ObjectRelator: Enabling Cross-View Object Relation Understanding Across Ego-Centric and Exo-Centric Perspectives

Supplementary Material

We first present the implementation details in Sec. 1, followed by additional results in Sec. 2, Sec. 3, and conclude with a discussion of limitations and future work in Sec. 4.

1. Implementation Details

1.1. Data Processing on Ego-Exo4D

Frame Extraction. We follow the same frame extraction process as the baselines, i.e., XSegTx, XView-Xmem, as provided by Ego-Exo4D [13]. Specifically, for both ego and exo views, we sample one frame every 30 frames in chronological order to ensure time synchronization. Meanwhile, since the resolution of ego and exo is different, we adopt different scaling ratios. For ego video, we scale its resolution from (1408, 1408) to (704, 704). While the exo video is scaled from (2160, 3840) to (540, 960).

Train/Val/Test Sets. As described in Sec. 4, we follow the same dataset splits as Ego-Exo4D [13]. By default, we retain only the ego-exo pairs where the object is visible in both views (in some cases, an object may be visible in one view but fully occluded in the other). When it comes to evaluating the VA, we test models on all cases. To construct the **SmallTrain** set, we sample a subset of the **FullTrain** set at a fixed frequency of 1/3. In the end, for both Ego2Exo and Exo2Ego tasks, we have the **FullTrain**, **SmallTrain**, and **Val** sets. We further report the number of frame images (No. Img) and the average number of objects per frame (Avg. Obj) for each set in Tab. A. All splits used in this work will be released to the community to facilitate reproduction and comparison.

	Ego2Exo			Exo2Ego		
	FullTrain	SmallTrain	Val	FullTrain	SmallTrain	Val
No. Img	110118	36706	31205	123381	41127	36073
Avg. Obj	2.4	2.4	2.4	2.3	2.3	2.3

Table A. Number of images and average object per image of sets.

1.2. Data Processing on HANDAL-X

We adapt the HANDAL-X from HANDAL [14] as a new cross-view image segmentation benchmark. The vanilla HANDAL dataset is designed for category-level object pose estimation and affordance prediction, specifically tailored for robotics applications. It emphasizes manipulable objects that are ideal for robot manipulators to grasp, such as pliers, utensils, and screwdrivers. The dataset comprises 308,000 annotated image frames extracted from 2,200 videos featuring 212 distinct objects categorized into 17 groups.

The HANDAL dataset provides multi-view images capturing objects from a full 360-degree perspective, along with object-centered masks, enabling the construction of a cross-view dataset. Specifically, HANDAL defines a training and testing split for each object, and we adopt the same partitioning strategy for HANDAL-X. To enhance viewpoint variation, we construct query-target pairs at 100-frame intervals within each object's training set. The same approach is applied to the test set. We then aggregate the training pairs from all objects to form the final training set, and similarly, we merge all test pairs to create the final test set. As a result, our final HANDAL-X dataset consists of 44,202 training samples and 14,074 test samples.

1.3. Description Generation

Using LLaVA for Object Descriptions. Due to the lack of semantic information in the initial benchmark, we propose to utilize the pretrained vision language model LLaVA [40] for generating the text description to support the subsequence MCFuse module. Concretely, we merge the mask of the object of interest with the correspondence image to create the input. This approach not only directs the model's attention to the masked area but also preserves the full contextual information, such as the background, which aids the model in better understanding the object. As for the text prompt, we use *"Identify the single object covered by the green mask without describing it. Note that it is not a hand. Format your answer as follows: The object covered by the green mask is"*. By providing this text prompt and the masked image to LLaVA, we generate the corresponding object descriptions.

