Supplementary Material on Bridging the Projection Gap: Overcoming Projection Bias Through Parameterized Distance Learning

Chong Zhang[1], Mingyu Jin[1], Qinkai Yu[2], Haochen Xue[1]
Shreyank N Gowda[3], and Xiaobo Jin[1] †

[1] Xi'an Jiaotong-Liverpool University
[2] University of Liverpool,
[3] University of Oxford

1 Affine Transformation Fusion Schematic

For Affine Transformation Fusion, as shown in Fig. 1, we projected language-conditioned channel-wise scaling parameters α and shifting parameters β from text vector C from two MLPs (Multilayer Perceptron).

$$\alpha = MLP_1(C), \qquad \beta = MLP_2(C). \tag{1}$$

For any given input feature X from the backbone, we first conduct the channel-wise scaling operation with the scaling parameter α, then apply the channel-wise shifting operation with the shifting parameter β,

$$ATF(X, C) = \alpha X + \beta = MLP_1(C) \cdot X + MLP_2(C). \tag{2}$$

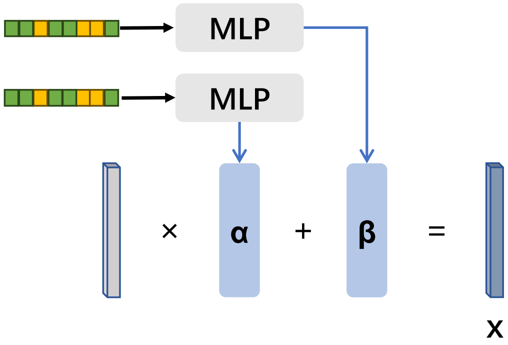


Fig. 1: Affine Transformation Fusion schematic diagram

† Corresponding author: Xiaobo Jin. Email: xiaobo.jin@xjtlu.edu.cn

2 Mahalanobis Metric and Euclidean Metric

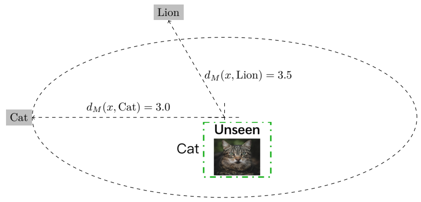


Fig. 2: Our method corrects the projection bias problem of the model through Mahalanobis distance: Mahalanobis distance-based inference correctly assigns the unknown sample from unseen classes to its true class Cat.

Fig. 2 shows our motivation for using Mahalanobis distance instead of Euclidean distance in reasoning through a specific example of two similar classes, Lion and Cat. First, we project unknown samples and semantic vectors into the same common space through a deep network model. According to the Euclidean distance, it can be seen that the unknown sample is closer to the seen class Lion, and then the unknown sample from the unseen class Cat will be mistakenly classified into the Lion class. However, according to the Mahalanobis distance, the unknown sample is closer to the class Cat. Note that the Mahalanobis distance from the dotted points of the ellipse to the center of the ellipse is equal. Still, the category Lion is outside the dotted points of the ellipse, so the Mahalanobis distance between the category Lion and the ellipse's center is farther. Therefore, our method can alleviate the biased problem of projection learning in inference to some extent if our deep model learns a less accurate projection representation.

3 Comparison with pre-2019 methods

Tab. 2 gives the performance comparison between our method and the method before 2019 as a supplement to the experimental results of the main text. It can be seen that the methods in 2019 mainly focus on the performance of seen classes (S columns), while the generalization on unseen (U columns) has been greatly improved. For example, on CUB, our method has greatly improved on invisible classes, reaching 62.1, compared with 31.73 of the f-CLSWGAN method.

4 Ablation Study

Impact of $\lambda_{\mathbf{VAE}}$, $\lambda_{\mathbf{MSE}}$ and $\lambda_{\mathbf{M}}$ Tab. 1 gives a more detailed data comparison of Figure 3. Each parameter takes $\{0.1, 0.5, 0.8\}$, while keeping the other two parameter values as 1, so that the impact of each parameter on the model can be quantitatively analyzed.

Table 1: Quantitative results of the impact of parameters λ_{VAE}, λ_{MSE}, and λ_{M} on the model

$(\lambda_{\mathrm{VAE}}, \lambda_{\mathrm{MSE}}, \lambda_{\mathrm{M}})$	**CUB**			**AWA2**		
	U	**S**	**H**	**U**	**S**	**H**
$(0.1, 1.0, 1.0)$	16.1	24.3	19.4	13.6	27.7	18.2
$(0.5, 1.0, 1.0)$	24.2	27.6	25.8	17.6	31.7	22.6
$(0.8, 1.0, 1.0)$	54.3	**76.4**	63.5	59.3	67.3	63.0
$(1.0, 0.1, 1.0)$	49.3	51.7	50.5	47.5	57.1	51.9
$(1.0, 0.5, 1.0)$	48.4	52.6	50.4	42.3	49.9	45.8
$(1.0, 0.8, 1.0)$	58.7	57.2	64.5	57.7	67.6	62.3
$(1.0, 1.0, 0.1)$	15.7	38.3	22.3	21.4	39.9	27.9
$(1.0, 1.0, 0.5)$	22.3	32.1	26.3	27.1	34.4	30.3
$(1.0, 1.0, 0.8)$	61.4	69.8	65.3	63.1	71.6	67.1
$(1.0, 1.0, 1.0)$	**62.1**	74.6	**67.8**	**64.9**	**79.1**	**71.3**

