Model-free Domain Adaptation for Concealed Multimodal Large-Language Models

Supplementary Material

A. Detailed Descriptions of proposed Method

A.1. Efficient Prediction Technique for Multimodal Large-language Model

To obtain a prediction from an MLLM F, we need to calculate the classification likelihood for each class y by

$$p_{\text{MLLM}}(y|\mathbf{X},\mathbf{t}) = \prod_{i=1}^{N} F(o_i^y|\mathbf{X},\mathbf{t},o_{<i}^y). \tag{1}$$

However, this class likelihood calculation cannot be conducted in parallel, so it requires huge computational costs as the number of classes increases. To avoid this problem, we limit the calculation to a part of the whole class set by using CLIP prediction results. Specifically, we first obtain the classification likelihood for all classes $\{y\}_{y=1}^{C}$as

$$p_{\text{CLIP}}(y|\mathbf{X}) = \sigma(\text{cossim}(E_{\text{vision}}(\mathbf{X}), E_{\text{text}}(\mathbf{t}_y))). \tag{2}$$

We then obtain the top-k class predictions of CLIP, i.e., top-$k_y[p_{\text{CLIP}}(y|X)]$. We obtain the classification results of the MLLM by calculating the likelihood only for top-$k_y[p_{\text{CLIP}}(y|\mathbf{X})]$ classes; namely,

$$\hat{y}_{\text{MLLM}} = \underset{y' \in \text{top-}k_y[p_{\text{CLIP}}(y|\mathbf{X})]}{\text{argmax}} p_{\text{MLLM}}(y'|\mathbf{X}). \tag{3}$$

We set k to 5 during the prompt training. We also used this technique during the test phase by setting k to 10 to facilitate efficient research.

A.2. Label Propagation Algorithm

We used the label propagation algorithm described in a previous study [4], from which we basically borrowed the formulations. The label propagation algorithm $\mathcal{A}_{\text{LP}}$ is given a labeled set D_L, an unlabeled set D_U, and the visual feature vector set V as inputs. We first obtain a reliable label matrix Y using D_L and D_U as

$$Y_{ij} := \begin{cases} 1 & \text{if } (\mathbf{X}_i, \hat{y}_{\text{MLLM}_i}) \in D_L \wedge \hat{y}_{\text{MLLM}_i} = j, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

We calculate a sparse affinity matrix A using $V = \{\mathbf{v}_i\}_{i=1}^{N}$ as

$$a_{ij} = \begin{cases} [\mathbf{v}_i^\top \mathbf{v}_j]_+^\gamma, & \text{if } i \neq j \wedge \mathbf{v}_i \in \text{NN}_m(\mathbf{v}_j) \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where NN_m denotes the set of m nearest neighbors and γ denotes the parameter. We set $\gamma = 3$ for the default setting. We calculate a symmetric adjacency matrix W by $W = A + A^\top$ then, calculate the normalized counterpart as $\mathcal{W} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where D is a dgree matrix $D = \text{diag}(W\mathbf{1}_N)$ and $\mathbf{1}_N$ is the all-ones vector. The label-propagated results can be obtained by calculating

$$Z = (I - \beta\mathcal{W})^{-1} Y. \tag{6}$$

However, calculating the inverse matrix incurs high computational costs. Thus, we use the conjugated gradient method and solve the following problem:

$$(I - \beta\mathcal{W})Z = Y. \tag{7}$$

After obtaining the matrix Z, the soft label of sample i is calculated by

$$\hat{\mathbf{p}}_i = \sigma(\mathbf{z}_i/\tau_{\text{LP}}), \tag{8}$$

where τ_{LP} is the temperature parameter. We set $\tau_{\text{LP}} = 0.01$ by default.

In addition to the above label propagation algorithm, we use an approach to adjust the sample size. When we apply the label propagation algorithm to the whole dataset for each training epoch, we cannot fully exploit the training results in the label propagation due to the scarce execution frequency. When we apply the label propagation algorithm to each training batch, we cannot obtain satisfactory results due to the too small sample size. Therefore, we apply the label propagation algorithm to each chunk of several batches. We additionally introduce a parameter N_C to adjust the chunk size. Note that the data size N is calculated as $N = N_C \times B$, where B denotes the batch size.

Among the settings of the label propagation algorithm, the tunable hyper-parameters are the nearest neighbor size m and N_C. We set m to 30 and N_C to 300 for DomainNet experiments and m to 10 and N_C to 100 for PACS, considering the whole dataset size.

B. Detailed Experimental Results

B.1. Comparison Results.

We provide the full version of Tab. 4 in Tab. A. A new finding from this table is that the first case has a significantly larger standard error than the others. We observed in the course of conducting our experiments that, in very few cases, the training normally progresses by chance, even in the absence of the domain adaptation training. However, in most cases, the training failed, making it very unstable. This result

Table A. **Detailed Comparison Results.** We highlighted in colors the cases where performance is improved and is declined. The best scores are highlighted in **bold**.

