Model Debiasing via Gradient-based Explanation on Representation

Jindi Zhang
jd.zhang@my.cityu.edu.hk
City University of Hong Kong
Hong Kong SAR

Luning Wang
wangluning2@huawei.com
Hong Kong Research Center, Huawei
Hong Kong SAR

Dan Su
dasu@nvidia.com
NVIDIA Research
Hong Kong SAR

Yongxiang Huang
huang.yongxiang2@huawei.com
Hong Kong Research Center, Huawei
Hong Kong SAR

Caleb Chen Cao
goupcaleb@gmail.com
The Hong Kong University of Science and Technology
Hong Kong SAR

Lei Chen
leichen@cse.ust.hk
The Hong Kong University of Science and Technology
Hong Kong SAR

ABSTRACT

Machine learning systems produce biased results towards certain demographic groups, known as the fairness problem. Recent approaches to tackle this problem learn a latent code (i.e., representation) through disentangled representation learning and then discard the latent code dimensions correlated with sensitive attributes (e.g., gender). Nevertheless, these approaches may suffer from incomplete disentanglement and overlook proxy attributes (proxies for sensitive attributes) when processing real-world data, especially for unstructured data, causing performance degradation in fairness and loss of useful information for downstream tasks. In this paper, we propose a novel fairness framework that performs debiasing with regard to both sensitive attributes and proxy attributes, which boosts the prediction performance of downstream task models without complete disentanglement. The main idea is to, first, leverage gradient-based explanation to find two model focuses, 1) one focus for predicting sensitive attributes and 2) the other focus for predicting downstream task labels, and second, use them to perturb the latent code that guides the training of downstream task models towards fairness and utility goals. We show empirically that our framework works with both disentangled and non-disentangled representation learning methods and achieves better fairness-accuracy trade-off on unstructured and structured datasets than previous state-of-the-art approaches.

KEYWORDS

fairness, model debiasing, representation learning, gradient-based explanation

1 INTRODUCTION

Machine learning systems are reported to generate preferential predictions for some demographic groups and prejudiced predictions for others in many high-stake fields, such as loan offers, exam grading, school admission, and parole approval [1, 6, 21, 29]. This is known as the fairness problem in machine learning. Such fairness problems may cause long-term and high impacts on the life of vulnerable groups [5].

To tackle the fairness problem, early studies use adversarial training and regularization to force the model not to pay attention to sensitive information during training [8, 12, 25, 33, 35, 36]. And other works focus on learning fair (debiased) representations for downstream tasks [15, 16, 19, 24, 31, 34]. These methods usually specify the task attributes or the sensitive attributes before training, resulting in inflexibility [3].

To increase flexibility, the up-to-date approaches are to leverage disentangled representation learning methods to learn the disentangled latent code in which every dimension only contains one factor of variation and is optimized to be independent of each other [2, 3, 11, 13], and then remove the dimensions correlated with sensitive attributes before using the code to train downstream task models [3, 24].

However, because it is extremely difficult to enumerate all the factors of variation in real-world data [20], the number of the latent code dimensions is usually smaller than the real number of factors of variation. This results in incomplete disentanglement in the latent code, which poses two major challenges when we process real-world data, in particular, unstructured data such as images, with debiasing methods based on disentangled representation learning.

- First, it is challenging to avoid information loss for downstream tasks during the debiasing process. Since the latent code is usually incompletely disentangled, critical information for downstream tasks can be lost when we remove the dimensions correlated with sensitive attributes from the latent code, causing degradation in prediction accuracy.
- Second, it is challenging to cover all sensitive information in proxy attributes[1] (proxies for sensitive attributes) while debiasing downstream task models. Because of incomplete disentanglement in the latent code, sensitive information encoded in proxy attributes may not exist only in those removed dimensions but also in the remaining dimensions. This results in fairness degradation of downstream task models.

In this work, we aim to address the aforementioned two challenges by exploring methods that do not rely on complete disentanglement and can better cover sensitive information. To this end, we propose a novel fairness framework named DVGE (**D**ebiasing **v**ia **G**radient-based **E**xplanation), as depicted in Figure 1. Specifically, to address the first challenge, DVGE does not remove latent code

[1] For example, when the sensitive attribute is gender, corresponding proxy attributes can be hair length, beard, etc.

dimensions, which causes problems in locating sensitive information and debiasing downstream task models. To locate sensitive information and simultaneously address the second challenge, we exploit gradient-based explanations to highlight the importance of each latent code dimension when a model predicts sensitive attributes using the latent code. To debias downstream task models, we propose to perturb the latent code with the model focuses derived from gradient-based explanations. Overall, our main idea is to exploit gradient-based explanation to 1) obtain the model focus for predicting sensitive attributes, which we refer to as sensitive focus, and 2) obtain the model focus for predicting downstream task attributes, which we refer to as downstream task focus, and 3) use the two focuses to guide the training of downstream task models. Specifically, we propose *bidirectional perturbation* which uses the downstream task focus to positively perturb the latent code so that models pay more attention to downstream task information, and uses the sensitive focus to reversely perturb the latent code so that models pay less attention to sensitive information.

Compared with methods based on adversarial training, DVGE is more flexible, because it separates encoding and debiasing, so that the encoder does not need retraining when sensitive attributes or downstream tasks are changed. DVGE is also less tricky to train, since it debiases via perturbation instead of adversary. Compared with methods based on disentangled representation learning, DVGE better covers sensitive information with XAI explanations and reduces useful information loss without removing latent code dimensions.

As for evaluation, we conduct extensive experiments to compare our framework with previous state-of-the-art approaches by considering disentangled and non-disentangled VAE-based representation learning methods, on both real-world unstructured dataset (CelebA [18]) and structured dataset (South German Credit [9]). We measure the extent of fairness with two standard metrics, demographic parity (DP) [7, 32] and equal opportunity (EO) [10], against model accuracy. The results show that DVGE achieves better fairness-accuracy trade-off than the state-of-the-art approaches.

Our contributions are summarized as follows.

- We propose a novel fairness framework DVGE, to address the problem of the loss of useful downstream task information and the problem of overlooking sensitive information from proxy attributes, when debiasing models with incompletely disentangled latent code.
- We introduce to exploit gradient-based explanation to obtain model focuses related to sensitive information and downstream task information, and propose *bidirectional perturbation* to guide the model training for fairness purpose with the focuses.
- By extensive experiments, we show that our framework leads to better fairness-accuracy trade-off on both unstructured and structured real-world datasets compared to previous state-of-the-art approaches.

2 RELATED WORK

In this section, we review the works related to our paper, namely, debiasing methods in machine learning, variational autoencoders, and gradient-based explanations.

