PromptStyler: Prompt-driven Style Generation for Source-free Domain Generalization
— Supplementary Material —

Junhyeong Cho[1] Gilhyun Nam[1] Sungyeon Kim[2] Hunmin Yang[1,3] Suha Kwak[2]
[1]ADD [2]POSTECH [3]KAIST
https://PromptStyler.github.io

In this supplementary material, we provide more method details (Section A), analyses on Terra Incognita (Section B), evaluation results (Section C) and discussion (Section D).

A. Method Details

This section provides more details of the chosen vision-language model (Section A.1) and design choices for learning style word vectors (Section A.2).

A.1. Large-scale vision-language model

We choose CLIP [13] as our pre-trained vision-language model which is a large-scale model trained with 400 million image-text pairs. Note that the proposed method is broadly applicable to the CLIP-like vision-language models [7, 16] which also construct hyperspherical joint vision-language spaces using contrastive learning methods. Given a batch of image-text pairs, such models jointly train an image encoder and a text encoder considering similarity scores obtained from image-text pairings.

Joint vision-language training. Suppose there is a batch of M image-text pairs. Among all possible $M \times M$ pairings, the matched M pairs are the positive pairs and the other $M^2 - M$ pairs are the negative pairs. CLIP [13] is trained to maximize cosine similarities of image and text features from the positive M pairs while minimizing the similarities of such features from the negative $M^2 - M$ pairs.

Image encoder. CLIP [13] utilizes ResNet [6] or ViT [4] as its image encoder. Given an input image, the image encoder extracts its image feature. After that, the image feature is mapped to a hyperspherical joint vision-language space by ℓ_2 normalization.

Text encoder. CLIP [13] utilizes Transformer [14] as its text encoder. Given an input text prompt, it is converted to word vectors via a tokenization process and a word lookup procedure. Using these word vectors, the text encoder generates a text feature which is then mapped to a hyperspherical joint vision-language space by ℓ_2 normalization.

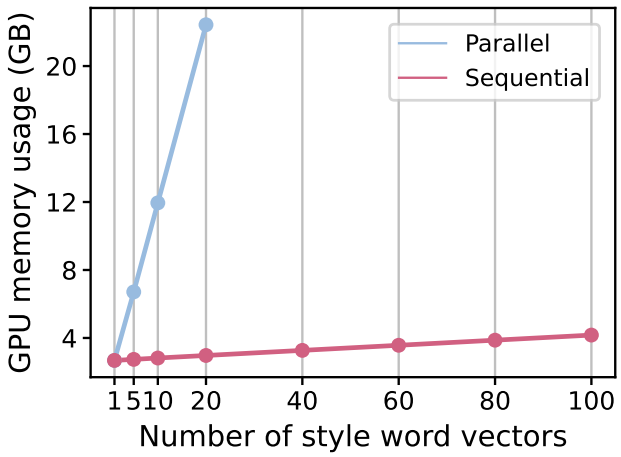

Figure A1: GPU memory usage when learning K style word vectors for the target task OfficeHome [15] (65 classes) with respect to the design choices, *Sequential* or *Parallel*.

Zero-shot inference. At inference time, zero-shot CLIP [13] synthesizes classifier weights via the text encoder using N class names pre-defined in the target task. Given an input image, the image encoder extracts its image feature and the text encoder produces N text features using the N class names. Then, it computes cosine similarity scores between the image feature and text features, and selects the class name which results in the highest similarity score as its classification output.

A.2. Empirical justification of our design choice

As described in Section 3.1 of the main paper, there are two possible design choices for learning K style word vectors: (1) learning each style word vector $\mathbf{s}_i$ in a sequential manner, or (2) learning all style word vectors $\{\mathbf{s}_i\}_{i=1}^{K}$ in a parallel manner. We choose the former mainly due to its much less memory overhead. As shown in Figure A1, we could sequentially learn $\sim$100 style word vectors with $\sim$4.2 GB memory usage. However, it is not possible to learn more than 21 style word vectors in a parallel manner using a single

(a) Dog (b) Cat (c) Squirrel

Figure B1: Several examples from the Terra Incognita [1] dataset. We visualize class entities using red bounding boxes, since they are not easily recognizable due to their small sizes and complex background scenes.

