EXPLOITING CATEGORY NAMES FOR FEW-SHOT CLASSIFICATION WITH VISION-LANGUAGE MODELS

Taihong Xiao[1], **Zirui Wang**[3], **Liangliang Cao**[3], **Jiahui Yu**[2], **Shengyang Dai**[2], **Ming-Hsuan Yang**[1,2]
{txiao3, mhyang}@ucmerced.edu {jiahuiyu, sydai}@google.com {ziruiw, llcao}@apple.com
[1] University of California, Merced [2] Google [3] Apple

ABSTRACT

Vision-language foundation models pretrained on large-scale data provide a powerful tool for many visual understanding tasks. Notably, many vision-language models build two encoders (visual and textual) that can map two modalities into the same embedding space. As a result, the learned representations achieve good zero-shot performance on tasks like image classification. However, when there are only a few examples per category, the potential of large vision-language models is often underperformed, mainly due to the gap between a large number of parameters and a relatively small amount of training data. This paper shows that we can significantly improve the performance of few-shot classification by using the category names to initialize the classification head. With the proposed category name initialization method, our model obtains the state-of-the-art performance on a number of few-shot image classification benchmarks (e.g., 87.37% on ImageNet and 96.08% on Stanford Cars, both using five-shot learning).

1 INTRODUCTION

In recent years, large vision-language models have opened doors to many new applications and provided new thoughts to existing problems. The advantages of large vision-language models are blessed by learning from largely available images with surrounding texts, as well as exploring the capacity of transformer network (Dosovitskiy et al., 2021) to model web-scale image-text data. Radford et al. first proposed CLIP for vision-language modeling, which was followed by numerous works, including ALIGN (Jia et al., 2021), LiT (Zhai et al., 2022), Flamingo (Alayrac et al., 2022), Florence (Yuan et al., 2021), CoCa (Yu et al., 2022), etc. The development of vision-language models provides novel perspectives of thinking of few-example learning.

This paper considers the problem of few-shot classification in the new light of large vision-language models. Researchers have found that models pretrained from ImageNet can be easily transferred by finetuning on a new classification task (Huh et al., 2016). Similarly, we can take the vision encoder from the pretrained vision-language model and finetune it with a few examples. Since state-of-the-art vision-language models were pretrained on billions of web images and texts, such finetuning often outperforms the models trained on ImageNet with better robustness and generalization capabilities.

Despite the capability of the text branch in pretrained vision-language models, it is not optimally utilized when directly fine-tuning the vision component for downstream image classification tasks. Additionally, the large size of these models can lead to overfitting when trained on limited data. In addition to the above approach, we exploit another source of information in vision-language models that traditional models have overlooked. Such new information comes from the category names in downstream image classification tasks. Because vision-language models can generate powerful representations for both images and texts, we will show that by utilizing semantic category names, vision-language models can be transferred better with few examples in downstream tasks.

As summarized in Figure 1, this paper explores several scenarios: (1) randomly initializing a classification head; (2) initializing a classification head with category names; (3) initializing a classification head with other heuristics such as class digits or even non-English category names. Note that (1) corresponds to the scenario when we only know the category ID (e.g., class 0, class 1, ..., class N) without knowing the meaning of each category. However, (2) implicitly parses the information

from category names such as "tench" and "goldfish". These label names could be processed by the pretrained language model to provide a better initialization for the model adaption. As a comparison to (2), (3) provides different types of category name information. The main difference between scenario (1) and the others is that (1) does not utilize text/language information from the categories. In scenario (1), the backbone network is initialized from the pretrained model weights, and the classification head is randomly initialized. We set (1) to be our baseline as it is the most common model adaptation method. For the other scenarios, we leverage the pretrained language model to parse the text information in the provided categories. Specifically, we pair all category names with prompts and extract the average text embedding as the weight to initialize the classification head. The second scenario is called *category name initialization* (CNI), and it has achieved the best performance among all these scenarios when finetuning using one-shot ImageNet data, as shown in Figure 1.

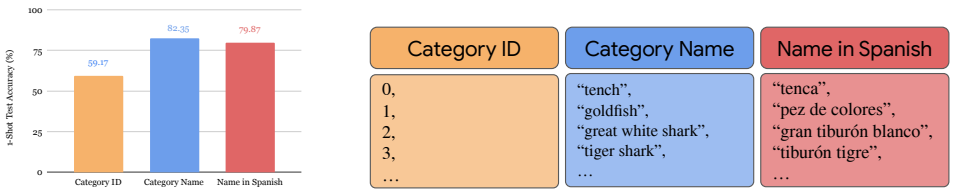


Figure 1: Comparing one-shot classification accuracy on ImageNet using different category information.

2 Approach

In this paper, we take a recent state-of-the-art vision language model – CoCa to illustrate our approach. Unlike other recent vision-language models, CoCa adopts an encoder-decoder model architecture to learn generic vision and multimodal representations. As shown in Figure 2 (a), CoCa encodes images to latent representation via an encoder network (e.g., vision transformer (ViT) (Dosovitskiy et al., 2021)). An image pooler is appended after the image encoder to customize the image representations for different tasks and training objectives. On the other hand, CoCa uses a unimodal decoder to extract text-only embeddings and cascades multimodal decoder layers cross-attending to image embeddings to learn multimodal image-text representations. Here, we focus on reusing these two components to initialize for few-shot learning.

