Data-Efficient Language Shaped Few-shot Image Classification

Zhenwen Liang
KAUST
zhenwen.liang@kaust.edu.sa

Xiangliang Zhang ✉
University of Notre Dame, KAUST
xzhang33@nd.edu
xiangliang.zhang@kaust.edu.sa

Abstract

Many existing works have demonstrated that language is a helpful guider for image understanding by neural networks. We focus on a language-shaped learning problem in a few-shot setting, i.e., using language to improve few-shot image classification when language descriptions are only available during training. We propose a data-efficient method that can make the best usage of the few-shot images and the language available only in training. Experimental results on dataset ShapeWorld and Birds show that our method outperforms other state-of-the-art baselines in language-shaped few-shot learning area, especially when training data is more severely limited. Therefore, we call our approach data-efficient language-shaped learning (DF-LSL).

1 Introduction

Few-shot image classification is well aligned with the practical application scenarios where labeled images are costly to acquire. Building effective few-shot image classifiers is challenged by the difficulty to improve the classifier generalizability given few labeled images in each class. Recent efforts have been dedicated to design metric-based approaches (Snell et al., 2017; Sung et al., 2018), augmentation-based methods, (Mehrotra and Dukkipati, 2017; Wang et al., 2018; Xian et al., 2019), and meta-learning methods (Finn et al., 2017, 2018; Sun et al., 2019).

Another stream of work introduces language information to guide the image classification (Andreas et al., 2018; Mu et al., 2020), because nature languages are a kind of reflection of the world and convey rich information and knowledge for understanding the visual patterns. In this paper, we target on addressing the few-shot image classification by efficiently using the language description as a guide during the training of image classification model. Different from (Elhoseiny et al., 2013) and (Andreas et al., 2018), we aim to deal with a more challenging scenario where we have no language information during testing period. All language descriptions are only available during training. Our study shares the same setting with only one recent work in (Mu et al., 2020). We design a different model that can make the best usage of the few images available for each class and the language information in the training process. The key difference is two-fold. First, all few-shot images are asked to participate in the language-shaping stage to enhance the guidance on image understanding. Second, extra supervision tasks are introduced to enlarge the communication channel between language description and images.

Our proposed approach, named data-efficient language shaped learning (DF-LSL), is shown in extensive evaluation to perform better than state-of-the-art baselines. Comparing to the strongest baseline LSL in (Mu et al., 2020), our proposed method DF-LSL has 1.8% higher accuracy on the CUB benchmark dataset, and 0.5%-1.8% higher accuracy on the ShapeWorld benchmark dataset.

2 Related Work

Image Few-Shot Learning. Due to the difficulty to acquire a large number of labeled images, few-shot classification draws increasing attention in machine learning, which can be roughly categorized into three different approaches. The first one is metric-based approaches, which learn a model to represent images with latent features, such as (Snell et al., 2017; Sung et al., 2018). Secondly, some approaches (Mehrotra and Dukkipati, 2017; Wang et al., 2018; Xian et al., 2019) use augmentation-based method to generate more useful samples of features, feeding the model with more knowledge. The last kind of approaches is called meta-based methods (Finn et al., 2017, 2018; Sun et al., 2019), which is motivated by meta-learning, using an inner-loop and outer-loop to achieve fast adaption on new tasks.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4680–4686
November 7–11, 2021. ©2021 Association for Computational Linguistics

Learning from Other Domains: Zero-shot Learning. Zero-shot learning is a kind of problem setup where a model needs to predict the class of samples, without giving samples in those classes in training phase. Usually, a zero-shot model has to utilize some side information from other domains, to learn about those zero-shot classes. There are approaches using attribute information to give descriptions about the unseen classes (Lampert et al., 2009; Atzmon and Chechik, 2018). There are also approaches (Elhoseiny et al., 2013; Srivastava et al., 2018) try to transfer language information into zero-shot image classification and achieved good results.