Method	Configuration		Accuracy (%)				
	Source Domain	Domain Description	Location100	Location38	Location43	Location46	Avg.
ResNet-50 [6] with pre-trained weights on ImageNet [2]							
SelfReg [8]	✓	–	48.8±0.9	41.3±1.8	57.3±0.7	**40.6**±0.9	47.0
GVRT [11]	✓	–	**53.9**±1.3	**41.8**±1.2	**58.2**±0.9	38.0±0.6	**48.0**
ResNet-50 [6] with pre-trained weights from CLIP [13]							
ZS-CLIP (C) [13]	–	–	8.4±0.0	13.7±0.0	32.5±0.0	23.3±0.0	19.5
ZS-CLIP (PC) [13]	–	✓	9.9±0.0	28.3±0.0	32.9±0.0	24.0±0.0	23.8
PromptStyler	–	–	**13.8**±1.7	**39.8**±1.3	**38.0**±0.4	**30.3**±0.3	**30.5**

Table B1: Top-1 classification accuracy on the Terra Incognita [1] dataset. Compared with existing domain generalization methods which utilize source domain data, zero-shot methods using CLIP [13] show unsatisfactory results on this dataset.

RTX 3090 GPU (24 GB Memory) due to its large memory overhead. In detail, learning 20 and 21 style word vectors takes 22.4 GB and 23.5 GB, respectively. The large memory overhead caused by the parallel learning design substantially limits the number of learnable style word vectors.

To be specific, PromptStyler with the parallel learning design needs to generate K style features, KN style-content features, and N content features for learning K style word vectors at the same time; these features are used to compute the style diversity loss $\mathcal{L}_{\text{style}}$ and the content consistency loss $\mathcal{L}_{\text{content}}$ for learning all the style word vectors in a parallel manner. Note that the large memory overhead is mainly caused by the KN style-content features. Suppose we want to learn 80 style word vectors for the target task OfficeHome [15] (65 classes). Then, we need to synthesize $5200(= 80 \times 65)$ style-content features. Even worse, we need to generate $27600(= 80 \times 345)$ style-content features for the target task DomainNet [12] (345 classes). On the other hand, PromptStyler with the sequential learning design only requires i style features, N style-content features, and N content features for learning i-th style word vector, where $1 \leq i \leq K$. For scalability, we chose the sequential learning design since it could handle a lot of learnable style word vectors and numerous classes in the target task.

B. Analyses on Terra Incognita

As described in Section 5 of the main paper, the quality of the latent space constructed by a large-scale pre-trained model significantly affects the effectiveness of PromptStyler. To be specific, the proposed method depends on the quality of the joint vision-language space constructed by CLIP [13]. Although our method achieves state-of-the-art results on PACS [9], VLCS [5], OfficeHome [15], and DomainNet [12], its performance on Terra Incognita [1] is not satisfactory. This section provides more analyses on the dataset.

Table B1 shows that PromptStyler outperforms zero-shot CLIP [13] for all domains in the Terra Incognita dataset [1]. However, its accuracy on this dataset is lower compared with existing domain generalization methods [8, 11] which utilize several images from the dataset as their source domain data. This unsatisfactory result might be due to the low accuracy of CLIP on the dataset. We suspect that images in the Terra Incognita dataset (Fig. B1) might be significantly different from the domains that CLIP has observed. The distribution shifts between CLIP training dataset and the Terra Incognita dataset might be extreme, and thus such distribution shifts could not be entirely covered by our method which exploits CLIP latent space. We hope this issue could be alleviated with the development of large-scale models.

Method	Configuration		Accuracy (%)				
	Source Domain	Domain Description	Art Painting	Cartoon	Photo	Sketch	Avg.
ResNet-50 [6] with pre-trained weights on ImageNet [2]							
GVRT [11]	✓	–	**87.9**±0.3	78.4±1.0	**98.2**±0.1	75.7±0.4	85.1
SelfReg [8]	✓	–	**87.9**±1.0	**79.4**±1.4	96.8±0.7	**78.3**±1.2	**85.6**
ResNet-50 [6] with pre-trained weights from CLIP [13]							
ZS-CLIP (C) [13]	–	–	88.9±0.0	94.4±0.0	99.3±0.0	79.8±0.0	90.6
ZS-CLIP (PC) [13]	–	✓	90.8±0.0	93.3±0.0	**99.4**±0.0	79.3±0.0	90.7
PromptStyler	–	–	**93.7**±0.1	**94.7**±0.2	**99.4**±0.0	**84.9**±0.1	**93.2**
ViT-B / 16 [4] with pre-trained weights from CLIP [13]							
ZS-CLIP (C) [13]	–	–	96.4±0.0	98.9±0.0	**99.9**±0.0	87.7±0.0	95.7
ZS-CLIP (PC) [13]	–	✓	97.2±0.0	**99.1**±0.0	**99.9**±0.0	88.2±0.0	96.1
PromptStyler	–	–	**97.6**±0.1	**99.1**±0.1	**99.9**±0.0	**92.3**±0.3	**97.2**
ViT-L / 14 [4] with pre-trained weights from CLIP [13]							
ZS-CLIP (C) [13]	–	–	97.2±0.0	99.5±0.0	99.9±0.0	93.8±0.0	97.6
ZS-CLIP (PC) [13]	–	✓	99.0±0.0	**99.7**±0.0	99.9±0.0	**95.5**±0.0	98.5
PromptStyler	–	–	**99.1**±0.0	**99.7**±0.0	**100.0**±0.0	**95.5**±0.1	**98.6**