Performance of Surrogate MLLM: MiniGPT-4

	Methods	DA	CMPL	CMGA		Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Average
0.	No Adapt.	-	-	-	Accuracy	68.00	40.20	55.90	14.20	74.62	58.27	51.87
1.	TVP† [8]				Accuracy Gain Δ	2.74±0.10 - 65.26	1.61±0.52 - 38.59	41.67±19.97 - 14.23	0.39±0.02 - 13.81	80.23±0.14 + 5.61	1.34±0.30 - 56.93	21.06±3.28 - 30.54
2.	TVP†+DA	✓			Accuracy Gain Δ	72.94 ±0.27 + 4.94	42.50 ±0.20 + 2.30	64.00 ±0.24 + 8.10	19.64 ±0.01 + 5.44	80.58 ±0.11 + 5.96	63.90 ±0.18 + 5.63	57.26 ±0.02 + 5.40
3.	2. + CMPL	✓	✓		Accuracy Gain Δ	**73.40**±0.16 + 5.40	**43.78**±0.04 + 3.58	**65.11**±0.18 + 9.21	23.44±0.24 + 9.24	**81.00**±0.23 + 6.38	64.77±0.14 + 6.50	**58.58**±0.03 + 6.72
4.	2. + CMGA	✓		✓	Accuracy Gain Δ	72.76±0.17 + 4.76	41.77±0.56 + 1.57	63.39±0.41 + 7.49	19.30±0.26 + 5.10	80.25±0.13 + 5.63	63.39±0.33 + 5.12	56.81±0.08 + 4.95
5.	**MTDA-VP**	✓	✓	✓	Accuracy Gain Δ	73.34±0.05 + 5.34	43.76±0.15 + 3.56	64.80±0.31 + 8.90	**23.46**±0.22 + 9.26	80.75±0.30 + 6.13	**64.93**±0.10 + 6.66	58.51±0.18 + 6.64

Performance of Concealed MLLM: InstructBLIP

	Methods	DA	CMPL	CMGA		Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Average
0.	No Adapt.	-	-	-	Accuracy	66.04	36.16	52.09	13.26	73.64	54.48	49.28
1.	TVP† [8]				Accuracy Gain Δ	45.04±4.14 - 21.00	19.47±1.78 - 16.69	41.48±13.39 - 10.61	7.37±0.19 - 5.89	74.06±0.38 + 0.42	16.61±3.16 - 37.87	34.00±2.48 - 15.27
2.	TVP†+DA	✓			Accuracy Gain Δ	69.20±0.62 + 3.16	38.38±0.51 + 2.22	54.58±0.70 + 2.49	17.39±0.16 + 4.13	73.55±0.31 - 0.09	57.37±0.54 + 2.89	51.74±0.25 + 2.46
3.	2. + CMPL	✓	✓		Accuracy Gain Δ	69.73±0.42 + 3.69	**41.07**±0.04 + 4.91	57.82±0.35 + 5.73	18.87±0.17 + 5.61	73.53±0.59 - 0.11	58.25±1.28 + 3.77	53.21±0.23 + 3.93
4.	2. + CMGA	✓		✓	Accuracy Gain Δ	68.98±0.41 + 2.94	35.88±2.09 - 0.28	54.62±1.22 + 2.53	17.16±0.40 + 3.90	73.56±0.24 - 0.08	58.51±0.83 + 4.03	51.45±0.42 + 2.17
5.	**MTDA-VP**	✓	✓	✓	Accuracy Gain Δ	**70.12**±0.38 + 4.08	40.05±0.56 + 3.89	**59.42**±0.98 + 7.33	**20.28**±0.42 + 7.02	**74.24**±0.82 + 0.60	**58.92**±0.13 + 4.44	**53.84**±0.28 + 4.56

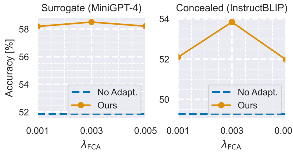

Figure A. $\mathcal{L}_{\text{FCA}}$ analysis.

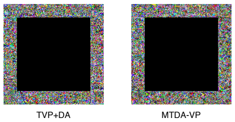

Figure B. Qualitative analysis of prompts.

indicates that the domain adaptation loss makes the training stable and successfully avoids prediction collapse.

A more in-depth look at the effect of CMGA shows that it varies with and without the use of CMPL. Specifically, the performance of the concealed model is slightly reduced by CMGA without CMPL, but CMGA with CMPL demonstrates an improvement in the concealed model. This indicates that a sufficient number of pseudo-labeled samples is required for CMGA to be effective. The samples utilized in the pseudo-label training without CMPL are limited to a part of the whole sample. The role of CMGA is to accelerate the pseudo-label training in those samples. When the label information is not enough, CMGA may fail to exploit its usefulness, but rather disturb the total training. In fact, CMGA caused the performance degradation in the surrogate model in this case. Since CMPL provides high-quality pseudo labels for all samples, CMGA successfully demonstrates its full potential.

