Appendix of Any-Shift Prompting for Generalization over Distributions

A. Derivations of any-shift prompting

In the main paper, we provide the modeling of our any-shift prompting. Here we provide further derivations of the optimizations of the prior and posterior distributions.

To model the information of training and test distributions and their relationships, we propose any-shift prompting within a hierarchical framework. We introduce training and test prompts as latent variables in the hierarchical probabilistic architecture, the prediction function of the CLIP model is then formulated as:

$$
\begin{aligned}
&p_{\Phi,\theta}(\mathbf{y}_t|\mathbf{x}_t,\mathcal{Y}_t,\mathcal{D}_s)\\
&=\int\int p(\mathbf{y}_t,\mathbf{v}_t,\mathbf{v}_s|\mathbf{x}_t,\mathcal{Y}_t,\mathbf{x}_s,\mathbf{y}_s,\mathcal{Y}_s)d\mathbf{v}_t d\mathbf{v}_s\\
&=\int\int p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{v}_t,\mathcal{Y}_t)p(\mathbf{v}_t,\mathbf{v}_s|\mathbf{x}_t,\mathcal{Y}_t,\mathcal{D}_s)d\mathbf{v}_t d\mathbf{v}_s\\
&=\int\int p_{\Phi}(\mathbf{y}_t|\mathbf{x}_t,\mathbf{v}_t,\mathcal{Y}_t)p_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathbf{x}_t,\mathcal{Y}_t)p(\mathbf{v}_s|\mathcal{D}_s)d\mathbf{v}_t d\mathbf{v}_s,
\end{aligned}
\tag{1}
$$

where the prior distribution of the training and test prompts is factorized as

$$
p(\mathbf{v}_t,\mathbf{v}_s|\mathbf{x}_t,\mathcal{Y}_t,\mathcal{D}_s){=}p_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathbf{x}_t,\mathcal{Y}_t)p(\mathbf{v}_s|\mathcal{D}_s). \tag{2}
$$

$p(\mathbf{v}_s|\mathcal{D}_s)$ is learned from the training data $\mathcal{D}_s$ sampled from training distribution $p(\mathbf{x}_s,\mathbf{y}_s)$. $p_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathbf{x}_t,\mathcal{Y}_t)$ denotes the test prompt, which aggregates both training information from $\mathbf{v}_s$ and test information from the test image $\mathbf{x}_t$ and class names $\mathcal{Y}_t$. The test prompt exploits the relationships between training and test distributions by the transformer inference network $\boldsymbol{\theta}$. $\mathbf{v}_t$ is then utilized into the frozen image and text encoders $\Phi=\{\Phi_I,\Phi_T\}$ to generalize the CLIP model to the test data.

To optimize the model for generating the probabilistic training and test prompts, we further introduce variational inference to approximate the true posterior $p(\mathbf{v}_t,\mathbf{v}_s|\mathcal{D}_t,\mathcal{Y}_t,\mathcal{D}_s)$ into eq. (1), which is factorized as:

$$
q_{\boldsymbol{\theta}}(\mathbf{v}_t,\mathbf{v}_s|\mathcal{D}_t,\mathcal{Y}_t,\mathcal{D}_s){=}q_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathcal{D}_t,\mathcal{Y}_t)p(\mathbf{v}_s|\mathcal{D}_s), \tag{3}
$$

where $\mathcal{D}_t$ consists of test input-output pairs sampled from the test distribution $p(\mathbf{x}_t,\mathbf{y}_t)$. The variational posterior shares the same inference model $\boldsymbol{\theta}$ with the prior distribution. By integrating eq. (3) into eq. (1), the evidence lower bound (ELBO) of the log-likelihood $\log p_{\Phi,\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t,\mathcal{Y}_t,\mathcal{D}_s)$ is derived as:

$$
\begin{aligned}
&\log p_{\Phi,\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t,\mathcal{Y}_t,\mathcal{D}_s)\\
&=\log\int\int p(\mathbf{y}_t|\mathbf{x}_t,\mathbf{v}_t,\mathcal{Y}_t)p(\mathbf{v}_t,\mathbf{v}_s|\mathbf{x}_t,\mathcal{Y}_t,\mathcal{D}_s)d\mathbf{v}_t d\mathbf{v}_s\\
&=\log\int\int p(\mathbf{y}_{t'}|\mathbf{x}_t,\mathbf{v}_t,\mathcal{Y}_t)q_{\boldsymbol{\theta}}(\mathbf{v}_t,\mathbf{v}_s|\mathcal{D}_t,\mathcal{Y}_t,\mathcal{D}_s)\\
&\qquad\frac{p(\mathbf{v}_t,\mathbf{v}_s|\mathbf{x}_t,\mathcal{Y}_t,\mathcal{D}_s)}{q(\mathbf{v}_t,\mathbf{v}_s|\mathcal{D}_t,\mathcal{Y}_t,\mathcal{D}_s)}d\mathbf{v}_t d\mathbf{v}_s\\
&=\log\int\int p(\mathbf{y}_{t'}|\mathbf{x}_t,\mathbf{v}_t,\mathcal{Y}_t)q_{\boldsymbol{\theta}}(\mathbf{v}_t,\mathbf{v}_s|\mathcal{D}_t,\mathcal{Y}_t,\mathcal{D}_s)\\
&\qquad\frac{p_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathbf{x}_t,\mathcal{Y}_t)p(\mathbf{v}_s|\mathcal{D}_s)}{q_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathcal{D}_t,\mathcal{Y}_t)p(\mathbf{v}_s|\mathcal{D}_s)}d\mathbf{v}_t d\mathbf{v}_s\\
&=\log\int\int p(\mathbf{y}_{t'}|\mathbf{x}_t,\mathbf{v}_t,\mathcal{Y}_t)q_{\boldsymbol{\theta}}(\mathbf{v}_t,\mathbf{v}_s|\mathcal{D}_t,\mathcal{Y}_t,\mathcal{D}_s)\\
&\qquad\frac{p_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathbf{x}_t,\mathcal{Y}_t)}{q_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathcal{D}_t,\mathcal{Y}_t)}d\mathbf{v}_t d\mathbf{v}_s\\
&\geq\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{v}_t,\mathbf{v}_s)}\big[\log p_{\Phi}(\mathbf{y}_t|\mathbf{x}_t,\mathbf{v}_t,\mathcal{Y}_t)\big]\\
&\qquad-\mathbb{D}_{\mathrm{KL}}\big[q_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathcal{D}_t,\mathcal{Y}_t)||p_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathbf{x}_t,\mathcal{Y}_t)\big],
\end{aligned}
\tag{4}
$$

where the expectation of the log-likelihood is calculated on the variational posterior distribution $q_{\boldsymbol{\theta}}(\mathbf{v}_t,\mathbf{v}_s|\mathcal{D}_t,\mathcal{Y}_t,\mathcal{D}_s)$.

Our goal is to maximize the log-likelihood of the test data $\log p_{\Phi,\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t,\mathcal{Y}_t,\mathcal{D}_s)$, i.e., maximize the ELBO in eq. (4), which is equivalent to minimize the negative log-likelihood. Therefore, minimizing the loss function to optimize our any-shift prompting becomes minimizing:

$$
\begin{aligned}
&-\log p_{\Phi,\boldsymbol{\theta}}(\mathbf{y}_t|\mathbf{x}_t,\mathcal{Y}_t,\mathcal{D}_s)\\
&\quad\leq\mathbb{E}_{q_{\boldsymbol{\theta}}(\mathbf{v}_t,\mathbf{v}_s)}\big[-\log p_{\Phi}(\mathbf{y}_t|\mathbf{x}_t,\mathbf{v}_t,\mathcal{Y}_t)\big]\\
&\quad+\mathbb{D}_{\mathrm{KL}}\big[q_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathcal{D}_t,\mathcal{Y}_t)||p_{\boldsymbol{\theta}}(\mathbf{v}_t|\mathbf{v}_s,\mathbf{x}_t,\mathcal{Y}_t)\big].
\end{aligned}
\tag{5}
$$

