Prompt Learning with Optimal Transport for Vision-Language Models Supplementary Material

Anonymous Author(s)
Affiliation
Address
email

A. More Details of Optimal Transport

The Optimal Transport [11] is initially introduced to find a transportation plan to move simultaneously several items at a minimal cost, such as moving a pile of sand to fill all the holes. Recently, it is widely used for the comparison of distributions. Mathematically, given two probability density function U and V over space $\mathcal{X}$ and $\mathcal{Y}$, the OT (Wasserstein) distance [16] can be defined as

$$D_{OT}(U, V) = \inf_{\Gamma} \int_{\mathcal{X} \times \mathcal{Y}} \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}) d\gamma(\boldsymbol{x}, \boldsymbol{y}), \tag{1}$$

where $\boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y})$ is the cost between two points in the space $\mathcal{X} \times \mathcal{Y}$, and Γ denotes the set of transport plans between support points $\boldsymbol{x}$ and $\boldsymbol{y}$ (e.g. $\gamma(\boldsymbol{x}, \boldsymbol{y})$). We can regard two probability density functions U and V as piles and holes and $\boldsymbol{C}$ is the cost function of moving a unit of sand.

In our problem of multiple prompts learning, we formulate the sets of visual features and prompt features as two discrete distributions as

$$U = \sum_{m=1}^{M} u_m \delta_{\boldsymbol{f}_m} \qquad \text{and} \qquad V = \sum_{n=1}^{N} v_n \delta_{\boldsymbol{g}_n}, \tag{2}$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the discrete probability vectorss that sum to 1, and $\delta_{\boldsymbol{f}}$ is a Dirac delta function placed at support point $\boldsymbol{f}$ in the embedding space. Given two support points $\boldsymbol{f}_m$ and $\boldsymbol{g}_n$, the cost function is written as $\boldsymbol{C}(\boldsymbol{f}_m, \boldsymbol{g}_n) = 1 - \text{sim}(\boldsymbol{f}_m, \boldsymbol{g}_n) = 1 - \frac{\boldsymbol{f}_m^T \boldsymbol{g}_n}{||\boldsymbol{f}_m|| \cdot ||\boldsymbol{g}_n||}$. For simply, in this discrete situation, $\boldsymbol{C} \in \mathbb{R}^{M \times N}$ is a cost matrix in which each point denotes the cost between $\boldsymbol{f}_m$ and $\boldsymbol{g}_n$. Then, the total distance of these two distributions is written as:

$$< \boldsymbol{T}, \boldsymbol{C} >= \sum_{m=1}^{M} \sum_{n=1}^{N} \boldsymbol{T}_{m,n} \boldsymbol{C}_{m,n}, \tag{3}$$

where the $\boldsymbol{T} \in \mathbb{R}^{M \times N}$ is a matrix of transport plan, which is learned to minimize the total distance. Each point $\boldsymbol{T}_{m,n}$ in $\boldsymbol{T}$ is a weight of local cost $\boldsymbol{C}_{m,n}$.

The optimization problem of optimal transport is formulated as:

$$\begin{aligned} d_{OT}(\boldsymbol{u}, \boldsymbol{v} | \boldsymbol{C}) &= \underset{\boldsymbol{T}}{\text{minimize}} < \boldsymbol{T}, \boldsymbol{C} > \\ \text{subject to} \quad & \boldsymbol{T}\mathbf{1} = \boldsymbol{u}, \boldsymbol{T}^T \mathbf{1} = \boldsymbol{v}, \boldsymbol{T} \geq 0. \end{aligned} \tag{4}$$

These constraints of $\boldsymbol{T}$ is used to match its marginal distributions and original discrete distributions in Eq (2). In our framework, we treat visual features $\boldsymbol{f}_m$ and prompt features $\boldsymbol{g}_n$ equally and thus $\boldsymbol{u} = \mathbf{1}_{M \times 1}/M$ and $\boldsymbol{v} = \mathbf{1}_{N \times 1}/N$.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Algorithm 1: Prompt Learning with Optimal Transport

Input: Training few-shot image data: $\mathbf{X} = \{\boldsymbol{x}\}$, pretrained CLIP model f and g. number of prompts N, entropy parameter λ, maximum number of iterations in inner and outer loops T_{in}, T_{out}.