2.1 Debiasing Methods in Machine Learning

The methods for debiasing can be categorized as pre-processing methods, in-processing methods, and post-processing methods. Pre-processing methods aim for generating unbiased data for training by transforming the input data. Many recent pre-processing methods focused on learning discrimination-free encodings or embeddings for various tasks [3, 15, 24]. As for in-processing methods, they try to remove discrimination from models during training via objective functions, fairness constraints, or through adversarial training [8]. Furthermore, post-processing methods are proposed to audit predictions and may reassign labels with regard to fairness measurements after the training process [4]. Our proposed framework falls into the category of pre-processing methods.

2.2 Variational Autoencoders (VAEs)

VAEs are exploited to generate new unseen data that complies with the original distribution for generation tasks [23]. The main idea of VAEs is to learn a Gaussian distribution from training data and force the decoded data to have a similar distribution. Following the vanilla VAE [14], many variations of VAEs are proposed for different purposes, such as disentanglement [11, 13], recommendation [17], fairness [3, 24], etc. In this paper, we demonstrate that our fairness framework achieves better fairness-accuracy trade-off by considering both non-disentangled VAE (VanillaVAE [14]) and disentangled VAE (FactorVAE [13]).

2.3 Gradient-based Explanations

Gradient-based explanation methods are one of the primary approaches to explaining machine learning models, along with prototype-based methods, perturbation-based methods, etc. They produce local explanations for individual data points. The generated explanation (also known as saliency map or sensitivity map) by exploiting input-gradient highlights which parts in an input data point the model focuses on for making the prediction. Such an attempt is first made by Simonyan et al. [27]. Following their work, a number of variations are proposed, such as Grad-CAM [26], SmoothGrad [28], FullGrad [30]. In this work, we leverage gradient-based explanations to perturb the latent code for boosting the fairness and accuracy of downstream task models.

3 BACKGROUND

Here, we briefly introduce the background of two group fairness notions that we consider in this paper.

In this paper, we consider two commonly used group fairness notions, demographic parity (DP) [7, 32] and equal opportunity (EO) [10]. Let us first consider a simple example, in which we train a model $\hat{y} = g(x)$ to predict the label $y \in \{0, 1\}$, where $\hat{y}$ is the prediction, x denotes the input data, $s \in \{s_1, s_2\}$ denotes sensitive attributes in the input.

Demographic Parity. The definition of DP is that the model prediction is independent of sensitive attributes. In other words, the probability that a member of any subgroup (s_1 or s_2) receives the same prediction, 0 or 1 in our example, is completely the same. Based on the definition, the distance to demographic parity Δ_{DP} is

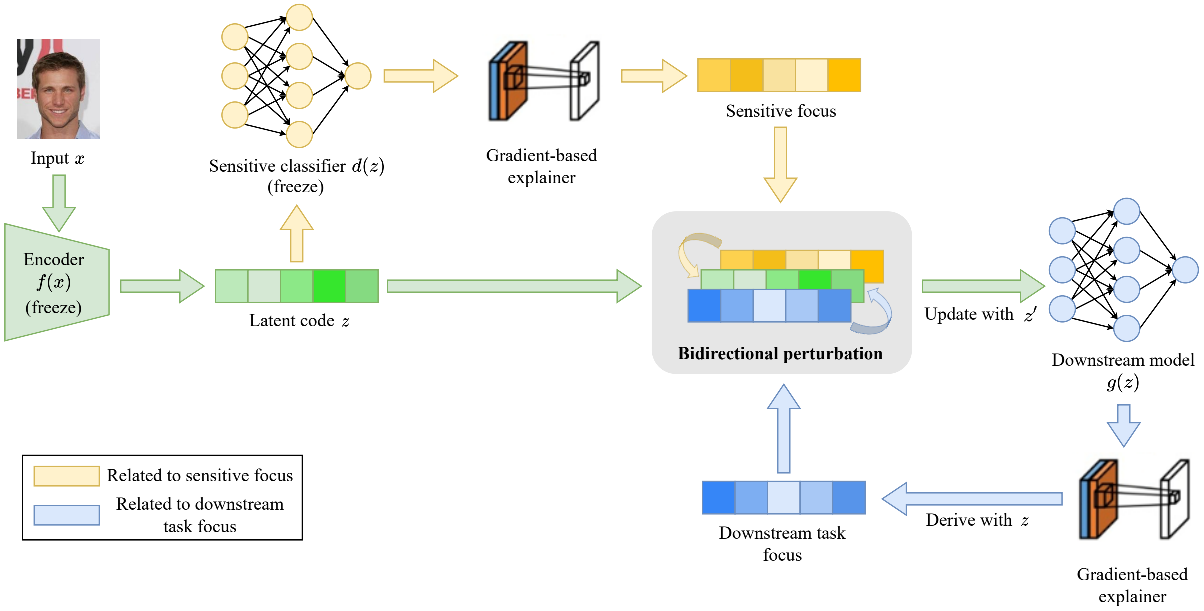


Figure 1: The overview of DVGE training procedure. First, the latent code z is generated with a trained VAE. Then, by feeding z to a trained sensitive classifier and the downstream task model being trained, the sensitive focus and the downstream task focus are derived from the gradient-based explanations on them. After the bidirectional perturbation perturbs z with the sensitive focus and the downstream task focus, the perturbed latent code z' is used to update the downstream task model.

used to measure how fair a model is as

$$\Delta_{DP} = |P(\hat{y} = 1|s = s_1) - P(\hat{y} = 1|s = s_2)|. \tag{1}$$

When $\Delta_{DP} = 0$, the demographic disparity is satisfied.

Equal Opportunity. Equal opportunity indicates that the true positive rate(TPR) of a model remains the same with respect to each subgroup. This is mathematically equivalent to that each subgroup has the same false negative rate (FNR). We can also use the distance to EO Δ_{EO} to measure the extent of fairness of a model as

$$\Delta_{EO} = |P(\hat{y} = 1|s = s_1, y = 1) - P(\hat{y} = 1|s = s_2, y = 1)| \tag{2}$$

or

$$\Delta_{EO} = |P(\hat{y} = 0|s = s_1, y = 1) - P(\hat{y} = 0|s = s_2, y = 1)|. \tag{3}$$

This definition underlines the idea that the qualified members of each subgroup should have the same probability to receive positive or negative predictions.

4 THE PROPOSED FAIRNESS FRAMEWORK

In this paper, we design a new fairness framework DVGE, as illustrated in Figure 1, by considering sensitive information from both sensitive attributes and proxy attributes. The framework does not depend on complete disentanglement. Instead, it leverages gradient-based explanation to obtain model focuses for predicting sensitive attributes and downstream task labels, and uses the proposed bidirectional perturbation to perturb the latent code for guiding the training of downstream task models.