Table C1: Comparison with state-of-the-art domain generalization methods in terms of per-domain top-1 classification accuracy on PACS [9]. We repeat each experiment using three different seeds, and report average accuracies with standard errors. ZS-CLIP (C) denotes zero-shot CLIP using "[class]" as its text prompt, and ZS-CLIP (PC) indicates zero-shot CLIP using "a photo of a [class]" as its text prompt. Note that PromptStyler does not use any source domain data and domain descriptions.

Method	Configuration		Accuracy (%)				
	Source Domain	Domain Description	Caltech	LabelMe	SUN09	VOC2007	Avg.
ResNet-50 [6] with pre-trained weights on ImageNet [2]							
SelfReg [8]	✓	–	96.7±0.4	**65.2**±1.2	73.1±1.3	76.2±0.7	77.8
GVRT [11]	✓	–	**98.8**±0.1	64.0±0.3	**75.2**±0.5	**77.9**±1.0	**79.0**
ResNet-50 [6] with pre-trained weights from CLIP [13]							
ZS-CLIP (C) [13]	–	–	99.2±0.0	62.4±0.0	69.0±0.0	73.5±0.0	76.0
ZS-CLIP (PC) [13]	–	✓	99.4±0.0	65.0±0.0	71.7±0.0	84.2±0.0	80.1
PromptStyler	–	–	**99.5**±0.0	**71.2**±0.2	**72.0**±0.0	**86.5**±0.3	**82.3**
ViT-B / 16 [4] with pre-trained weights from CLIP [13]							
ZS-CLIP (C) [13]	–	–	99.7±0.0	61.8±0.0	70.1±0.0	73.9±0.0	76.4
ZS-CLIP (PC) [13]	–	✓	**99.9**±0.0	68.9±0.0	**74.8**±0.0	85.9±0.0	82.4
PromptStyler	–	–	**99.9**±0.0	**71.5**±0.3	73.9±0.2	**86.3**±0.1	**82.9**
ViT-L / 14 [4] with pre-trained weights from CLIP [13]							
ZS-CLIP (C) [13]	–	–	**99.9**±0.0	59.3±0.0	71.0±0.0	79.9±0.0	77.5
ZS-CLIP (PC) [13]	–	✓	**99.9**±0.0	70.9±0.0	**72.9**±0.0	86.0±0.0	**82.4**
PromptStyler	–	–	**99.9**±0.0	**71.1**±0.7	71.8±1.0	**86.8**±0.0	**82.4**

Table C2: Comparison with state-of-the-art domain generalization methods in terms of per-domain top-1 classification accuracy on VLCS [5]. We repeat each experiment using three different seeds, and report average accuracies with standard errors. ZS-CLIP (C) denotes zero-shot CLIP using "[class]" as its text prompt, and ZS-CLIP (PC) indicates zero-shot CLIP using "a photo of a [class]" as its text prompt. Note that PromptStyler does not use any source domain data and domain descriptions.

