Appendix

A. Implementation Details

Here we introduce the implementation details. All experiments are implemented in PyTorch [43].

A.1. Hyperparameters

The following hyperparameters in this paragraph are used across all experiments on all datasets. We use Adam [26] to optimize the biased attribute hyperplane h_{b}, where β_1, and β_2 are 0.9, and 0.999, respectively. The number of iterations for both Hessian Penalty [44] and our method is 1000.

In the experiments on disentanglement datasets (Sec. 5.1), we set the batch size to 64. The learning rate is 10^{-3}. The coefficient of *orthogonalization penalty* (see λ in Eq. 5) is 10. In terms of traversal images, we set the number of traversal steps N (see N in Eq. 3) to 20, where the step sizes $\{\alpha_i \mid i = 1 \dots N\}$ (see α_i in Eq. 3) are numbers evenly spaced in the interval $[-2, 2]$.

In the experiments on face images (Sec. 5.2), the learning rate is 10^{-1}. The coefficient of *orthogonalization penalty* is 100. In the experiments on images from other domains (Sec. 5.3), the learning rate is 10^{-3}. In both previous experiments, we set the batch size to 1 and set the number of traversal steps N (see N in Eq. 3) to 6, where the step sizes $\{\alpha_i \mid i = 1 \dots N\}$ (see α_i in Eq. 3) are numbers evenly spaced in the interval $[-3, 3]$.

A.2. Projecting Latent Code to Hyperplane

We use the offset o_b for projecting randomly sampled latent code $\mathbf{z}$ to the hyperplane h_{b}. To explain that o_b is jointly optimized with $\mathbf{w}_{\mathsf{b}}$ when minimizing the *total variation loss* $\mathcal{L}_V$, we describe the complete algorithm for computing *total variation loss* $\mathcal{L}_V$.

We define the biased attribute's hyperplane h_{b} as $\mathbf{w}_{\mathsf{b}}^T \mathbf{x} + o_{\mathsf{b}} = 0$. Therefore, the projected sampled latent code $\mathbf{z}_{\text{proj}}$ can be computed by:

$$\mathbf{z}_{\text{proj}} = \mathbf{z} - \frac{\mathbf{w}_{\mathsf{b}}^T \mathbf{z} + o_{\mathsf{b}}}{||\mathbf{w}_{\mathsf{b}}||^2} \mathbf{w}_{\mathsf{b}}. \tag{6}$$

Then, the projected latent code $\mathbf{z}_{\text{proj}}$ is used to sample traversal latent codes $\{\mathbf{z}_{\text{proj}} + \alpha_i \frac{\mathbf{w}_{\mathsf{b}}}{||\mathbf{w}_{\mathsf{b}}||} \mid i = 1 \dots N\}$, which are fed to generative model G for synthesizing traversal images:

$$\mathbf{I}(\mathsf{b} = b_i) = G(\mathbf{z}_{\text{proj}} + \alpha_i \frac{\mathbf{w}_b}{||\mathbf{w}_b||}). \tag{7}$$

Next, the traversal images $\{\mathbf{I}(\mathsf{b} = b_i) \mid i = 1 \dots N\}$ are fed to the classifier for predicting the target attributes $\{P(\hat{\mathsf{t}} \mid \mathbf{I}(\mathsf{b} = b_i)) \mid 1 \dots N\}$. Finally, the *total variation loss* can be computed by:

$$\begin{aligned} &\mathcal{L}_V(\mathbf{w}_{\mathsf{b}}, o_{\mathsf{b}}) = \\ &-\log \frac{1}{N-1} \sum_{i=1}^{N-1} |P(\hat{\mathsf{t}} \mid G(\mathbf{z}_{\text{proj}} + \alpha_{i+1} \frac{\mathbf{w}_{\mathsf{b}}}{||\mathbf{w}_{\mathsf{b}}||})) \\ &\qquad - P(\hat{\mathsf{t}} \mid G(\mathbf{z}_{\text{proj}} + \alpha_i \frac{\mathbf{w}_{\mathsf{b}}}{||\mathbf{w}_{\mathsf{b}}||}))|. \end{aligned} \tag{8}$$

Note that the *total variation loss* $\mathcal{L}_V$ is a function of both $\mathbf{w}_{\mathsf{b}}$ and o_{b} because $\mathbf{z}_{\text{proj}}$ is computed based on both of them. Therefore, we can optimize both $\mathbf{w}_{\mathsf{b}}$ and o_{b} when minimizing the *total variation loss* $\mathcal{L}_V$.

Algorithm 1: Compute Ground-truth Hyperplanes

Input: $\{\mathbf{I}, \mathbf{a}\}$: set of pairs of image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ and attribute label $\mathbf{a} \in \mathbb{R}^J$ (J: number of attributes); $E : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^d$: image encoder of VAE-based model.

Output: $Q \in \mathbb{R}^{d \times J}$: normal vectors of attribute hyperplane; $\mathbf{o} \in \mathbb{R}^J$: offsets of attribute hyperplane

$\{\mathbf{z}\} \coloneqq E(\{\mathbf{I}\})$ /* Encode all images into the latent space. */

randomly initialize a matrix $W \in \mathbb{R}^{d \times J}$

randomly initialize a vector $\mathbf{o} \in \mathbb{R}^J$

for *each iteration* **do**

  $Q \coloneqq \text{QR-decomposition}(W)$

  $\mathbf{p} \coloneqq \text{sigmoid}(Q^T \mathbf{z} + \mathbf{o})$

  $l \coloneqq \text{BCE}(\mathbf{p}, \mathbf{a})$ or $\text{MSE}(\mathbf{p}, \mathbf{a})$ /* use binary cross-entropy (BCE) loss for binary attribute and mean squared error for continuous valued attribute */

  $W, \mathbf{o} \coloneqq \text{Adam}(l)$ /* update with Adam optimizer */

A.3. Ground-truth Hyperplanes on Disentanglement Datasets

Here we describe how to compute the ground-truth hyperplane for all attributes in the dataset. We do not follow previous works [3, 23, 46] to compute the hyperplane via SVM [9] or logistic regression [41] because we observe a strong correlation of variation from different attributes even with the orthogonalization trick introduced by Balakrishnan *et al.* [3]. We suspect that the correlation exists because the hyperplanes are computed individually. Therefore, we propose a new method to optimize all hyperplanes jointly. The algorithm is described in Algorithm 1.

Suppose the dataset has J attributes (*i.e.*, $J = 4, J = 5$ for SmallNORB [32] and dSprites [20], respectively) and

the dimension of VAE-based model's latent space is d. We are also given the dataset's images $\{\mathbf{I}\}$ and corresponding attributes labels $\{\mathbf{a} \in \mathbb{R}^J\}$. Then we encode the images $\{\mathbf{I}\}$ to latent codes $\{\mathbf{z}\}$ by the pre-trained encoder of the VAE-based model. We will use latent codes $\{\mathbf{z}\}$ paired with the attribute labels $\{\mathbf{a} \in \mathbb{R}^J\}$ as training data to compute the ground-truth hyperplanes.

After obtaining the training data, we initialize a matrix $W \in \mathbb{R}^{d \times J}$, where the j-th column in W represents the normal vector of j-th attribute's hyperplane. Similarly, we initialize a vector $\mathbf{o} \in \mathbb{R}^J$, where j-th value in $\mathbf{o}$ represents the offset of j-th attribute's hyperplane. Then, we perform QR-decomposition on W to obtain the orthogonal matrix $Q = \text{QR-decomposition}(W)$. Next, Q and $\mathbf{o}$ are used to classify the latent codes $\{\mathbf{z}\}$. Supervised by the attribute labels $\{\mathbf{a}\}$, we optimize W and $\mathbf{o}$ iteratively via Adam optimizer. After the optimization, Q and $\mathbf{o}$ are the ground-truth normal vectors and offsets of hyperplanes for all attributes in the given dataset. Since we use five different VAE-based models to compute latent codes $\{\mathbf{z}\}$, the ground-truth hyperplane of the same attribute is different for different VAE-based models.