B. Details of setting and implementations

B.1. Details of datasets and settings

Covariate shift. We conduct the experiments on covariate shifts in two settings, multiple training distributions and single training distributions. The experiments on multiple training distributions are conducted on domain generalization datasets `PACS`, `VLCS`, `Office-Home`, and `DomainNet`, which contain multiple domains of images with the same label space. `PACS` [16] includes images of 7 classes from four different domains, *photo*, *art-painting*, *cartoon*, and *sketch*. VLCS [7] consists of images of 5 classes and four different datasets, *Pascal-VOC2007* [6], *LabelMe* [21], *Caltech101* [10], and *SUN* [2]. Office-Home also contains four domains, *art*, *clipart*, *product*, and *real-world*, while the images are from 65 categories, which is much more than PACS and VLCS. DomainNet is even larger, which consists of images from six domains and 345 categories. The domains are *clipart*, *inforgraph*, *painting*, *quickdraw*, *real*, and *sketch*. We follow the "leave-one-out protocol" [16] on these datasets, where we select one domain as the test distribution, and the other domains are treated as the training distributions. The model is trained on the training distributions and evaluated on the test one. We treat each domain at the test distribution individually for evaluation and report the averaged results on all test distributions in Table 2 in the main paper. The detailed results of each test distribution are reported in the following section.

The experiments on single training distribution follow the domain generalization in Zhou *et al*. [28], where the model is trained on ImageNet (1,000 categories) and evaluated on the other four variants ImageNet-V2 [20], ImageNet-(S)ketch [26], ImageNet-A [13], and ImageNet-R [12] with the same label space. Most of the above datasets have shifts in the images, i.e., marginal input distributions $p(\mathbf{x})$. Therefore, we use these datasets for the evaluation of our method across covariate shift.

Label shift. We conduct the experiments on label shift following the base-to-new classification setting in Zhou *et al*. [29]. In this case, the distribution shifts occur in the marginal output distribution $p(\mathbf{y})$, where the "new" classes have $p(\mathbf{y}_c)$=0 during training. We use eleven benchmarks with label shift. The benchmarks includes general classification datasets ImageNet [4] and Caltech101 [8]; fine-grained classification datasets OxfordPets [19], StanfordCars [15], Flowers102 [18], Food101 [1], and FGVCAircraft [17]; scene recognition dataset SUN397 [27]; action recognition dataset UCF101 [25]; texture classification dataset DTD [3]; and satellite image recognition EuroSAT [11]. We follow the same base-new classes split and evaluation set in Zhou *et al*. [28].

Concept shift. We approximate the concept shift by relabeling the ImageNet dataset with the superclasses in [22]. The model is trained on the original classes and evaluated on the superclasses. In this case, the marginal input distribution $p(\mathbf{x})$ is the same while the conditional distributions $p(\mathbf{y}|\mathbf{x})$ are different between training and test data.

Conditional shift. For conditional shift, we evaluate the proposed method on two subpopulation datasets, Living-17 and Entity-30 [22], which contain images of 17 animal categories and images of 30 entities, respectively. We follow the training and test split in [9], where the training and test distributions have the same overall classes but contain different subpopulations of those classes. In this case, the marginal output distributions $p(\mathbf{y})$ of training and test data are the same, while the input distributions are changed according to different categories, i.e., $p(\mathbf{x}|\mathbf{y})$ are different. Therefore, we treat the setting as conditional shift.

Domains	Classes
Source 1	0 - 2, 3 - 8, 9 - 14, 21 - 31
Source 2	0 - 2, 3 - 8, 15 - 20, 32 - 42
Source 3	0 - 2, 9 - 14, 15 - 20, 43 - 53
Target	0 - 64

Table 1. **Classes split for joint distribution shifts** on Open-Office-Home. We use the numbers to denote the class names. The setting contains both covariate and label shifts, leading to joint shifts on $p(\mathbf{x}, \mathbf{y})$.

Joint shift. To evaluate the proposed method on joint shift, we conduct experiments on Office-Home under the open domain generalization setting [24], which we refer to as Open-Office-Home. We split the label space of the 65 classes and make various label spaces across different domains. The split of classes is shown in Table 1. Therefore, there are both covariate shift and label shift between the training and test distributions, which we treat as the joint shift on $p(\mathbf{x}, \mathbf{y})$.

