Supplementary Material
Bi-directional Distribution Alignment for Transductive Zero-Shot Learning

We present additionally (1) the dataset and implementation details, (2) the explanation of the feature pre-tuning network, (3) an examination of different feature spaces, and (4) the comparison of BBSE and CPE for class prior estimation.

1. Dataset and Implementation

1.1. Dataset

We conduct experiments using four benchmark datasets. The Animals with Attributes 1&2 (AWA1 [2] & AWA2 [10]) contain 30,475&37,322 samples from a total of 50 classes, and the dimension of the attribute vector is 85. The Caltech UCSD Bird 200 (CUB) [8] consists of 11,788 fine-grained images of 200 bird species with an attribute size of 312. The SUN Scene classification (SUN) [6] dataset has 14,340 samples selectecd from 717 scenes with an attribute size of 102. More details are shown in Table 1.

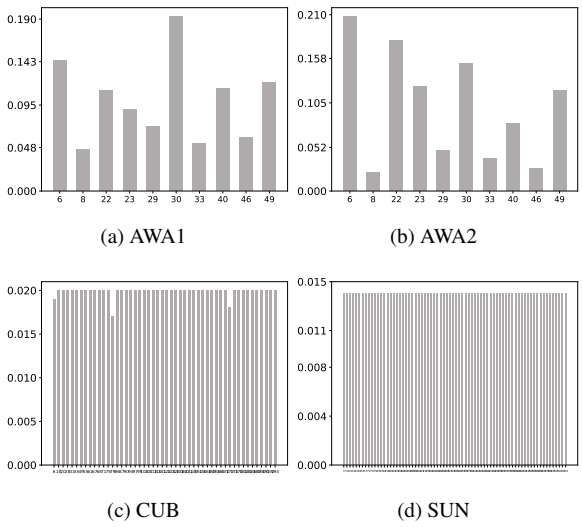
Dataset	N	att.	stc.	$\|\mathcal{Y}^s\|$	$\|\mathcal{Y}^u\|$
AWA1	30,475	85	-	40	10
AWA2	37,322	85	-	40	10
CUB	11,788	312	1,024	150	50
SUN	14,340	102	-	645	72

Table 1. Statistics of the four datasets. 'att.' denotes the attribute size, 'stc.' is the dimension of semantic information extracted from descriptive sentences [7], $\|\mathcal{Y}^s\|$ and $\|\mathcal{Y}^s\|$ correspond to the numbers of the seen and unseen classes, respectively.

Figure 1 displays the class distribution prior estimated from the class information of the testing samples from the unseen classes, i.e., the percentage of the samples contained by each class, for the four datasets. AWA1 and AWA2 have unbalanced class priors, while CUB and SUN have class priors close to a uniform distribution. AWA2 has more samples from those popular classes like 'horse' and 'dolphin'.

1.2. Implementation

In the training of all our modules, we use AdamW optimizer [4] with a learning rate of 0.001 and (β_1, β_2) is set as (0.5, 0.999). The encoder $\boldsymbol{E}$, decoder $\boldsymbol{G}$ and regressor $\boldsymbol{R}$ in Bi-VAEGAN are all two-layer MLPs, in which the hidden layer output has 4,096 dimensions and the inner activation layer is LeakyReLU. The conditional visual critic D, unconditional visual critic D^u, and attribute critic D^a



(a) AWA1 (b) AWA2

(c) CUB (d) SUN

Figure 1. The unseen class prior computed from test data.

are two-layer MLPs where the output of the last layer is a scalar, and the WGAN gradient penalty coefficient is set as 10. The level-1 and level-2 trainings proceed alternatively. We conduct one-step level-1 training for every five steps of level-2 training to accelerate the training speed. The training epochs for AWA1, AWA2, CUB, and SUN are set to be 300, 300, 600, and 400, respectively. In the inference stage, the synthesized feature number of each class are set to be 3000, 3000, 400, and 400 for for AWA1, AWA2, CUB, and SUN, respectively. The classifier f is a single fully connected (FC) layer and its output dimension is equal to the number of unseen classes for TZSL or the number of both seen and unseen classes for generalized TZSL.