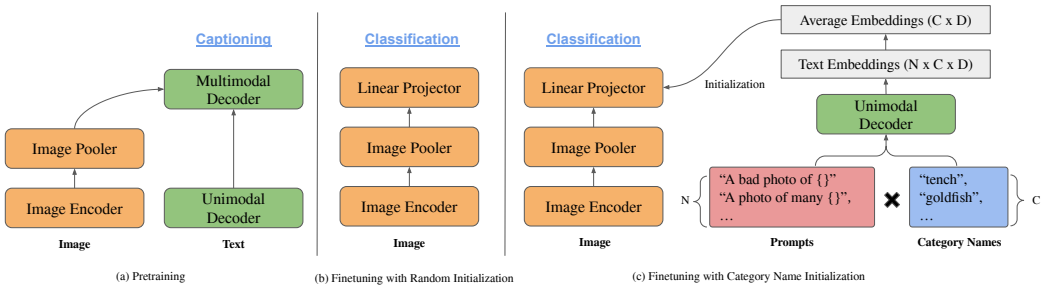


Figure 2: An overview of the CoCa pretraining and finetuning. (a) The pretraining of CoCa relies on mapping image and text pairs into the same space for embedding alignment, where the image and text embeddings are extracted through an image encoder and a unimodal text decoder, respectively. The image pooler is used to customize the image embedding for different tasks. (b) We append a randomly initialized linear projector to the image pooler and initialize the image encoder from pretrained weights. (c) We construct text sequences by pairing all C category names with N different prompts. Via the pretrained unimodal decoder, we can compute the text embeddings for all text sequences (with a total number of $N \times C$), each of them is D-dimensional vector. The normalized average embeddings can be used for initializing the weight in the linear projector.

Random initialization. One straightforward model adaption approach is to add a randomly initialized linear projector upon the pretrained model and selectively finetune the model (all or part of

the layers), as depicted in Figure 2 (b). Following the approach used by CLIP (Radford et al., 2021) and CoCa (Yu et al., 2022), we first use an image pooler to obtain the aggregated image embedding $H \in \mathcal{R}^D$ and then apply a linear projector to get the prediction $Y \in \mathcal{R}^C$,

$$Y = \text{softmax}(WH + b), \tag{1}$$

where $W \in \mathcal{R}^{C \times D}$ and $b \in \mathcal{R}^C$ are learnable weight and bias of the linear projector. Here W and b are randomly initialized, while the image encoder and generative image pooler are initialized from the pretrained weights.

Category name initialization. We argue that the above random initialization ignores the potential of the language model for model adaptation. In contrast, we propose the category name initialization to maximize the capacity of the pretrained unimodal decoder. First, we pair all category names (whose total number is C) with N different prompts as the text inputs. For example, pairing the category name "tench" with a prompt "A bad photo of {}" gives us a text sequence "A bad photo of tench". Next, we compute the text embeddings for all these $N \times C$ text sequences via the unimodal decoder. As the text embedding for each text input is a D-dimensional vector, we can obtain a text embedding tensor with a shape of $N \times C \times D$. We then compute the average over different prompts and perform the normalization to obtain the average embeddings of shape $C \times D$. Unlike random initialization, we initialize the weight W by the average embeddings and bias b by a zero vector in the linear projector. To enable zero-shot inference of the category name initialized model, we use the pretrained model weights to initialize the image encoder and the image pooler.

3 EXPERIMENTS

We conduct our finetuning experiments on several widely-used image classification datasets, including ImageNet (Deng et al., 2009), ImageNet-V2 (Recht et al., 2019), ImageNet-R (Hendrycks et al., 2021a), ImageNet-A (Hendrycks et al., 2021b), ImageNet-Sketch (Wang et al., 2019), Cifar100 (Krizhevsky, 2009), Oxford Flowers (Nilsback & Zisserman, 2008), Stanford Cars (Krause et al., 2013), Country-211 (Radford et al., 2021), Food-101 (Bossard et al., 2014), FGVC Aircraft (Maji et al., 2013), EuroSAT (Helber et al., 2019), and Oxford-IIIT Pets (Parkhi et al., 2012). For different few-shot settings, we randomly sample a particular portion of data from these datasets. For example, one-shot ImageNet means that we only select one image from the ImageNet training data for each category. The evaluation is still performed using the whole testing set. Following the existing benchmark (Li et al., 2022), we use the same text prompts[1] for evaluating all methods.

We use the pretrained CoCa model and apply category name initialization. Following the standard benchmark evaluation (Li et al., 2022), we then compare our method against the previous works, including MAE (He et al., 2022), CLIP (Radford et al., 2021), CLIP+CoOp (Zhou et al., 2022a), WiSE-FT (Wortsman et al., 2022), and Flamingo-3B (Alayrac et al., 2022) on ImageNet and its variants, including ImageNet-V2 (Recht et al., 2019), ImageNet-R (Hendrycks et al., 2021a), ImageNet-A (Hendrycks et al., 2021b) and ImageNet-Sketch (Wang et al., 2019). As shown in Table 1, the CoCa-2B model has achieved state-of-the-art few-shot classification results on all these benchmarks. Surprisingly, the one-shot performance of CoCa-base is even better than the performance of some recent methods finetuned on the whole dataset.