Language Related Learning. Nature language shapes the way we know about the world, and thus has been introduced to assistant various tasks. For example, language descriptions are generated to explain the decisions of neural network (Belle, 2017) for improving the explainability of deep learning methods. Language can also provide guidance during learning. This idea is applied on many different learners such as monte-carlo framework (Branavan et al., 2012) and reinforcement learning (Harrison et al., 2018). Moreover, (Andreas et al., 2018; Mu et al., 2020) try to use language information to guide the image classification, which is also our study purpose.

3 Problem Statement and Preliminaries

3.1 Problem Statement

In the problem of few-shot image classification shaped with language description, a model is expected to learn to classify images based on small training sets, which is also called the support set of labels. Following the common few-shot learning setting, the model is trained through a set of N-way K-shot tasks. In each task, we have a support set with N classes, and each class contains K support samples $\{x^s_{n,1}, ..., x^s_{n,K}\}$, where s denotes the support set and n denotes the class index. The trained model is applied to predict the label of a test set (called query set), which has M query images with the ground-truth labels $\{(x^q_1, y^q_1), ...(x^q_M, y^q_M)\}$, where q denotes the query image and y is the ground truth label represented by a one-hot vector in N dimensions. When running on each task, the prediction loss on the query set is often defined by comparing the predicted label $\hat{y}^q$ with the ground-truth label y^q.

Besides the image data, we have also D language descriptions for every class n, which can be denoted as $W_n = \{w^n_1, ..., w^n_D\}$. Language information is only available during training. The learning target is to make the model be able to predict correctly the label of query set by using only K-shot images in the support set, with the guidance of available language descriptions in training.

3.2 Language-Shaped Learning

Language-shaped learning (LSL) method proposed in (Mu et al., 2020) share the same problem setting with our approach. LSL borrows the idea from a metric-based method (Snell et al., 2017), using a backbone network to extract class prototypes from support images, then making prediction by running similarity function S between prototypes and query images. Let f_θ be the feature extraction network with parameters θ. The prototype for class n is:

$$z_n = \frac{1}{K} \sum_{k=1}^{K} f_\theta(x^s_{n,k}). \tag{1}$$

Following (Snell et al., 2017), z_n is used to classify the query image by $p(\hat{y}^q = n|x^q) \propto S(z_n, f_\theta(x^q))$. In addition, z_n is used to generate the language description of class n as an auxiliary task. During the training of LSL, a classification loss is minimized jointly with a language loss.

$$L^{lan}(\theta, \phi) = -\sum_{n=1}^{N} \sum_{d=1}^{D} \log g_\phi(w^n_d \mid z_n) \tag{2}$$

$$L^{image}(\theta) = -\sum_{m=1}^{M} \log p(\hat{y}^q_m = n \mid x^q_m) \tag{3}$$

where g_ϕ is a language model to generate language descriptions.

4 The Proposed Method

Our proposed DF-LSL method has two key differences from LSL, as shown in Figure 1. The details are discussed next.

4.1 Multiple Prototypes

In LSL, language descriptions are generated by the averaged prototypes z_n. However, the conversation between the language descriptions and images should be open to all support images, rather than only the "averaged" image. Same for the classification of query images, all support images should