4.1 Latent Code

As our work follows the idea of using representation learning to debias machine learning models with the flexibility to cope with different sensitive attributes and downstream tasks, we first train a VAE $f(x)$ to encode the input data x into latent code z by maximizing the Evidence Lower Bound (ELBO) [14]. The VAE used in DVGE is fixed after training. Our framework works with both disentangled and non-disentangled VAEs, which we show in the experiments.

4.2 Sensitive Focus

Sensitive focus is the model focus for predicting sensitive attributes s with latent code z. We derive it using a gradient-based explanation, since this explanation is input-specific and assigns an importance score to each latent code dimension based on gradients, which can be easily used for perturbation. Given a trained sensitive classifier $d(z)$ which takes the latent code z as input to predict sensitive attributes s, the gradient-based explanation e_{sens} for its prediction is calculated as

$$e_{sens} = \psi(\nabla_z d(z) \odot z), \tag{4}$$

where $\psi(\cdot)$ is a post-processing operation for a gradient-based explanation, e.g., scale and taking the absolute value, $\nabla_z d(z)$ is the model gradient with regard to z, and $\odot$ denotes element-wise multiplication. As $\psi(\cdot)$ and $\odot$ are only for the visualization purpose of the explanation, the essence of the explanation is $\nabla_z d(z)$, we define the sensitive focus as

$$F_{sens} = \nabla_z d(z). \tag{5}$$

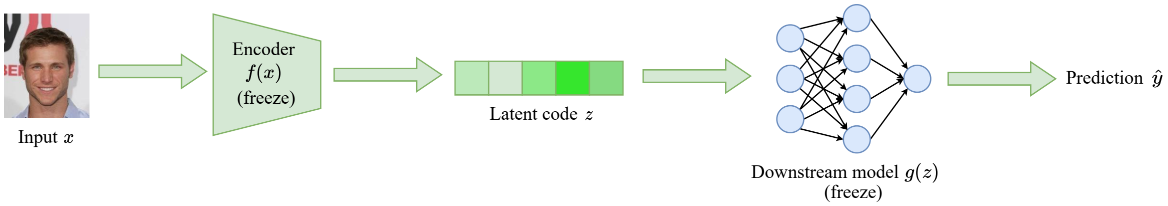


Figure 2: The overview of DVGE inference procedure. After training, DVGE does not perform bidirectional perturbation on the latent code during inference.

Please note that $\nabla_z d(z)$ is computed via backpropagation with the predicted sensitive attributes $\hat{s} = d(z)$. Thus, computing sensitive focus does not require access to real sensitive attributes.

Sensitive Information Coverage. Since the sensitive classifier $d(z)$ is trained to make use of every dimension of the latent code z to make predictions about sensitive attributes s, any sensitive information or shortcut information linking to s is exploited by it for the prediction. In addition, the gradient-based explanation can highlight all this information in **every dimension** of z, so the defined sensitive focus in our framework covers the sensitive information from both sensitive attributes and proxy attributes in the latent code.

Flexibility w.r.t. Changing Sensitive Attributes. As the sensitive focus is obtained via gradient-based explanation on sensitive classifier $d(z)$, when different sensitive attributes are required, we only need to change to a new $d(z)$ for predicting the new version of s, while reusing the latent code z.

4.3 Downstream Task Focus

The downstream task focus is the model focus for predicting downstream task label y with the latent code z. We obtain this focus directly from the gradient-based explanation of the downstream task model during its training process. Similar to Section 4.2, given a downstream task model $g(z)$ and the latent z, the gradient-based explanation of the model is calculated as

$$e_{task} = \psi(\nabla_z g(z) \odot z), \tag{6}$$

and we define the downstream task focus as

$$F_{task} = \nabla_z g(z). \tag{7}$$

The downstream task focus is for boosting the accuracy performance of the downstream task model while debiasing.

4.4 Bidirectional Perturbation

We perform bidirectional perturbation by perturbing the latent code z with the sensitive focus F_{sens} and the downstream task focus F_{task} to guide the training of the downstream task model for the purpose of fairness and prevention of downstream task accuracy degradation. The perturbed latent code z' is calculated as

$$\begin{aligned} z' &= z + Clip_\epsilon\{\eta_1 * F_{sens} - \eta_2 * F_{task}\} \\ &= z + Clip_\epsilon\{\eta_1 * \nabla_z d(z) - \eta_2 * \nabla_z g(z)\}, \end{aligned} \tag{8}$$

where η_1 and η_2 are the hyperparameters for controlling the intensity of debiasing and accuracy boosting, respectively, and

$$Clip_\epsilon\{v\} = \begin{cases} \epsilon, \text{ if } v > \epsilon, \\ max(v, -\epsilon), \text{ otherwise,} \end{cases} \tag{9}$$

where ϵ is non-negative and denotes the threshold for the distortion caused by bidirectional perturbation. $Clip_\epsilon\{\cdot\}$ is designed to prevent bidirectional perturbation introducing too much information distortion on latent code dimensions.

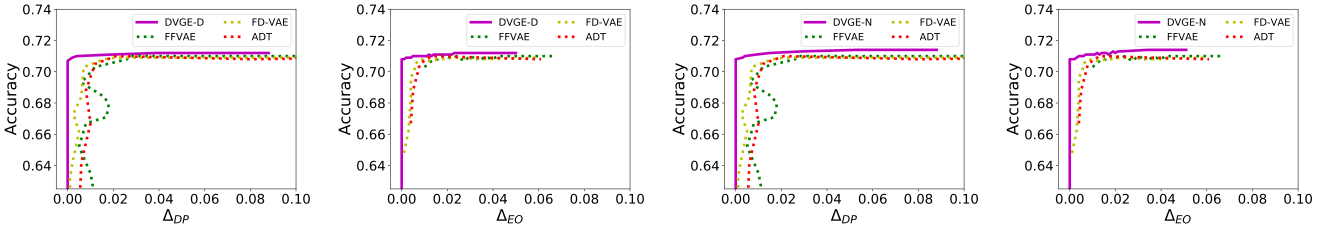
Rationale behind Bidirectional Perturbation. Since backpropagation gradients indicate the directions to optimize the objective function, in the equation 8, $+\nabla_z d(z)$ (sensitive focus) is for updating the latent code z in the reverse direction of optimizing sensitive information prediction, while $-\nabla_z g(z)$ (downstream task focus) is for updating z towards the direction of optimizing downstream task model, so that the downstream task model is guided to pay less attention to sensitive information and more attention to downstream task information. DVGE uses the perturbed latent code z' to update the downstream task model.