Output: The parameters of prompts $\{\boldsymbol{vec}_{l,n}|_{l=1,n=1}^{L,N}\}$

Initialize $\{\boldsymbol{vec}_{l,n}\}$
for $t_{out} = 1, 2, \ldots, T_{out}$ in the outer loop **do**
    Obtain a visual feature set $\boldsymbol{F} \in \mathbb{R}^{M \times C}$ with the visual encoder of CLIP;
    Generate prompt feature set $\boldsymbol{G}_k \in \mathbb{R}^{N \times C}$ of each class with the textual encoder;
    Calculate the cost matrix $\boldsymbol{C}_k = \mathbf{1} - \boldsymbol{F}^T \boldsymbol{G}_k \in \mathbb{R}^{M \times N}$ of each class
    Calculate the OT distance with an inner loop: Initialize the $\boldsymbol{v}^0 = \mathbf{1}$, $\delta = 0.01$ and $\Delta_v = \infty$
    for $t_{in} = 1, 2, \ldots, T_{in}$ **do**
        Update $\boldsymbol{u}^{t_{in}} = \boldsymbol{u}/((exp(-\boldsymbol{C}/\lambda)\boldsymbol{v}^{t_{in}-1})$
        Update $\boldsymbol{v}^{t_{in}} = \boldsymbol{v}/((exp(-\boldsymbol{C}/\lambda)^T \boldsymbol{u}^{t_{in}})$
        Update $\Delta_v = \sum |\boldsymbol{v}^{t_{in}} - \boldsymbol{v}^{t_{in}-1}|/N$
        if $\Delta_v < \delta$ **then**
            break
        end if
    end for
    Obtain optimal transport plan as $\boldsymbol{T}_k^* = \text{diag}(\boldsymbol{u}^t) exp(-\boldsymbol{C}_k/\lambda) \text{diag}(\boldsymbol{v}^t)$,
    Calculate the OT distance $d_{OT}(k) =< \boldsymbol{T}_k^*, \boldsymbol{C}_k >$
    Calculate the classification probability $p_{ot}(y = k|\boldsymbol{x})$ with the OT distance
    Update the parameters of prompts with cross-entropy loss L_{CE}
end for
return $\{\boldsymbol{vec}_{l,n}|_{l=1,n=1}^{L,N}\}$

As directly optimizing the above objective is always time-consuming, we apply the Sinkhorn distance [3] to use an entropic constraint for fast optimization. The optimization problem with a Lagrange multiplier of the entropy constraint is:

$$\begin{aligned} & d_{OT,\lambda}(\boldsymbol{u}, \boldsymbol{v}|\boldsymbol{C}) = \underset{\boldsymbol{T}}{\text{minimize}} < \boldsymbol{T}, \boldsymbol{C} > -\lambda h(\boldsymbol{T}) \\ & \text{subject to} \qquad \boldsymbol{T}\mathbf{1} = \boldsymbol{u}, \boldsymbol{T}^T\mathbf{1} = \boldsymbol{v}, \end{aligned} \tag{5}$$

where $h(\cdot)$ is entropy and $\lambda \geq 0$ is a hyper-parameter. Then we can have a fast optimization solution with a few iterations as:

$$\boldsymbol{T}^* = \text{diag}(\boldsymbol{u}^t) exp(-\boldsymbol{C}/\lambda) \text{diag}(\boldsymbol{v}^t), \tag{6}$$

where t denotes iteration and in each iteration $\boldsymbol{u}^t = \boldsymbol{u}/((exp(-\boldsymbol{C}/\lambda)\boldsymbol{v}^{t-1})$ and $\boldsymbol{v}^t = \boldsymbol{v}/((exp(-\boldsymbol{C}/\lambda)^T \boldsymbol{u}^t)$, with the initiation $\boldsymbol{v}^0 = \mathbf{1}$. The detailed algorithm is shown in Algorithm 1

B. Dataset Details

The datasets we used in the experiments follow CoOp [19], which include 11 datasets for few-shot visual recognition and 4 ImageNet-based datasets for generalization (robustness) evaluation. The details of each dataset are shown in Table 1, including the number of classes, the sizes of training and testing sets, and original tasks.

C. Few-shot Recognition Accuracy

In Section 4.3.1, we provide a line chart to show and compare the performance of PLOT and CoOp. In this section, we provide detailed performance results on all 11 few-shot recognition datasets in Table 2, where we use gray for our method and white for CoOp. To highlight, we respectively use dark cyan and light cyan to represent the performance of PLOT and CoOp on the average of all 11 datasets. We repeat all experiments 3 times and report the mean and standard deviation in the table.

Table 1: The detailed statistics of datasets used in experiments.

Dataset	Classes	Training size	Testing size	Task
Caltech101 [5]	100	4,128	2,465	Object recognition
DTD [2]	47	2,820	1,692	Texture recognition
EuroSAT [6]	10	13,500	8,100	Satellite image recognition
FGVCAircraft [10]	100	3,334	3,333	Fine-grained aircraft recognition
Flowers102 [12]	102	4,093	2,463	Fine-grained flowers recognition
Food101 [1]	101	50,500	30,300	Fine-grained food recognition
ImageNet [4]	1,000	1.28M	50,000	Object recognition
OxfordPets [13]	37	2,944	3,669	Fine-grained pets recognition
StanfordCars [9]	196	6,509	8,041	Fine-grained car recognition
SUN397 [18]	397	15,880	19,850	Scene recognition
UCF101 [15]	101	7,639	3,783	Action recognition
ImageNetV2 [14]	1,000	-	10,000	Robustness of collocation
ImageNet-Sketch [17]	1000	-	50,889	Robustness of sketch domain
ImageNet-A [8]	200	-	7,500	Robustness of adversarial attack
ImageNet-R [7]	200	-	30,000	Robustness of multi-domains

Table 2: The few-shot visual recognition accuracy on 11 datasets.