In addition to ImageNet and variants, we show that our method can achieve state-of-the-art few-shot performance on other image classification benchmarks, including Cifar100 (Krizhevsky, 2009), Oxford Flowers (Nilsback & Zisserman, 2008) and Stanford Cars (Krause et al., 2013), Country-211 (Radford et al., 2021), Food-101 (Bossard et al., 2014), FGVC Aircraft (Maji et al., 2013), EuroSAT (Helber et al., 2019), and Oxford-IIIT Pets (Parkhi et al., 2012). As a comparison, we choose MAE (He et al., 2022), CAE (Chen et al., 2022), MoCo-v3 (Chen et al., 2021), DeiT (Touvron et al., 2021), ViT (Dosovitskiy et al., 2021) and CLIP (Radford et al., 2021). We can observe from Table 2 that our CoCa-2B model outperforms many other approaches even with less training data. Such good performance benefits a lot from the category name initialization, as it provides a good starting point so that the model could achieve better performance using a few examples.

[1] https://github.com/Computer-Vision-in-the-Wild/Elevater_Toolkit_IC/blob/main/vision_benchmark/datasets/prompts.py

Table 1: Few-shot results on ImageNet and its variants. We use IN as the abbreviation for ImageNet, and CNI for category name initialization. The second column means how much training data per class is used for finetuning. 0 shot means the pretrained vision-language model is directly evaluated without finetuning. Full means the entire training set has been used. All the numbers under the last five columns denote the top-1 test accuracy.

Model	Shot	IN	IN-V2	IN-R	IN-A	IN-Sketch
MAE	full	-	-	66.50	76.70	50.90
CLIP (ViT-B/16)	0	68.40	62.60	77.60	50.00	48.20
	full	79.90	69.80	70.80	46.40	46.90
CLIP (ViT-L/14)	0	76.20	70.10	88.90	77.2	60.20
	full	85.20	75.80	85.30	76.10	58.70
CLIP+Adapter (ResNet-50)	0	55.50	-	-	-	-
	1	58.10	-	-	-	-
	4	59.50	-	-	-	-
CLIP+CoOp (ViT-B/16)	0	58.18	-	-	-	-
	1	58.00	-	-	-	-
	4	60.01	-	-	-	-
Tip-Adapter-F (ResNet-50)	0	60.33	-	-	-	-
	1	61.32	-	-	-	-
	4	62.52	-	-	-	-
WiSE-FT (ViT-L/14)	full	85.30	76.90	89.80	79.70	63.00
Flamingo-3B	1	70.90	-	-	-	-
	5	72.70	-	-	-	-
Flamingo-80B	1	71.90	-	-	-	-
	5	77.30	-	-	-	-
CoCa-base	0	82.26	76.22	93.16	76.17	71.12
CoCa-base+CNI (Ours)	1	82.35	76.47	93.37	77.00	71.61
	5	83.58	77.23	93.22	77.23	71.35
CoCa-2B	0	86.09	80.39	96.19	89.39	77.12
CoCa-2B+CNI (Ours)	1	86.15	80.57	96.62	90.12	77.49
	5	87.37	81.66	96.41	89.68	77.39

Table 2: Comparing with the state-of-the-art on multiple classification benchmarks. CNI stands for category name initialization. Our model obtains the state-of-the-art few-shot learning performance with less training data than many other works.

Model	Shot	Cifar100	Oxford Flowers	Stanford Cars	Country-211	Food-101	FGVC Aircraft	EuroSAT	Oxford-IIIT Pets
MAE	5	21.20	50.90	6.30	2.80	7.70	7.00	64.60	17.20
	20	43.50	71.90	25.50	4.40	30.40	29.90	74.10	60.00
	full	68.30	72.00	37.20	10.10	65.10	39.10	94.80	81.60
CAE	5	38.30	70.30	8.70	3.50	18.60	14.30	76.70	37.30
	20	55.10	81.20	27.50	5.50	35.70	32.60	89.00	63.30
	full	78.90	81.20	40.40	11.40	67.40	40.80	96.70	79.80
MoCo-v3	5	60.50	79.50	13.40	4.80	36.60	11.80	77.10	76.20
	20	75.50	89.50	49.50	7.60	59.30	38.20	84.80	86.40
	full	85.30	89.50	63.00	13.70	78.00	48.00	95.90	91.40
DeiT	5	61.50	82.70	27.60	4.40	41.90	24.10	62.50	87.80
	20	73.70	92.70	68.80	6.20	61.50	34.10	90.70	91.90
	full	89.60	92.40	83.00	14.10	84.50	59.30	98.20	93.90
ViT	5	75.40	99.20	27.60	6.80	59.00	22.70	70.00	89.60
	20	84.00	99.20	53.90	11.50	81.70	40.50	86.50	92.60
	full	89.80	99.20	67.50	16.60	89.60	47.80	96.00	94.80
CLIP	5	71.10	94.20	73.60	21.70	89.70	36.00	76.70	90.50
	20	75.40	96.8	73.60	25.20	90.60	48.10	86.60	92.30
CoCa-2B	0	77.19	92.04	94.37	42.15	94.79	44.83	49.74	97.88
CoCa-2B+CNI	1	77.89	98.45	95.29	42.44	94.91	58.33	75.06	97.93
	5	78.62	99.25	96.08	44.52	95.50	69.29	85.78	98.12