No Reliance on Complete Disentanglement. Even if the factors of variation for sensitive information are not fully disentangled (mixed with other factors of variation in the latent code dimensions), our framework still can debias the downstream task model, since the sensitive focus covers the sensitive information in *every dimension* of the latent code z, and our framework leverages the sensitive focus to perturb *every dimension* of z.

Inference. After finishing training the downstream task model with our framework, we do not perform the bidirectional perturbation on the latent code during inference as shown in Figure 2. The reason is that the model after training has learned to pay more attention to downstream task information and less attention to sensitive information in the latent code.

5 EXPERIMENTS

We conduct extensive experiments to evaluate our framework while comparing it with previous state-of-the-art approaches. To show the flexibility of our framework, we consider different sensitive attributes individually and jointly on structured dataset and unstructured dataset. To demonstrate that our framework does not rely on complete disentanglement, we consider both non-disentangled and disentangled VAEs. Furthermore, we use an ablation study to demonstrate that our framework has better coverage on sensitive information.



(a) DVGE-D, Δ_{DP}

(b) DVGE-D, Δ_{EO}

(c) DVGE-N, Δ_{DP}

(d) DVGE-N, Δ_{EO}

Figure 3: Fairness-accuracy trade-off comparison results for Experiment 1: CelebA dataset, sensitive attribute = “Male", task label = “Oval_Face".

5.1 Experiment Setups

5.1.1 DVGE-D and DVGE-N. To demonstrate that our framework does not rely on complete disentanglement to debias downstream task models, we implement DVGE with one disentangled VAE (FactorVAE [13]) and one non-disentangled VAE (VanillaVAE [14]), respectively. And we denote them as **DVGE-D** and **DVGE-N**. For more implementation details, please refer to Appendix B.

5.1.2 Baselines. We consider three state-of-the-art debiasing approaches as baselines in the experiments.

- Adversarial Training (ADT) [8]: The model based on ADT consists of three parts, i.e., feature encoder, sensitive branch, and downstream task branch. ADT debiases the model by updating the feature encoder with the reverse loss of the sensitive branch.
- FFVAE [3]: Based on previous disentangled representation learning methods, FFVAE tries to explicitly separate sensitive dimensions from non-sensitive dimensions in the latent code by learning the sensitive latent part with supervised learning.
- FD-VAE [24]: FD-VAE separates the latent code into three portions, i.e., sensitive dimensions, downstream-task-related dimensions, and mutual-information dimensions. FD-VAE trains the downstream task model using the latent code without sensitive dimensions while trying to exclude sensitive information from mutual-information dimensions with adversarial training.

Before training the encoder, ADT and FD-VAE require to specify the sensitive attributes and the downstream task attribute, while FFVAE requires to specify the sensitive attributes. In contrast, our framework does not require to specify either of them and has the highest flexibility. Since the debiasing process in our framework is not based on adversarial training, DVGE is more stable and easier to train than the baselines.

5.1.3 Datasets. In the experiments, we use two commonly used datasets. One is an unstructured dataset, which is CelebA[2] [18], while the other is a structured dataset, which is South German Credit[3] [9]. CelebA has 202,599 facial images, each of which is associated with 40 attributes, such as “Attractive", “Male", “Young". And all attributes are in binary form. As for the structured dataset, South German Credit has 1,000 entries with 21 attributes. The first 20 attributes are the information about the loan applicants (gender, age, income, etc.), and the last one is the loan application result. Since some attributes in South German Credit are in category form, we convert them into numerical form for convenience.

5.1.4 Metrics. In the experiments, we compare our framework with the baselines on the fairness-accuracy trade-off. Specifically, we consider two common fairness metrics, the distance to demographic parity Δ_{DP} and the distance to equal opportunity Δ_{EO} (refer to Section 3). We calculate the fairness metrics against the accuracy (Acc.) of the downstream task model, and plot the Pareto fronts of them to show the fairness-accuracy trade-off. **Better fairness-accuracy trade-off indicates higher accuracy with lower Δ_{DP} and Δ_{EO}.** In order to obtain the fairness-accuracy trade-off for our framework and the baselines, we sweep a range of the value combinations of hyperparameters in their objective functions.

5.2 Experiment Results on CelebA

On CelebA, we select three different combinations of sensitive attributes and downstream tasks for the experiments on the unstructured dataset. Because of the flexibility, DVGE uses the same latent code encoder for the following three different experiments.

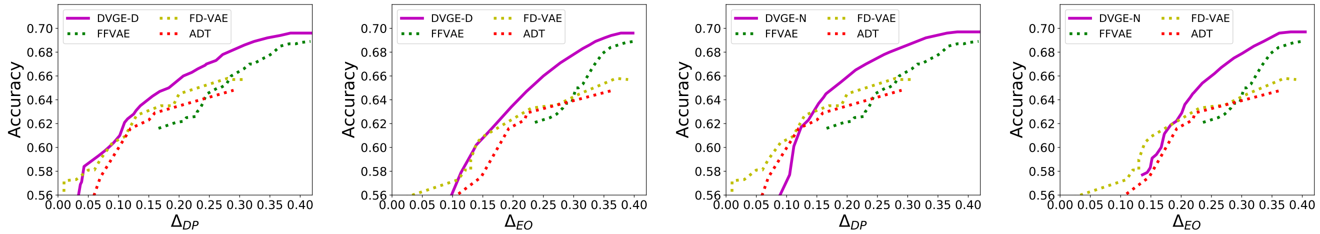
5.2.1 Experiment 1. We choose “Male" as the sensitive attribute and set the downstream task to predict the label “Oval_Face". For this task, if we had a perfect classifier (100% accuracy), its Δ_{DP} would be 0.094, indicating almost no fairness problem in this task. When there is no fairness problem, the accuracy of the downstream task model should not vary with Δ_{DP} or Δ_{EO}. This experiment is designed to verify that DVGE has no negative impacts on tasks without fairness problems.

As we can observe in Figure 3, the accuracy of the downstream task model barely changes when Δ_{DP} or Δ_{EO} increases for our framework and the baselines. In addition, our framework can maintain the model accuracy even when Δ_{DP} and Δ_{EO} are very close to 0. We can also observe that our framework achieves slightly better accuracy than FFVAE and FD-VAE, because they remove dimensions of the latent code and suffers from incomplete disentanglement, resulting in information loss for downstream tasks.