The used hyper-parameters for reporting results are r=1 for L_2 normalization, λ=1, α=1, β=10 and γ=10, where the setting of α, γ and the WGAN critic training are the same as TF-VAEGAN [5]. In level-1 training, λ is less sensitive and thus set to 1. Values of r and β are searched within $\{1, 2, 5, \cdots, 100\}$ and $\{0.01, 0.1, 1, 10, 100\}$, respectively. Due to the unavailability of a test split in the datasets, we report our results on the validation split, consistent with previous works [5, 9]. For conventional Zero-Shot Learning (ZSL) and Generalized Zero-Shot Learning (GZSL), a more rigorous setting is desired, especially under the impact of current large models.

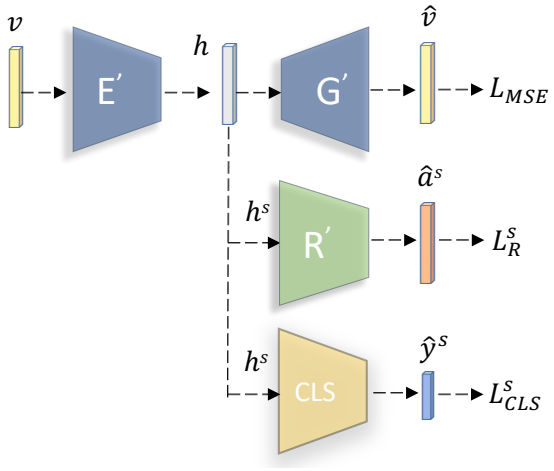


Figure 2. The used feature pre-tuning network.

2. Feature Pre-tuning Network

2.1. Used Approach

For the CUB dataset, we pre-tune the pre-trained features using a supervised neural network of which the architecture is shown in Figure 2. It builds on an auto-encoder network ($\boldsymbol{E}'$ and $\boldsymbol{G}'$) and consists of two supervised modules that work in the latent space, acting as a regressor ($\boldsymbol{R}'$) and a classifier (CLS). '′' denotes it is a different module from the one in the main text. The input and latent features share the same feature dimension, i.e., 2,048 for the pre-trained ResNet-101. Only the seen classes receive supervision from the two supervised modules. The training objective for feature pre-tuning is,

$$\min_{\boldsymbol{E}',\boldsymbol{G}',\boldsymbol{R}',CLS} L_{MSE} + L^s_{\boldsymbol{R}'} + L^s_{CLS}, \tag{1}$$

where

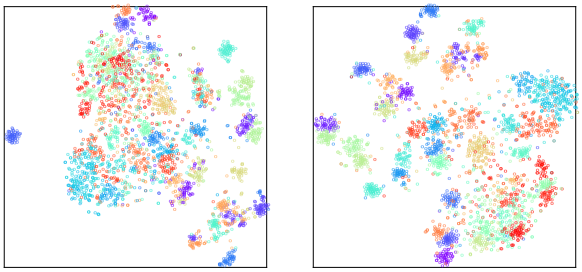
$$L^s_{CLS}(\mathcal{V}^s) = \mathbb{E}[\log(P(y|v^s))]. \tag{2}$$

The latent features are extracted by the encoder $\boldsymbol{E}'$ after training for 15 epochs for both the seen and unseen classes. These replace the original visual features to be used as the input of Bi-VAEGAN.

2.2. Result Comparison

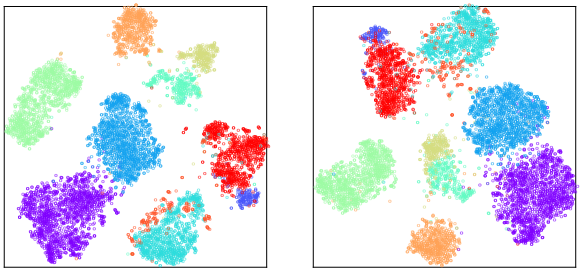
Figures 3 and 4 visualize the tuned and untuned features for the CUB and AWA2 datasets, using the visulization tool t-SNE? . The tuned features exhibit more clear cluster structure for the cross-domain dataset CUB. It should be noted that our feature pre-tuning network will not be beneficial for datasets that already have a satisfactory cluster structure, and somehow the cluster property could be damaged.

