Supplementary Material of Visual Prompting via Partial Optimal Transport

Mengyu Zheng[1,2], Zhiwei Hao[2,3], Yehui Tang[2], and Chang Xu[1⋆]

[1] School of Computer Science, Faculty of Engineering, The University of Sydney
[2] Huawei Noah's Ark Lab
[3] School of information and Electronics, Beijing Institute of Technology
mzhe4259@uni.sydney.edu.au, haozhw@bit.edu.cn, yehui.tang@huawei.com, c.xu@sydney.edu.au

A Datasets and Backbones Specifications

Pre-trained Backbones. In our experiments, we use a total of six backbones, with their specifics outlined in Table 1. Specifically, under the VP framework, all these six backbones are utilized. The results for Vit-B pretrained on ImageNet-22K and ResNet50 are presented in the manuscript while the results for Vit-B/Vit-L pretrained on ImageNet-1K, ConvNeXt-B and Swin-B are presented in this supplementary. Meanwhile, Swin-B is also utilized under VPT framework.

Downstream Datasets. Detailed information about the downstream datasets used under two framework groups is displayed in the table. Specifically, within the VP framework, we utilized ten datasets. For datasets without publicly available splits, we followed [8] for the allocation of training and validation sets. Additionally, the five datasets we used within the VPT framework are from Fine-Grained Visual Classification tasks. For datasets without publicly available splits, we followed [9].

Table 1: Specifications of six pre-trained backbones employed in this paper. All of these backbones are pre-trained on ImageNet [3].

Backbone	**Pre-trained Dataset**	**# params (M)**	**Feature dim** d
vit-B/16 [4]	ImageNet-1k	85	768
vit-B/16 [4]	ImageNet-22k	85	768
swin-B [12]	ImageNet-22k	88	1024
ResNet50 [6]	ImageNet-22k	25	2048
vit-L/16 [4]	ImageNet-1k	307	1024
ConvNeXt-B/16 [13]	ImageNet-2k	88	1024

⋆ Corresponding author

Table 2: Discription of various datasets evaluated.

Dataset	# Classes	# Train	# Val	# Test	Description
Evaluated under VP framework					
DTD [2]	47	1880	1880	1880	texture database
CUB-200-2011 [17]	200	5394	600	5794	images of birds
NABirds [16]	555	21536	2393	24633	images of birds
Stanford Dogs [10]	120	10800	1200	8580	images of dogs
Oxford Flowers [15]	102	1020	1020	6149	images of flowers
Food101 [1]	101	60600	15150	25250	images of food
CIFAR100 [11]	100	40000	10000	10000	real-life object
CIFAR10 [11]	10	40000	10000	10000	real-life object
GTSRB [7]	43	21312	2526	12630	traffic signs
SVHN [14]	10	58605	14652	26032	numbers in real-life
Evaluated under VPT framework					
CUB-200-2011 [17]	200	5394	600	5794	images of birds
NABirds [16]	555	21536	2393	24633	images of birds
Oxford Flowers [15]	102	1020	1020	6149	images of flowers
Stanford Dogs [10]	120	10800	1200	8580	images of dogs
Stanford Cars [5]	196	7329	815	8041	images of cars

B More Results on Other Backbones

We assess our OTLM on Swin-B pretrained on ImageNet-22K and Vit-B pretrained on ImageNet-1K, presented in Table 3 and Table 4, respectively. We train the source models for 100 epochs. And we also assess OTLM on ViT-L pretrained on ImageNet-1K and ConvNeXt-B pretrained on ImageNet-22K, presented in Tabe 5, we train these two source models for 50 epochs. As shown in Table 3, the performance of the original VP with Swin-B as the backbone is not ideal. All LM methods, especially OTLM, significantly enhance VP's performance on all ten downstream tasks. Moreover, OTLM notably outperforms both FLM and ILM. Additionally, as mentioned in experiment section, pre-trained datasets with smaller output dimensions make label mapping simpler. Therefore, compared to backbones pre-trained on ImageNet-22K, the performance of frequency-based label mapping methods improves, as shown in Table 4. Nevertheless, performance of OTLM remains significantly superior to other label mapping methods across all datasets.

C Data Efficiency

To assess the data efficiency of various label mapping methods, we present the detailed results when reducing training samples per clasee from 4k to 4 on CIFAR10 dataset using Vit-B pretrained on ImageNet-22K in Tabel 6.

Table 3: Comparison of performance on 10 downstream datasets under the VP framework. The utilized source model is Swin-B pretrained on ImageNet-22K. The highest performance achieved among prompt-based methods is highlighted in **bold**.