5.2.2 Experiment 2. The sensitive attribute for this experiment is also “Male", but we change the task label to “Attractive". Δ_{DP} for a perfect classifier in this task would be 0.398, indicating a serious

[2] https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[3] https://archive.ics.uci.edu/ml/datasets/South+German+Credit



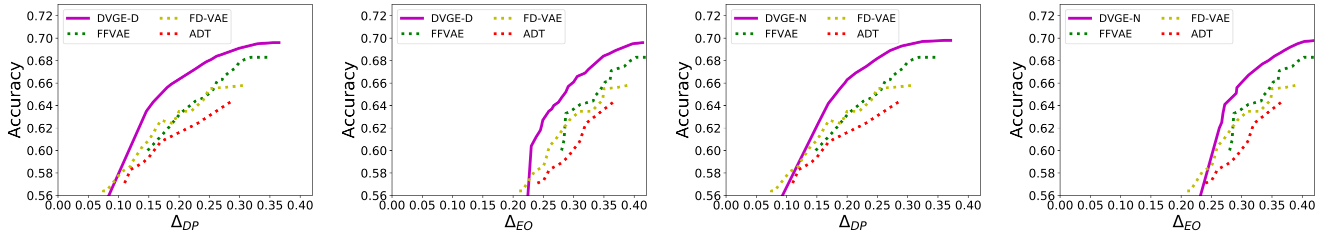
(a) DVGE-D, Δ_{DP}

(b) DVGE-D, Δ_{EO}

(c) DVGE-N, Δ_{DP}

(d) DVGE-N, Δ_{EO}

Figure 4: Fairness-accuracy trade-off comparison results for Experiment 2: CelebA dataset, sensitive attribute = "Male", task label = "Attractive".



(a) DVGE-D, Δ_{DP}

(b) DVGE-D, Δ_{EO}

(c) DVGE-N, Δ_{DP}

(d) DVGE-N, Δ_{EO}

Figure 5: Fairness-accuracy trade-off comparison results for Experiment 3: CelebA dataset, sensitive attribute = "Male" $\wedge$ "Young", task label = "Attractive".

fairness problem. This experiment evaluates DVGE when debiasing in the setting of single sensitive attributes.

As we can see from the experiment results in Figure 4, our framework outperforms the baselines by a relatively large margin. For example in Figure 4(a), DVGE-D almost always achieves higher accuracy than the baselines when at the same Δ_{DP}. More importantly, our framework achieves similar fairness-accuracy trade-off with a non-disentangled VAE setting (DVGE-N in Figure 4(c) and 4(d)) as with disentangled VAE setting, which demonstrates that our framework does not rely on complete disentanglement for debiasing.

5.2.3 Experiment 3. In order to demonstrate the flexibility and superiority of our framework in the case of multiple sensitive attributes, we consider the conjunction of two sensitive attributes in this experiment. Specifically, the sensitive attributes are "Male" and "Young", denoted as "Male" $\wedge$ "Young"[4], and the task is still to predict the label "Attractive". Here, we train a sensitive classifier to jointly distinguish the two sensitive attributes from the latent code. Δ_{DP} for a perfect classifier in this task would be 0.445, suggesting an even more serious fairness problem than those in previous tasks.

We depict the results for this experiment in Figure 5. As we can observe, our framework overall achieves better fairness-accuracy trade-off than the baselines. For example in Figure 5(b), when achieving the same Δ_{EO}, DVGE-D always hits higher downstream task accuracy than other baselines. Even when Δ_{DP} or Δ_{EO} moves close to 0, and the gaps of the fairness-accuracy trade-off between our framework and the baselines get smaller, our framework still outperforms or is on par with the baselines.

5.3 Experiment Results on South German Credit

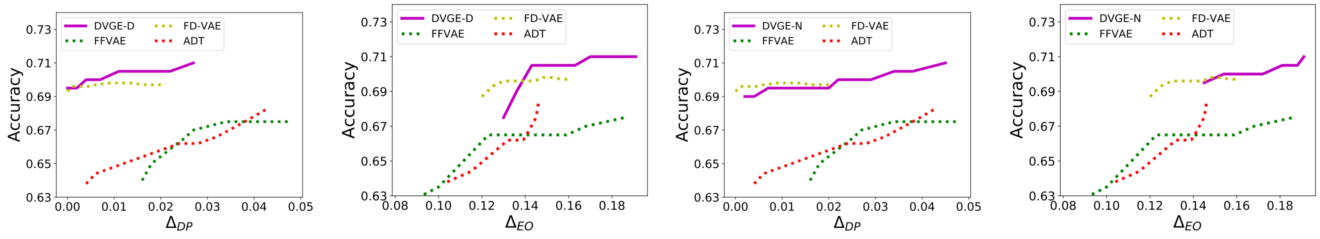
On South German Credit, we choose two different combinations of sensitive attributes for the experiments on the structured dataset. DVGE uses the same latent code encoder for the following two different experiments.

5.3.1 Experiment 4. We select "age" as the sensitive attribute and the downstream task is to predict the label of "credit_risk" in this experiment. Δ_{DP} of a perfect classifier in this task would be 0.188. This experiment is designed for testing our framework when dealing with single sensitive attributes.

The experiment results are demonstrated in Figure 6. As we can observe, when the fairness metric is Δ_{DP}, both our framework and the baselines can largely reduce the unfairness of the downstream task model, but our framework achieves much higher accuracy than FFVAE and ADT. When we measure with Δ_{EO}, FFVAE and ADT achieve lower values of Δ_{EO}, but their downstream task accuracy is still lower than our framework. And our framework performs on par with or slightly better than FD-VAE.

5.3.2 Experiment 5. In this experiment, we evaluate our framework when debiasing in the setting of multiple sensitive attributes in structured dataset. We consider the conjunction of "age" and

[4] $\wedge$ represents logical *and*.



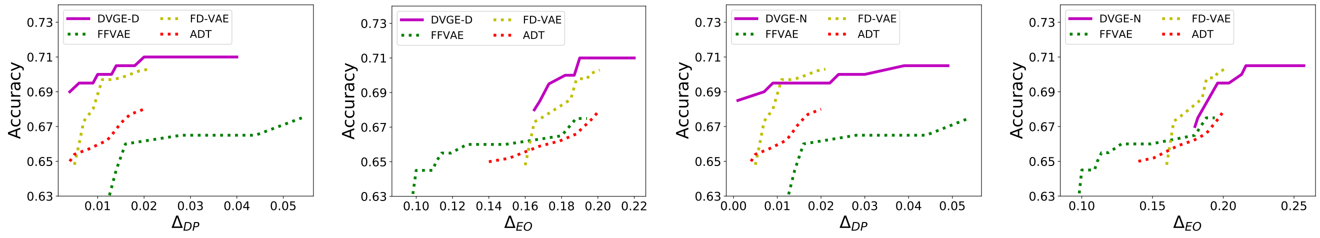
(a) DVGE-D, Δ_{DP}

(b) DVGE-D, Δ_{EO}

(c) DVGE-N, Δ_{DP}

(d) DVGE-N, Δ_{EO}

Figure 6: Fairness-accuracy trade-off comparison results for Experiment 4: South German Credit dataset, sensitive attribute = "age", task label = "credit_risk".