4 RELATED WORK

Recently, there has been increasing interest in utilizing the vision-language model for visual zero-shot learning, a related problem of few-shot learning. CLIP (Radford et al., 2021) is a pioneering work in large-scale vision-language modeling. Unlike previous works in vision-language representation (Donahue et al., 2015; Vinyals et al., 2015), CLIP collects image-text pairs from the Web, which contains diversified semantics in a weakly supervised fashion. In addition, CLIP is built on large-scale contrastive learning, which maps images and text into the same subspace. Through this, the model can map textual class names with images hence performing image classification in a zero-shot manner. The approach of CLIP was followed by ALIGN (Jia et al., 2021), Flamingo (Alayrac et al., 2022), LiT (Zhai et al., 2022), Florence (Yuan et al., 2021), FLAVA (Singh et al., 2022), SimVLM (Wang et al., 2022) and CoCa (Yu et al., 2022). Among these works, ALIGN, Florence, FLAVA, and LIT are based on contrastive learning. Flamingo chooses to optimize a generative loss with gated cross-attention layers. At last, CoCa integrates contrastive and generative loss into one framework. Although training CoCa seems the most challenging among all these vision-language works, it obtains consistently better results in many tasks.

In the literature, CLIP, LiT, ALIGN, Florence, FLAVA, and CoCa have demonstrated promising results with zero-shot learning. However, the potential of these models for few-shot learning is not well exploited. Li et al. (2022) construct a benchmark and toolkit named Elevater for evaluating the transferability of vision-language models using different training samples. Radford et al. (2021) point out that using few training examples could improve the effectiveness robustness while undermining the relative robustness. Most few-shot learning algorithms are trained exclusively on image data, which ignores the valuable text information that can be used to enhance the learning process. However, Flamingo has emerged as a promising approach for addressing this issue. Flamingo utilizes few-shot interleaved prompts that incorporate gated cross-attention layers to improve few-shot learning.

Alayrac et al. (2022) propose context optimization (CoOp) as a means of modeling text in prompts through continuous representations. Zhou et al. (2022b) propose CoCoOp, which extends CoOp by further learning a lightweight neural network to generate an input-conditional token (vector) for each image. In addition, there are a series of prior-based methods that utilize CLIP priors with a cache model. CLIP-Adapter (Gao et al., 2021) combines zero-shot visual or language embeddings with corresponding finetuning features to improve performance. TIP-Adapter (Zhang et al., 2022) constructs adapters using a key-value cache model from few-shot training sets and updates their prior knowledge through feature retrieval. TIP-X (Udandarao et al., 2022) further constructs an affinity matrix by measuring the KL divergence between test and few-shot samples, which removes direct reliance on the uncalibrated image-image similarities. APE (Zhu et al., 2023) explores the trilateral affinities between the test image, prior cache model, and textual representations and only enable a lightweight category-residual module to be trained. Among these approaches, TIP-Adapter, TIP-X, and APE are training-free, while CoOp, CoCoOp, CLIP-Adapter, and APE-T (Zhu et al., 2023) are training required.

5 CONCLUSION

This paper has studied the few-shot classification problem using large vision-language models. Since it is hard to optimize large vision-language models with a few training examples, we propose exploring category names to initialize the classification head, significantly improving performance. In addition, we have also investigated the condition when the category names help. We demonstrate that borrowing other non-perfect category names or even names from a foreign language could also help the few-shot classification of vision-language models, which is better than randomly initializing the classification head. This paper obtains state-of-the-art few-shot performance on numerous benchmarks, including ImageNet, ImageNet-V2, ImageNet-R, ImageNet-A, ImageNet-Sketch, Cifar100, Oxford Flowers, Stanford Cars, Country-211, Food-101, FGVC Aircraft, EuroSAT, and Oxford-IIIT Pets. Our few-shot classification result is even better than many previous works that have employed the whole training set.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Neural Information Processing Systems (NeurIPS)*, 2022. 1, 3, 5

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 - mining discriminative components with random forests. In *European Conference on Computer Vision (ECCV)*, 2014. 3

Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *ArXiv*, 2202.03026, 2022. 3

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 9620–9629, 2021. 3

Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3008–3017, 2020. 9

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009. 3, 12

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625–2634, 2015. 5

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 3

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *ArXiv*, 2110.04544, 2021. 5

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988, 2022. 3

Patrick Helber, Benjamin Bischke, Andreas R. 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*, 12:2217–2226, 2019. 3

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 8320–8329, 2021a. 3

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Xiaodong Song. Natural adversarial examples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15257–15266, 2021b. 3

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *ArXiv*, abs/1608.08614, 2016. 1

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Learning Representations (ICLR)*, pp. 4904–4916, 2021. 1, 5

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 554–561, 2013. 3

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 3

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 10

Chengkun Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Yong Jae Lee, Houdong Hu, Zicheng Liu, and Jianfeng Gao. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *ArXiv*, abs/2204.08790, 2022. 3, 5

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *ArXiv*, abs/1306.5151, 2013. 3

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008. 3

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3498–3505, 2012. 3

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. 1, 3, 5, 10

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, 2019. 3

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning (ICML)*, pp. 4596–4604. PMLR, 2018. 9

Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, Mia X Chen, Ye Jia, Anjuli Kannan, Tara Sainath, Yuan Cao, Chung-Cheng Chiu, et al. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *ArXiv*, abs/1902.08295, 2019. 9

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15638–15650, 2022. 5, 10

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herv'e J'egou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, 2021. 3

Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. *ArXiv*, abs/2211.16198, 2022. 5

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3156–3164, 2015. 5

Haohan Wang, Songwei Ge, Eric P. Xing, and Zachary Chase Lipton. Learning robust global representations by penalizing local predictive power. In *Neural Information Processing Systems (NeurIPS)*, 2019. 3

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2022. 5

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7959–7971, 2022. 3

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *ArXiv*, abs/2205.01917, 2022. 1, 3, 5

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel C. F. Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *ArXiv*, abs/2111.11432, 2021. 1, 5

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18123–18133, 2022. 1, 5

Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. In *European Conference on Computer Vision (ECCV)*, 2022. 5

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal on Computer Vision (IJCV)*, 130:2337–2348, 2022a. 3, 9

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 (CVPR)*, 2022b. 5

Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. *ArXiv*, abs/2304.01195, 2023. 5

A APPENDIX

A.1 OPTIMIZATION

We use the Adafactor optimizer (Shazeer & Stern, 2018) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay ratio of 0.01. All input images are first rescaled to 580×580 and then randomly cropped to the size of 540×540. We further apply RandAugment (Cubuk et al., 2020) and label smoothing in our data preprocessing pipeline. Our model is implemented in the Lingvo framework using Tensorflow (Shen et al., 2019).

A.2 ANALYSIS OF CATEGORY NAME INITIALIZATION

In this section, we delve deeper into how the proposed category name initialization helps in few-shot learning with large vision-language models. Vision-language models are adept at zero-shot inference without knowing any class names from downstream tasks. However, the zero-shot performance heavily depends on the domain gap and data distribution, thus varying on different downstream tasks. By leveraging a few training examples from the target domain, the pretrained vision-language models can adapt to the target domain.

Improvement upon zero-shot performance. We first examine how category name initialization improves zero-shot performance. As illustrated in Table 1 and Table 2, category name initialization enhances performance across all datasets. The improvement in performance from zero-shot to five-shot varies depending on the dataset. For instance, CoCa-2B on ImageNet saw a 1.32% increase in performance, whereas EuroSAT saw a 36.04% increase. CoCa's impressive zero-shot performance on ImageNet is leaving less room for few-shot learning. Nonetheless, the performance gain achieved through our category name initialization is noteworthy, as some other methods may not achieve comparable improvements, which will be discussed below. We also contend that our few-shot performance is not solely attributable to the strong pretrained CoCa model but also to our proposed category name initialization. For example, CoCa-2B's zero-shot performance on EuroSAT is 49.74%, which is lower than that of most other approaches. However, with our category name initialization, it achieves 85.78%, outperforming other approaches in the five-shot setting.

Table 3: Comparing other fine-tuning methods on ImageNet and its variants. We use IN as the abbreviation for ImageNet, and CNI for category name initialization. The second column means how much training data per class is used for finetuning. 0 shot means the pretrained vision-language model is directly evaluated without finetuning. All the numbers under the last five columns denote the top-1 test accuracy.

Model	Shot	IN	IN-V2	IN-R	IN-A	IN-Sketch
CoCa-base	0	82.26	76.32	93.16	76.17	71.43
CoCa-base+Linear Probing	1	57.49	54.20	69.19	53.38	47.94
	5	79.33	73.18	90.02	73.18	68.03
CoCa-base+Full Fintuning	1	43.77	41.64	55.98	40.31	33.29
	5	60.90	54.32	71.20	54.34	49.25
Coca-base+CoOp	1	79.85	73.21	89.88	76.42	65.81
	5	81.01	75.81	92.58	76.55	71.27
CoCa-base+CNI	1	82.35	76.47	93.37	77.00	71.61
	5	83.47	77.23	93.22	77.23	71.35

Comparing with other fine-tuning methods. In order to further validate the efficacy of category name initialization, we compare it with several other finetuning methods. CoCa-base is chosen as the pretrained vision-language model and experiments are carried out on ImageNet with different finetuning methods, such as linear probing, full finetuning, CoOp (Zhou et al., 2022a), and category name initialization. As demonstrated in Table 3, all finetuning methods, except category name initialization, fail to improve over zero-shot CoCa when one or five training examples per class are used. Furthermore, full finetuning underperforms linear probing due to the fact that the number of training examples is not commensurate with the number of trainable parameters in few-shot learning.

Although CoOp demonstrated better performance compared to linear probing and full finetuning, its one- or five-shot performance is slightly inferior to zero-shot CoCa. This suggests that CoCa's few-shot performance is not significantly improved by learning contextual prompts. On the other hand, category name initialization effectively improves the few-shot performance, which is challenging when the zero-shot performance of CoCa is significantly higher than that of other counterparts such as CLIP (Radford et al., 2021), FLAVA (Singh et al., 2022), and so on.