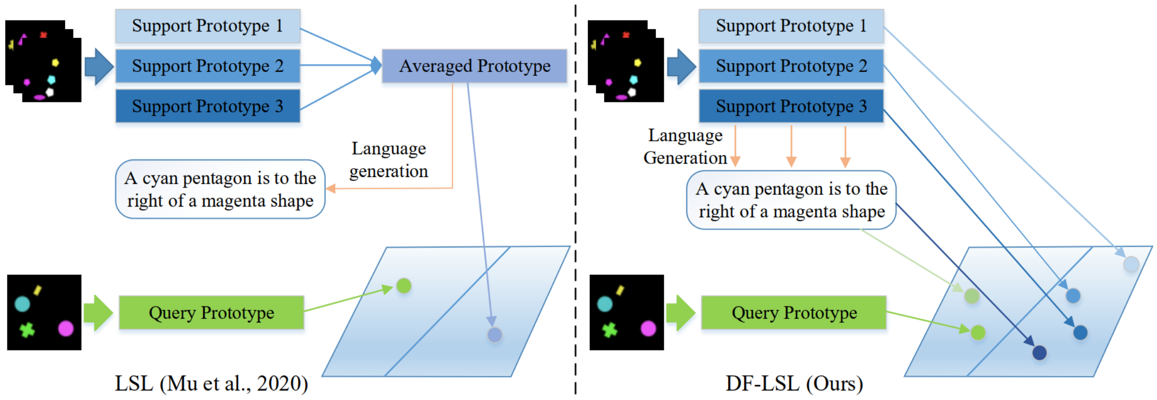

Figure 1: The overall framework of our proposed DF-LSL and the LSL framework in (Mu et al., 2020). The key differences are: 1) we let every support image in one class contribute to the image classification and language generation, rather than using the averaged prototype of the class; and 2) we introduce additional supervising tasks, using language descriptions to classify support and query images (the green and blue arrows pointing from the language description to the classification space).

be allowed to participate the classifier hyperplane construction. Therefore, we create a "prototype" for each image in the support set. Mathematically, Eq. (1) will be redefined as:

$$z_n^k = f_\theta(x_{n,k}^s) \tag{4}$$

where z_n^k is the k-th prototype of class n. Then the language loss and classification loss of our approach become:

$$L^{lan}(\theta, \phi) = -\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{d=1}^{D} \log g_\phi(w_d^n \mid z_n^k) \tag{5}$$

$$L_1^{image}(\theta) = -\sum_{m=1}^{M}\sum_{k=1}^{K} \log p(\hat{y}_m^q = n \mid x_m^q) \tag{6}$$

where the prediction $p(\hat{y}^q = n|x^q) \propto S(z_n^k, f_\theta(x^q))$.

4.2 Extra Supervision Tasks

To enlarge the communication channel between language description and images, we introduce extra supervision tasks to further take advantages of the language information. Suppose that the language description can be mapped to a representation vector $h_\gamma(w_d^n)$, e.g., by GRU. We use $h_\gamma(w_d^n)$ to classify the support and query images. In this way, the visual patterns and the language information are aligned in double directions, rather than the single direction in LSL. The corresponding introduced loss functions are:

$$L_2^{image}(\theta, \gamma) = -\sum_{m=1}^{M} \log p(\hat{y}_m^q = n \mid x_m^q, w_d^n) \tag{7}$$

where $p(\hat{y}_m^q = n \mid x_m^q, w_d^n) \propto S(h_\gamma(w_d^n), f_\theta(x_m^q))$.

$$L_3^{image}(\theta, \gamma) = -\sum_{k=1}^{K} \log p(\hat{y}_k^s = n \mid x_k^s, w_d^n) \tag{8}$$

where $p(\hat{y}_k^s = n \mid x_k^s, w_d^n) \propto S(h_\gamma(w_d^n), f_\theta(x_k^s))$.

4.3 Training Criteria

To sum up, our proposed DF-LSL has three different kinds of image classification loss and one language loss, defined in Eq. (5-8). The overall loss function used in training is the summation of all these four loss functions:

$$L_{final} = L_1^{image} + L_2^{image} + L_3^{image} + \lambda L_{lan} \tag{9}$$

where λ controls the weight of language loss.

5 Experiments

In general, we implement by the same settings as LSL (Mu et al., 2020) for the sake of fair comparison. For predictions, we average the probabilities across k prototypes. We use ShapeWorld (Kuhnle and Copestake, 2017) and CUB (Wah et al., 2011) dataset to evaluate our method. The details of experimental settings and model descriptions can be found in our source code[1].

5.1 Datasets

ShapeWorld. ShapeWorld (Kuhnle and Copestake, 2017) dataset is firstly proposed in visual question answering field. Each image has several

[1] https://github.com/derderking/DF-LSL

Dataset	ShapeWorld			CUB
Backbone	VGG16	Conv4	ResNet-18	Conv4
Meta	60.59 ± 1.07	50.91 ± 1.10	58.73 ± 1.08	73.05 ± 0.72
L3	66.60 ± 1.18	62.28 ± 1.09	67.90 ± 1.07	66.98 ± 0.82
LSL	67.29 ± 1.03	63.25 ± 1.06	68.76 ± 1.02	73.52 ± 0.79
DF-LSL (Ours)	$\mathbf{69.06 \pm 1.07}$	$\mathbf{64.55 \pm 1.04}$	$\mathbf{69.25 \pm 1.01}$	$\mathbf{75.37 \pm 0.76}$