(a) DVGE-D, Δ_{DP}

(b) DVGE-D, Δ_{EO}

(c) DVGE-N, Δ_{DP}

(d) DVGE-N, Δ_{EO}

Figure 7: Fairness-accuracy trade-off comparison results for Experiment 5: South German Credit dataset, sensitive attribute = "age" $\wedge$ "foreign_worker", task label = "credit_risk".

"foreign_worker" as sensitive attributes. The downstream task is still to predict the label of "credit_risk".

As we can observe in Figure 7, the Pareto fronts suggest similar experiment results as those in the setting of a single sensitive attribute in Section 5.3.1. When the extent of fairness is measured by Δ_{DP}, our framework outperforms the baselines by a large margin. When the extent of fairness is measured by Δ_{EO}, the fairness-accuracy trade-off of our framework is still comparable to that of the baselines.

5.4 Ablation

To further evaluate the coverage on sensitive information in our framework, we conduct ablation experiments on CelebA [18]. Specifically, we use the latent code perturbed by our framework to retrain sensitive classifiers. We vary the hyperparameter η_1 (sensitive focus) while setting $\eta_2 = 0$, and observe the highest accuracy that the retrained sensitive classifiers can achieve. Here, we use the highest accuracy of the retrained sensitive classifiers to indicate the coverage on sensitive information. The rationale of this measurement is that better coverage leads to less sensitive information in the perturbed latent code, and further the sensitive classifiers retrained with it are less accurate. **In turn, lower accuracy of the retrained sensitive classifiers indicates better coverage on sensitive information.** For comparison, we retrain sensitive classifiers using the latent code with sensitive dimensions removed [3] and the latent code without removal, respectively. The encoders we use here are a disentangled VAE (FactorVAE [13]) and a non-disentangled VAE (VanillaVAE [14]).

First, we consider a single sensitive attribute "Male". The ablation results are shown in Table 1. As we can observe, when η_1 increases for DVGE, the highest accuracy of the retrained sensitive classifier decreases accordingly. Furthermore, when η_1 increases to only 0.2, DVGE achieves better coverage on sensitive information than the approach based on removing sensitive dimensions. Second, we consider two sensitive attributes, "Male" and "Young". The results are in Table 2. As we can see, when η_1 increases to only 0.3, the highest accuracy of the retrained sensitive classifier with DVGE is lower than that with the approach based on removing sensitive dimensions. The ablation results demonstrate that the sensitive focus in DVGE effectively covers sensitive information.

5.5 Discussions

First, from the experiments above, we can observe that DVGE overall achieves better fairness-accuracy trade-off than the baselines. Second, the ablation study shows that the sensitive focus in our framework effectively covers sensitive information in the latent code. Third, we can also observe that DVGE-D generally performs better than DVGE-N from all the experiments above.

6 CONCLUSION

In this paper, we targeted at the fairness problem in machine learning and followed the idea of using representation learning to tackle

Table 1: Debiasing performance of DVGE in the setting of single sensitive attribute

Encoder	No removal	Sens. dim.	DVGE with η_1									
		removed	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Disentangled	0.798	0.736	0.767	0.735	0.706	0.682	0.675	0.661	0.655	0.658	0.650	0.648
Non-disentangled	0.804	0.746	0.769	0.733	0.705	0.692	0.686	0.682	0.682	0.674	0.671	0.668

Table 2: Debiasing performance of DVGE in the setting of multiple sensitive attributes

Encoder	No removal	Sens. dim.	DVGE with η_1									
		removed	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Disentangled	0.752	0.690	0.732	0.707	0.680	0.661	0.644	0.638	0.637	0.633	0.633	0.631
Non-disentangled	0.757	0.704	0.736	0.709	0.683	0.664	0.657	0.653	0.653	0.651	0.653	0.649

it. To overcome the problem of downstream task accuracy degradation and the problem of insufficient coverage on sensitive information, we proposed DVGE that exploits the gradient-based explanation to obtain the model focuses for respectively predicting sensitive attributes and downstream task labels, and perturbs the latent code with the focuses for the purposes of fairness and prevention of downstream task accuracy degradation. We experimentally demonstrated that our framework achieves better fairness-accuracy trade-off and better coverage on sensitive information while not relying on complete disentanglement for debiasing.

REFERENCES

[1] Toon Calders and Sicco Verwer. 2010. Three naive bayes approaches for discrimination-free classification. *Data mining and knowledge discovery* 21, 2 (2010), 277–292.

[2] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942* (2018).

[3] Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. 2019. Flexibly fair representation learning by disentanglement. In *ICML*. PMLR, 1436–1445.

[4] Sen Cui, Weishen Pan, Changshui Zhang, and Fei Wang. 2021. Towards model-agnostic post-hoc adjustment for balancing ranking fairness and algorithm utility. In *Proceedings of the 27th ACM SIGKDD*. 207–217.

[5] Alexander D'Amour, Hansa Srinivasan, James Atwood, Pallavi Baljekar, D Sculley, and Yoni Halpern. 2020. Fairness is not static: deeper understanding of long term fairness via simulation studies. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 525–534.

[6] Julia Dressel and Hany Farid. 2018. The accuracy, fairness, and limits of predicting recidivism. *Science advances* 4, 1 (2018), eaao5580.

[7] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 1 (2016), 2096–2030.

[9] U Groemping. 2019. South German credit data: Correcting a widely used data set. *Rep. Math., Phys. Chem., Berlin, Germany, Tech. Rep* 4 (2019), 2019.

[10] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016), 3315–3323.

[11] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

[12] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. 2019. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9012–9020.

[13] Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. In *ICML*. PMLR, 2649–2658.

[14] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[15] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. 2019. ifair: Learning individually fair data representations for algorithmic decision making. In *2019 ieee 35th ICDE)*. IEEE, 1334–1345.

[16] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. 2019. Operationalizing individual fairness with pairwise fair representations. *arXiv preprint arXiv:1907.01439* (2019).

[17] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 305–314.

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of ICCV*.

[19] Francesco Locatello, Gabriele Abbati, Tom Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. 2019. On the fairness of disentangled representations. *arXiv preprint arXiv:1905.13662* (2019).

[20] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*. PMLR, 4114–4124.

[21] Frank Marcinkowski, Kimon Kieslich, Christopher Starke, and Marco Lünich. 2020. Implications of AI (un-) fairness in higher education admissions: the effects of perceived AI (un-) fairness on exit, voice and organizational reputation. In *Proceedings of the 2020 ACM FAccT*. 122–130.