B.2. Implementations and hyperparameters

For all experiments, we train and evaluate the model on a single NVIDIA V100 GPU. We use the same backbone and transformer inference network for all datasets. The backbone is the frozen CLIP model with ViT-B/16 as the image encoder. The transformer inference network consists of a 2-layer transformer and 2 MLP layers to generate the distribution of the test prompt. There are also two trainable vectors as the mean and variance of the probabilistic training prompt and trainable position embeddings for image and text features respectively. The sampled test prompt is then fed into both the image and text encoders to generalize the features and classifiers. We provide an illustration in Figure 1. Note that the test prompt is utilized as tokens of the image and text encoders. To make it the same size as the inputs, we use two linear layers to project the test prompt to the image path and text embedding space, respectively.

Except for the architecture and settings shared by all datasets, we also provide the specific hyperparameters for different datasets. Batch size is a hyperparameter that varies per dataset (Tables 2 and 3). For the experiments of label shift (eleven datasets) and the others based on ImageNet (ImageNet-based covariate shift and concept shift), we use the same learning rate $2e-3$ as Zhou *et al*. [28] with SGD. The dataset-specific batch size and epochs are provided in Table 2. For the covariate shift datasets

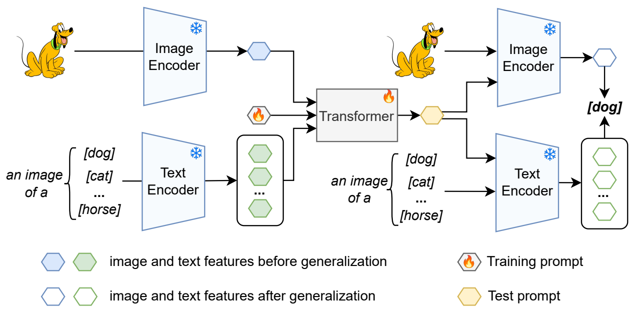

Figure 1. **Overall framework of generating the any-shift prompt and generalizing the CLIP model.**

	`ImageNet`	`Caltech101`	`OxfordPets`	`StanfordCars`	`Flowers102`	`Food101`	`FGVC`	`SUN397`	`DTD`	`EuroSAT`	`UCF101`
Learning rate	$2e-3$										
Optimizer	SGD										
Batch Size	1	4	8	6	4	4	4	2	8	10	4
Epochs	10	30	30	30	30	30	30	30	30	30	30

Table 2. **Dataset-specific hyper-parameters for label shift datasets and ImageNet-based datasets**. The ImageNet-based covariate shift, label shift, and concept shift datasets use the same hyperparameters.

	`PACS`	`VLCS`	`Office-Home`	`Open-Office-Home`	`DomainNet`	`Living-17`	`Entity-30`
Learning rate	$5e-4$						
Optimizer	Adam						
Training iterations	3,000 iterations				10,000 iterations	30 epochs	
Batch Size	32	32	8	8	2	32	16

Table 3. **Dataset-specific batch sizes for common domain generalization datasets and conditional shift datasets.**

		Accuracy				
Method	Iterations	Art	Clipart	Product	Real	***Mean***
CLIP baseline	-	79.32	67.70	86.93	87.46	80.35
Transformer adapter	20,000	78.76	64.62	87.98	84.83	79.05
Any-shift prompt	3,000	**83.40**	**72.53**	**91.24**	**90.84**	**84.50**

Table 4. **Benefits of generalization with any-shift prompting.** Directly training a transformer as an adapter of the image and textual features still easy to lead to overfitting. By aggregating the training, test, and relationship information into the prompt, any-shift prompting achieves better generalization.

`PACS`, `VLCS`, `Office-Home`, `DomainNet` and joint shift dataset `Open-Office-Home`, we train the model with $5e-4$ learning rate and 3000 iterations by Adam optimizer.