	Fine-tuning		**Prompt-based**			
Dataset	FF	LP	VP	FLM-VP	ILM-VP	OTLM
DTD	72.4	73.6	28.0	56.1	51.1	**62.2**
CUB200	89.7	88.6	5.4	54.5	57.9	**60.9**
NAbirds	86.8	85.2	3.2	46.8	47.7	**52.9**
StanfordDogs	86.2	86.8	12.5	81.9	82.4	**86.6**
Flowers102	98.3	99.4	13.7	89.5	90.8	**95.5**
Food101	91.7	88.2	19.8	62.9	64.9	**66.1**
CIFAR100	73.3	61.6	24.6	68.5	71.4	**77.9**
CIFAR10	98.3	96.3	76.9	93.4	92.9	**95.1**
GTSRB	97.1	93.8	64.3	68.8	72.3	**76.8**
SVHN	91.2	43.5	74.7	77.6	78.5	**81.9**
Average	88.50	81.70	32.29	69.97	71.00	**75.58**

Table 4: Comparison of performance on 10 downstream datasets under the VP framework. The utilized source model is ViT-B pretrained on ImageNet-1K. The highest performance achieved among prompt-based methods is highlighted in **bold**.

	Fine-tuning		**Prompt-based**			
Dataset	FF	LP	VP	FLM-VP	ILM-VP	OTLM
DTD	70.6	68.7	47.8	53.0	54.2	**57.4**
CUB200	84.7	83.5	40.6	53.5	56.8	**63.2**
NAbirds	72.3	69.3	13.8	30.5	34.6	**38.8**
StanfordDogs	84.6	84.4	61.9	71.1	70.3	**73.7**
Flowers102	98.3	97.7	56.5	76.7	74.7	**82.0**
Food101	83.0	78.5	55.7	62.3	63.4	**64.1**
CIFAR100	87.5	77.6	54.4	63.7	64.0	**64.3**
CIFAR10	97.4	92.9	92.9	93.8	93.7	**94.9**
GTSRB	96.8	65.6	86.0	86.1	86.4	**89.0**
SVHN	96.9	61.1	87.8	88.0	88.2	**88.7**
Average	87.21	77.93	59.74	67.87	68.67	**71.61**

Table 5: Comparison of performance on 3 downstream datasets under the VP framework. The utilized source models are ViT-L pretrained on ImageNet-1K and ConvNeXt-B pretrained on ImageNet-22K. The highest performance achieved among prompt-based methods is highlighted in **bold**.

Model	**ViT-L-1k**			**ConvNeXt-B-22k**		
Method	FLM-VP	ILM-VP	OTLM	FLM-VP	ILM-VP	OTLM
CIFAR10	92.9	91.7	**94.9**	79.1	81.3	**83.2**
CIFAR100	50.6	51.4	**72.4**	18.6	23.7	**29.9**
StanfordDogs	62.3	64.5	**87.9**	14.8	32.4	**62.5**

Table 6: Training samples reduce.

Method | 100%(4000)	10%(400)	1%(40)	0.1%(4)
FLM 94.6	88.5	70.6	12.7
ILM 94.7	88.5	69.8	10.9
OTLM **95.2**(+0.5)	**93.6**(+5.1)	**80.3** (+9.7)	**18.4**(+5.7)

D Discussion on Tunable Parameters and Running Speed

Tunable parameters. We present the amount of trainable parameters of various methods on Vit-B pretrained on ImageNet-22K in Table 7. VP incorporates a minimum of trainable parameters with the input images without access to the model instead of tuning a great number of parameters of pretrained model. As shown in Table 7. Compared to FF and LP, the amount of parameters that VP needs to train is negligible. Furthermore, compared to VP, existing label mapping methods, especially our OTLM, do not introduce additional parameters yet significantly enhance model performance. Therefore, it is evident that a VP method equipped with an appropriate label mapping strategy can significantly improve training efficiency and greatly reduce the memory for model storage.

Table 7: Total trainable parameters in the input prompt or model finetuning for all 10 downstream datasets under VP framework on Vit-B pretrained on ImageNet-22K. '×' denotes the multiple of the tunable parameter amount relative to the total amount of pre-trained Vit-B pretrained on ImageNet-22K encoder parameters (85.8M).

	FF	**LP**	**VP**	**FLM**	**ILM**	**OTLM**
Total params	10.01×	0.43×	0.01×	0.01×	0.01×	0.01×

Running speed. Additionally, we report the average execution time for 1 epoch training under the VP framework in Table 8. OTLM shares the similar run time with ILM which is slightly behind FLM.

Table 8: Running time(s/epoch) under VP framework on DTD dataset with source model Vit-B pretrained on ImageNet-22K.