The computed ground-truth hyperplanes by the method mentioned above are used for evaluation. However, we cannot use them in the *orthogonalization penalty*. The reason is that the orthogonal matrix Q ensures the orthogonalization among the hyperplanes, which cannot be realized in the real-world setting where the biased attribute is *unknown* so that we cannot let the hyperplanes of the target attribute and known attributes be orthogonal with the *unknown* biased attribute. Therefore, the normal vectors for the *orthogonalization penalty* are computed in a different way. Suppose the b-th attribute is selected as the ground-truth biased attribute under an experimental setting (recall that an experimental setting is a triplet of (target attribute, biased attribute, generative model)). Then, we remove the b-th column in the optimized W to obtain a new matrix $W' \in \mathbb{R}^{J-1}$. Next, we apply the QR-decomposition to obtain Q' (*i.e.*, $Q' = \text{QR-docomposition}(W')$), so that column vectors in $Q' \in \mathbb{R}^{J-1}$ are used as normal vectors of hyperplanes for target attribute and known attributes in the *orthogonalization penalty*.

A.4. Pseudo-ground-truth of Biased Attribute on Face Images

As described in the main paper, since CelebA [34] and FFHQ [23] are in-the-wild datasets, we do not know the real ground-truth biased attributes. Therefore, for the quantitative evaluation, we obtain the pseudo-ground-truth of the biased attribute as follows: First, we assume a larger set of attributes as the potential biased attribute by adding *age*, *smile*, *eyeglasses*, and *pose* attributes into consideration. Then, we obtain the ground-truth hyperplanes of all five attributes (four attributes mentioned before plus the *gender* attribute) in the latent space of StyleGAN from Shen *et al.* [46]. Next, for each pair of a target attribute and a possible biased attribute (pairs of identical attributes are excluded), we generate traversal images based on the biased attribute, test them on a target attribute classifier, and record the total variation (TV). We pick the attribute that produces the largest total variation (TV) as the pseudo-ground-truth for the target attribute.

A.5. Generative Models

Experiments on Disentanglement Datasets We use the same set of hyperparameters with disentanglement-lib [36] to train VAE-based models and we use Disentanglement-PyTorch [1] as the code base for training. The dimension of latent space of all VAE-based generative models (vanilla VAE [27], β-VAE [20], DIP-VAE-I, and DIP-VAE-II [30]) is 10. The image size is 64×64. The optimizer for training VAE-based models is Adam ($\beta_1 = 0.9, \beta_2 = 0.999$) and the learning rate is 10^{-4}. The training steps is 3×10^5.

Experiments on Face Images We use two generative models: two StyleGANs[23] models trained on CelebA-HQ [22] and FFHQ [23], respectively. The weights are obtained from the officially released code from Shen *et al.* [46]. The dimension of latent space of models mentioned above is 512. The latent space of all models is the $\mathcal{W}$-space. The synthesized image size is 1024×1024 and we resize it to 64×64 before feed them to the classifier.

Experiments on Images from Other Domains StyleGAN [23] and StyleGAN2 [24] trained on the images from each category in LSUN [55] are used as generative models and weights are obtained from Shen *et al.* [47]. The size of the synthesized images is 256×256. We use $\mathcal{Z}$-space as the latent space of the generated model, where the dimension is 512.

A.6. Baseline Methods

Adaptation of Baseline Methods We choose unsupervised disentanglement methods as baselines. Here, we describe how to adapt them for *unknown biased attribute discovery task*. First, a trained unsupervised disentanglement method will predict a set of hyperplanes in the latent space. Then, we remove the hyperplane whose normal vector has the largest absolute value of cosine similarity with the normal vector of ground-truth target attribute hyperplane because it can be regarded as the predicted hyperplane for the target attribute. Then, we use each of the remaining hyperplanes to generate batches of traversal images, feed them to the classifier, and record the average total variation (TV, which will be introduced in Sec. B) over each batch. The hyperplane with

the largest average TV is selected as the predicted biased attribute hyperplane.

Results of VAE-based Method We create 480 different experiment settings in the experiments on disentanglement datasets, where each setting is a triplet of (target attribute, biased attribute, generative model). To help readers better understand how to compute the results for the VAE-based method, we use the following example to illustrate. Under the setting (*shape*, *scale*, β-VAE), we use the predicted axis-aligned hyperplane of β-VAE as the prediction for the VAE-based method. In other words, under each setting, the result of the VAE-based method depends on the generative model used in the experiment setting.

A.7. Attributes Preprocessing

We preprocess the attributes' values in disentanglement datasets if they are not binary-valued or continuous-valued. Here we introduce the details of attributes preprocessing.

For *shape* and *category* attributes in the disentanglement datasets, we choose a subset of shapes or categories as the positive class and others belong to the negative class. Therefore, we convert the *shape* and *category* attributes to binary-valued attributes. Concretely, for *category* attribute in Small-NORB [32], we choose "square" and "ellipse" as the positive class and "heart" as the negative class. For *shape* attribute in dSprites [20], we choose "four-legged animals" and "human figures" as the positive class and "airplanes," "trucks," and "cars" as the negative class.

A.8. Training Biased Classifiers on Disentanglement Datasets

In order to ensure that the classifier is biased to the chosen biased attribute (*i.e.*, ground-truth biased attribute), following the method in [54], we sample the disentanglement dataset with skewed distribution to train the classifier. Formally, we denote the biased attribute as b and the target attribute as t. First, for the sampling purpose, we transform the target attribute and the biased attribute to the binary-valued attributes if they are continuous-valued attributes. We achieve this by considering the values less than the medium value as the positive class and the values greater or equal than the medium value as the negative class. Note that such binary-valued attributes are only used for sampling and will not be used for training. Second, we uniformly sample the binary value of the target attribute (*i.e.*, $\mathsf{t}=0$ or $\mathsf{t}=1$). Next, we sample the value of the biased attribute based on the following skewed conditional distribution:

$$
\begin{aligned}
P(\mathsf{b}=0 \mid \mathsf{t}=1) &= S \\
P(\mathsf{b}=1 \mid \mathsf{t}=1) &= 1-S \\
P(\mathsf{b}=0 \mid \mathsf{t}=0) &= 1-S \\
P(\mathsf{b}=1 \mid \mathsf{t}=0) &= S,
\end{aligned}
$$

where S is the "skewness" of the conditional distribution for sampling the biased attribute. We set $S=0.9$ in all experiments on the disentanglement dataset. The ablation study on "skewness" is shown in Sec. C.3. After sampling the values for the biased attribute and the target attribute, we use them to uniformly sample the data with the sampled values in terms of the biased attribute and the target attribute from the dataset.