Inference network	Art	Clipart	Product	Real	***Mean***
CLIP baseline	79.32	67.70	86.93	87.46	80.35
Averaging	82.27	70.91	89.95	89.66	83.20
MLP	82.48	71.09	90.18	89.73	83.37
Transformer	**83.40**	**72.53**	**91.24**	**90.84**	**84.50**

Table 5. **Ablations on the aggregation methods.** The transformer inference network performs best since it better encodes the relationships between different information.

For the conditional shift dataset conditional shift datasets `Living-17` and `Entity-30`, we use the same learning rate $5e-4$ and Adam optimizers for 30 epochs. The details are shown in Table 3.

C. More ablations and comparisons

Benefits of generalization with prompts In our any-shift prompting, we generate the test prompt by aggregating the training information and the test information by a transformer inference network. The test information is from the image and textual features of the CLIP model. In addition to generating the prompt for the CLIP model, another way to achieve generalization is directly adapting the image and textual features by the transformer network and making predictions by the image and textual features. To show the benefits of generalization with our any-shift prompting, we conduct an experiment that adapts the image and textual features using the same transformer inference network, which we refer to as "Transformer adapter". The experimental results on `Open-Office-Home` are reported in Table 4. The transformer adapter performs even worse than the CLIP baseline since it is still easy to overfit the training distribution. Moreover, the transformer adapter requires much more training costs (20,000 iterations) than any-shift prompting (3,000 iterations). The results demonstrate both the effectiveness and efficiency of our any-shift prompting for generalization across distribution shifts.

Benefits of the transformer inference network We also conduct experiments on `Open-Office-Home` with different methods for aggregating the training and test information. We generate the test prompt by directly averaging the training prompts, the test image feature, and textual features. In addition, we also use an MLP network to replace the transformer network to generate the test prompt from the averaged features. As shown in Table 5, the transformer inference network achieves the best performance, demonstrating the effectiveness of considering the relationships between different information for aggregation.

Comparison on cross-dataset shift. Following Zhou *et al.* [28], we conduct experiments on the cross-dataset setting, where the model trained on `ImageNet` is evaluated on the other 10 datasets shown in Table 6. In this case, there are

	Source	Target										
	ImageNet	Caltech101	OxfordPets	StanfordCars	Flowers102	Food101	FGVCAircraft	SUN397	DTD	EuroSAT	UCF101	*Average*
CoOp [29]	**71.51**	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
CoCoOp [28]	71.02	94.43	90.14	65.32	71.88	86.06	22.94	67.36	45.73	45.37	68.21	65.74
TPT [23]	68.98	68.98	47.75	87.79	66.87	68.04	94.16	84.67	65.50	24.78	42.44	65.10
BPL [5]	70.70	93.67	90.63	65.00	70.90	**86.30**	24.93	**67.47**	46.10	45.87	68.67	65.95
MaPLe [14]	70.72	93.53	90.49	65.57	72.23	86.20	24.74	67.01	46.49	48.06	68.69	66.30
This paper	71.05	**94.57**	**90.79**	**66.90**	**72.30**	86.17	**25.16**	67.32	**47.35**	**50.25**	**69.52**	**67.03**

Table 6. **Comparison of prompt learning methods in the cross-dataset transfer setting.** Our method achieves the best overall performance on 10 test datasets.

Method	Photo	Art	Cartoon	Sketch	*Mean*
CLIP	99.94	97.41	98.98	88.19	96.13
CLIP-D	99.94	97.61	99.02	90.03	96.65
CoOp	99.70	97.56	98.59	89.95	96.45
CoCoOp	99.94	98.09	99.19	90.77	97.00
TPT	99.82	97.68	98.92	92.58	97.25
This paper	**99.94**	**98.86**	**99.32**	**94.53**	**98.16** $\pm$ 0.4

Table 7. **Detailed comparisons on `PACS` with covarate shift.**

Method	VOC	LabelMe	Caltech	SUN	*Mean*
CLIP	84.32	68.26	98.61	74.52	81.43
CLIP-D	82.60	68.76	98.76	72.68	80.70
CoOp	85.86	68.51	98.94	76.72	82.51
CoCoOp	86.03	70.45	99.12	77.96	83.39
TPT	86.20	71.05	99.46	80.60	84.33
This paper	**88.14**	**72.65**	**100.00**	**85.37**	**86.54** $\pm$ 0.4