Table 2 demonstrates the effect of feature pre-tuning on AWA2 and CUB datasets. We name the simplified model



(a) untuned (CUB) (b) pre-tuned (CUB)

Figure 3. Visualization of vanilla and pre-tuned CUB features.



(a) untuned (AWA2) (b) pre-tuned (AWA2)

Figure 4. Visualization of vanilla and pre-tuned AWA2 features.

that only contains $\boldsymbol{G}$, $\boldsymbol{D}$, and $\boldsymbol{D}^u$ as a Simple-GAN. Both Simple-GAN and Bi-VAEGAN use L_2 feature normalization. A key observation is that for CUB, feature pre-tuning introduces a noticeable improvement for both models, i.e., $+8.8$ and $+1.2$ respectively, when using the less informative AK1 knowledge. Notably, Simple-GAN significantly benefits from this straightforward strategy and performs comparably to the untuned Bi-VAEGAN, e.g., 76.9% vs. 76.8%. This shows that despite the fact that no additional super-

Model	AWA2	CUB$^{\text{AK1}}$	CUB$^{\text{AK2}}$
w/o pre-tuning			
Simple-GAN	92.7	68.1	79.8
Bi-VAEGAN	**95.8**	76.8	**82.8**
w/ pre-tuning			
Simple-GAN	88.9 (-3.8)	76.9 ($+8.8$)	80.3 ($+0.5$)
Bi-VAEGAN	90.0 (-5.8)	**78.0** ($+1.2$)	82.0 (-0.8)

Table 2. The effect of feature pre-tuning on AWA2 and CUB. Simple-GAN is a simplified version of Bi-VAEGAN. All performances are shown in percentage (%). CUB$^{\text{AK1}}$ conditions on the original attribute information (AK1) while CUB$^{\text{AK2}}$ conditions on the semantic embedding (AK2) extracted from the fine-grained visual description.

vision (regressor) is applied, the visual feature alignment for the tuned features is substantially simpler. We could conclude that the tuned features can lead to a better inter-class discriminability, which enables an easier alignment between the auxiliary and visual spaces when the class distribution prior is known.

Another observation is that Simple-GAN benefits less from the feature pre-tuneing ($+0.5$) when it conditions on the more informative AK2. Bi-VAEGAN also shows a small performance drop (-0.8) with the feature pre-tuning. We could conclude that the pre-tuned features are less effective when the auxiliary information is already strong enough. Besides, for the AWA2 dataset, pre-tuning decreases the inter-class discriminability as shown in Figure 4, and a significant performance drop (-3.8, -5.8) is observed. These indicate that feature pre-tuning is not a completely free-lunch approach and that cross-domain datasets may benefit more from it. Transductive regressor could also achieve a competitive knowledge transfer for the cross-domain dataset. It is easier to provide a better alignment since it doest not change the original features extracted from the powerful backbone. Overall, both the transductive regressor method and the feature pre-tuning offer advantages of their own and may complement one another in complex real-world circumstances.

3. Feature Augmentation

As a bi-directional distribution alignment technique for TZSL, our Bi-VAEGAN allows the regressor and generator to independently solve the TZSL problem. In the inference phase, we compare the performance of using four different feature spaces, i.e., attribute space $\mathcal{A}$, hidden space $\mathcal{H} \in \mathbb{R}^{4096}$ corresponding to the hidden representation of the regressor, visual space $\mathcal{V}$ and the augmented multi-modal space $\mathcal{A} \times \mathcal{H} \times \mathcal{V}$. To conduct inference on $\mathcal{A}$, we have two straightforward choices: (1) Use only the transductively trained $\boldsymbol{R}$ and infer for the test unseen data $\boldsymbol{R}(V^u)$ using a 1-nearest neighbor (1-NN) classifier. (2) Use both $\boldsymbol{G}$ and $\boldsymbol{R}$, synthesize the labeled unseen set $\langle \boldsymbol{R}(\hat{V}^u_{\boldsymbol{G}}), \hat{Y}^u_{\boldsymbol{G}} \rangle$ in attribute space, train a neural network classifier using the labeled set that includes the synthesized examples and infer for $\boldsymbol{R}(V^u)$ using this classifier. A similar method of inference can also be applied to the hidden space when this option is chosen.