Method	Configuration		Accuracy (%)				
	Source Domain	Domain Description	Art	Clipart	Product	Real World	Avg.
ResNet-50 [6] with pre-trained weights on ImageNet [2]							
SelfReg [8]	✓	–	63.6±1.4	53.1±1.0	76.9±0.4	78.1±0.4	67.9
GVRT [11]	✓	–	**66.3**±0.1	**55.8**±0.4	**78.2**±0.4	**80.4**±0.2	**70.1**
ResNet-50 [6] with pre-trained weights from CLIP [13]							
ZS-CLIP (C) [13]	–	–	69.9±0.0	46.8±0.0	77.7±0.0	79.8±0.0	68.6
ZS-CLIP (PC) [13]	–	✓	71.7±0.0	52.0±0.0	81.6±0.0	82.6±0.0	72.0
PromptStyler	–	–	**73.4**±0.1	**52.4**±0.2	**84.3**±0.1	**84.1**±0.1	**73.6**
ViT-B / 16 [4] with pre-trained weights from CLIP [13]							
ZS-CLIP (C) [13]	–	–	80.7±0.0	64.6±0.0	86.3±0.0	88.0±0.0	79.9
ZS-CLIP (PC) [13]	–	✓	82.7±0.0	67.6±0.0	89.2±0.0	89.7±0.0	82.3
PromptStyler	–	–	**83.8**±0.1	**68.2**±0.0	**91.6**±0.1	**90.7**±0.1	**83.6**
ViT-L / 14 [4] with pre-trained weights from CLIP [13]							
ZS-CLIP (C) [13]	–	–	86.2±0.0	73.3±0.0	92.0±0.0	92.2±0.0	85.9
ZS-CLIP (PC) [13]	–	✓	87.2±0.0	73.8±0.0	93.0±0.0	93.4±0.0	86.9
PromptStyler	–	–	**89.1**±0.1	**77.6**±0.1	**94.8**±0.1	**94.8**±0.0	**89.1**

Table C3: Comparison with state-of-the-art domain generalization methods in terms of per-domain top-1 classification accuracy on OfficeHome [15]. We repeat each experiment using three different seeds, and report average accuracies with standard errors. ZS-CLIP (C) denotes zero-shot CLIP using "[class]" as its text prompt, and ZS-CLIP (PC) indicates zero-shot CLIP using "a photo of a [class]" as its text prompt. Note that PromptStyler does not use any source domain data and domain descriptions.

Method	Configuration		Accuracy (%)						
	Source Domain	Domain Description	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg.
ResNet-50 [6] with pre-trained weights on ImageNet [2]									
SelfReg [8]	✓	–	60.7±0.1	**21.6**±0.1	49.4±0.2	12.7±0.1	60.7±0.1	51.7±0.1	42.8
GVRT [11]	✓	–	**62.4**±0.4	21.0±0.0	**50.5**±0.4	**13.8**±0.3	**64.6**±0.4	**52.4**±0.2	**44.1**
ResNet-50 [6] with pre-trained weights from CLIP [13]									
ZS-CLIP (C) [13]	–	–	53.1±0.0	39.2±0.0	52.7±0.0	**6.3**±0.0	75.2±0.0	47.1±0.0	45.6
ZS-CLIP (PC) [13]	–	✓	53.6±0.0	39.6±0.0	53.4±0.0	5.9±0.0	76.6±0.0	48.0±0.0	46.2
PromptStyler	–	–	**57.9**±0.0	**44.3**±0.0	**57.3**±0.0	6.1±0.1	**79.5**±0.0	**51.7**±0.0	**49.5**
ViT-B / 16 [4] with pre-trained weights from CLIP [13]									
ZS-CLIP (C) [13]	–	–	70.7±0.0	49.1±0.0	66.4±0.0	**14.8**±0.0	82.7±0.0	63.1±0.0	57.8
ZS-CLIP (PC) [13]	–	✓	71.0±0.0	47.7±0.0	66.2±0.0	14.0±0.0	83.7±0.0	63.5±0.0	57.7
PromptStyler	–	–	**73.1**±0.0	**50.9**±0.0	**68.2**±0.1	13.3±0.1	**85.4**±0.0	**65.3**±0.0	**59.4**
ViT-L / 14 [4] with pre-trained weights from CLIP [13]									
ZS-CLIP (C) [13]	–	–	78.2±0.0	53.0±0.0	70.7±0.0	21.6±0.0	86.0±0.0	70.3±0.0	63.3
ZS-CLIP (PC) [13]	–	✓	79.2±0.0	52.4±0.0	71.3±0.0	**22.5**±0.0	86.9±0.0	71.8±0.0	64.0
PromptStyler	–	–	**80.7**±0.0	**55.6**±0.1	**73.8**±0.1	21.7±0.0	**88.2**±0.0	**73.2**±0.0	**65.5**

Table C4: Comparison with state-of-the-art domain generalization methods in terms of per-domain top-1 classification accuracy on DomainNet [12]. We repeat each experiment using three different seeds, and report average accuracies with standard errors. ZS-CLIP (C) denotes zero-shot CLIP using "[class]" as its text prompt, and ZS-CLIP (PC) indicates zero-shot CLIP using "a photo of a [class]" as its text prompt. Note that PromptStyler does not use any source domain data and domain descriptions.