B. Evaluation Metric - Total Variation (TV)

B.1. Definition of TV

We also report the results with another evaluation metric – Total Variation (TV), which is formally defined by:

$$
\begin{aligned}
\mathrm{TV}(\mathbf{w}_\mathsf{b}, o_\mathsf{b}) = \frac{1}{N-1}\sum_{i=1}^{N-1} |P(\hat{\mathsf{t}} \mid G(\mathbf{z}_{\text{proj}} + \alpha_{i+1}\frac{\mathbf{w}_\mathsf{b}}{||\mathbf{w}_\mathsf{b}||})) \\
- P(\hat{\mathsf{t}} \mid G(\mathbf{z}_{\text{proj}} + \alpha_i \frac{\mathbf{w}_\mathsf{b}}{||\mathbf{w}_\mathsf{b}||}))|.
\end{aligned} \tag{9}
$$

Compared with the definition of *total variation loss* (Eq. 8), TV removes the $-\log$. Intuitively, TV captures the averaged absolute difference of classifier's predictions between each pair of consecutive steps. Therefore, larger TV values indicate larger variations of target attribute classifier's predictions on the traversal images. We add TV metric results of Tab. 1, 2, 3 in Tab. 4, 5, 6, respectively. We also report the TV results of the ground-truth biased attribute and the target attribute hyperplanes on disentanglement datasets and face image datasets in Tab. 7 and Tab. 8, respectively. To compute the TV of the target attribute hyperplane, we replace $\mathbf{w}_\mathsf{b}$ with $\mathbf{w}_\mathsf{t}$ in Eq. 9. Note that the TV values of predicted biased attribute from all methods are larger than the TV values of ground-truth biased attribute and ground-truth target attribute. We believe the reason is that all methods' predicted biased attribute hyperplanes are not perfectly orthogonal w.r.t. the ground-truth target attribute. Hence, both biased attribute and target attribute will vary in the traversal images, leading to larger classifier prediction variations than the ground-truth hyperplanes that only have single-attribute variations.

B.2. TV as an Unfairness Metric

Note that two cases of biased attribute prediction will cause large TV values:

1. the prediction is close to ground-truth of the biased attribute;

2. the prediction is close to the target attribute (*i.e.*, trivial solution explained in Sec. 4.3).

dataset	method	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{b}\rangle\rvert \uparrow$	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert \downarrow$	$\Delta\cos \uparrow$	%leading $\uparrow$	TV
SmallNORB	VAE-based	0.21±0.21	0.16±0.13	0.05±0.20	16.67%	0.15±0.07
	Hessian Penalty	**0.24±0.16**	0.26±0.16	-0.02±0.24	**31.67%**	0.14±0.05
	Ours	0.23±0.18	**0.10±0.11**	**0.12±0.21**	**51.67%**	0.13±0.05
dSprites	VAE-based	0.11±0.14	0.13±0.14	-0.01±0.16	22.00%	0.09±0.05
	Hessian Penalty	**0.23±0.15**	0.25±0.15	-0.02±0.21	**41.00%**	0.09±0.04
	Ours	0.17±0.14	**0.13±0.11**	**0.05±0.18**	**37.00%**	0.07±0.04

Table 4: Tab. 1 in the main paper with Total Variation (TV) results. The TV metric is introduced in Sec. B. We do not bold the TV results because the baseline methods achieve larger $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert$ (see explanations in Sec. B). The table shows mean and standard deviation results averaged over all 480 experiment settings on SmallNORB [32] and dSprites [20] datasets. Top-2 results under %leading metric are bolded. ↑: larger value means better result. ↓: smaller value means better result. Note that $\Delta\cos$ is the major evaluation metric that jointly considers the first two metrics. Our method achieves better performance than two baseline methods.

	$\mathcal{L}_H$	$\mathcal{L}_\perp$	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{b}\rangle\rvert \uparrow$	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert \downarrow$	$\Delta\cos \uparrow$	TV
SmallNORB			0.25±0.15	0.27±0.18	-0.02±0.23	0.15±0.05
		✓	0.23±0.18	**0.10±0.11**	**0.12±0.21**	0.13±0.05
	✓		**0.27±0.16**	0.28±0.17	-0.01±0.25	0.10±0.03
	✓	✓	0.25±0.17	0.15±0.13	0.10±0.24	0.10±0.03
dSprites			0.20±0.13	0.21±0.13	-0.01±0.18	0.07±0.04
		✓	0.17±0.14	**0.13±0.11**	**0.05±0.18**	0.07±0.04
	✓		**0.21±0.13**	0.21±0.13	0.00±0.18	0.06±0.04
	✓	✓	**0.21±0.13**	0.19±0.13	0.01±0.18	0.06±0.04

Table 5: Tab. 2 in the main paper with Total Variation (TV) results. The TV metric is introduced in Sec. B. The table shows the ablation study on *orthogonalization penalty* ($\mathcal{L}_\perp$) and Hessian Penalty [44] ($\mathcal{L}_H$). ✓denotes the penalty is used. Note that all rows used $\mathcal{L}_V$. We incorporate $\mathcal{L}_H$ into our method. Although adding $\mathcal{L}_H$ helps to improve $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{b}\rangle\rvert$, it seriously harms the $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert$. Overall, our final method (second row in each dataset) performs the best in $\Delta\cos$.

Therefore, **we regard TV as an *unfairness metric* only when the method has smaller** $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert$ (*i.e.*, nontrivial solution). In other words, larger TV value does not mean better result if the method also has larger $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert$. For example, in Tab. 4, although baseline methods have larger TV results, they also have larger $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert$. Therefore, baseline methods' biased attribute predictions are not more unfair than our method, but rather closer to the trivial solution. However, comparing the TV result with Hessian Penalty on face image datasets in Tab. 6, our method achieves larger TV and smaller $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert$ simultaneously. Therefore, on face image datasets, our method not only accurately predicts the biased attribute with larger unfairness results (*i.e.*, TV), but also avoids the trivial solution.

C. Ablation Studies

C.1. Ablation Study on Skewed Dataset for Training Generative Models

In the experiments on the disentanglement datasets, the distribution of the generative model's training set is balanced, meaning that the distribution of the target attribute and the distribution of the biased attribute are independent of each other. This setting may not be feasible when such a balanced training set is unavailable. To study the performance of our method and the baseline methods when the balanced training data is unavailable, we train the generative models with the *same* skewed distribution (*i.e.*, $S = 0.9$) in the training set for training the biased classifier (see Sec. A.8 in this supplementary material). To obtain accurate ground-truth for evaluation, we still use the balanced dataset to compute the ground-truth hyperplanes (method for computing the ground-truth hyperplanes is shown in Sec. A.3). The results on SmallNORB [32] dataset are shown in Tab. 10. Our method outperforms baseline methods in all metrics with the generative models trained on training data in the skewed distribution. While baseline methods' performances become worse when using the generative models trained on skewed data, our method can still maintain consistent performances. The results demonstrate that baseline methods perform worse when the training data is in a skewed distribution, and our method can better discover the biased attribute and achieve

	method	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{b}\rangle\rvert \uparrow$	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert \downarrow$	$\Delta\cos \uparrow$	TV
CelebA	$\mathcal{L}_H$	0.02±0.02	0.02±0.0003	0.0005±0.02	0.02±0.01
	Ours	**0.06±0.01**	**0.002±0.001**	**0.06±0.01**	**0.11±0.008**
FFHQ	$\mathcal{L}_H$	0.05±0.01	0.01±0.008	0.03±0.004	0.05±0.001
	Ours	**0.17±0.11**	**0.002±0.002**	**0.17±0.11**	**0.11±0.001**

Table 6: Tab. 3 in the main paper with Total Variation (TV) results. The table shows the results on CelebA [34] and FFHQ [23] datasets. We omit %leading since our method leads in all experiment settings (*i.e.*, %leading (Ours) = 100 %). We bold the TV results because our method also achieves smaller $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert$ (see explanation in Sec. B).

dataset	GT Biased TV	GT Target TV
SmallNORB	0.06±0.06	0.05±0.04
dSprites	0.03±0.04	0.04±0.03

Table 7: Total variation (TV) of the ground-truth biased attribute (GT Biased TV) and the ground-truth target attribute (GT Target TV) on SmallNORB and dSprites datasets.

classifier	StyleGAN	PGT Biased TV	GT Target TV
CelebA	CelebA-HQ	0.07	0.05
	FFHQ	0.07	0.04
FFHQ	CelebA-HQ	0.10	0.04
	FFHQ	0.06	0.11

Table 8: Total variation (TV) of the pseudo-ground-truth of biased attribute (PGT Biased TV) and ground-truth target attribute (GT Target TV) on face image datasets. The first two columns denote the training datasets of the target attribute classifier and StyleGAN, respectively.

