PRISM: Reducing Spurious Implicit Biases in Vision-Language Models with LLM-Guided Embedding Projection

Supplementary Material

https://github.com/MahdiyarMM/PRISM

Mahdiyar Molahasani*, Azadeh Motamedi*, Michael Greenspan, Il-Min Kim, Ali Etemad
Queen's University, Canada
{m.molahasani, 19am43, michael.greenspan, ilmin.kim, ali.etemad}@queensu.ca

A. Notation

Here, we summarize the key notations used throughout the paper:

- x: An input image.
- $\{t_y\}_{y=1}^K$: A set of K text prompts corresponding to class labels.
- $\phi_\text{I}(\cdot)$: CLIP's image encoder that maps an image to a d-dimensional embedding.
- $\phi_\text{T}(\cdot)$: CLIP's text encoder that maps text to a d-dimensional embedding.
- d: Dimensionality of the shared embedding space.
- $\langle \cdot, \cdot \rangle$: Inner product in the embedding space.
- $\hat{y}$: Predicted class label computed as

$$\hat{y} = \arg\max_k \langle \phi_\text{I}(x), \phi_\text{T}(t_k) \rangle.$$

- $h_\text{core}(x)$: The core features of image x that are essential for correct classification.
- $h_\text{spu}(x)$: The spurious features of image x that may capture irrelevant cues.
- f: A function that combines core and spurious features, i.e.,

$$\phi_\text{I}(x) = f\big(h_\text{core}(x), h_\text{spu}(x)\big).$$

- a: A spurious attribute (e.g., background type).
- y: The true class label.
- $g = (a, y)$: The group identity, with

$$g \in \mathcal{G} = A \times Y,$$

  where A is the set of spurious attributes and $Y = \{1, \ldots, K\}$ is the set of class labels.
- $\mathcal{T}_{a,y}$: Scene descriptions corresponding to (a, y)
- $\ell(\cdot, \cdot)$: The classification loss function.
- $\theta \in \Theta$: The parameters of the model.
- $\mathcal{F}$: A pre-trained large language model (LLM) used to detect spurious attributes.
- $\mathcal{A}$: The embedding of spurious attributes, computed as

$$\mathcal{A} = \phi_\text{T}(\mathcal{F}(\{t_y\}_{k=1}^K)).$$

- P: A projection operator applied to the embedding space to reduce the impact of spurious cues.
- $\mathcal{L}_\text{LD}$: The Latent space Debiasing Loss used to learn P_learn, which balances intra-class invariance and inter-class separation.
- m: The margin hyperparameter in $\mathcal{L}_\text{LD}$.

*These authors contributed equally.

Model	Waterbirds					CelebA				
	WG ↑	Acc ↑	ΔWG ↑	ΔAcc ↑	Gap ↓	WG ↑	Acc ↑	ΔWG ↑	ΔAcc ↑	Gap ↓
Baseline										
Zero-shot	38.3%	92.8%	–	–	54.5%	75.8%	82.4%	–	–	6.6%
Methods using images for debiasing										
DFR (Sub) [16]	66.1%	92.9%	27.8%	0.1%	26.8%	80.9%	91.7%	5.1%	9.3%	10.8%
DFR [16]	54.2%	90.3%	15.9%	–2.5%	36.1%	89.9%	91.3%	14.1%	8.9%	**1.4%**
FairerCLIP [7]	75.4%	84.3%	37.1%	–8.5%	8.9%	81.5%	85.0%	5.7%	2.6%	3.5%
Data-free methods										
VisualDistiller [6]	44.2%	**93.1%**	5.9%	**0.3%**	48.9%	–	–	–	–	–
Orth-Proj [5]	48.1%	83.6%	9.8%	–9.2%	35.5%	61.4%	86.4%	–14.4%	4.0%	25.0%
Orth-Cali [5]	**74.0%**	78.7%	**35.7%**	–14.1%	**4.7%**	82.2%	84.4%	6.4%	2.0%	2.2%
PRISM-Mini (ours)	62.0%	91.7%	23.7%	–1.1%	29.7%	70.3%	79.2%	–5.5%	–3.2%	8.9%
PRISM (ours)	70.2%	91.9%	31.9%	–0.9%	21.7%	**83.3%**	**89.0%**	**7.5%**	**6.6%**	5.7%

Table A1. The performance of PRISM compared with baselines on CLIP-RN50 on Waterbirds and CelebA datasets. Top performers in each column are **bolded**, second bests are underlined.

Model	Attribute Labels (Need)	Task Info at Inf. (Need)	LLM at Inf. (Need)	Inference Speed (Relative)	Explicit Debiasing (Objective)	Worst Group AUROC (CelebA %)
Comparison of PRISM variants and BendVLM						
Zero-shot	No	No	No	Fast	None	72.8
BendVLM [10]	Yes	Yes	Yes	Slow	Yes (inference constraint)	77.2
PRISM-mini (ours)	No	No	No	Fast	Yes (simple projection)	82.6
PRISM (ours)	No	No	No	Fast	Yes (LD loss)	**84.0**

Table A2. Comparison of Zero-shot CLIP, PRISM-mini, PRISM, and BendVLM in terms of practical requirements, design differences, and performance on the CelebA dataset.

B. Prompts

To find the spurious correlations, we use the following prompt: "`Provide a list of potential bias attibutes associated with the following zero-shot classification using CLIP:` <$\{t_k\}$>".
Then, we use generate a set of controlled scene descriptions using the following: "`based on these classes` <$\{t_k\}$> `and these spurious attributes` <A>, `create` <n> `scene descriptions where the class and attributes are marked as "*class*" and "*attribute*" and can be later replaced with a class or attribute from its corresponding list. Generate a list for each class and a list for each attributes separately. Use this example as a guide: <"Example of Panda and Camel with spurious connection with Desert and Bamboo">`".

C. Additional experiments

Table A1 presents the results for the CLIP-RN50 model, including only the baselines that report for CLIP-RN50. On the CelebA dataset, PRISM attains the best performance in both WG and Acc metrics. However, it does not outperform the state-of-the-art methods on the Waterbirds dataset. This is partly because Waterbirds is smaller in scale and has a stronger background–label correlation, making a less powerful backbone like ResNet-50 more susceptible to entangling these features. In this case, finding spurious features without any predefined prompts, is more difficult for PRISM.

To further highlight the effectivenss of our method, We also compare PRISM with BendVLM. While both PRISM and BendVLM aim to mitigate biases in vision-language models, they differ fundamentally in several aspects that make PRISM more practical, efficient, and effective. Specifically: (i) PRISM does not require any labeled image data for training, unlike BendVLM which relies on annotated images. (ii) PRISM needs no task-specific information (e.g., class labels or number of classes) at inference time. (iii) PRISM is faster at inference by applying a single fixed projection learned during training, whereas BendVLM recomputes a projection per test input. (iv) PRISM requires LLM access only once during training, while BendVLM needs repeated LLM queries at inference, resulting in higher cost and latency. (v) PRISM explicitly introduces an LD loss that enforces intra-class invariance and inter-class separability. (vi) As summarized in Table X, PRISM achieves better performance across key benchmarks.