B.2. Additional Analysis

Quantitative Analysis of FCA. We provide additional analysis of our method, i.e., the quantitative analysis of the feature consistency alignment (FCA) and the qualitative analysis of the visual prompts. Since FCA was originally proposed in the base method, TVP [8], we employed it as a default component. To confirm the effect of FCA on our method, we conducted the quantitative analysis by varying the size of λ_{FCA}. As shown in Fig. A, it is beneficial to employ FCA to enhance the model transferability of the visual prompts, but we found that too strong FCA training may hinder the other training effect of our method.

Qualitative Analysis of the training visual prompts. We qualitatively analyzed the trained visual prompts by actually visualizing them (Fig. B). Unfortunately, we could not find any interpretable difference between the prompts trained on different training objectives. We suppose that since the visual prompting techniques [1] are mechanically the same as the adversarial perturbations, the difference in the visual prompts is not interpretable to people.

Analysis on the large model gap. The concealed MLLMs in the real-world applications are completely black-box, and there may be a large gap between them and the open-source model in terms of model components such as the network architectures and training recipes. To verify the effectiveness of our method under such cases, we conducted the experi-

Table B. **Analysis on the large model gap.** We highlighted in colors the cases where performance is improved and is declined.

Surrogate MLLM: MiniGPT-4							
	C	I	P	Q	R	S	Ave.
No Aadpt.	45.97	26.80	38.46	10.76	56.57	40.93	36.58
MTDA-VP	45.76	30.63	42.03	14.05	54.34	42.30	38.19
Gain∆	-0.21	+3.83	+3.57	+3.29	-2.23	+1.37	+1.60
Surrogate MLLM: Instruct-BLIP							
	C	I	P	Q	R	S	Ave.
No Aadpt.	45.97	26.80	38.46	10.76	56.57	40.93	36.58
MTDA-VP	45.76	30.69	42.15	13.72	54.04	42.56	38.15
Gain∆	-0.21	+3.89	+3.69	+2.96	-2.53	+1.63	+1.57

ments by setting Qwen2-VL-7B [7] model as the concealed model[1], which is newer than the surrogate models (MiniGPT-4 and InstructBLIP) and has a different model structure. The experimental results using DomainNet dataset are shown in Tab. B. The gain becomes smaller than in the smaller model gap cases, but the performance of the concealed model is still improved, which demonstrates the effectiveness of our method. In this experiment, we validate the effectiveness of our method for Qwen2-VL by using it as the concealed model, but the experimental result suggests that the reversed pattern may also be effective (i.e., using Qwen2-VL as the surrogate model). Building more robust algorithms against the model gaps will be our future work. In the practical cases, using an open-source MLLM that is expected to have a small model gap from the target concealed MLLM as the surrogate model could be one solution (e.g., using Gemma [6] as the surrogate MLLM when the concealed model is Gemini [5]).

C. Extension of MTDA-VP

As we have mentioned in the limitations, the MTDA-VP proposed in this paper is verified with simple classification tasks. However, our method possesses sufficient potential as a good base method for solving various other tasks. For example, our method can be applicable to the generative tasks (e.g., VQA) with some minor modifications. Specifically, in CMPL, we use CLIP-Score [3] for quality assessment of the MLLM outputs to obtain reliable pseudo labels, and SFDA training in CLIP branch can be substituted by image-text contrastive learning (positive pair loss = discriminability, negative pair loss = diversity), which is also employed in training the generative model like BLIP.

D. Broader Impacts

Our method primarily aims to improve the performance of the concealed MLLMs by adding visual prompts to the query image. From another viewpoint, this method modifies the image to make the models behave differently from the developer's intent, which also applies to TVP [8] as this is the base of our method. As discussed in Adversarial Reprogramming (AR) [2], there are concerns that these could lead to ethical issues such as abuse of the models. However, unlike AR, our method (including TVP) does not directly use the model itself, so we consider that it is difficult to manipulate the model as desired, and the abuse is kept to a minimum.

References

[1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 2

[2] Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. In *Proc. ICLR*, 2019. 3

[3] Hessel et al. Clipscore: A reference-free evaluation metric for image captioning. In *Proc. EMNLP*, 2021. 3

[4] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proc. CVPR*, pages 5070–5079, 2019. 1

[5] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 3

[6] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024. 3

[7] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 3

[8] Yichi Zhang, Yinpeng Dong, Siyuan Zhang, Tianzan Min, Hang Su, and Jun Zhu. Exploring the transferability of visual prompting for multimodal large language models. In *Proc. CVPR*, pages 26562–26572, 2024. 2, 3

[1] We newly implemented the evaluation code of Qwen2-VL-7B following TVP code (https://github.com/zycheiheihei/Transferable-Visual-Prompting) to align the evaluation method with the main experiments.