	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{b}\rangle\rvert \uparrow$	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert \downarrow$	$\Delta\cos \uparrow$
CelebA	0.08±0.02	0.004±0.002	0.07±0.02
FFHQ	0.08±0.05	0.01±0.005	0.07±0.04

Table 9: Results of different random initializations of biased attribute hyperplane on face images. The results on two datasets (CelebA and FFHQ) are averaged over three random seeds. The generator is StyleGAN pretrained on FFHQ.

better disentanglement w.r.t. the target attribute.

C.2. Ablation Study on Known Attributes

In the *orthogonalization penalty*, users can provide a set of known attributes K to exclude the case that the discovered biased attribute is identical with one of the known attributes. Here we also show the results on SmallNORB [32] dataset when known attributes are not provided (*i.e.*, $K = \varnothing$) in Tab 11. The results show that known attributes make performance in $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{b}\rangle\rvert$ slightly worse and improve the performance in $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert$. We regard that the known attributes introduce a stronger constraint for discovering the biased attribute, but they are still helpful for better disentanglement w.r.t. the target attribute.

C.3. Ablation Study on Skewness of Distribution of Data for Training Classifiers

We conduct the ablation study on the distribution of data for training the classifiers on SmallNORB [32] dataset. As results shown in Tab. 12, higher skewness (*i.e.*, S) (introduced in Sec. A.8) leads to better results in $\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{b}\rangle\rvert$, indicating that the higher skewness makes it easier for discovering the biased attribute. The results show that our method can beat baseline methods in both metrics under all skewness settings.

D. Ablation Study on Differnt Random Initializations of Biased Attribute Hyperplane

To investigate whether our method is sensitive to different random initialization of biased attribute hyperplanes, we conduct experiments with three different random seeds on face images with the same setting introduced in Sec. 5.2. The mean and standard deviation of results over different random seeds are reported in Tab. 9, which shows that our method is robust to different random initializations of biased attribute hyperplane.

E. Detailed Experimental Results

Quantitative Results in Fig. 3 We show the quantitative results of the Fig. 3 in Tab. 13. The results prove that our method can more accurately predict the biased attribute and keep a better disentanglement w.r.t. the target attribute, which is also reflected in Fig. 3.

Detailed Results on Face Image Datasets We show the detailed results of experiments on face image datasets in Tab. 14. Our method achieves better results in all experimental settings than the Hessian Penalty method. Especially, our method can achieve much better results when using the StyleGAN trained on CelebA-HQ to discover the biased attribute in the classifier trained on FFHQ (see 6th row in Tab. 14).

distribution	method	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{b}\rangle\rvert \uparrow$	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert \downarrow$	$\Delta\cos \uparrow$	%leading $\uparrow$	TV
balanced	VAE-based	0.21±0.21	0.16±0.13	0.05±0.20	16.67%	0.15±0.07
	Hessian Penalty	**0.24±0.16**	0.26±0.16	-0.02±0.24	31.67%	0.14±0.05
	Ours	0.23±0.18	**0.10±0.11**	**0.12±0.21**	**51.67%**	0.13±0.05
skewed	VAE-based	0.17±0.18	0.19±0.16	-0.01±0.22	28.33%	0.10±0.04
	Hessian Penalty	0.24±0.17	0.29±0.18	-0.04±0.23	11.67%	0.09±0.04
	Ours	**0.24±0.16**	**0.13±0.11**	**0.11±0.17**	**60.00%**	0.09±0.04

Table 10: Ablation study of data distribution of training data for generative models on SmallNORB [32] dataset. "Balanced" means that the distribution of generative model's training data is balanced and "skew" denotes that the distribution of generative model's training data is skewed (*same* skewness with the training data of the classifier). For the generative model whose training set is skewed, the TV results of the ground-truth biased attribute and target attribute are 0.02±0.02, 0.03±0.02, respectively.

	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{b}\rangle\rvert \uparrow$	$\lvert\cos\langle\hat{\mathbf{w}}_\mathsf{b}, \mathbf{w}_\mathsf{t}\rangle\rvert \downarrow$	$\Delta\cos \uparrow$	TV
$K = \varnothing$	**0.26±0.17**	0.12±0.12	**0.14±0.21**	0.13±0.04
$K \neq \varnothing$	0.23±0.18	**0.10±0.11**	0.11±0.22	0.13±0.05

Table 11: Ablation study on the known attributes in the *orthogonalization penalty* on SmallNORB [32] dataset. $K = \varnothing$ means that the set of known attributes K is not used in *orthogonalization penalty* and only the target attribute is used. $K \neq \varnothing$ denotes that all of known attributes and the target attribute are used in *orthogonalization penalty*.

We suspect that the biased attribute and the target attribute are less correlated in CelebA-HQ than FFHQ, making it easier for discovering the biased attribute in the classifier trained on FFHQ dataset. We also include a discussion on this in Sec. H.4 and Sec. H.6.

F. Qualitative Results

Here we show more qualitative results on face images and images from other domains.

F.1. Qualitative Comparisons on Face Images

We show more qualitative comparisons on face images in Fig. 7. Compared with Hessian Penalty [44], our method can accurately discover the biased attribute *pose* while keeping the disentanglement w.r.t. the target attribute *gender*. In contrast, the Hessian Penalty [44] method predicts target attribute *gender* in Fig. 7 (a) and Fig. 7 (b), which are trivial solutions.

F.2. Discovering Other Biased Attributes on Face Images

By setting all considered attributes as known attributes in the *orthogonalization penalty*, our method can discover biased attributes other than the known attributes. We additionally show more results in Fig. 8. Our method can discover *lighting* and *bald or hair length* biased attributes, which are not the known attributes. We found that the male images have variations in terms of the *bald* attribute, and female images have variations in terms of the *hair length* attribute based on the very same predicted biased attribute hyperplane. Since *bald* and *hair length* are all closely related, we merge them together and regard it as *bald or hair length* attribute. We also admit that our method cannot achieve perfect disentanglement with other attributes such as *beard* in the first row of Fig. 8 (a). We will discuss this problem in Sec. H.6.

F.3. Images from Other Domains

We additionally show more qualitative results on the discovered biased attribute for classifiers on images from other domains in Fig. 11. Our method can successfully discover unnoticeable biased attributes such as *is Eiffel Tower*, *layout*, *number of beds*, *buildings in the background*, *shade of fur color* for tower, conference room, bedroom, bridge, and cat classifiers, respectively.