Distribution	Accuracy (%)				
	PACS	VLCS	OfficeHome	DomainNet	Avg.
$\mathcal{U}(0.00,\ 0.20)$	93.1	**82.6**	**73.8**	49.2	**74.7**
$\mathcal{N}(0.00,\ 0.20^2)$	93.0	81.0	73.6	**49.5**	74.3
$\mathcal{N}(0.20,\ 0.02^2)$	93.1	82.5	73.5	49.3	74.6
$\mathcal{N}(0.00,\ 0.02^2)$	**93.2**	82.3	73.6	**49.5**	**74.7**

Table C5: Effects of the distributions used for initializing style word vectors. Uniform or Normal distribution is used.

C. Evaluation Results

Per-domain accuracy. As shown in Table C1–C4, we provide per-domain top-1 classification accuracy on domain generalization benchmarks including PACS [9] (4 domains and 7 classes), VLCS [5] (4 domains and 5 classes), OfficeHome [15] (4 domains and 65 classes) and DomainNet [12] (6 domains and 345 classes); each accuracy is obtained by averaging results from experiments repeated using three different random seeds. Interestingly, compared with zero-shot CLIP [13] which leverages a photo domain description ("a photo of a [class]"), our PromptStyler achieves similar or better results on photo domains, *e.g.*, on the VLCS dataset which consists of 4 photo domains. Note that the description has more domain-specific information and more detailed contexts compared with the naïve prompt ("[class]").

Different distributions for initializing style word vectors. Following prompt learning methods [18, 19], we initialized learnable style word vectors using zero-mean Gaussian distribution with 0.02 standard deviation. To measure the effect of the used distribution for the initialization, we also quantitatively evaluate PromptStyler using different distributions for initializing style word vectors. As shown in Table C5, the proposed method also achieves similar results when initializing style word vectors using different distributions.

D. Discussion

PromptStyler aims to improve model's generalization capability by simulating various distribution shifts in the latent space of a large-scale pre-trained model. To achieve this goal, our method leverages a joint vision-language space where text features could effectively represent their relevant image features. It does not mean that image and text features should be perfectly interchangeable in the joint vision-language space; a recent study has demonstrated the modality gap phenomenon of this joint space [10]. However, thanks to the cross-modal transferability in the joint vision-language space [17], the proposed method could still be effective, *i.e.*, we could consider text features as proxies for image features while training a linear classifier (Fig. 3 of the main paper).

When our method is implemented with CLIP [13] and we adopt ArcFace [3] as our classification loss $\mathcal{L}_{\text{class}}$, there is another interesting interpretation of the proposed method. As described in Section A.1, CLIP text encoder synthesizes classifier weights using class names for zero-shot inference and then it computes cosine similarity scores between the classifier weights and input image features. Similarly, our method computes cosine similarity scores between classifier weights of the trained classifier (Fig. 3 of the main paper) and input image features. From this perspective, the proposed method improves the decision boundary of the synthesized classifier used in zero-shot CLIP by generating diverse style-content features and then training a linear classifier using the style-content features. In other words, the trained classifier could be considered as an improved version of the synthesized classifier used in zero-shot CLIP.

References

[1] Sara Beery, Grant van Horn, and Pietro Perona. Recognition in Terra Incognita. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[5] Chen Fang, Ye Xu, and Daniel N. Rockmore. Unbiased Metric Learning: On the Utilization of Multiple Datasets and Web Images for Softening Bias. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[6] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[7] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[8] Daehee Kim, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. SelfReg: Self-supervised Contrastive Regularization for Domain Generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[9] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, Broader and Artier Domain Generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[10] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the Gap: Understanding the Modal-

ity Gap in Multi-modal Contrastive Representation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[11] Seonwoo Min, Nokyung Park, Siwon Kim, Seunghyun Park, and Jinkyu Kim. Grounding Visual Representations with Texts for Domain Generalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[12] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment Matching for Multi-Source Domain Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[15] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep Hashing Network for Unsupervised Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. Vision-Language Pre-Training with Triple Contrastive Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[17] Yuhui Zhang, Jeff Z. HaoChen, Shih-Cheng Huang, Kuan-Chieh Wang, James Zou, and Serena Yeung. Diagnosing and Rectifying Vision Models using Language. In *International Conference on Learning Representations (ICLR)*, 2023.

[18] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional Prompt Learning for Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[19] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to Prompt for Vision-Language Models. In *International Journal of Computer Vision (IJCV)*, 2022.