Comparison Among Different Options. We compare our way of generating text descriptions with two other options. Visual examples are provided in Fig. A. Specifically: 1) **Without Object Masking (Fig. A(a)):** The entire image is used as input without adding an object mask, and the prompt is: *"Could you help describe the main object of the image?"*. Results show that when the image contains multiple objects, the model struggles to identify the correct one. Attempts to zoom in on the target object lead to reduced resolution, making the image difficult for the model to interpret. 2) **With Object Masking but different prompt (Fig. A(b)):** The object is masked, and the model is prompted with *"Please describe the object covered by the green mask."* While this approach focuses the model's attention on the masked area, it often incorrectly predicts the object as a human hand, which is not considered an object of

(a) option1

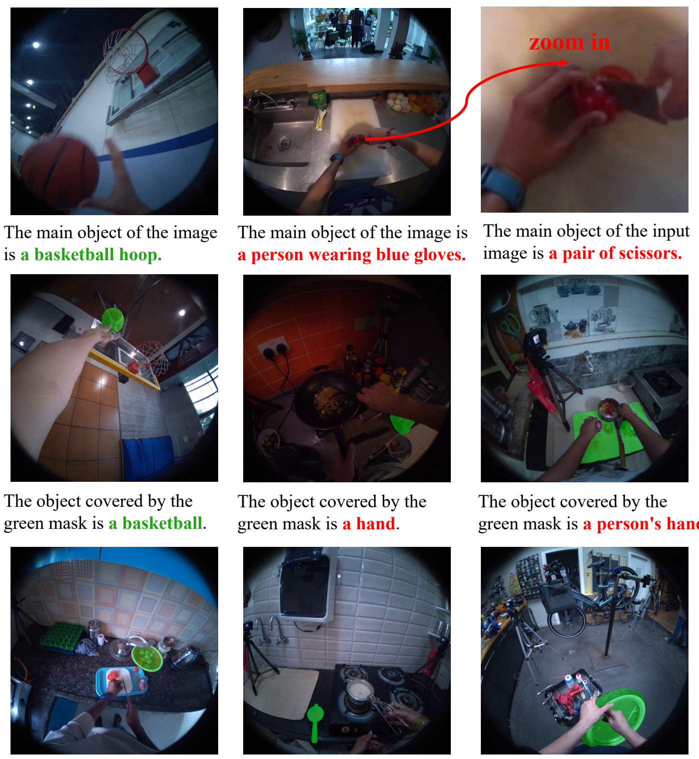
"Could you help describe the main object of the image?"

(b) option2

"Please describe the object covered by the green mask."

(c) ours

"Identify the single object covered by the green mask without describing it. Note that it is not a hand. Format your answer as follows: The object covered by the green mask is"



The main object of the image is **a basketball hoop.**

The main object of the image is **a person wearing blue gloves.**

The main object of the input image is **a pair of scissors.**

The object covered by the green mask is **a basketball**.

The object covered by the green mask is **a hand**.

The object covered by the green mask is **a person's hand**.

The object covered by the green mask is **a bowl**.

The object covered by the green mask is **a spoon**.

The object covered by the green mask is **a bicycle wheel**.

Figure A. Comparisons among various options for text generation.

interest in this benchmark. 3) **Our Approach (Fig. A(c)):** We use an object-masked image as input and a prompt that explicitly excludes the human hand. Results show significant improvement, as the generated descriptions align with the target object categories in most cases.

Error Rate and Ambiguity Robustness Analysis. To further quantitatively assess the quality of our generated text descriptions, we randomly sample 50 frames from each scenario and evaluate both error rate and ambiguity robustness. For the error rate, a description is considered correct if it accurately reflects the object in the image; otherwise, it is counted as an error. We report results using both human and GPT-4o as judges. For ambiguity robustness, we run inference using our trained model on the subset of samples where the descriptions were judged correct by humans. We then report the IoU scores and compare them with those obtained using ground-truth text as input. Results are summarized in Tab. B.