Table 1: Test accuracy (%) with 95% confidence interval of different visual backbones on ShapeWorld dataset, and Conv4 on CUB dataset.

non-overlapping shapes, and the language descriptions are related to the special information between two shapes. Following the same setting in LSL, we set K, the number of image samples per class to 4, and a language description is associated with that class (a universal description for those four images). Query set has positive and negative samples, where positive samples can match that language description while the negative ones cannot. Our entire dataset contains 9000 training tasks, 1000 validation tasks and 4000 tasks. No augmentation method was employed on this dataset, because the images in ShapeWorld dataset are classified accroding to their colors, shapes and positions. Cropping, flipping and color jittering will be harmful to those properties. Due to the special case of binary classification, we simply apply dot-product operation to be our similarity metric S, then use a sigmoid function to scale the similarity value from 0 to 1, which will be a suitable representation for probability.

Caltech-UCSD Birds. Images in ShapeWorld dataset are synthetic by computers and only contain several basic shapes and a black background. In real world scenarios, we have more complicated shapes and more noisy background information. Moreover, we only have one language description per class in ShapeWorld, which is not enough for us to analysis the influence of the amount of language information. Therefore, we perform our experiments on another challenging dataset Caltech-UCSD Birds (CUB) (Wah et al., 2011), which contains 200 bird species and their images. All language descriptions are from (Reed et al., 2016), and describe each bird image with ten different sentences. For the purpose of pre-processing and augmentation, we apply pixel normalization, color jittering, horizontal flipping and random cropping. We use a matrix $\mathbf{W}$ as the similarity function, which means $S(a, b)$ can be calculated by $a^T W b$.

5.2 Network Architecture

Image Prototype Model. Image prototype model extracts prototypes from images. The first model is frozen ImageNet-pretrained VGG-16 (Simonyan and Zisserman, 2015) with two fully-connected layers and one ReLU activation. The second model has a simple structure with 4 convolutional blocks (Conv4) (Chen et al., 2019). Another image prototype model is a deeper approach called ResNet-18 (He et al., 2016).

Language Prototype Model. Model h_γ maps language descriptions into prototype space. In our approach, we take the last hidden states of a gated recurrent unit (GRU) (Cho et al., 2014) as language descriptions' prototypes. Empirically, we set the dimension of hidden state to 512.

Language Generation Model. To generate language descriptions by an image prototype, we need a model g_ϕ, which is also a 512-dimensional GRU. Teacher forcing was employed during training, making the model coverage faster.

5.3 Experiment Results

The evaluation metric used in our experiments is accuracy among all test tasks, with a 95% confidence interval and K = 4. Baselines contain **Meta, L3** and **LSL**. **Meta** (Snell et al., 2017) is the prototypical network without the usage of language information. **L3** (Andreas et al., 2018) is the abbreviation of learning with latent language, which applies a decoder to generate language description, then uses generated language description to help image classification. However, the generated language description could have mistakes and be harmful to the classification result. **LSL** (Mu et al., 2020) is the state-of-the-art language shaped few-shot learning model, which was introduced in Section 3. Table 1 shows that our proposed DF-LSL outperforms all the baselines among three different visual

K-shot	LSL	DF-LSL (Ours)
K = 2	67.05 ± 0.83	$\mathbf{69.34} \pm \mathbf{0.86}$
K = 4	73.52 ± 0.79	$\mathbf{75.37} \pm \mathbf{0.76}$
K = 8	78.14 ± 0.69	$\mathbf{79.30} \pm \mathbf{0.63}$
K = 16	79.60 ± 0.65	$\mathbf{80.18} \pm \mathbf{0.64}$

Table 2: Test accuracy (%) with 95% confidence interval of LSL and DF-LSL at different K on CUB dataset. The smaller K is (the fewer images are available in training), the larger gain DF-LSL has over LSL.

backbones f_θ on ShapeWorld dataset and Conv4 backbone on CUB dataset.