	FLM-VP	**ILM-VP**	**OTLM**
Running time	20	24	24

E Visualization

Visualize mapping results. we present the comparison mapping results on CIFAR10 dataset under the VP framework using model Vit-B pretrained on

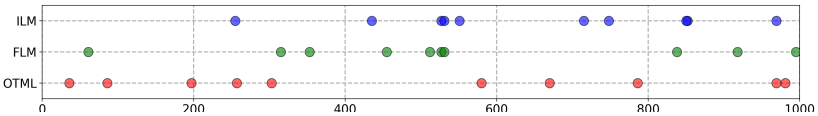

Fig. 1: Visualize mapping results of three label mapping methods .

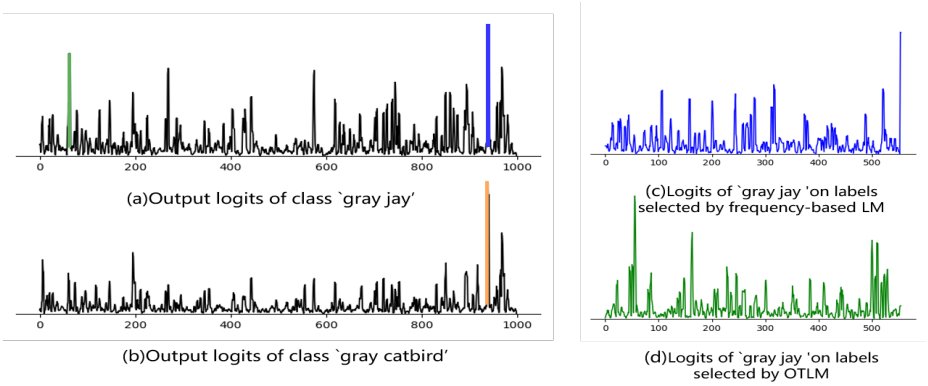

Fig. 2: Explanation of the difference between frequency-based LM and OTLM. (a) The blue line represents the mapping of the 'gray jay' selected from the source labels by FLM. The green line represents the mapping of the 'gray' jay selected from the source labels by OTLM. (b) The orange line indicates the logit for the class 'gray catbird' on the selected label of FLM for 'gray jay'. (c) and (d) report the logits of 'gray jay' on labels selection results by frequency-based LM and OTLM, respectively.

ImageNet-1K in Figure 1. Different label mapping strategies will yield different mapping results, thereby directly affecting the model's accuracy on downstream datasets.

Explanation of Different Label Mapping Strategies. We report the LM strategies of frequency-based LM and OTLM on NABirds dataset using a Vit-B model pretrained on ImageNet-1K at the first epoch in Figure 2. Specifically, the results of the average logit for the classes 'gray jay' and 'gray catbird' at the model's output are shown in subfigures (a) and (b), respectively. The average logits for class 'gray jay', based on labels selected by frequency-based LM and OTLM, are presented in subfigures (c) and (d). As Figure 2 illustrates, the frequency-based LM, following a greedy strategy, chooses the source label with the highest logits as the mapping for 'gray jay' (blue line). However, this causes a significant transport cost since other target labels also have high logits (orange line). In contrast, OTLM, employing a linear programming strategy, selects the source label with the minimal transport cost (green line). Although it may not have the highest logit in the output, as shown by subfigure (d), it has the highest logit among the selected labels. Therefore, frequency-based LM focuses solely on local selection results, while OTLM not only identifies the source label with the highest association but also ensures the transport cost is minimized.

References

1. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13. pp. 446–461. Springer (2014) 2
2. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3606–3613 (2014) 2
3. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition (2009) 1
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021) 1
5. Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Fei-Fei, L.: Fine-grained car detection for visual census estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31 (2017) 2
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2016) 1
7. Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., Igel, C.: Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In: The 2013 international joint conference on neural networks (IJCNN). pp. 1–8. Ieee (2013) 2
8. Huang, Q., Dong, X., Chen, D., Zhang, W., Wang, F., Hua, G., Yu, N.: Diversity-aware meta visual prompting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10878–10887 (2023) 1
9. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European Conference on Computer Vision. pp. 709–727. Springer (2022) 1
10. Khosla, A., Jayadevaprakash, N., Yao, B., Li, F.F.: Novel dataset for fine-grained image categorization: Stanford dogs. In: Proc. CVPR workshop on fine-grained visual categorization (FGVC). vol. 2. Citeseer (2011) 2
11. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) 2
12. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: International Conference on Computer Vision. pp. 9992–10002 (2021) 1
13. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022) 1
14. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011) 2
15. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian conference on computer vision, graphics & image processing. pp. 722–729. IEEE (2008) 2
16. Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S.: Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In: Proceedings of the

IEEE conference on computer vision and pattern recognition. pp. 595–604 (2015) 2
17. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011) 2