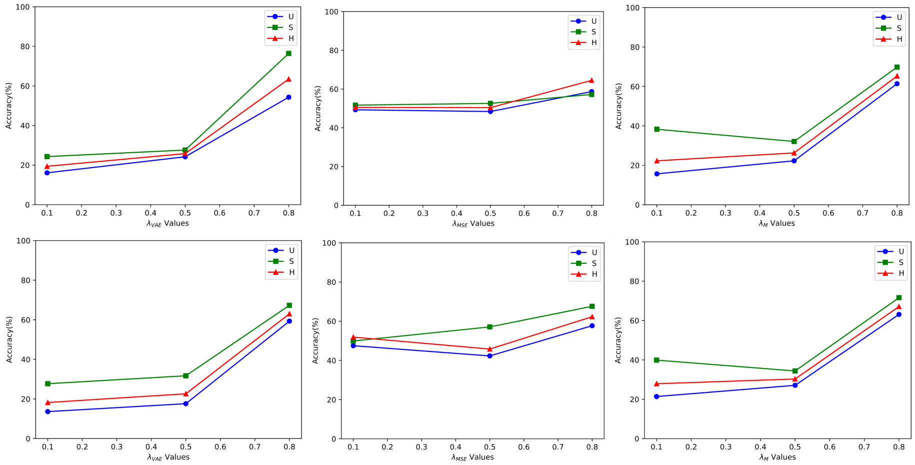


Fig. 3: Performance of various λ_{VAE}, λ_{MSE} and λ_{M} ratios on CUB dataset (above row) and AWA2 dataset (below row).

Table 2: Comparison of our method with pre-2019 state-of-the-art methods on four datasets

Model	**CUB**			**AWA1**			**AWA2**			**SUN**		
	U	S	H	U	S	H	U	S	H	U	S	H
ALE [2015]	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9	21.8	33.1	26.3
DEM [2017]	19.6	57.9	29.2	32.8	84.7	47.3	30.5	81.4	45.1	19.6	57.9	29.2
f-CLSWGAN [2018]	31.73	64.34	42.50	61.41	59.63	60.51	29.85	76.60	42.96	42.6	36.6	39.4
Our model + ResNet50	57.1	**81.6**	67.2	62.9	**83.1**	**71.6**	62.2	**82.3**	70.9	39.6	**52.7**	45.9
Our model + ViT-B	**62.1**	74.6	**67.8**	**67.2**	76.3	71.5	64.9	79.1	**71.3**	45.7	49.8	**47.7**

Impact on Dimensions of Feature Vectors Regarding feature vector dimensions of image and semantic information, we take the hidden layer of each pre-trained model as the default output vector. At the same time, we also set the dimensions of the visual features to 500 and 1000 on the data sets CUB and AWA2 to explore the impact of the model parameters on the model performance, as shown in Fig. 4.

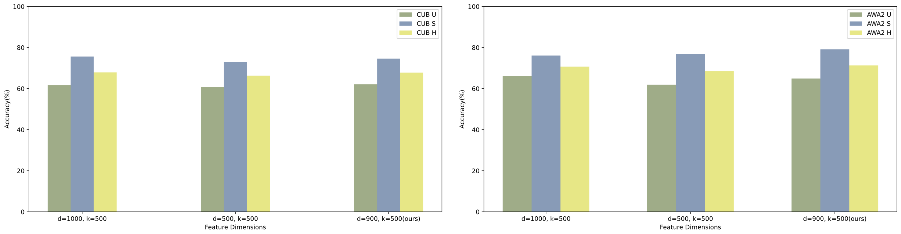


Fig. 4: Impact of changes in the visual dimensions relative to the semantic dimensions of the dataset on model performance.

It can be observed from Fig. 4 that the performance of the zero-shot experiments on the CUB and AWA2 datasets is less affected when the semantic dimension k is fixed and the visualization dimension d increases or decreases relative to the baseline value of 900. On both datasets, increasing or decreasing the dimensionality of the visual features resulted in a slight improvement or decrease, but overall, the impact on model performance is not very large.

With one parameter fixed, we also experimented with several sets of parameters to observe how the performance of the algorithm changes with the parameters, as shown in Tab. 3, where the lengths of the latent semantic vector and visual vector are set to D_s and D_v, respectively. It can be seen that these two parameters have a very slight impact on the model performance.