Discussion. Table 3 shows the TZSL top-1 accuracy on three datasets using different spaces to conduct inference. The observation could be summarized as, (1) $\boldsymbol{R}$ could be served as an individual module to conduct TZSL inference, but it is much less discriminative that $\boldsymbol{G}$. (2) When using $\boldsymbol{G}$ to conduct inference, a multi-modal space is preferred and the rank of spaces' discriminability is $\mathcal{H} > \mathcal{V} > \mathcal{A}$. We attribute the hidden space absorbing the knowledge of both transductive generator and regressor and the larger dimensionality is also preferred to alleviate the hubness problem.

Module	Space	AWA2	CUB$^{\text{AK1}}$	CUB$^{\text{AK2}}$	SUN
$\boldsymbol{R}$	$\mathcal{A}$	73.2	64.5	45.0	52.6
$\boldsymbol{G}$	$\mathcal{V}$	94.2	75.0	81.8	71.8
$\boldsymbol{R}, \boldsymbol{G}$	$\mathcal{A}$	89.8	65.6	67.3	53.2
	$\mathcal{H}$	**95.8**	**77.2**	82.7	73.8
	$\mathcal{A} \times \mathcal{H} \times \mathcal{V}$	**95.8**	76.8	**82.8**	**74.2**

Table 3. TZSL results of Bi-VAEGAN using different feature spaces.

4. BBSE vs. CPE for Class Prior Estimation

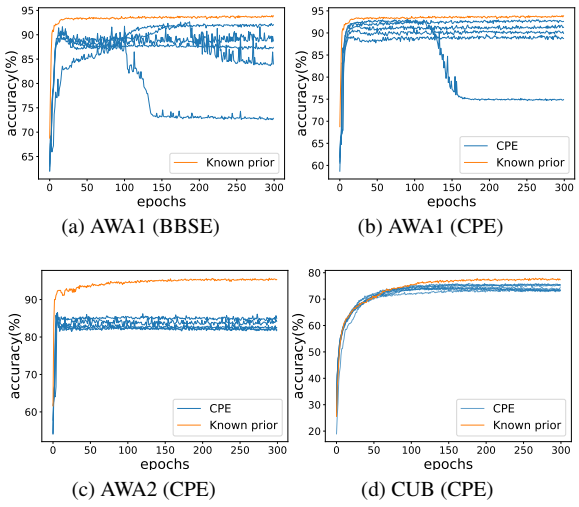
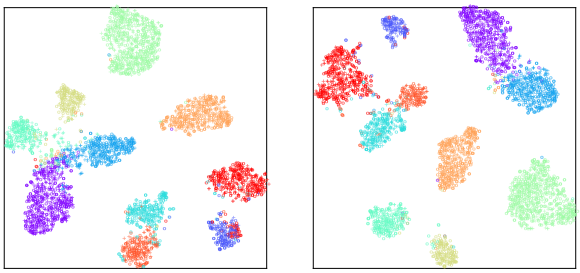


Figure 5. Training accuracy of Bi-VAEGAN using different random seeds when class prior is unknown.

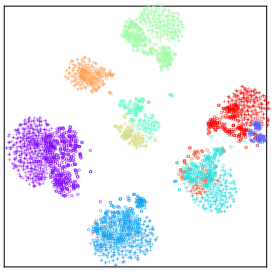
Here we explain the black box shift estimation (BBSE) [3] approach for class prior estimation. It attempts to solve the problem in label shift setting [1] and we consider the TZSL problem in a discrete form i.e., $y \in Y^u = \{0, 1, 2, ..., N_u - 1\}$. We view our synthesized joint distribution $p^u_G(\hat{\boldsymbol{v}}, y)$ as the source domain and the unknown joint distribution $p^u(\boldsymbol{v}, y)$ as the target domain. Under the label shift assumption, i.e., $p^u_G(\hat{\boldsymbol{v}}|y) = p^u(\boldsymbol{v}|y)$, we can approximate the unseen prior via the normalized confusion matrix $\boldsymbol{C}_{\hat{y},y} := p^u_{\boldsymbol{G}}(\hat{y}|y)$ of synthesized features, where $\hat{y} = f(\hat{\boldsymbol{v}})$ is the predicted label using hypothesis f. Following [3], when the label shift condition is held and the confusion matrix is invertible, the following equation holds,

$$\begin{aligned} p^u(\hat{y}) &= \sum_{y \in Y^u} p^u(\hat{y}|y) p^u(y) = \sum_{y \in Y^u} p^u_{\boldsymbol{G}}(\hat{y}|y) p^u(y), \\ &= \sum_{y \in Y^u} \boldsymbol{C}_{\hat{y},y} p^u(y), \end{aligned} \tag{3}$$