Table 8. **Detailed comparisons on `VLCS` with covarate shift.**

Method	Art	Clipart	Product	Real	*Mean*
CLIP	79.32	67.70	86.93	87.46	80.35
CLIP-D	80.47	68.83	87.93	88.80	81.51
CoOp	80.99	69.52	88.69	89.28	82.12
CoCoOp	81.78	70.09	89.32	89.89	82.77
TPT	82.45	71.18	90.03	90.15	83.45
This paper	**83.70**	**73.00**	**92.50**	**91.44**	**85.16** $\pm$ 0.6

Table 9. **Detailed comparisons on `Office-Home`.**

Method	Clipart	Painting	Real	Infograph	Quickdraw	Sketch	*Mean*
CLIP	68.12	56.18	78.82	46.36	14.32	60.69	54.08
CLIP-D	70.83	58.02	80.52	48.85	16.39	62.84	56.24
CoOp	74.39	61.18	83.26	51.88	16.67	65.52	58.82
CoCoOp	74.82	61.56	83.98	**52.68**	17.47	66.10	59.43
TPT	75.09	62.77	84.67	52.65	17.28	66.98	59.90
This paper	**76.08**	**66.62**	**85.03**	52.56	**18.05**	**67.26**	**60.93** $\pm$ 0.4

Table 10. **Detailed comparisons on `DomainNet`.**

different distribution shifts for different test datasets. Compared with the other prompt learning methods, e.g., CoOp [29], CoCoOp [28], BPL [5], MaPLe [14], and test-time tuning method TPT [23], our method shows improvement on 8 of the 10 datasets, as well as the averaged result.

Detailed results on covariate shift We also report the detailed comparisons of each test distribution on the four covariate shift datasets. The results of `PACS`, `VLCS`, `Office-Home`, and `DomainNet` are provided in Table 7, 8, 9, and 10, respectively. Our method achieves the best performance on most of the test distributions.

Inference efficiency. Since our method only uses a single feedforward pass for generating the test prompts and making predictions, the inference time cost per iteration on a single V100 GPU (0.13s) is slightly higher than other prompt tuning methods like CoOp (0.10s) and CoCoOp (0.11s), and faster than TPT (0.25s), which has 1-step optimization at test time.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461. Springer, 2014. 2

[2] Myung Jin Choi, Joseph J Lim, Antonio Torralba, and Alan S Willsky. Exploiting hierarchical context on a large database of object categories. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 129–136. IEEE, 2010. 2

[3] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 2

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

[5] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor G Turrisi da Costa, Cees GM Snoek, Georgios Tzimiropoulos, and Brais Martinez. Bayesian prompt learning for image-language model generalization. In *IEEE International Conference on Computer Vision*, pages 15237–15246, 2023. 4

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 2

[7] Yuming Fang, Weisi Lin, Zhenzhong Chen, Chia-Ming Tsai, and Chia-Wen Lin. A video saliency detection model in compressed domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(1):27–38, 2013. 2

[8] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178. IEEE, 2004. 2

[9] Saurabh Garg, Nick Erickson, James Sharpnack, Alex Smola, Sivaraman Balakrishnan, and Zachary Chase Lipton. Rlsbench: Domain adaptation under relaxed label shift. In *International Conference on Machine Learning*, pages 10879–10928. PMLR, 2023. 2

[10] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 2

[11] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 2

[12] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *IEEE International Conference on Computer Vision*, pages 8340–8349, 2021. 2

[13] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 2

[14] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multimodal prompt learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 4

[15] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. 2

[16] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *IEEE International Conference on Computer Vision*, pages 5542–5550, 2017. 1, 2

[17] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 2

[18] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 2

[19] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505. IEEE, 2012. 2

[20] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019. 2

[21] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008. 2

[22] Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*, 2020. 2

[23] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *Advances in Neural Information Processing Systems*, pages 14274–14289, 2022. 4

[24] Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9624–9633, 2021. 2

[25] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[26] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, 2019. 2

[27] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010. 2

[28] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 2, 3, 4

[29] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2, 4