Supplementary Materials: RegionAligner: Bridging Ego-Exo Views for Object Correspondence via Unified Text-Visual Learning

Yuhao Su
Northeastern University
su.yuh@northeastern.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

1. More Qualitative Results

Figure 1 presents qualitative comparisons between our method and the baseline methods (Osprey+Lisa and DCAMA) across various scenarios. In the soccer scenario, both Osprey+Lisa and DCAMA detect the soccer ball but also include extraneous objects—Osprey+Lisa captures a yellow rounded object due to its similar shape, while DCAMA includes white socks because of their similar color. In the health scenario, Osprey+Lisa fails to detect the timer, whereas DCAMA includes additional regions, such as a white instruction manual. Our method consistently performs the best across all cases, highlighting the effectiveness of our proposed solution.

2. More Implementation Details

We refine large VLM outputs by filtering out commonly irrelevant objects (tripod cameras, floor, ceiling). For object detection, we discard overly large boxes (covering more than 90% of the frame), set both the box and text confidence thresholds to 0.15, and apply non-maximum suppression (NMS) with an IoU threshold of 0.5. We use CLIP-B [8] to extract text features with dimension being 512. For data processing, following [1], we downsample all images to 480×480 with padding at the bottom and top. During training, inputs are further interpolated to 384×384 to accommodate DCAMA's architecture requirements, while predictions and evaluations are conducted at the original 480×480 resolution. No additional data augmentation is applied in our experiments. For training protocol, our model is implemented in PyTorch 1.15.1 with CUDA 11.7 and trained on 4 NVIDIA Tesla V100 GPUs. We initialize our DCAMA model with pre-trained weights from COCO dataset [6]. We optimize using SGD with momentum 0.9 and weight decay 5e-4. The learning rate is set to 5e-4, and following [1], we assign weights of 10:1 to positive and negative pixels respectively in the $\mathcal{L}_{bce}$. To stabilize the training process, we apply gradient clipping.

For training PSALM, we use the same preprocessed data as DCAMA. The model is trained bidirectionally for 3 epochs, initialized from publicly released checkpoints. Both PSALM and RegionAligner (PSALM) are trained using PyTorch 2.0.1 with CUDA 11.8 on 2 NVIDIA H100 GPUs, with a batch size of 10 per GPU. We use a learning rate of 6e-5 and enable bfloat16 (bf16) precision. Text-visual fusion is not explicitly implemented, as PSALM natively supports multimodal integration.

3. Failure Mode Analysis

We analyze failure cases in Figure 2, focusing on *severe occlusion under dense co-location*. In a chopping scenario, the board is heavily occluded by vegetables, bowls, and the person's elbow. Under such extreme stacking, both methods fail to recover the full chopping board. These cases highlight that annotation noise and extreme occlusion remain challenging for ego-exo correspondence, pointing to directions for improving robustness.

4. More Baseline and Backbone Analysis

		Ego-to-Exo		Exo-to-Ego	
Method	Pattern	IoU	Cont. Acc.	IoU	Cont. Acc.
Osprey + READ [7, 11]	zero-shot	4.72	0.082	7.48	0.105
Osprey + LISA [4, 11]	zero-shot	5.75	0.113	9.75	0.171
BAM [5]	trained	13.88	0.278	19.47	0.236
DCAMA [10]	trained	21.66	0.391	27.42	0.329
Ours	trained	**25.35**	**0.463**	**30.36**	**0.378**

Table 1. Comparison with additional baselines.