Category name initialization vs random initialization. To gain a deeper understanding of the advantages of category name initialization, we compared it with random initialization. Figure 3 provides a more detailed comparison of the optimization process using the two initialization methods. By meticulously tuning the parameters, we set the initial learning rate to 1e-5 for category name initialization and 5e-5 for random initialization. It is evident that category name initialization results in a better starting model with higher test accuracy compared to random initialization. Furthermore, the model utilizing category name initialization converges at a quicker pace than random initialization. This can be attributed to the fact that the test accuracy while using random initialization continues to increase even after 250 epochs, whereas the accuracy achieved with category name initialization plateaus around 200 epochs.

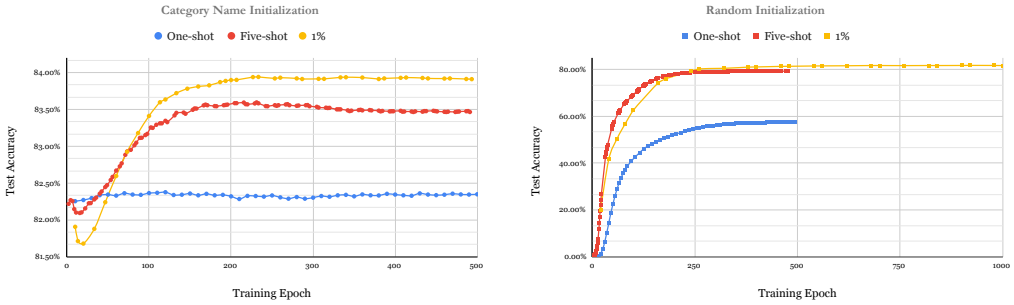


Figure 3: Comparison of test accuracy over the training epoch. We finetune the CoCa-base model with category name initialization or random initialization. Category name initialization provides better initial test accuracy and helps the model converge better and faster than random initialization. The test accuracy when using random initialization keeps increasing after 250 epochs, while the one using category name initialization converges around 200 epochs.

A.3 Comparing different initialization approaches

In the real world, we cannot guarantee that every classification task is associated with perfect category names. Sometimes we may only have digital labels like class "1", "2", ...; sometimes, the users may not speak fluent English. In these scenarios, we want to test how the model performs with different versions of category names.

Table 4 compares the performance of using no category names (i.e., random initialization) with different variants of category names. The simplest situation is to use digits (class 1, 2, ...) as category names. This approach has little semantic information and thus provides no help with the few-shot performance. In contrast, category names in English and other languages significantly help the few-shot recognition. This is surprising because CoCa was trained in English-only text with limited knowledge of other languages. However, thanks to sentence piece tokenizer (Kudo & Richardson, 2018) and token sharing, our method can still benefit from foreign language transfer to perform better than random initialization, even though these foreign language names' performance is not as good as English names.

Motivated by the above observation, we suspect only initialization with partial category information can still help. To verify this, we randomly choose 50% of the category names as initialization while using random initialization for the rest. Table 5 shows using 50% of the names can still boost the one-shot accuracy from random initialization 59.17% to 66.82%, and five-shot accuracy from 79.33% to 80.67%. This suggests that our method is a promising tool when within-domain labels are not complete or come in different languages.

Table 4: Comparison of category name initialization using digits or different languages. We use the same pretrained CoCa-base model for all category name initialization. The numbers below are top-1 test accuracy on ImageNet.

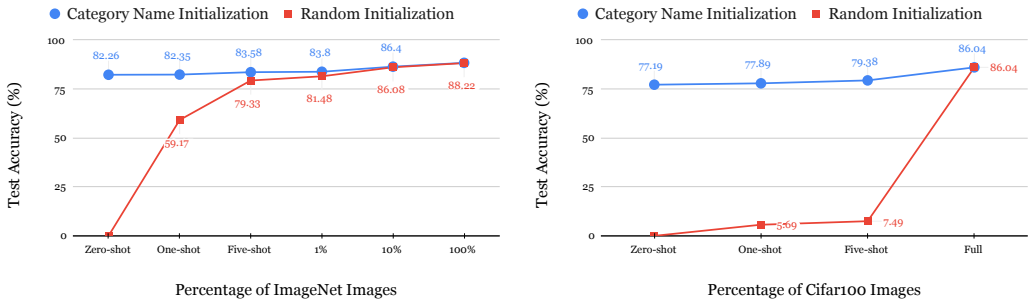
Category name Initialization	Zero-shot	One-shot	Five-shot
No	N/A	59.17	79.33
Digits	0.10	53.60	78.75
Korean	22.89	53.71	79.53
Russian	43.59	53.43	79.55
Germany	29.24	63.15	79.90
Spanish	34.38	79.87	80.05
English	82.26	82.35	83.58

Table 5: Comparing the performance of using all category names or using 50% of names (the other half will be initialized with random vectors) for initialization. The numbers below are top-1 test accuracy on ImageNet.

Initialization	Zero-shot	One-shot	Five-shot
No category name	N/A	59.17	79.33
50% category names	44.36	66.82	80.67
100% category names	82.26	82.35	83.58

A.4 When does the effect of category name initialization diminish?

To demonstrate the effectiveness of category name initialization, we set a baseline by using random initialization for comparison. We perform finetuning over different pretrained vision-language models using different numbers of training images. Specifically, we finetune CoCa-base on ImageNet and CoCa-2B on Cifar100. As shown in Figure 4, category name initialization outperforms random initialization over different datasets, model architectures, and numbers of training data. The contribution of category name initialization diminishes as more training data is provided.



