Omniview-Tuning: Boosting Viewpoint Invariance of Vision-Language Pre-training Models — Appendix

Shouwei Ruan[1], Yinpeng Dong[2,5], Hanqing Liu[3], Yao Huang[1], Hang Su[2,4], and Xingxing Wei[1,3⋆]

[1] Institute of Artificial Intelligence, Beihang University, Beijing 100191, China
[2] Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center, THBI Lab, Tsinghua-Bosch Joint ML Center, Tsinghua University, Beijing, 100084, China
[3] Hangzhou Innovation Institute, Beihang University, Hangzhou 311228, China
[4] Zhongguancun Laboratory, Beijing, 100080, China
[5] RealAI
{shouweiruan,hqliu,y_huang,xxwei}@buaa.edu.cn, {dongyinpeng,suhangss}@tsinghua.edu.cn

A Evaluation on OpenFlamingo

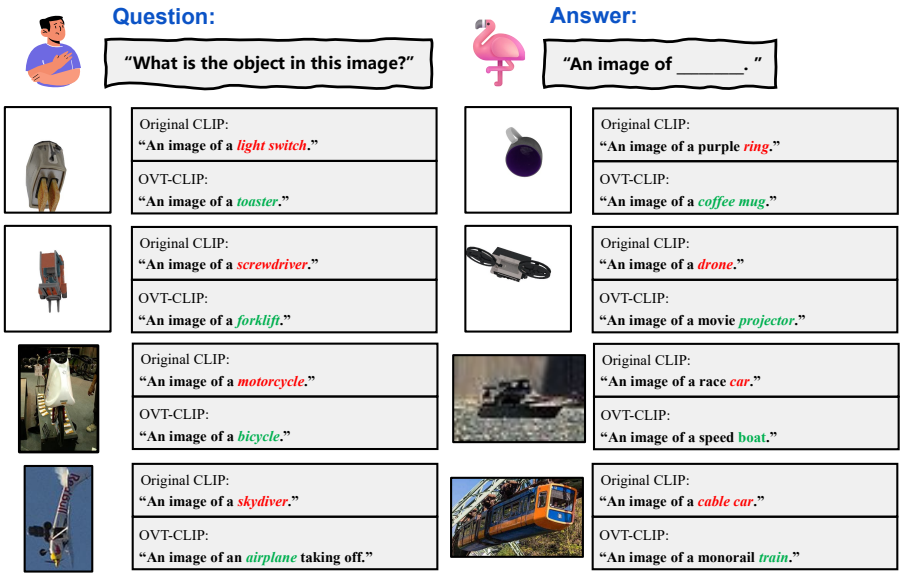

Fig. A.1: The answers generated by OpenFlamingo-3B using our OVT-CLIP and the original OpenAI CLIP as vision encoder, where ***red texts*** indicates incorrect category descriptions, and ***green texts*** represents correct.

⋆ Corresponding author.

Table A.1: VQA accuracy (%) of OpenFlamingo under clean distribution samples and viewpoint-OOD samples from Real-world and Synthetic domains. We utilize the MPNet [3] to calculate the similarity between generated descriptions and ground-truth labels, considering predictions successful if they exceed the similarity threshold β.

Model	Visual Encoder	Real-world Domain						Synthetic Domain					
		OOD-CV (iid)			OOD-CV (Pose)			IM3D			ImageNet-View.+		
		β@1.0	β@0.5	β@*Adp.*	β@1.0	β@0.5	β@*Adp.*	β@1.0	β@0.5	β@*Adp.*	β@1.0	β@0.5	β@*Adp.*
OF-3b	OpenAI CLIP(ViT-L/14)	40.1	**93.3**	62.6	37.7	**87.0**	48.8	49.2	78.3	59.1	24.6	55.8	34.2
	OVT-CLIP(ViT-L/14)	**40.7**	92.7	**63.4**	**38.0**	82.5	**49.9**	**50.7**	**79.4**	**61.2**	**30.0**	**62.4**	**42.0**
OF-4b	OpenAI CLIP(ViT-L/14)	45.6	**93.8**	66.4	43.3	**84.8**	48.8	**50.4**	**79.2**	60.7	25.3	56.5	35.6
	OVT-CLIP(ViT-L/14)	**47.6**	**93.8**	**68.4**	**44.4**	81.5	**51.7**	50.1	76.7	**62.2**	**29.8**	**61.0**	**43.5**

In this study, we integrate our improved OVT-CLIP into OpenFlamingo [1] to evaluate its performance in the Visual Question Answering (VQA) task, leveraging the same evaluation datasets and metrics outlined in Sec. 5.2 for consistency. Our experimental setup involves a comparative analysis between the baseline OpenAI CLIP model (ViT-L/14) and our improved OVT-CLIP (ViT-L/14). For OpenFlamingo's text prompts, we employ a question-and-answer format, with the questions template as "What is the object in this image?" and the answers template as "This is an image of <>."