To strengthen baseline coverage, we include additional comparisons in Table 1. Specifically, we evaluate a prototype-based VRS method (BAM [5]), and zero-shot VLM-based segmentation approaches, Osprey [11], combined with either LISA [4] or the more recent READ [7] method. All trained models use 10% of the training data

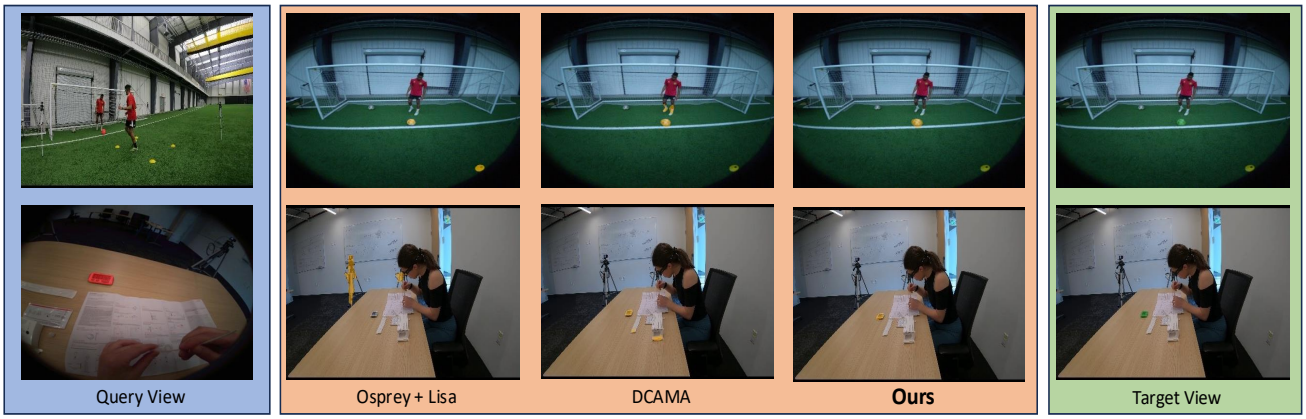

Figure 1. Additional qualitative comparisons of ego-exo correspondence predictions. Input query masks (red) highlight the object of interest (soccer ball, timer from top to bottom, shown in the leftmost frame). Model predictions (orange) from baseline methods and our approach are shown in the next frames. Ground truth target masks (green) are displayed in the rightmost frame. Our method consistently outperforms baseline methods, achieving more accurate segmentation across all scenarios.

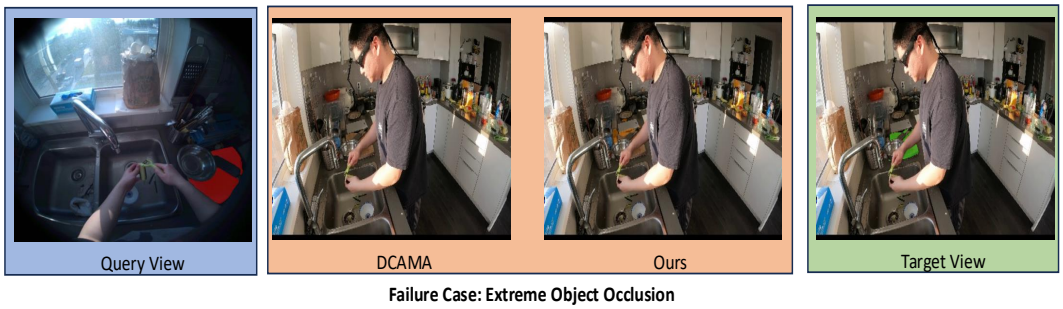

Figure 2. Failure cases. Multiple objects (vegetables, bowls, and an elbow) occlude the chopping board, preventing full recovery by either method.

following our ablation setup and are initialized with COCO-pretrained weights, with evaluation on 10% of the test data.

Among zero-shot methods, Osprey+READ underperforms Osprey+LISA in the ego–exo setting despite READ being a more recent VLM based segmentation method. This suggests that the zero-shot transfer ability of VLM-based segmentation methods remains limited without explicit ego–exo training. Among trained methods, BAM performs significantly worse than DCAMA, indicating that prototype-based VRS methods are less effective in ego–exo setting. Overall, these comparisons validate the soundness of our baseline selection and highlight the strong performance of RegionAligner relative to prior trained methods.