(a) CoCa-base + CNI on ImageNet

(b) CoCa-2B + CNI on Cifar100

Figure 4: Comparison of test accuracy over different percentages of training images. Category name initialization outperforms random initialization over different datasets, model architectures, and numbers of training data.

A.5 Model distillation

First, to verify the scalability of CoCa, we carry out few-shot experiments using two different pretrained CoCa architectures: CoCa-base and CoCa-2B, under different numbers of training data. Abandoning the uni-modal and multi-modal text decoders, CoCa-base and CoCa-2B contain 96M and 1B parameters for downstream image classification tasks. As shown in Table 6, we can observe the trend that bigger models do better and more shots help.

Table 6: Few-shot results of different CoCa-models on ImageNet.

Model	Zero-shot	One-shot	Five-shot	1%
CoCa-2B	86.19	86.15	87.37	87.90
CoCa-base	82.26	82.35	83.58	83.80
+ distillation	-	-	-	84.81

As large models give better performance, it is intuitive to consider knowledge distillation, i.e., taking the prediction of a teacher model as guidance to train a student model. We take the finetuned CoCa-2B using 1% ImageNet images as the teacher model and CoCa-base as our student model. In addition to 1% labeled ImageNet images, we also leverage other unlabeled images for knowledge distillation. The teacher model weights are frozen during the finetuning process, and the student model weights are updated according to two loss objectives. The first is the supervised loss, where we compute the cross entropy between the student model prediction and labels on the 1% labeled ImageNet images. The other one is the distillation loss computed over all unlabeled data. In contrast to few-shot finetuning, where we finetune only the last few layers, we here finetune the whole student model because the distillation loss is computed over many unlabeled images. As shown in Table 6, we have achieved a 1.01% accuracy gain (from 83.80% to 84.81%) for CoCa-base by distilling from a bigger finetuned teacher model – CoCa-2B.

A.6 Ablation Studies

In this section, we analyze several important factors that influence the few-shot performance. We use CoCa-base for our ablation study in the below.

Finetuning layers. We first use ImageNet (Deng et al., 2009) to study the performance of the CoCa-base model by selecting different finetuning layers in several few-shot learning scenarios. We consider random initialization as the baseline for comparison. In our notation, P stands for the image pooler, and L stands for the linear projector. For both category name initialization and random initialization, we try three different optimization strategies: 1) optimize L; 2) optimize P + L; 3) optimize All layers. Note that we have extensively tried various hyper-parameters (e.g., initial learning rate) and present the best number for each setting.

Table 7: Comparison of different finetuning layers for random initialization. P: image pooler; L: linear projector; All: all layers. The best performance of each column is in **bold**.

Finetuning Layers	One-shot	Five-shot	1%	100%
L	49.38	69.64	76.53	85.62
P + L	**57.49**	**79.33**	**81.48**	**88.22**
All	43.77	60.90	79.75	86.03

Table 8: Comparison of different finetuning layers for category name initialization. P: image pooler; L: linear projector; All: all layers. The best performance of each column is in **bold**.

Finetuning Layers	One-shot	Five-shot	1%	100%
L	82.35	81.03	81.67	86.16
P + L	**82.35**	**83.58**	**83.91**	88.25
All	82.28	82.63	83.63	**88.35**

As shown in Table 7, finetuning the image pooler and linear projector gives the best performance under all settings compared to the other two optimization strategies in random initialization.

To improve the performance of few-shot learning, we try category name initialization. In contrast to random initialization, we initialize the linear projector using the computed average text embeddings of the category names. Table 8 illustrates that the few-shot recognition performance is significantly

improved. Besides, we can also observe that the finetuning P + L is the best optimization strategy for few-shot settings, while finetuning all layers works better given more training data.

Learning rates. We analyze the influence of the initial learning rate on few-shot learning. We set a batch size of 512, freeze the image encoder, and adopt a cosine learning rate schedule for the final three layers. Figure 5 presents the top-1 test accuracy on ImageNet using different initial learning rates. Figure 5 presents the top-1 test accuracy on ImageNet using different initial learning rates. A small initial learning rate (5e-6) results in a slow convergence rate, while a larger learning rate (5e-5) achieves faster convergence. However, despite reaching the highest test accuracy within 1000 training steps, the finetuning becomes unstable as the test accuracy declines right after the peak value. Conversely, using an even larger learning rate (5e-4) could prevent the surging phase, resulting in a downward trend of test accuracy. By contrast, selecting an appropriate learning rate (1e-5) is the key to stable and rapid few-shot finetuning. Unfortunately, there is no mathematical formula for determining the optimal initial learning rate since it varies across different datasets and depends on the batch size. We can adjust the initial learning rate by trial and observation, and these four test accuracy curves could indicate whether to enlarge or reduce the initial learning rate.

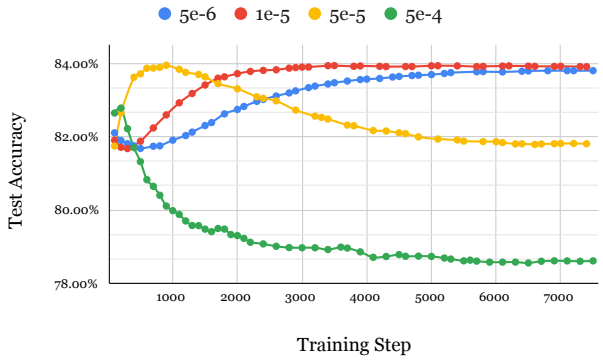


Figure 5: The top-1 test accuracy of finetuning CoCa-base on 1% ImageNet using different initial learning rates.