G. User Study

We conduct the user study to verify that our method can find unknown-biased attributes that are difficult for the baseline method on more images comparable to Fig. 5, 6. Ten subjects are asked to name the attribute of traversal images synthesized by $\mathcal{L}_H$ and our method without knowing the images are generated by which method. The user can also say "cannot tell" the attribute from the traversal image (denoted as "cannot tell" in Fig. 9 and Fig. 10) when the users find it hard to interpret the variation of the traversal images. If the user regards multiple attributes in the traversal images, we ask the users to name the most salient one. For a fair

skewness	method	$\lvert\cos\langle\hat{\mathbf{w}}_\mathbf{b}, \mathbf{w}_\mathbf{b}\rangle\rvert \uparrow$	$\lvert\cos\langle\hat{\mathbf{w}}_\mathbf{b}, \mathbf{w}_\mathbf{t}\rangle\rvert \downarrow$	$\Delta\cos \uparrow$	%leading $\uparrow$	TV
$S = 0.99$	VAE-based	0.22±0.23	0.16±0.14	0.06±0.22	10.00%	0.14±0.06
	Hessian Penalty	0.21±0.17	0.23±0.16	-0.01±0.25	23.33%	0.13±0.04
	Ours	**0.25±0.19**	**0.11±0.11**	**0.14±0.22**	**48.33%**	0.12±0.04
$S = 0.95$	VAE-based	0.20±0.22	0.15±0.13	0.05±0.20	13.33%	0.14±0.06
	Hessian Penalty	**0.25±0.17**	0.20±0.16	0.05±0.24	35.00%	0.13±0.04
	Ours	0.23±0.18	**0.10±0.10**	**0.13±0.21**	**51.67%**	0.12±0.04
$S = 0.9$	VAE-based	0.21±0.21	0.16±0.13	0.05±0.02	16.67%	0.15±0.07
	Hessian Penalty	**0.24±0.16**	0.26±0.16	-0.02±0.25	31.67%	0.14±0.05
	Ours	0.23±0.18	**0.10±0.11**	**0.12±0.21**	**51.67%**	0.13±0.05
$S = 0.75$	VAE-based	0.20±0.22	0.15±0.13	0.05±0.21	13.33%	0.15±0.07
	Hessian Penalty	**0.23±0.17**	0.19±0.15	0.03±0.25	40.00%	0.14±0.05
	Ours	**0.23±0.18**	**0.11±0.10**	**0.12±0.21**	**46.67%**	0.13±0.04

Table 12: Ablation study on the skewness of distribution of data for training the classifiers on SmallNORB [32] dataset.

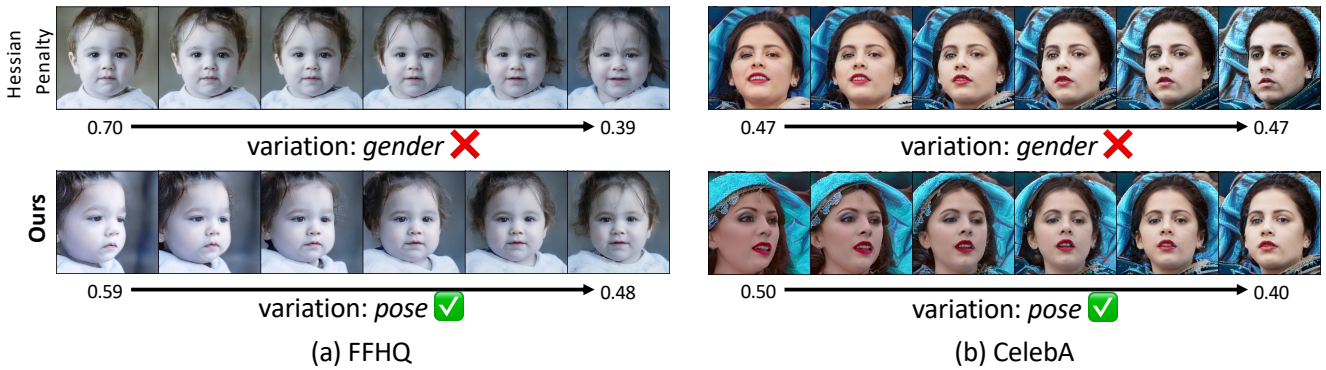

Figure 7: Additional qualitative comparison of the traversal images of predicted biased attributes synthesized by StyleGAN [23] trained on FFHQ [23] dataset. Numbers below the image is the predicted probability of "male" from the *gender* classifier. The training datasets of the target attribute classifier of (a) and (b) are FFHQ and CelebA, respectively. The Hessian Penalty method predicts *gender* as the biased attribute, which is a trivial solution for a *gender* classifier. In (b), although the traversal images from Hessian Penalty also vary in terms of the *skin tone* attribute, it has very little variation in terms of the classifier's prediction (*i.e.*, 0.47 to 0.47). In comparison, our method can correctly predict the pseudo-ground-truth biased attribute *pose* on two datasets.

	method	$\lvert\cos\langle\hat{\mathbf{w}}_\mathbf{b}, \mathbf{w}_\mathbf{b}\rangle\rvert \uparrow$	$\lvert\cos\langle\hat{\mathbf{w}}_\mathbf{b}, \mathbf{w}_\mathbf{t}\rangle\rvert \downarrow$	$\Delta\cos \uparrow$	TV
SmallNORB	VAE-based	0.08	0.19	-0.12	0.16
	$\mathcal{L}_H$	**0.14**	0.33	-0.19	0.18
	Ours	0.13	**0.07**	**0.06**	0.13
dSprites	VAE-based	**0.08**	0.17	-0.10	0.05
	$\mathcal{L}_H$	**0.08**	0.53	-0.45	0.05
	Ours	0.05	**0.04**	**0.01**	0.01

Table 13: Quantitative results of qualitative results shown in Fig. 3.

comparison, we use the *same* sampled latent vector $\mathbf{z}$ for two methods to synthesize traversal images (40 face traversal images, 40 other-domain traversal images). To further let the Hessian Penalty method find other biased attributes, we remove $|K|$ (cardinality of set K) predicted hyperplanes which are top-$|K|$ similar (*i.e.*, high absolute value of cosine similarity) with the known-biased attributes K, which is a similar procedure as we introduce how to adapt the baseline method in Sec. A.6. After collecting the user study results, we compute the percentage of each attribute named by the user. For example, "68%" of the *bald / hair length* attribute of our method in Fig. 9 (a) means that among all the named attributes on the traversal images generated in the experiment setting (a), 68% of them are *bald / hair length* attributes. All other attributes that are rarely named by the users are merged into "others" in Fig. 9 and Fig. 10.

Discovered Other Biased Attributes on Face Images The results of the user study on finding other biased at-

classifier	StyleGAN	method	$\lvert\cos\langle\hat{\mathbf{w}}_b, \mathbf{w}_b\rangle\rvert \uparrow$	$\lvert\cos\langle\hat{\mathbf{w}}_b, \mathbf{w}_t\rangle\rvert \downarrow$	$\Delta\cos \uparrow$	TV
CelebA	CelebA-HQ	$\mathcal{L}_H$	0.0007	0.02	-0.02	0.007
		Ours	**0.07**	**0.001**	**0.07**	**0.11**
	FFHQ	$\mathcal{L}_H$	0.04	0.02	0.02	0.03
		Ours	**0.05**	**0.003**	**0.05**	**0.12**
FFHQ	CelebA-HQ	$\mathcal{L}_H$	0.03	0.007	0.03	0.05
		Ours	**0.28**	$\mathbf{10^{-5}}$	**0.28**	**0.11**
	FFHQ	$\mathcal{L}_H$	**0.06**	0.02	0.03	0.05
		Ours	**0.06**	**0.004**	**0.06**	**0.11**

Table 14: Detailed results of each experiment setting on face images. The first two columns denote the training datasets of the target attribute classifier and StyleGAN, respectively. Our method beats the baseline method under every experiment setting.