Dataset	Methods	1 shot	2 shots	4 shots	8 shots	16 shots
Caltech101	PLOT	89.83 ± 0.33	90.67 ± 0.21	90.80 ± 0.20	91.54 ± 0.33	92.24 ± 0.38
	CoOp	87.51 ± 1.02	87.84 ± 1.10	89.52 ± 0.80	90.28 ± 0.42	91.99 ± 0.31
DTD	PLOT	46.55 ± 2.62	51.24 ± 1.95	56.03 ± 0.43	61.70 ± 0.35	65.60 ± 0.82
	CoOp	43.62 ± 1.96	45.35 ± 0.31	53.94 ± 1.37	59.69 ± 0.13	62.51 ± 0.25
EuroSAT	PLOT	54.05 ± 5.95	64.21 ± 1.90	72.36 ± 2.29	78.15 ± 2.65	82.23 ± 0.91
	CoOp	52.12 ± 5.46	59.00 ± 3.48	68.61 ± 3.54	77.08 ± 2.42	83.69 ± 0.47
FGVCAircraft	PLOT	17.90 ± 0.09	18.94 ± 0.44	22.36 ± 0.42	26.17 ± 0.29	31.49 ± 0.89
	CoOp	8.59 ± 5.79	16.52 ± 2.38	20.63 ± 2.46	26.63 ± 0.86	31.43 ± 0.96
Flowers102	PLOT	71.72 ± 0.97	81.19 ± 0.79	87.82 ± 0.20	92.43 ± 0.25	94.76 ± 0.34
	CoOp	67.98 ± 1.98	77.58 ± 1.46	86.10 ± 1.05	91.27 ± 0.83	94.49 ± 0.40
FOOD101	PLOT	77.74 ± 0.47	77.70 ± 0.02	77.21 ± 0.43	75.31 ± 0.30	77.09 ± 0.18
	CoOp	74.25 ± 1.52	72.61 ± 1.33	73.49 ± 2.03	71.58 ± 0.79	74.48 ± 0.15
ImageNet	PLOT	59.54 ± 0.16	60.64 ± 0.06	61.49 ± 0.23	61.92 ± 0.09	63.01 ± 0.13
	CoOp	56.99 ± 1.03	56.40 ± 0.87	58.48 ± 0.47	60.39 ± 0.57	61.91 ± 0.17
OxfordPets	PLOT	87.49 ± 0.57	86.64 ± 0.63	88.63 ± 0.26	87.39 ± 0.74	87.21 ± 0.40
	CoOp	85.99 ± 0.28	82.22 ± 2.15	86.65 ± 0.97	85.36 ± 1.00	87.02 ± 0.89
StanfordCars	PLOT	56.60 ± 0.36	57.52 ± 0.71	63.41 ± 0.29	67.03 ± 0.50	72.80 ± 0.75
	CoOp	55.81 ± 1.67	58.41 ± 0.43	62.74 ± 0.16	67.64 ± 0.06	73.60 ± 0.19
SUN397	PLOT	62.47 ± 0.43	61.71 ± 0.65	65.09 ± 0.43	67.48 ± 0.04	69.96 ± 0.24
	CoOp	60.12 ± 0.82	59.60 ± 0.76	63.24 ± 0.63	65.77 ± 0.02	68.36 ± 0.66
UCF101	PLOT	64.53 ± 0.70	66.83 ± 0.43	69.60 ± 0.67	74.45 ± 0.50	77.26 ± 0.64
	CoOp	62.13 ± 1.14	64.05 ± 0.99	67.79 ± 0.71	72.71 ± 0.50	76.90 ± 0.50
Average	PLOT	62.59 ± 1.13	65.23 ± 0.72	68.60 ± 0.52	71.23 ± 0.51	73.94 ± 0.54
	CoOp	59.56 ± 2.06	61.78 ± 1.39	66.47 ± 1.29	69.85 ± 0.69	73.33 ± 0.42

D. Computation Cost Evaluation

As shown in Table 3, we provide the comparison of the training time and inference seed of the baseline method CoOp [19] and our PLOT with the different number of prompts. We report the one-epoch time training on the 1-shot setting of the Food101 [1] dataset and the number of images processed by the model in 1 second. Taking $N = 4$ as an example, PLOT only reduces the 9.2%

Table 3: The training and inference time comparison.