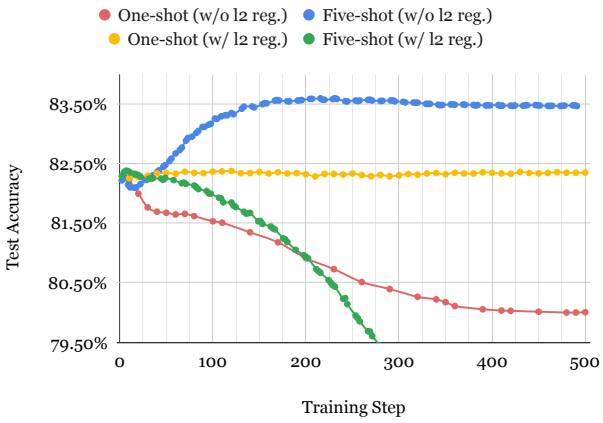
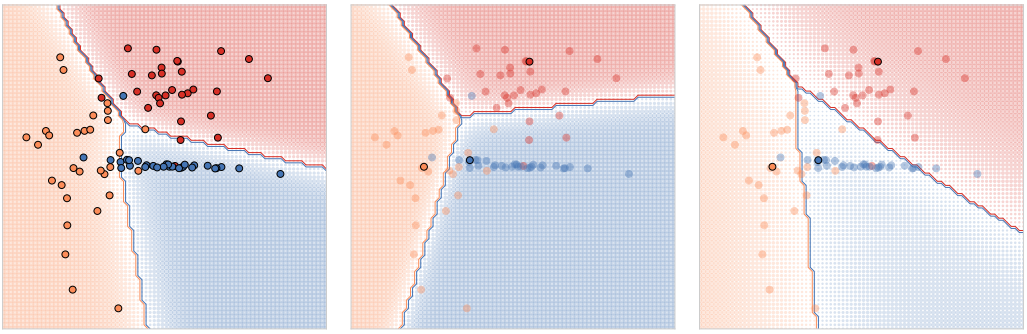


Figure 6: The effect of l2 weight regularization for one-shot and five-shot learning. We plot the top-1 test accuracy of CoCa-base on ImageNet vs. the training step. The l2 weight regularization is beneficial to one-shot learning but harmful to five-shot learning.

L2 weight regularization. Among all the few-shot settings, one-shot learning is the most special and intriguing one. Figure 6 shows that the one-shot test accuracy (red) on ImageNet decreases even using category name initialization in finetuning in contrast to the five-shot one (blue). The

pretrained feature gets distorted as the decision boundary could not be refined better using only one image per class. To address this issue, we leverage l2 weight regularization for one-shot learning. The test accuracy (yellow) increases stably from 82.26% to 82.35%. Though the performance gain is small, it is still nontrivial as the information provided by one-shot data is very limited to help a pretrained model. On the other hand, we find that applying the l2 weight regularization in five-shot learning could ruin the model adaptation, as depicted by the green curve. The reason is the l2 weight regularization, as an additional constraint, prevents the model from learning new knowledge from training data when the data contains relatively sufficient information for refining the decision boundary of the pretrained model.

L2 weight regularization. Out of all the few-shot settings, one-shot learning is the most unique and intriguing. As illustrated in Figure 6, the one-shot test accuracy (in red) on ImageNet decreases even with category name initialization during finetuning, unlike the five-shot accuracy (in blue). Using only one training image per class can easily distort the decision boundary, as illustrated in Figure 7. We plot the decision boundary in Figure 7 for an illustration. Without l2 regularization, the decision boundary of the finetuned model is easily distorted by the limited training examples, resulting in a degradation from zero-shot performance. However, by applying l2 weight regularization for one-shot learning, the decision boundary does not deviate much compared with the decision boundary of the pretrained model. This is reflected in the steady increase of test accuracy from 82.26% to 82.35%, as depicted by the yellow curve in Figure 6. Although the performance gain is small, it is still noteworthy since the information provided by one-shot data is limited in helping a pretrained model. On the other hand, applying l2 weight regularization in five-shot learning could adversely affect the model adaptation, as shown by the green curve. The reason is that l2 weight regularization, acting as an additional constraint, restricts the model from learning new knowledge from the training data when sufficient information is available to refine the decision boundary of the pretrained model. It should be noted that all of the aforementioned phenomena are dependent on utilizing category name initialization. The decision boundary will lack discriminative power if category name initialization is not used. Therefore, adding l2 weight regularization would have no meaningful effect.



(a) Pretrained model (b) No l2 regularization (c) L2 regularization

Figure 7: Visualization of decision boundary in one-shot learning. From left to right, The first subfigure displays the decision boundary of the pretrained model. In contrast, the second and third subfigures show the finetuned model without and with l2 weight regularization, respectively. Each model was trained using only one training example per class, with three classes retained for simplicity. The decision boundary does not shift significantly when finetuning on the one-shot dataset with l2 regularization. This indicates that the model's generalization ability is improved, as it is less likely to overfit to the training examples.