problem in attribute synthesis, Fig. 13 (c) shows moving towards the z value of the cluster center directly for Gaussian; Fig. 13 (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. 13 (b) shows that we can combine the UDVs to discover new attribute values.



(a) Find the boundaries with SVM

(b) Combine UDVs to find new colors

(c) Move to cluster center

(d) Move with UDVs

Figure 13: Towards controllable exploration direction

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 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 the 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. 12 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.

D.2 More details of Data Augmentation experiments

We design a letter image classification experiment with *Fonts* (Ge et al., 2020a) 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 (Table. 5). 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. We first train the resnet18 classifier on D_L and D_S, calculating the test accuracy on the test dataset D_{test}. 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. We compare two synthesized datasets by training classifiers with the same settings and calculating the corresponding testing accuracy (Table 5). The result shows controllable synthesis can improve downstream tasks as a data augmentation method.

Table 5: Controllable augmentation performance (the $\star$ means that synthesized images with new attributes are added into the 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%	**77%**	N/A

D.3 Bias Elimination for Fairness.

We design three datasets, a biased training dataset $\mathcal{D}^{\text{B}}$, two unbiased dataset: $\mathcal{D}^{\text{UB}}$ for training and $\mathcal{D}^{\text{T}}$ for test. In $\mathcal{D}^{\text{B}}$, we entangle the two attributes letter and background color as dataset bias. $\mathcal{D}^{\text{B}}$ consists of three-part: G1, G2, and G3. The details of those datasets can be found in Table. 6. The number of colors represents the number of background colors for each letter.

For G1 of $\mathcal{D}^{\text{B}}$, we randomly select 15 letters. For each letter, we randomly select one background color. When we add images to G1, only the images with the selected background color can be added to G1, but font color, size, font are unlimited. For G2, we randomly select 15 letters that are different from G1. For every letter, we randomly select three background colors and perform the same collection process as G1. For G3, we use the last 22 letters and every letter has all of 6 background colors and do the same collection process as G1.

Table 6: Bias elimination dataset setting

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