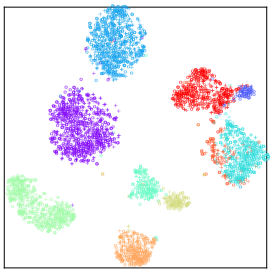


(a) AWA1 + CPE (T1-91.5%)

(b) AWA1 (T1-93.9%)



(c) AWA2 + CPE (T1-85.6%)



(d) AWA2 (T1-95.8%)

Figure 6. Visualization of real/synthesized unseen visual feature using Bi-VAEGAN. Left column uses CPE strategy and class prior is unknown. The right column is trained with given real class prior. '$\circ$' means the real feature and '+' means the syntehsized feature.

thus $p^u(y)$ is computed as,

$$p^u(y) = \sum_{\hat{y} \in Y^u} \boldsymbol{C}^{-1}_{y,\hat{y}} p^u(\hat{y}). \tag{4}$$

To compute the confusion matrix $\boldsymbol{C}$ we synthesize two labeled unseen set $\langle \hat{V}^u_{\boldsymbol{G}}, \hat{Y}^u_{\boldsymbol{G}} \rangle_1$ and $\langle \hat{V}^u_{\boldsymbol{G}}, \hat{Y}^u_{\boldsymbol{G}} \rangle_2$. We train the hypothesis on one labeled set and compute the confusion matrix on the other set. Note that as the training process goes, the confusion matrix tends to be an identity matrix and the BBSE estimation collapse to $p^u(y) \leftarrow p^u(\hat{y})$.

Discussion. We display the BBSE and our CPE's training accuracy curves on AWA1 in Figure 5a and Figure 5b. It might be observed that BBSE is more vulnerable to seed selection and that it more readily results in a poor convergence. This observation can be explained as that the label shift assumption is too strong for prior estimation, so that the neural network classifier performs more unstably. The non-parametric K-means technique tends to provide a more moderate and reliable estimation since CPE avoids directly employing the black-box neural network classifier and utilizes it as an initialization approximation of the class center instead.

Figure 6 shows the t-SNE visualizations using CPE when the class prior is unknown. For the more evenly balanced AWA1, our CPE provides a satisfactory alignment between the real and the syntehsized features, and there is only a minor accuracy gap with the known prior scenario (91.5% vs. 93.9%). For the more unbalanced AWA2 dataset, the domain between the synthesized and real features shifts noticeably, and the performance disparity with the know-prior scenario increases to (85.6% vs. 95.8%). This supports the argument of Corollary 3.1 that the class prior is crucial to the alignment of the conditional distribution for the TZSL. However, it is still unclear how to proceed with a more accurate class prior estimation when the real prior is highly unbalanced. Different from the widely studied problems of *covariate shift* and *label shift* in domain adaptation [3, 11], the unknown prior TZSL is less well-organized and is more similar to a cross-modal *generalized label shift* problem.

References

[1] Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary C. Lipton. A unified view of label shift estimation. In *NeurIPS*, 2020. 3

[2] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 36(3):453–465, 2013. 1

[3] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *ICML*, pages 3122–3130. PMLR, 2018. 3, 4

[4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*. OpenReview.net, 2019. 1

[5] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *ECCV*, pages 479–495. Springer, 2020. 1

[6] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, pages 2751–2758. IEEE, 2012. 1

[7] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, pages 49–58, 2016. 1

[8] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 1

[9] Jiamin Wu, Tianzhu Zhang, Zheng-Jun Zha, Jiebo Luo, Yongdong Zhang, and Feng Wu. Self-supervised domain-aware generative network for generalized zero-shot learning. In *CVPR*, pages 12767–12776, 2020. 1

[10] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 41(9):2251–2265, 2018. 1

[11] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. On learning invariant representations for domain adaptation. In *ICML*, volume 97, pages 7523–7532. PMLR, 2019. 4