To demonstrate our proposed DF-LSL is able to use information more efficiently, we conduct experiments with different K-shot settings on CUB dataset. As Table 2 shows, DF-LSL surpasses LSL among all pairs. Furthermore, it is worth noting that as K increases from 2, the performance gap between LSL and DF-LSL decreases, which indicates that data-efficient training method is more effective when training data is more severely limited.

5.4 Ablation Study

Our ablation study experimental results are shown in Table 3. All ablated models are trained and evaluated in the same way and we compare them between LSL and DF-LSL. **A1** is the model which does not apply multiple prototypes for image classification task, while **A2** does not have multiple prototypes for language generation task. The comparison between the results of **A1** and **A2** shows that both multiple prototypes based classification and generation can contribute to the performance improvement.

B1 and **B2** are designed for examining the contribution of two new supervising tasks with loss in Eq. (7) and (8). Specifically, **B1** is trained without the loss of Eq. (7), which is the task of classifying query images by using language descriptions. Similarly, **B2** does not use language descriptions to predict the label of support images (without Eq. (8)). As shown in Table 3, DF-LSL outperforms both B1 and B2. It is interesting that B1 is better than B2 on ShapeWorld dataset, but worse on CUB dataset. This is because the number of support images is more than that of query images on CUB dataset, while tasks of ShapeWorld dataset contain more query images.

For encouraging the future research, we also report two failed attempts, which are model **C1** and **C2** in Table 3. Similar to language generation task, which uses image prototypes generate language descriptions, we create a new task of using image prototypes to generate the original images in C1. This idea is inspired by back-translation technique that is commonly used in Neural Machine Translation. The potential reason of C1's failure is that retrieving images from prototypes is not helpful for classification. C2 contains a new binary classifier that uses image prototypes to classify text descriptions, where the negative descriptions are sampled from other classes. Since we cannot determine if descriptions for another class are true or false for current class on ShapeWorld dataset, we skip this evaluation. On ShapeWorld, the accuracy of C2 is not as good as DF-LSL, because C2 provides a wrong way of using image prototypes, which are in fact specially designed for image classification task. To sum up, adding new tasks during training always takes risks. We have to carefully plug them into our model, and fine-tune many parameters such as the weights of new loss. Therefore, although we fail to use C1 and C2, we will keep exploring in our future work to find other useful addition tasks.

Model	ShapeWorld	CUB
LSL	67.29 ± 1.07	73.05 ± 0.72
A1	68.39 ± 1.02	73.80 ± 0.69
A2	68.20 ± 1.02	73.89 ± 0.69
B1	68.86 ± 1.01	74.38 ± 0.71
B2	68.63 ± 1.02	74.57 ± 0.72
C1	67.01 ± 1.03	71.08 ± 0.70
C2	67.25 ± 1.06	–
DF-LSL	$\mathbf{69.06} \pm \mathbf{1.07}$	$\mathbf{75.37} \pm \mathbf{0.76}$

Table 3: Test accuracy of different ablated models. We use VGG16 backbone on ShapeWorld dataset and Conv4 backbone on CUB dataset.

6 Conclusion

This paper proposes a data-efficient language shaped learning (DF-LSL) model, which aims to improve the few-shot image classification model by language information. Experiment results show that the overall performance of our approach surpasses all other baselines on two benchmark datasets. This verifies the effectiveness of the proposed two key innovations in DF-LSL.

References

Jacob Andreas, Dan Klein, and Sergey Levine. 2018. Learning with latent language. In *NAACL-HLT*, pages 2166–2179.

Yuval Atzmon and Gal Chechik. 2018. Probabilistic AND-OR attribute grouping for zero-shot learning. In *UAI*, pages 382–392.

Vaishak Belle. 2017. Logic meets probability: Towards explainable ai systems for uncertain worlds. In *IJCAI*, pages 5116–5120.

SRK Branavan, David Silver, and Regina Barzilay. 2012. Learning to win by reading manuals in a monte-carlo framework. *Journal of Artificial Intelligence Research*, 43:661–704.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. In *ICLR*.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734.

Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. 2013. Write a classifier: Zero-shot learning using purely textual descriptions. In *ICCV*, pages 2584–2591.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML 2017, Sydney, NSW, Australia, 6-11 August*, volume 70, pages 1126–1135.

Chelsea Finn, Kelvin Xu, and Sergey Levine. 2018. Probabilistic model-agnostic meta-learning. In *NIPS/NeurIPS*, pages 9516–9527.

Brent Harrison, Upol Ehsan, and Mark O. Riedl. 2018. Guiding reinforcement learning exploration using natural language. In *AAMAS*, pages 1956–1958.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*, pages 770–778.

Alexander Kuhnle and Ann Copestake. 2017. Shapeworld-a new test methodology for multimodal language understanding. *arXiv preprint arXiv:1704.04517*.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958.

Akshay Mehrotra and Ambedkar Dukkipati. 2017. Generative adversarial residual pairwise networks for one shot learning. *arXiv preprint arXiv:1703.08033*.

Jesse Mu, Percy Liang, and Noah D. Goodman. 2020. Shaping visual representations with language for few-shot classification. In *ACL 2020, Online, July 5-10*, pages 4823–4830.

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

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NIPS/NeurIPS*, pages 4077–4087.

Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2018. Zero-shot learning of classifiers from natural language quantification. In *ACL (Volume 1: Long Papers)*, pages 306–316.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. 2019. Meta-transfer learning for few-shot learning. In *CVPR*, pages 403–412.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*, pages 1199–1208.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset.

Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. 2018. Low-shot learning from imaginary data. In *CVPR*, pages 7278–7286.

Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. 2019. f-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, pages 10275–10284.

A Appendix

ShapeWorld. ShapeWorld dataset is firstly proposed in visual question answering field. Each image has several non-overlapping shapes, and the language descriptions are related to the special information between two shapes. Following the same setting in LSL, we set K, the number of image samples per class to 4, and a language description is associated with that class (a universal description for those four images). Query set has positive and negative samples, where positive samples can match that language description while the negative ones cannot. Our entire dataset contains 9000 training tasks, 1000 validation tasks and 4000 tasks. No augmentation method was employed on this dataset, because the images in ShapeWorld dataset are classified accroding to their colors, shapes and positions. Cropping, flipping and color jittering will be harmful to those properties.

Due to the special case of binary classification, we simply apply dot-product operation to be our similarity metric S, then use a sigmoid function to scale the similarity value from 0 to 1. We train for 80 epochs with Adam optimizer and the learning rate is set to 0.001. The batch size during training is 64, and the weight of language generation loss is set to 20.

Caltech-UCSD Birds. Images in ShapeWorld dataset are synthetic by computers and only contain several basic shapes and a black background. In real world scenarios, we have more complicated shapes and more noisy background information. Moreover, we only have one language description per class in ShapeWorld, which is not enough for us to analysis the influence of the amount of language information. Therefore, we perform our experiments on another challenging dataset Caltech-UCSD Birds (CUB), which contains 200 bird species and their images. Each bird image is described with ten different sentences. For the purpose of pre-processing and augmentation, we apply pixel normalization, color jittering, horizontal flipping and random cropping.

Without loss of fairness during comparison, we follow the same settings described in LSL. We have 5 classes in each task (5-way), 16 images in query set. The visual backbone of all the experiments on CUB dataset is set to Conv4, which has 3×3 convolution kernels, batch normalization layer, ReLU activation and max-pooling operation. In the end, Conv4 will transform an 84×84 image into a feature map with 1600 hidden dimensions. We use a 1600×1600 matrix **W** as the similarity function, which means $S(a, b)$ can be calculated by $a^T W b$. We train the model with Adam optimizer and a learning rate of 0.001. The weight of language generation loss λ in L_{final} is set to 5. The number of language descriptions per class is set to 20, and the number of query images per task is 16. The only different parameter in this paper is the number of support images per class, where LSL set it to 1 (1-shot). However, our proposed multiple prototypes setting requires a larger K than 1 where K stands for K-shot classification. Therefore, we apply 5-way 4-shot classification setting and re-produce the accuracy measurements for all baselines.