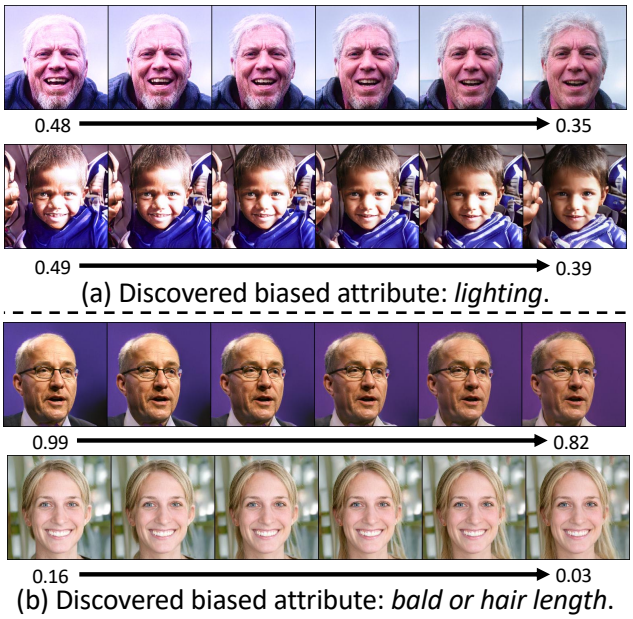

Figure 8: Additional qualitative results on discovered biased attributes by setting the set of known attributes K to all considered attributes generated by StyleGAN [23] trained on FFHQ [23] dataset. The target attributes of (a) and (b) is *gender*. The target attribute classifiers in (a) and (b) are trained on CelebA and FFHQ, respectively. Our method can successfully discover *lighting* and *bald or hair length* biased attributes.

tributes on face image datasets are shown in Fig. 9. For the Hessian Penalty method, by looking at the largest percentage among all attributes, most users agree that it still predicts the known biased attribute *pose* in experiment settings (a) and (d), and the users cannot tell the attribute in experiment setting (b). Although the Hessian Penalty can predict one unknown biased attribute *lighting* in experiment setting (c), its percentage (34%) is still lower than the percentage of *skin tone* attribute (36%) predicted by our method. In the other three experiment settings (a), (b), and (d), by looking at the largest percentage among all attributes, most users agree that our method predicts *bald / hair length* and *lighting* biased attributes. In conclusion, our method can find the biased attributes that are difficult for the baseline method.

Discovered Biased Attributes on Images from Other Domains We conduct the user study on four categories of images: cat, tower, conference room, and bedroom. The results of the user study on discovered biased attributes on images from other domains are shown in Fig. 10. For the Hessian Penalty method, the named attributes are either uniformly distributed (see Hessian Penalty results in Fig. 10 (a) (b)), or the user cannot tell the attributes from the traversal images (see Hessian Penalty results in Fig. 10 (b) (d)). Hence, it is hard to tell the biased attribute from the traversal images synthesized by the Hessian Penalty's biased attribute prediction. In contrast, by looking at the largest percentage numbers, most users agree that our method can find *shade of fur color*, *is Eiffel Tower*, *layout*, and *number of beds* biased attributes, meaning that users can easily tell the biased attribute from the traversal images generated by our method. In conclusion, our method can better discover the biased attributes in diverse domains of images, which are hard to be found by the baseline method.

H. Discussion

H.1. Why $\lvert\cos\langle\hat{\mathbf{w}}_b, \mathbf{w}_b\rangle\rvert$ results on face image datasets are smaller than the ones on disentanglement datasets?

One may question why the $\lvert\cos\langle\hat{\mathbf{w}}_b, \mathbf{w}_b\rangle\rvert$ results on the faces image datasets (Tab. 3) are smaller than the results on disentanglement datasets (Tab. 1). We suspect two reasons for it. First, the numbers of latent space dimensions used in the two experiments are different (512 vs. 10). It would be more difficult for face image experiments to optimize in latent space with larger dimensions (512). Second, discover-

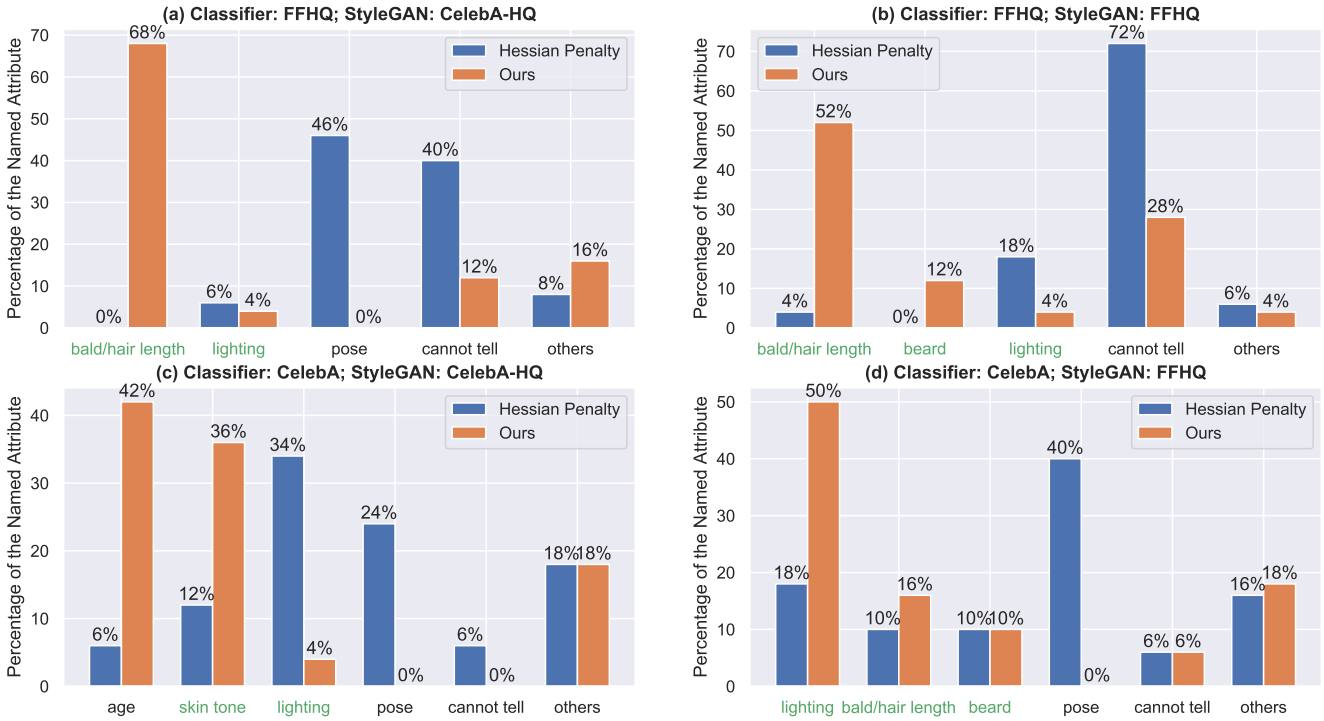

Figure 9: User study on face images. Four bar charts correspond to four experiment settings on face images. The title of each chart denotes the experiment setting. For example, "Classifier: FFHQ; StyleGAN: CelebA-HQ" means that the target attribute classifier is trained on FFHQ, and the StyleGAN is trained on CelebA-HQ. In each bar chart, the x-axis is the attribute named by the users. The y-axis is the percentage of the attribute out of all named attributes by users. A higher percentage means that users agree more on that attribute. The attributes in green are out of the known-biased attribute set K. Attributes in green with higher percentages and the attributes in black with lower percentages mean that the method can better find the unknown attributes. In experiment settings (a), (b), and (d), all users agree that our method find the biased attribute *bald/hair length*, whereas the Hessian Penalty method can only find known-biased attribute *pose* (in (a), (d)), or users cannot tell the attribute (see "cannot tell" in (b)). In the experiment setting (c), although the top-1 named attribute is a known attribute *age*, the named attribute with the second largest percentage is *skin tone*, whose percentage is still larger than the percentage of the top-1 attribute *lighting* from the Hessian Penalty method. We suspect the reason is that our method does not achieve perfect disentanglement between *age* and *skin tone*. In conclusion, our method can better find other unknown biased attributes than the Hessian Penalty method in all experiment settings on face images.

ing the biased attribute on face image datasets is harder than identifying biases on disentanglement datasets because the former datasets are in-the-wild datasets, whereas the latter synthetic datasets only contain finite sets of attributes.