The results across different data distributions are shown in Tab. A.1. It indicates that OVT-CLIP significantly outperforms the original OpenAI CLIP model in handling viewpoint-OOD data (OOD-CV(Pose) and ImageNet-V+) while preserving its performance on clean data distributions (OOD-CV(iid) and IM3D) across the 3B and 4B parameter scales in OpenFlamingo. Fig. A.1 highlights specific answer examples where OpenFlamingo, powered by OVT-CLIP, demonstrates remarkable precision in identifying object categories despite shifts in viewpoint. Building on these promising results, we will next focus on extending the application of OVT-CLIP to a broader spectrum of VLLMs to further bolster their resilience against viewpoint shifts, thereby enhancing their overall robustness and applicability in real-world scenarios.

B Additional Experimental results

B.1 Ablation study on λ and K

In this section, we conduct an ablation study focusing on key hyperparameters within the Omniview-Tuning (OVT) framework — the loss balance parameter λ and the number of outlier samples K set for each object during the maximization process. We train OVT-OpenCLIP (ViT-B/32) under different ablation settings, evaluating their average Top-1 accuracy across three data distributions as in Sec. 5.1. For the ablation experiments on λ, we fix K at 5, and for the ablation experiments on K, we set λ to 1.0. All other training parameters are set consistently across each experiments, ensuring all other training parameters remain consistent across each set of experiments.

Effects of λ: As a balancing parameter between $\mathcal{L}_{VC}$ and $\mathcal{L}_{ITC}$, λ critically influences the contribution ratio of these two loss terms during the fine-tuning

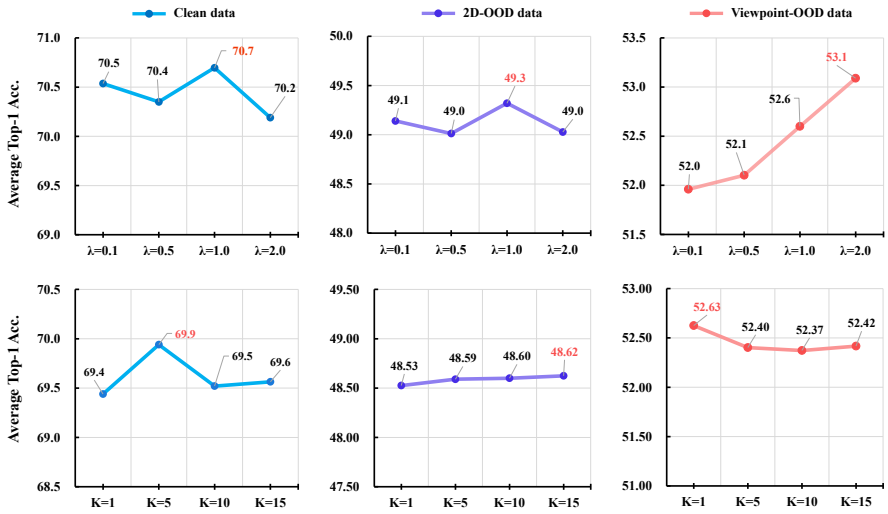

Fig. B.1: The curves of Top-1 average accuracy (%) for OVT-OpenCLIP (ViT-B/32) under various data distributions, with different settings of λ and K.

process. Specifically, higher λ values emphasize enhancing cross-viewpoint alignment, theoretically improving the model's performance on viewpoint shift samples. As illustrated in the first row of Fig. B.1, where the curve on the right shows the average accuracy for viewpoint-OOD data, increasing λ generally correlates with better performance. However, for clean and 2D-OOD samples, a higher λ value might lead to a performance decrement. Considering the performance across three data distributions, setting λ to 1.0 allows the model to achieve the most balanced performance. It not only realizes the highest average Top-1 accuracy on clean and 2D-OOD samples (70.7% and 49.3%, respectively) but also attains a 52.6% average Top-1 accuracy on viewpoint-OOD data.