Method	Type	IoU & C. A. (to Exo)		IoU & C. A. (to Ego)	
PSALM [12]	trained	29.61	0.483	37.88	0.459
RegionAligner (PSALM)	**trained**	**32.78**	**0.534**	**38.94**	**0.468**

Table 2. Ablation on two backbones.

Table 2 shows that our core principles (region filtering and guided supervision) can be applied to a different segmentation backbone, PSALM, and still yield a significant performance boost over its vanilla implementation, achieving results that are comparable or even superior to a fully trained DCAMA due to its larger parameter count (1.7B vs. 44M). This confirms the generality and robustness of our proposed techniques. .

5. Unsupervised Scale Analysis

Amount of top K% pairs	Query mask	IoU	Cont. Acc.	Query mask	IoU	Cont. Acc.
0% (zero-shot)	Ego	9.57	0.156	Exo	13.98	0.161
5%	Ego	12.51	0.195	Exo	17.01	0.191
10%	Ego	13.91	0.211	Exo	18.37	0.203
15%	Ego	14.75	0.224	Exo	19.44	0.215
20%	Ego	14.86	0.228	Exo	19.73	0.224

Table 3. Scaling analysis of the unsupervised module.

We further study how the unsupervised module scales as the proportion of pseudo-mask pairs increases. We use top 5% to top 20% of pseudo-mask pairs and evaluate on 10% of testing data. As shown in Table 3, performance improves substantially when increasing pairs from

5% to 15%, demonstrating the benefit of leveraging pseudo-masks. However, gains from 15% to 20% are marginal. This saturation arises because lower-ranked pseudo-mask pairs are noisier, and adding them dilutes supervision quality, leading to limited gains. A potential improvement would be to re-match pseudo-mask pairs with a previously trained model, which we leave as a future direction.

6. Object Size Analysis

Ego-to-Exo	DCAMA [10]		RegionAligner	
Quartile	IoU	Cont. Acc.	IoU	Cont. Acc.
1st quartile (smallest)	2.26	0.122	5.24	0.259
2nd quartile	9.82	0.289	19.15	0.464
3rd quartile	22.85	0.45	32.74	0.593
4th quartile (largest)	47.43	0.623	54.64	0.673
Exo-to-Ego	DCAMA [10]		RegionAligner	
Quartile	IoU	Cont. Acc.	IoU	Cont. Acc.
1st quartile (smallest)	8.92	0.182	13.52	0.246
2nd quartile	18.57	0.273	24.91	0.359
3rd quartile	30.61	0.376	37.42	0.439
4th quartile (largest)	48.15	0.474	54.89	0.559

Table 4. Performance across object size quartiles in the target view. RegionAligner yields larger relative gains on small and mid-sized objects.

We analyze performance across object sizes by sorting target objects into four quartiles by area. Table 4 shows that RegionAligner consistently outperforms DCAMA, with larger relative gains on smaller objects. For example, using ego query mask, the IoU improves from 9.82 to 19.15 in the second quartile, whereas IoU improves from 47.43 to 54.64 in the fourth, showing stronger relative benefits on smaller targets. These results suggest that our framework is especially effective in cluttered settings with small or mid-sized objects, while gains on larger objects are moderate.

7. Generalization Experiments

Method	Query mask	IoU	Cont. Acc.	Query mask	IoU	Cont. Acc.
DCAMA [10]	Ego	22.23	0.397	Exo	26.22	0.329
RegionAligner	Ego	**24.19**	**0.437**	Exo	**28.51**	**0.351**

Table 5. Generalization on cross-time dataset variant of Ego-Exo4D. RegionAligner consistently outperforms DCAMA baseline under temporal misalignment.