H.2. Why use $\Delta \cos$ as the major evaluation metric?

One may ask that why we use $\Delta \cos$ as the major evaluation metric. The reason is that our ultimate goal is to let the human interpret the biased attribute from the traversal images (see Fig. 1 (b)). Suppose that the traversal images contain the variations of two attributes: the biased attribute and the target attribute. In this case, although the classifier has large prediction variations among the traversal images, humans still cannot decide which attribute (biased attribute or target attribute) is the real cause for the prediction variations. Therefore, it would be better for humans to make a causal conclusion if the traversal images only contain the variation of the biased attribute and do not contain the variation of the target attribute.

H.3. "Small" Variation

One may regard the variation of predicted probabilities that do not switch the 0.5-threshold (*e.g.*, 0.48 to 0.35 in Fig. 8) are "small" for a binary classifier. However, we believe such "small" variation is still value for the *unknown bias discovery task* for the following reasons:

First, such variations still break the fairness criterion. Second, we do not think switching 0.5 threshold is necessary because 1) a larger threshold may be required for some safety-critical scenarios; 2) when a classifier needs to give the ranking of different input images (such as image retrieval task) based on the predicted probability, the threshold is not

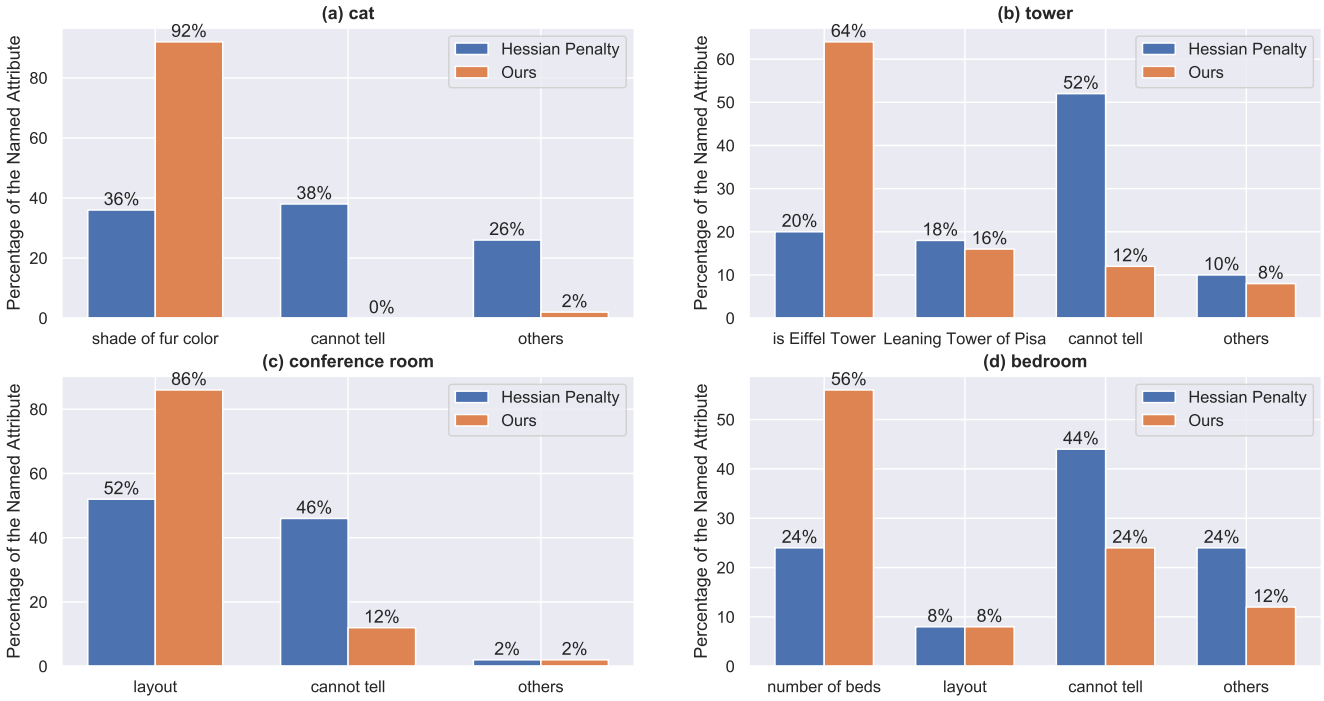

Figure 10: User study on other domains of images. Four bar charts correspond to four image domains (see titles of each chart). In each bar chart, the x-axis is the attribute named by the users. The y-axis is the percentage of the attribute out of all named attributes by users. A higher percentage means that users agree more on that attribute. The user study results show that the biased attributes predicted by the Hessian Penalty method are either uniformly distributed or cannot be told by the users (*e.g.*, "cannot tell" in (b) and (d)). In contrast, the biased attributes from our method are more concentrated on the one attribute (*e.g.*, *shade of fur color*). In conclusion, our method can better find the unknown biased attributes in diverse domains of images than the Hessian Penalty method.

needed, and the biased ranking may still lead to unfairness issue; 3) in a multi-class classification setting (*i.e.*, experiments on other domains of images), the threshold is not 0.5 and such "small" variation can alter the classifier's top-1 prediction. Third, such "small" variations can still provide insights to dataset curators to mitigate such biases, which could be learned by other networks.

H.4. Discussions on Orthogonalization Penalty

One may worry that the *orthogonalization penalty* may prevent the method from finding the biased attribute highly correlated with the target attribute, based on the assumption that the high correlation in the training set of the classifier leads to a high absolute value of cosine similarity between the biased attribute normal vector and the target attribute normal vector in the latent space of the generative model. For example, if the *hair length* biased attribute is highly correlated with the target attribute *gender* in the training set of the classifier (*e.g.*, "long hair female" and "short hair male" are overrepresented in the dataset), one may think that $\mathcal{L}_\perp$ may prevent our method from finding the *hair length* attribute hyperplane due to its high absolute cosine similarity with *gender* attribute hyperplane. We list our arguments and possible solutions to this question by the following points:

First, $\mathcal{L}_\perp$ is a soft penalty instead of a strict constraint, which still allows the correlation between the predicted biased attribute and the target one. In fact, we tried strict constraint but it is hard to optimize, which fails to predict biased attribute, resulting in low $|\cos\langle\hat{\mathbf{w}}_\mathrm{b}, \mathbf{w}_\mathrm{b}\rangle|$ values. That is also reflected by the ablation study on the soft orthogonalization penalty in Tab. 2, where even adding the soft orthogonalization penalty has already decreased the $|\cos\langle\hat{\mathbf{w}}_\mathrm{b}, \mathbf{w}_\mathrm{b}\rangle|$ results. Hence, the proposed soft *orthogonalization penalty* is still better than the strict orthogonalization constraint for solving the aforementioned issue.

Second, the ablation study on the skewness of the generative model's training data (Sec. C.1) has already proved that our method can still accurately predict the biased attribute even if the generative model's training data has the *same* skewness of classifier's training data (see results in Tab. 10). Note that the performance of two baseline methods drops when using the generative model trained on skewed data, whereas our method still maintains high performance.

Third, we observe that using a more disentangled latent space (*e.g.*, $\mathcal{W}$-space of StyleGAN) of the generative model can also mitigate this issue. We tried using the input noise

space ($\mathcal{Z}$-space) of the StyleGAN on face images, which has much lower results than using the $\mathcal{W}$-space of StyleGAN. For example, our method finds the *bald / hair length* biased attribute in the $\mathcal{W}$-space of StyleGAN, and we do not find such attribute in the $\mathcal{Z}$-space. As explained in Sec. 4 of the StyleGAN [23] paper, compared with the input noise space ($\mathcal{Z}$-space), $\mathcal{W}$-space is a more disentangled latent space, where hyperplanes are less correlated. Hence, using a more disentangled latent space can also address the issue mentioned above.