From the results, we acknowledge that the generated descriptions are not always perfect, with an accuracy around 80%. Given the complexity of ego-centric frames, such as small objects and motion blur, the overall quality remains strong. As shown in Tab. 2 and Tab. 3, the auto-generated text descriptions significantly contribute to performance improvements. Furthermore, the ambiguity robustness results indicate that our MCFuse module performs well.

Evaluation Type	Method	Metrics
Error Rate	Human Judge	22.1%
	GPT-4o Judge	16.9%
Ambiguity Robust	Our predicted text	54.45 IoU
	Ego-Exo4D GT text	55.44 IoU

Table B. Study on the quantity of generated text descriptions.

1.4. Training Details

Almost all models, including the base retrained PSALM, our ObjectRelator, and the ablation studies of ObjectRelator, share the same basic training setup: the pretrained PSALM model [75] is used as the initialization, AdamW is employed as the optimizer, the learning rate is select from 6e-5, 2e-4 with a cosine decay scheduler, the batch size is 12, and the image size is 1024×1024. By default, models without MCFuse are trained in a single stage (4 epochs) using all the data in the current training set. In contrast, models with MCFuse, such as our ObjectRelator, employ two training stages, **S1** and **S2**, with different epoch settings and data usage. The specific training configurations for our ObjectRelator are summarized in Tab. C.

	Ego2Exo/Exo2Ego/HANDAL-X Training	Joint Training
S1	$^{1}/_{20}$ set, epoch = 4	$^{1}/_{20}$ set, epoch = 3
S2	all set, epoch = 4	all set, epoch = 3

Table C. Training settings for ObjectRelator.

All training runs on 4xA100 GPUs, while testing is conducted on a single A100, A6000, or L4. Training on Ego-Exo4D Small TrainSet or HANDAL-X takes 5-6 hours, while Ego-Exo4D Full TrainSet takes around 20 hours.

2. More Results

2.1. More Ablations on MCFuse

To thoroughly assess the optimality of our design, we conduct additional ablations applying various options to MCFuse. These include testing: whether cross attention (CA) is the best method for fusing conditions; the impact of adopting a learnable residual connection; the usage of the learnable weight; and the most suitable placement for applying MCFuse. The results are presented in Tab. D.

Ablation Factor	Method	IoU↑
Fusion	Add	42.6
	CA w/o Params	42.1
	CA, S2	22.0
	CA+LearnResidual, S2	41.3
	CA+LearnResidual, S1-2 (Ours)	**43.2**
Residual Weight	$k_{lea} = 0.2$	42.1
	$k_{lea} = 0.5$	41.7
	$k_{lea} = 0.8$	43.1
	k_{lea} (Ours)	**43.2**
Placement of MCFuse	Before XObjAlign	43.6
	After XObjAlign (Ours)	**44.3**

Table D. Ablation study on MCFuse. Models are trained on Ego2Exo Small TrainSet.

For the **fusion**-related experiments, we designed four variants: "Add", "CA w/o Params", "CA, S2", "CA + LearnResidual, S2". The "Add" method simply adds the ego text and visual conditions as an easy test of our generated text descriptions and also the idea of multi-condition fusion. The "CA w/o Params" variant applies cross-attention without parameters, formulated as $(E_T^{ego} \cdot E_V^{egoT}) \cdot E_V^{ego}$, which helps evaluate model sensitivity to added parameters. "CA, S2" represents standard cross-attention fusion, as shown in Eq. 3.2. "CA + LearnResidual, S2" uses our MCFuse module. The "S2" denotes that the entire network is trained only once (i.e., our second training stage). Results show: 1) Even simple methods like "Add" and "CA w/o Params" improve upon the base model, validating MCFuse's core concept. 2) The notable drop with "CA, S2" suggests caution when introducing new parameters into other pre-trained modules, motivating our two-stage training strategy. This strategy is further validated by comparing "CA + LearnResidual, S2" with our MCFuse. 3) The improvement from "CA, S2" to "CA + LearnResidual, S2" supports protecting the more reliable visual prompt condition. For the **residual weight** experiments, we compare our learnable approach with fixed weights k_{lea} (0.2, 0.5, 0.8). Results indicate that learnable weight k_{lea} reduces manual tuning while effectively adapting to appropriate values. Regarding the **placement of MCFuse**, we compare put MCFuse "Before XObjAlign" with "After XObjAlign". Results show the latter is more effective, likely because applying the alignment loss $\mathcal{L}_{XObj}$ upon MCFuse may skew optimization toward better fusion objectives.