Settings	CoOp	PLOT(N=1)	PLOT(N=2)	PLOT(N=4)	PLOT(N=8)
Training Time (s)	1.127	1.135	1.148	1.182	1.267
Inference Time (images/s)	719.1	714.4	690.7	653.0	519.8

Table 4: The nearest words for 16 context vectors of all $N = 4$ prompts learned by PLOT. N/A means non-Latin characters.

Number	Prompt 1	Prompt 2	Prompt 3	Prompt 4
1	ag	pa	trying	gaz
2	flint	as	field	**white**
3	leaving	wit	N/A	t
4	sot	l	icons	ario
5	tint	N/A	eclub	safe
6	tar	yl	indiffe	class
7	attn	N/A	ts	represented
8	2	job	cold	attend
9	rollingstones	built	yeah	vie
10	N/A	brought	band	recognized
11	N/A	or	love	old
12	bel	j	late	stel
13	**head**	ag	industry	awhile
14	artifact	bad	N/A	ded
15	an	chie	across	these
16	5	in	actual	visiting

inference speed and requires an extra 4.9% training time, which is acceptable given the performance improvement.

E. The Interpretation of the Learned Prompts

The learned prompts are difficult to be understood by humans since the parameters are optimized in the continuous space [19]. CoOp proposes to use the word which is nearest to learned prompts in the embedding space to visualize the prompts. Following this manner, we show the nearest words of our learned prompts in Table 4. Similar to CoOp, most words can not be directly understood by human logic. However, we still find the relations between the learned prompts and the corresponding optimal transport plan. As shown in Figure 4 in the main paper, we can observe that the optimal transport plan for Prompt 1 always focuses on the "head", such as the head of "brambling", the head of "rooster", and even the head of "aircraft carrier". It is because the word "head" is in Prompt 1. Similarly, we can find that Prompt 4 prefers the white part of images, such as the white environment in the image of "brambling" and the snow in the image of "dog sled". It demonstrates that the learned multiple prompts focus on different characteristics of categories.

F. Visualization of the Failure Cases

To better understand the method and further discover the reason of the failure cases. we visualize the attention maps of some failure cases. As shown in Figure 1, we showed two failure examples with class "2000 AM General Hummer" in the StanfordCars dataset. During the training, we set the number of prompts as 4, but in these visualization results, we found that some of learned prompts remarkably coincide with each other. These prompts can be roughly divided into two classes: Foreground and Background. For example, in both images, prompt 2 (right top) and 3 (left down) focus on the foreground car, while the others focus on the background. It demonstrates that not all classes have multiple complementary attributes, which motivates us to go further to learn the dynamic local prompts numbers to reduce the computational load in the future.

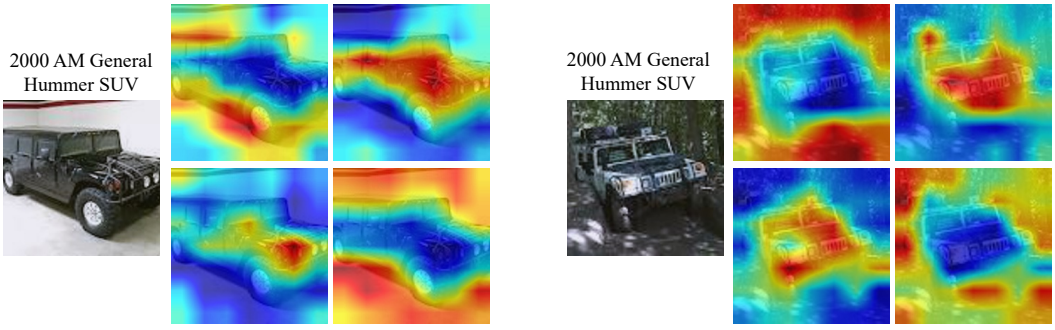

Figure 1: *Failure Visualization. We provide the heatmaps of transport plan T related to each prompt on 2 failure examples in the StanfordCars dataset.*

G. Code

The code used in our experiments is provided in the supplementary material. We implement the method with Pytorch 1.7. Our implementation is based on the publicly released code of CoOp [19].

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, pages 446–461, 2014.

[2] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014.

[3] Marco Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *NeurIPS*, volume 2, page 4, 2013.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[5] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPRW*, pages 178–178, 2004.

[6] Patrick Helber, Benjamin Bischke, Andreas 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(7):2217–2226, 2019.

[7] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[8] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

[9] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, pages 554–561, 2013.

[10] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[11] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

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

[13] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505, 2012.

[14] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019.

[15] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[16] Matthew Thorpe. Introduction to optimal transport. *Lecture Notes*, 2019.

[17] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *NeurIPS*, 32, 2019.

[18] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492, 2010.

[19] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021.