While several cross-view datasets [2, 9] are available, Ego-Exo4D is the only one that provides correspondence mask annotations. To further evaluate generalization, we construct a dataset variant using 10% of the data following our ablation setup. To create the cross-time variant, we perturb the synchronization of paired frames by replacing one view with a temporally adjacent frame (e.g., a pair originally aligned at time t is modified so that one view remains at t while the other is shifted to $t \pm 1$). This setting introduces additional challenges, as objects may undergo slight appearance or position changes due to the temporal shift.

As shown in Table 5, RegionAligner achieves clear gains over the DCAMA baseline (e.g., IoU +1.96 using ego query), with consistent improvements in correspondence accuracy. These results suggest that our method is more robust to temporal misalignment, reinforcing its ability to generalize beyond the strict synchronization assumption of the benchmark.

8. Leveraging Temporal Cues

Data	Query mask	IoU	Cont. Acc.	Query mask	IoU	Cont. Acc.
Standard	Ego	25.35	0.463	Exo	30.36	0.378
Standard + Cross-Time	Ego	**27.11**	**0.491**	Exo	**31.57**	**0.391**

Table 6. Performance when incorporating cross-time dataset variant into standard data. Temporal cues yield consistent improvements over the standard setting.

To examine the role of temporal information, we augment the standard 10% training set with the cross-time variant and train our model on the combined data. As shown in Table 6, incorporating cross-time pairs leads to consistent improvements for both ego and exo queries, indicating that even temporally misaligned frames provide useful supervisory signals. This preliminary experiment suggests that temporal cues can enhance performance for a spatial-only model, and motivates future work on dedicated approaches to better exploit temporal dynamics.

9. Efficiency Analysis

Vision–language model (VLM) inference is computationally expensive, so we report the runtime and resource cost in our setup. Using LLaVA-OneVision-0.5B on a single NVIDIA V100 GPU with the standard HuggingFace framework (batch size 1), we observe an average per-frame inference time of 1.8 seconds, requiring roughly 10 GB of GPU memory. Although this appears costly, VLM inference is a one-time preprocessing step. Moreover, with more advanced hardware such as NVIDIA H100 and optimized inference frameworks like vLLM [3] for batch processing, the runtime can be substantially reduced.

For training, our method introduces minimal additional parameters, adding only about 1.5M from the text-fuser module on top of the DCAMA baseline. This corresponds to approximately 1.7% of the total DCAMA model size, including the frozen visual encoder, showing that our approach remains lightweight and efficient. In contrast, fine-tuning large VLMs typically involves training billions of

parameters, making our method orders of magnitude lighter while still delivering strong performance.

References

[1] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024. 1

[2] Yifei Huang, Guo Chen, Jilan Xu, Mingfang Zhang, Lijin Yang, Baoqi Pei, Hongjie Zhang, Lu Dong, Yali Wang, Limin Wang, and Yu Qiao. Egoexolearn: A dataset for bridging asynchronous ego- and exo-centric view of procedural activities in real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22072–22086, 2024. 3

[3] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023. 3

[4] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024. 1

[5] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8057–8067, 2022. 1

[6] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *European Conference on Computer Vision*, 2014. 1

[7] Rui Qian, Xin Yin, and Dejing Dou. Reasoning to attend: Try to understand how <SEG>token works. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24722–24731, 2025. 1

[8] 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. *International Conference on Machine learning*, 2021. 1

[9] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022. 3

[10] Xinyu Shi, Dong Wei, Yu Zhang, Donghuan Lu, Munan Ning, Jiashun Chen, Kai Ma, and Yefeng Zheng. Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation. In *European Conference on Computer Vision*, pages 151–168. Springer, 2022. 1, 3

[11] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28202–28211, 2024. 1

[12] Zheng Zhang, Yeyao Ma, Enming Zhang, and Xiang Bai. Psalm: Pixelwise segmentation with large multi-modal model. In *European Conference on Computer Vision*, pages 74–91. Springer, 2025. 2