[22] Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. 2021. Disentangling Preference Representations for Recommendation Critiquing with ß-VAE. In *Proceedings of the 30th ACM CIKM*. 1356–1365.

[23] Achraf Oussidi and Azeddine Elhassouny. 2018. Deep generative models: Survey. In *2018 ISCV*. IEEE, 1–8.

[24] Sungho Park, Sunhee Hwang, Dohyung Kim, and Hyeran Byun. 2021. Learning Disentangled Representation for Fair Facial Attribute Classification via Fairness-aware Information Alignment. In *Proceedings of AAAI*, Vol. 35. 2403–2411.

[25] Stephen Pfohl, Ben Marafino, Adrien Coulet, Fatima Rodriguez, Latha Palaniappan, and Nigam H Shah. 2019. Creating fair models of atherosclerotic cardiovascular disease risk. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 271–278.

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

[27] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).

[28] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017).

[29] Helen Smith. 2020. Algorithmic bias: should students pay the price? *Ai & Society* 35, 4 (2020), 1077–1078.

[30] Suraj Srinivas and François Fleuret. 2019. Full-gradient representation for neural network visualization. *arXiv preprint arXiv:1905.00780* (2019).

[31] Chris Sweeney and Maryam Najafian. 2020. Reducing sentiment polarity for demographic attributes in word embeddings using adversarial learning. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 359–368.

[32] Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *2018 ieee/acm international workshop on software fairness (fairware)*. IEEE, 1–7.

[33] Yuyan Wang, Xuezhi Wang, Alex Beutel, Flavien Prost, Jilin Chen, and Ed H Chi. 2021. Understanding and Improving Fairness-Accuracy Trade-offs in Multi-Task Learning. *arXiv preprint arXiv:2106.02705* (2021).

[34] An Yan and Bill Howe. 2021. EquiTensors: Learning Fair Integrations of Heterogeneous Urban Data. In *Proceedings of the 2021 International Conference on Management of Data*. 2338–2347.

[35] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 335–340.

[36] Hantian Zhang, Xu Chu, Abolfazl Asudeh, and Shamkant B Navathe. 2021. OmniFair: A Declarative System for Model-Agnostic Group Fairness in Machine Learning. In *Proceedings of the 2021 International Conference on Management of Data*. 2076–2088.

A PREVIOUS DEBIASING APPROACHES VIA REMOVING SENSITIVE DIMENSIONS USING DISENTANGLED REPRESENTATION LEARNING

Debiasing by exploiting disentangled representation learning was first proposed by Creager et al. in FFVAE [3] and also used by FD-VAE [24]. These approaches begin with training an encoder $f(x)$ and a decoder using disentangled representation learning methods. Then, the encoder is used to produce disentangled latent code $z = f(x)$. Next, the latent code dimensions corresponding to the sensitive attributes z_s (also known as sensitive dimensions) are determined by calculating the correlation between each dimension of z and the sensitive attributes s or pre-designation. At last, these approaches use the latent code without sensitive dimensions $z \backslash z_s$ to train downstream task models $\hat{y} = g(z \backslash z_s)$. During inference, these approaches also need to remove sensitive dimensions from the latent code before feeding the code to downstream task models. In contrast, our framework DVGE does not need to make changes to the latent code during inference.

To further elaborate on these previous approaches, we perform a causal analysis on them by illustrating the structural causality model (SCM) of downstream tasks in Figure 8(a). As we can observe, because of $s \longrightarrow z$ and $p \longrightarrow z$, when we exploit the latent code z to predict the label y, both sensitive attributes s and proxy attributes p (proxies for s) are considered as confounders that cause biased predictions. Since there is no guarantee of complete disentanglement from current disentangled representation learning on real-world data [22], when previous debiasing approaches remove the dimensions correlated with sensitive attributes, the sensitive information from proxy attributes and some information from sensitive attributes is overlooked. As a result, in Figure 8(a), the link $p \longrightarrow z$ is not disconnected, still causing biased predictions. In our framework, we target at breaking both $s \longrightarrow z$ and $p \longrightarrow z$.

B IMPLEMENTATION DETAILS

The platform for all the experiments in this paper is an Ubuntu 20.04 system equipped with Nvidia V100 GPUs. The implementation is based on PyTorch.

There are basically three steps to implement DVGE. First, we train VAEs to produce the latent code. Then, we train a sensitive classifier with the latent code. Finally, we train the downstream task model with the latent code according to our framework.

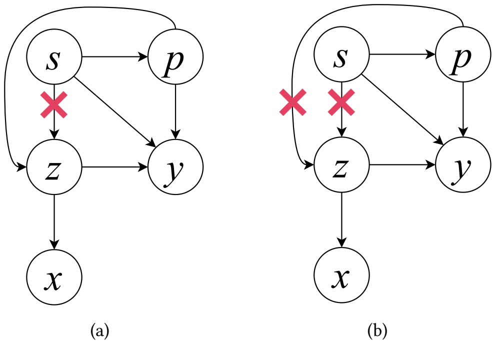


Figure 8: (a) Previous debiasing approaches using disentangled representation learning only break the link between the latent code z and sensitive attributes s in the structural causal model (SCM) when predicting downstream task attributes. (b) DVGE further breaks the link between the latent code z and proxy attributes p.

B.1 For CelebA

We resize the CelebA [18] images to the size of 64 × 64. For a fair comparison, we implement the encoder of VAEs (VanillaVAE [14], FactorVAE [13], FFVAE [3], and FD-VAE [24]) and the feature encoder of ADT [8] with the same architecture. In terms of implementing VAEs, we follow Kim et al. [13] and Creager et al. [3] to use a CNN for the encoder, a Deconvolutional Neural Network for the decoder, and an MLP for the discriminator. The detailed structure information is shown in Table 3. To train VAEs, we set the learning rate to 10^{-4} and use Adam optimizer with $\beta_1 = 0.9$ and $\beta_1 = 0.999$. To train the discriminator, we set the learning rate to 10^{-5} and use the Adam optimizer with $\beta_1 = 0.5$ and $\beta_1 = 0.9$. The batch size is 64, and we update them for 10^6 times (about 316 epochs). The input images are encoded into the latent codes with 10 dimensions. In terms of FFVAE, we designate the last one or two dimensions as sensitive dimensions. For FD-VAE, we designate the first three dimensions as downstream-task-related dimensions, the middle four dimensions as mutual-information dimensions, and the last three as sensitive dimensions. As for the hyperparameters of FD-VAE, we set them as in [24], As for the hyperparameters of other VAEs (γ for FactorVAE, α and γ for FFVAE), we sweep their values from 1.0 to 6.4.