Table 3: Performance comparison of the number of synthesized features on visual features

Model	**CUB**			**AWA2**		
	U	S	H	U	S	H
$D_v = 1000, D_s = 500$	61.7	**75.6**	**67.9**	**66.1**	76.1	70.7
$D_v = 500, D_s = 500$	60.8	72.9	66.3	61.9	76.8	68.5
$D_v = 900, D_s = 500$	**62.1**	74.6	67.8	64.9	**79.1**	**71.3**
$D_v = 900, D_s = 1000$	**62.6**	72.4	67.1	63.8	78.7	70.5
$D_v = 900, D_s = 700$	61.7	**74.9**	67.7	**65.2**	77.5	70.8
$D_v = 900, D_s = 500$	62.1	74.6	**67.8**	64.9	**79.1**	**71.3**

5 T-SNE Visualization Results Comparison

In this section, as shown in Fig. 5 and Fig. 6 below, we did four sets of T-SNE visualizations. The former group is a T-SNE visualization of the entire dataset features after extracting the data features using ResNet50 and ViT-B on the CUB dataset, respectively. The latter two groups are T-SNE visualization after visual feature extraction of our dataset with VAEGAN and our model, respectively.

To reduce the dimensionality of our extracted features to 3 dimensions, we have set the dimension of the embedding space(n_components) to 3. Additionally, we have set the random_state to 42, the initialization method of the embedding space to PCA embedding, the perplexity to 50, and the number of iterations(n_iter) of the optimization process to 2000.

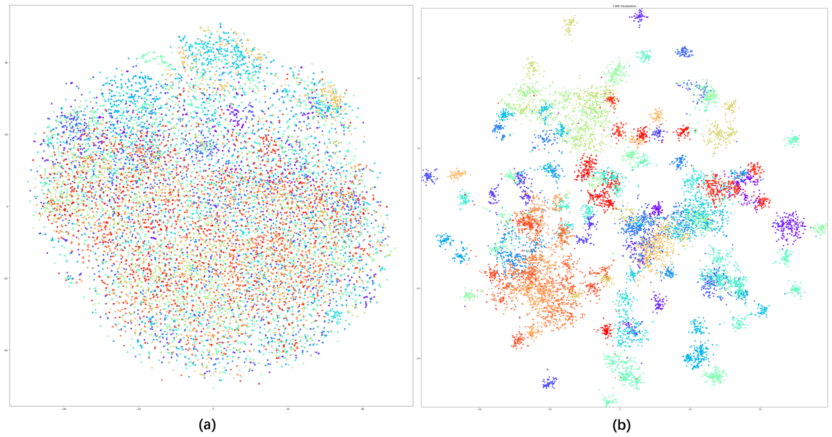
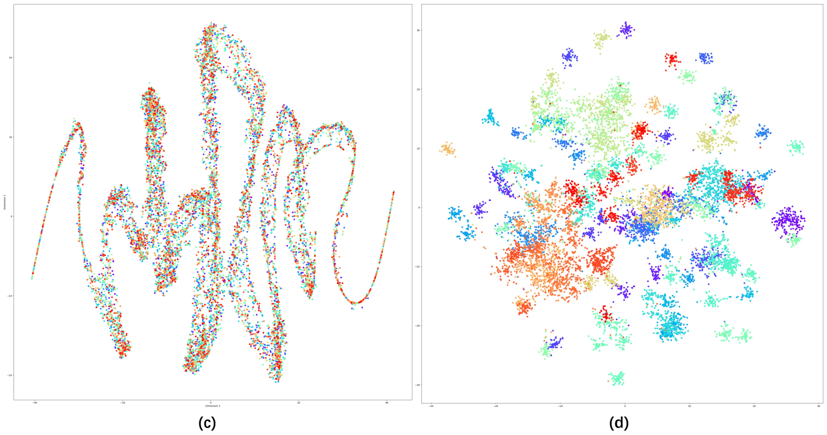

(a) (b)

Fig. 5: T-SNE visualization on CUB dataset. Fig(a). The visual feature extracted by ResNet50 backbone. Fig(b). The visual feature extracted by ViT-B backbone.

From the first set of results, ViT-B is effective in visual feature extraction thanks to ResNet50. After the former extracts the original visual features, the spatial distribution of visible and invisible classes is clearer, which lays the foundation for effective feature classification based on Mahalanobis distance in the next step. At the same time, it can be seen that ViT-B reduces the relative geometric relationship of entanglement between different classes, improves the performance and portability of the model, and plays a positive role in mapping from seen classes to unseen classes.

The second set of data is visualized by T-SNE to further explain why our method is significantly improved over the previous one. Our method significantly improves the visual features of both seen and unseen classes, making the figure more capable of enhancing visual features and reducing classification complexity, thus increasing the recognition and transferability of the model.



(c)

(d)

Fig. 6: T-SNE visualization on CUB dataset. Fig(c). The visual feature extracted by VAEGAN. Fig(d). The visual feature extracted by our model.