Effects of K: As shown in the second row of the curves in Fig. B.1, the model exhibits the best performance for the clean dataset when K=5, reaching an average Top-1 accuracy of 69.9%. For the 2D-OOD dataset, although there is a positive correlation between the K value and performance, the impact of the K value on performance is relatively minor, with less than 0.1% difference in performance between K=15 and K=1. On the viewpoint OOD dataset, smaller K values performed better. This can be attributed to the fact that when the number of focused outlier samples is reduced, these outliers are more likely to represent the most extreme viewpoint changes, thereby improving the model's generalization ability and consistency across different viewpoint-OOD data. Based on these experimental results, setting K to 5 is reasonable, achieving a more balanced performance across different data distributions.

Table B.1: Comparison between OVT and the random viewpoint sampling OVT versions (OVT-ROS and OVT-RO&AS) within OpenCLIP (ViT-B/32). We report the average Top-1/Top-5 zero-shot accuracy (%) under different data distributions.

Method	Total Avg.		Clean Avg.		2D-OOD Avg.		Viewpoint-OOD Avg.	
OVT-RAOS	56.8	82.8	69.7	91.4	48.8	76.0	51.9	81.0
OVT-ROS	56.9 (↑0.1)	82.9 (↑0.1)	70.2 (↑0.5)	91.7 (↑0.3)	49.1 (↑0.3)	76.2 (↑0.2)	51.5 (↓0.4)	80.8 (↓0.2)
OVT	**57.5 (↑0.7)**	**83.5 (↑0.7)**	**70.7 (↑1.0)**	**91.8 (↑0.4)**	**49.3 (↑0.5)**	**76.5 (↑0.5)**	**52.6 (↑0.7)**	**82.3 (↑1.3)**

B.2 Comparison with Random Viewpoints Sampling Baselines

Following the experimental logic of VIAT [2], we compare OVT with two potential baseline methods that employ random viewpoint sampling. In OVT, random viewpoint sampling primarily considers the following two scenarios:
(A) Random Outlier Viewpoint Sampling (OVT-ROS). The process of selecting outlier viewpoints is not based on a ranking of distance metrics, but rather involves randomly picking from all possible viewpoints of an object.
(B) Random Anchor & Outlier Viewpoint Sampling (OVT-RAOS). Building on baseline A, further involves randomly selecting anchor viewpoints. An anchor viewpoint can be any viewpoint on the same object, not specifically the central point of viewpoint embeddings. This setting corresponds to the naive cross-viewpoint alignment method described in Sec. 4.2, Eq. (6).

Based on the results from Tab. B.1, under the condition of the same number of sampled viewpoints, the OVT method employing the min-max optimization strategy outperforms the random sampling-based OVT baseline across various data distributions. In terms of overall average Top-1 accuracy, OVT achieves a 0.7% improvement over OVT-RAOS and a 0.6% increase compared to OVT-ROS. Particularly in the case of viewpoint-OOD data, the average accuracy of OVT improves by 0.7% compared to OVT-RAOS and by 1.3% compared to OVT-ROS, demonstrating its clear advantage.

B.3 Additional visualisation results

We provide more examples of zero-shot classification tasks, as shown in Fig. B.2.

C Explanation of the Evaluation Metrics

In our evaluation of image captioning and VQA tasks, we designed the "Description Accuracy" metric (as seen in Tab. 4 and Tab. A.1). This metric calculates the semantic similarity between the category-related vocabulary contained in the captions or answers and the ground-truth category labels by utilizing a word embedding model, and it counts the proportion of samples that exceed a certain similarity threshold. To clarify this process, we formally define Description Accuracy here. Let T^g be the category description text generated by the VLLMs, and T^t be the ground-truth text. We use MPNet [3], denoted as $\mathcal{M}$, to map these

Fig. B.2: Additional Visualization for zero-shot classification. Below each image, we show the predicted categories and their confidence levels (%) by the OpenCLIP(ViT-B/16) (*first row*) and by our improved OVT-OpenCLIP(ViT-B/16) (*second row*). ✓ indicates a correct prediction while ✗ indicating an incorrect one.