Sensitive classifiers, downstream task models, and branches of ADT share the same structure with the discriminator for VAEs as shown in Table 3. To train ADT, sensitive classifiers, and downstream task models, we set the learning rate to 10^{-5} and use Adam optimizer with $\beta_1 = 0.5$ and $\beta_1 = 0.9$. We train sensitive classifiers for 120 epochs, and downstream task models and ADT for 100 epochs. We sweep η_1 and η_2 from 0.1 to 2.0. And we set ϵ_i to $0.1 \times |z_i|$, where i is the index for the latent code dimensions.

B.2 For South German Credit

The values of some attributes in South German Credit [9] are continuous, while others are categorical. To balance the value ranges, we normalize the attributes whose values are continuous with the

Table 3: Structure of VAE, sensitive classifier, downstream task model, and ADT for CelebA

VAE Encoder, ADT Feature Encoder	VAE Decoder	Discriminator, Sensitive Classifier, Dowstream Task Model, and ADT Branches
Input 64 × 64 image	Input ∈ $\mathbb{R}^{10}$	Input ∈ $\mathbb{R}^{10}$
Conv2d(3,32,4,2,1) with ReLU	Conv2d(10,256,1) with ReLU	Linear(10,1000) with LeakyReLU(0.2)
Conv2d(32,32,4,2,1) with ReLU	ConvTrans2d(256,64,4) with ReLU	Linear(1000,1000) with LeakyReLU(0.2)
Conv2d(32,64,4,2,1) with ReLU	ConvTrans2d(64,64,4,2,1) with ReLU	Linear(1000,1000) with LeakyReLU(0.2)
Conv2d(64,64,4,2,1) with ReLU	ConvTrans2d(64,32,4,2,1) with ReLU	Linear(1000,1000) with LeakyReLU(0.2)
Conv2d(64,256,4,1) with ReLU	ConvTrans2d(32,32,4,2,1) with ReLU	Linear(1000,1000) with LeakyReLU(0.2)
Conv2d(256,2*10,1)	ConvTrans2d(32,3,4,2,1)	Linear(1000,2)

Table 4: Structure of VAE, sensitive classifier, downstream task model, and ADT for South German Credit

VAE Encoder, ADT Feature Encoder	VAE Decoder	Discriminator, Sensitive Classifier, Dowstream Task Model, and ADT Branches
Input ∈ $\mathbb{R}^{20}$	Input ∈ $\mathbb{R}^{10}$	Input ∈ $\mathbb{R}^{10}$
Linear(20,1000) with LeakyReLU(0.2)	Linear(10,1000) with LeakyReLU(0.2)	
Linear(1000,1000) with LeakyReLU(0.2)		
Linear(1000,1000) with LeakyReLU(0.2)		
Linear(1000,1000) with LeakyReLU(0.2)		
Linear(1000,1000) with LeakyReLU(0.2)		
Linear(1000,20)		Linear(1000,2)

Table 5: The debiasing performance of DVGE in the setting of single sensitive attribute

Encoder	No Removal	Sens. dim. removed	DVGE with η_1									
			0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Disentangled	0.798	0.718	0.772	0.726	0.677	0.633	0.595	0.557	0.528	0.500	0.478	0.458
Non-disentangled	0.804	0.722	0.770	0.719	0.667	0.621	0.581	0.545	0.513	0.489	0.465	0.447

Table 6: The debiasing performance of DVGE in the setting of multiple sensitive attributes

Encoder	No Removal	Sens. dim. removed	DVGE with η_1									
			0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Disentangled	0.752	0.670	0.732	0.697	0.662	0.627	0.599	0.572	0.549	0.530	0.513	0.496
Non-disentangled	0.757	0.677	0.736	0.701	0.663	0.628	0.596	0.571	0.548	0.528	0.512	0.496

maximal value, and convert the categorical attributes into $[0, 1]$. In terms of implementing VAEs, we use MLPs for the encoder, the decoder, and the discriminator. The structure of the feature encoder of ADT is the same as that of VAE encoders. The detailed structure information for implementation is shown in Table 4. The training parameters for VAEs and the discriminator, and the latent code configurations are the same as in Appendix B.1.

For South German Credit, sensitive classifiers, downstream task models, and branches of ADT also share the same structure with the discriminator for VAEs as shown in Table 4. Their training parameters are also the same as in Appendix B.1.

B.3 Gradient-based Explanation

For getting gradient-based explanations from sensitive classifiers and downstream task models, we follow Srinivas et al. [30] use predictions on the input to compute the loss for backpropagation, instead of the ground truth labels.

C MORE ON DEBIASING ABLATION

To further specifically evaluate how the sensitive focus influences the coverage on sensitive information in our framework, we design more ablation experiments on CelebA which are different from those in Section 5.4.

In these experiments, we do not retrain sensitive classifiers, but instead directly test the accuracy of the sensitive classifiers which achieve the best accuracy before with the perturbed latent code by our framework. The perturbed latent code is generated with different configurations of the hyperparameter η_1 but without being perturbed by downstream task focus ($\eta_2 = 0$). Then we observe the accuracy that the sensitive classifiers can achieve. Here, we use the accuracy of the sensitive classifiers to indicate the coverage on sensitive information. The lower the accuracy is, the better the coverage on sensitive information is. To compare with our framework, we also test sensitive classifiers with the modified latent code by the existing approach and the latent code without modifications, respectively. The VAEs used in these experiments are a disentangled VAE (FactorVAE) and a non-disentangled VAE (VanillaVAE).

First, we test with the setting of a single sensitive attribute which is set to "Male". The experiment results are demonstrated in Table 5. As we can observe, when we increase η_1 from 0.1 to 1.0, the accuracy of the sensitive classifier decreases from 0.772 to 0.458 for disentangled VAE, and from 0.770 to 0.447 for non-disentangled VAE, which suggests that the sensitive focus in our framework effectively covers the sensitive information with the setting of single sensitive attributes. In addition, when η_1 increases to only 0.2, our framework achieves comparable coverage on sensitive information with the approach based on removing sensitive dimensions. Second, we test with the setting of two sensitive attributes, which are set to "Male" and "Young". The ablation results are shown in Table 6. As we can see, the results are similar to those in Table 5. With η_1 increasing from 0.1 to 1.0, the accuracy of the sensitive classifier decreases accordingly for both disentangled VAE and non-disentangled VAE. And when η_1 is equal to or greater than 0.3, our framework outperforms the approach based on removing sensitive dimensions on the coverage on sensitive information. These results demonstrate that our framework has a good coverage on sensitive information with the setting of multiple sensitive attributes.