2.2. More Ablations on XObjAlign

Additionally, we provide further ablation studies on the XObjAlign module, including various metrics for computing the consistency constraint and different weights for $\mathcal{L}_{XObj}$. Particularly, as in Tab. E, we first compare the performance of using the Euclidean loss for XObjAlign with that of using cosine similarity, and then compare different weights for composing the sublosses $\mathcal{L}_{mask}$ and $\mathcal{L}_{XObj}$. Note that only the XObjAlign module is applied.

Ablation Factor	Method	IoU↑
-	Base PSALM [75]	39.7
Metrics for XObjAlign	Cosine	42.5
	Euclidean (Ours)	**43.8**
Loss Weights	$\mathcal{L} = \mathcal{L}_{mask} + 0.2 * \mathcal{L}_{XObj}$	41.2
	$\mathcal{L} = \mathcal{L}_{mask} + 0.5 * \mathcal{L}_{XObj}$	42.0
	$\mathcal{L} = \mathcal{L}_{mask} + \mathcal{L}_{XObj}$ (Ours)	**43.8**
	$\mathcal{L} = \mathcal{L}_{mask} + 10 * \mathcal{L}_{XObj}$	40.3

Table E. More ablation studies on XObjAlign and loss functions. Models are trained on Ego2Exo Small TrainSet.

Results show that: 1) Compared to the cosine similarity, the Euclidean loss is better. However, both of them clearly outperform the base PSALM. 2) Among different choices, the default one $\mathcal{L} = \mathcal{L}_{mask} + \mathcal{L}_{XObj}$ achieves the best result. The consistent improvement over the baseline observed across other weight ratios further demonstrates the robust positive impact of our XObjAlign module.

2.3. First-Frame Query Evaluation

Ego-Exo4D provides frame-level correspondence between ego and exo videos, while such extensive annotation might be impractical in the real world. Thus, we also test ObjectRelator under a more realistic scenario where only a first-frame query is provided. To adapt our model to this setting, we introduce a memory mechanism during inference. Taking Ego2Exo as an example, beyond the first frame—where the ego object mask serves as the query—the predicted results from previous exo frames are used as queries for subsequent exo frames. The results are presented in Tab. F.