texts into the embedding space and calculate the cosine similarity between these embedding vectors. Finally, Description Accuracy is defined as the proportion of samples that meet the condition under different similarity thresholds β as follow:

$$Acc@\beta = \frac{1}{N} \cdot \sum_{i=1}^{N} \sigma\left(\frac{\mathcal{M}(T_i^g) \cdot \mathcal{M}(T_i^t)}{\|\mathcal{M}(T_i^g)\| \cdot \|\mathcal{M}(T_i^t)\|} \geq \beta\right), \tag{C.1}$$

where $\sigma(\cdot)$ is an indicator function that returns 1 if the condition is true and 0 otherwise.

D Pseudo-Code and Computational Cost

To facilitate the understanding of the OVT training process, we provide the pseudocode for OVT as shown in Algorithm 1. In our experiments, the computational cost of the OVT fine-tuning process is primarily affected by the scale of the vision encoder and the batch size. Taking the MVCap dataset as an example, when using the ViT-B encoder, we set the batch size to 512. The outer maximization step of each fine-tuning cycle takes about 4 GPU hours, with the majority of this time occupied by the forward inference of multi-view embeddings while computing the anchor embeddings and outlier samples takes about 10 to 15 GPU minutes. The subsequent inner minimization step requires approximately 8 GPU hours. When using the ViT-L encoder and setting the batch size to 256, the maximization phase of each cycle takes about 20 GPU hours, and the minimization phase is about 40 GPU hours. The GPUs used in our experiments are the NVIDIA RTX 6000 Ada Generation with 48GB memory.

Algorithm 1: Omniview-Tuning Algorithm

Input: Multi-view image-text pairs $\tilde{\mathcal{D}} = \{\langle I_{ij}, T_{ij}\rangle \mid i = 1, 2, ..., N; j = 1, 2, ..., M_i\}$, learnable parameters $\mathbf{A}, \mathbf{B}, \boldsymbol{\theta}$, image encoder $E_{\mathbf{W_v}}$, text encoder $E_{\mathbf{W_t}}$, learning rate η, balance parameters λ, outlier sample size K.

Output: Optimal parameters $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\boldsymbol{\theta}}$.

Initialize $\mathbf{A}, \mathbf{B}, \boldsymbol{\theta}$;
for *Each fine-tuning epoch* **do**
  /* Inner Maximization Step */
  Calculate image embeddings $\tilde{z}^I_{ij}$ for each I_{ij} by Eq.(10) ;
  Calculate anchor embeddings $\tilde{z}^I_{C_i}$ for each object i by Eq.(8) ;
  Obtain outlier viewpoints indexes $\{j_1, j_2, ...j_K\} \leftarrow \max_{\{j_1,...j_K\}} d(\tilde{z}^I_{ij}, \tilde{z}^I_{C_i})$;
  $\mathcal{O} = \{O_i\}_{i=1}^N$; $O_i \leftarrow \{ij_1, ij_2, ..., ij_K\}$;
  /* Outer minimization step */
  for *Each mini-batch* **do**
    Calculate $\mathcal{L}_{ITC}$ by Eq.(3) ;
    if $\exists\ ij \in \mathcal{O}$ **then**
      Calculate $\mathcal{L}_{VC}$ by Eq.(7) ;
    else
      $\mathcal{L}_{VC} \leftarrow 0$
    end
    Calculate $\mathcal{L} \leftarrow \mathcal{L}_{ITC} + \lambda \cdot \mathcal{L}_{VC}$
    $\mathbf{A} \leftarrow \mathbf{A} + \eta \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{A}}$; $\mathbf{B} \leftarrow \mathbf{B} + \eta \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{B}}$; $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \cdot \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}$
  end
end

References

1. Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S., Sagawa, S., et al.: Openflamingo: An open-source framework for training large autoregressive vision-language models. arXiv preprint arXiv:2308.01390 (2023) 2
2. Ruan, S., Dong, Y., Su, H., Peng, J., Chen, N., Wei, X.: Towards viewpoint-invariant visual recognition via adversarial training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4709–4719 (2023) 4
3. Song, K., Tan, X., Qin, T., Lu, J., Liu, T.Y.: Mpnet: Masked and permuted pre-training for language understanding. Advances in Neural Information Processing Systems **33**, 16857–16867 (2020) 2, 4