Fourth, the assumption that attributes are highly correlated in the latent space may not hold when the training datasets of the generative model and the classifier are different. In other words, the biased attribute hyperplane may not be highly correlated with the target attribute hyperplane when the generative's model training data has a weaker skewness between the biased attribute and the target attribute.

Finally, another solution to the issue is using a generative model that trained on only one value of the target attribute. For example, as we did in the experiment on other domains of images (Sec. 5.3), we only use $\mathcal{L}_V$ and do not use $\mathcal{L}_\perp$ because the target attribute value will not change among the synthesized images (*i.e.*, a *cat* generator will only synthesize *cat* images, and will never synthesize *dog* images.). Although more generators need to be trained for each target attribute value, this could be another solution to the aforementioned problem as no target attribute exists in the generator's latent space, not to mention the correlation between the biased attribute and the target attribute.

H.5. Related Methods and Areas

The proposed *unknown biased attribute discovery task* can benefit many related methods and areas:

First, many supervised algorithmic de-biasing methods [54] require the well-defined biased attribute (or protected attribute) and corresponding labels to mitigate biases. The discovered unknown biases can serve as the definition of the biased attribute, and the corresponding labels can be further collected by humans.

Second, our method can also benefit dataset curation and audition process. The biases in the image classifiers may originate from the training data of the classifier. Therefore, users can balance the distribution of the dataset based on the discovered unknown biases to mitigate the dataset bias [49].

Third, our method also provides a unique perspective for the area of disentanglement methods. As discussed in [36], unsupervised disentanglement is theoretically impossible, and future works should explicitly present the inductive biases and weak supervision used in the framework. From a disentanglement perspective, our method empirically proves that the biases in a down-stream classifier can serve as a weak prior for finding the biased attribute in the latent space of the generative model.

Finally, our work can also give a new research direction to the adversarial attack methods [17, 37]. While most adversarial attack methods focus on adding uninterpretable pixel perturbation to the image, our method uses the traversal images to find the interpretable vulnerability of the deep neural networks.

H.6. Limitations and Future Direction

We honestly list some limitations of our method.

First, we do not achieve perfect disentanglement on in-the-wild datasets. For example, in Fig. 8, the discovered biased attribute *lighting* is entangled with the attribute *beard*. A possible solution is to obtain more hyperplanes of known attributes (*e.g.*, the *beard* attribute) for *orthogonalization penalty*.

Second, we admit that the biased attributes' searching space is decided by the coverage of attributes of the generative model's training data. However, we believe that this will not be a serious problem as long as the users can access either the same training data used by the classifier or even larger and diverse unlabeled datasets to train the generative model.

Lastly, our method only focuses on detecting one biased attribute at a time, while the target attribute could be affected by multiple biased attributes in real-world settings. A possible solution is extending our method by jointly optimizing multiple orthogonalized biased attribute hyperplanes.

The future directions of *unknown biased attribute discovery task* could be tackling our method's aforementioned limitations. In this work, we only study the counterfactual fairness criterion [12, 13, 21, 31]. Future works can explore more fairness criteria to define the biased attribute. One interesting direction is to discover unknown biases from object detectors or semantic segmentation networks rather than classifiers.

Furthermore, we base our method on recent advances in generative models that can synthesize photo-realistic images of faces, objects, and simple scenes. However, to the best of our knowledge, no generative models can synthesize photo-realistic images of complex scenes containing various objects and stuff (*e.g.*, images from MS-COCO dataset [33]). Developing methods for discovering unknown biases for models learned from complex scene images is a challenging but valuable research direction for *unknown biased attribute discovery task*.

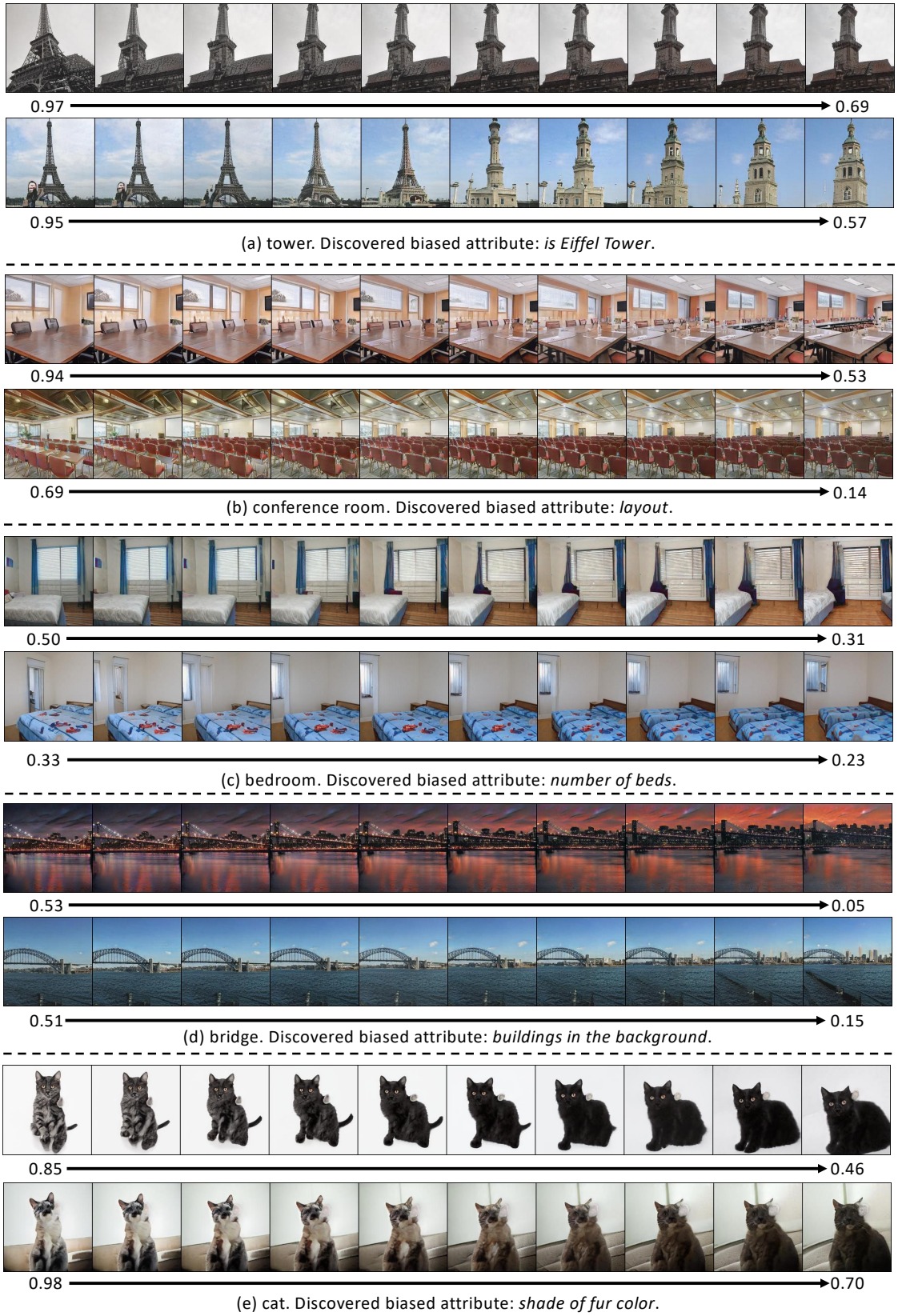

Figure 11: Additional qualitative results on the discovered biased attribute of classifiers for classifying *tower*, *conference room*, *bedroom*, *bridge*, and *cat* images. Numbers below images are predicted probability by the classifier.