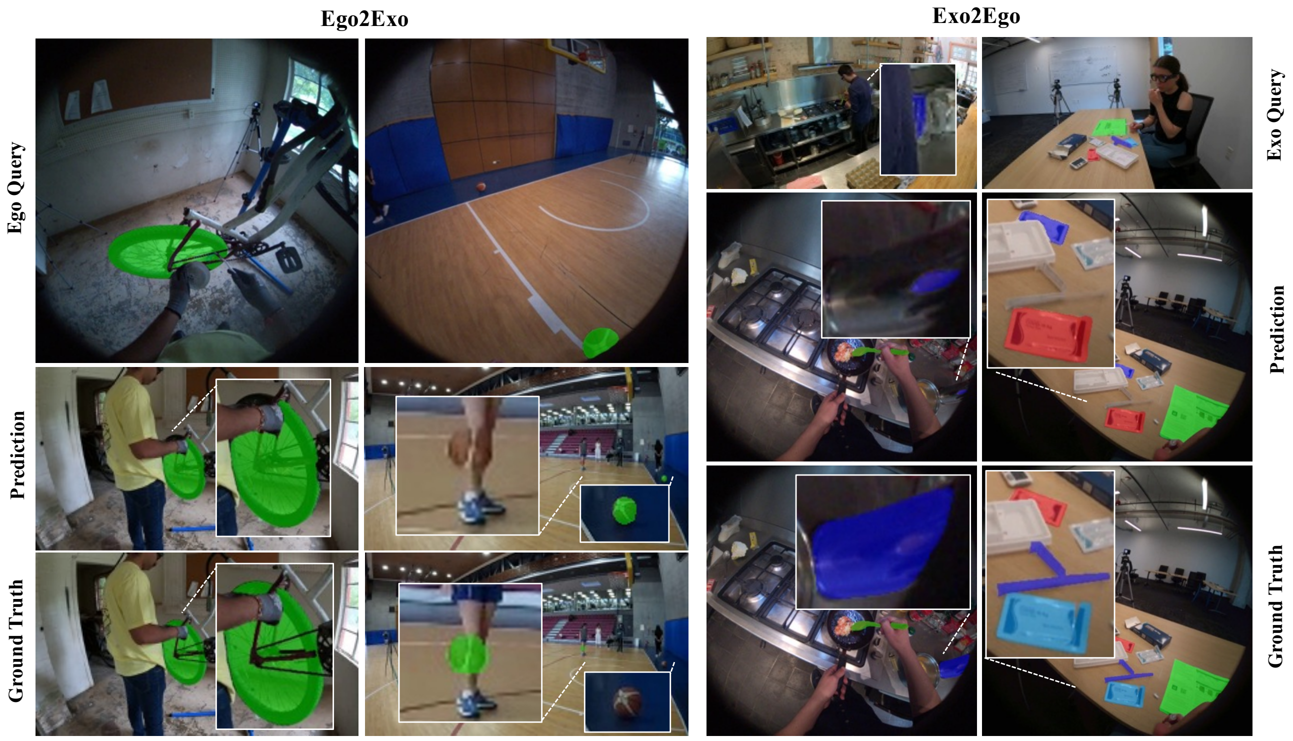


Figure B. Visualization of failure cases.

Method	Testing Query	Ego2Exo↑	Exo2Ego↑
ObjectRelator	Frame Level (Ours)	44.3	49.2
	1-st Frame + Memory	37.0	43.7

Table F. ObjectRelator with all frame-level queries vs. only 1st frame query. Model trained on Small TrainSet is tested.

Results show that though not as good as the frame-level queries, our ObjectRelator could still work under this case.

2.4. More Visualization Results

Due to space limitations, only a few examples are presented in Fig.4. We include additional visualization results in Fig.C (Ego2Exo) and Fig. D (Exo2Ego). For each subtask, we showcase diverse examples spanning all six scenarios: Cooking, BikeRepair, Health, Music, Basketball, and Soccer. Results demonstrate that our proposed ObjectRelator effectively segments cross-view objects in most cases, producing results that closely align with ground truth annotations. This highlights the robustness of our method in handling diverse scenarios, various object categories, and challenging cases, such as occlusion and novel viewpoints. We also provide visualization results (Fig. E) on HANDAL-X. 15 examples across diverse categories are demonstrated. Our model accurately predicts masks matching the ground truth, with only a few failures highlighted in red circles.

3. Failure Cases

We also analyze the failure cases produced by our method, with typical examples summarized in Fig. B. Results indicate that our method struggles in several scenarios: it fails to generate a complete mask when the object's surface is discontinuous or blends closely with the background (first and third columns), incorrectly identifies the object when multiple similar objects are present in the scene (second and fourth columns), and occasionally misses some objects entirely (fourth column).

4. Limitations and Future Work

In this paper, we propose ObjectRelator, a method designed to understand cross-view object relationships in terms of segmentation, validated on ego-exo perspectives and a relatively easier cross-view dataset. The approach primarily consists of multimodal condition fusion and SSL-based cross-view object alignment, built on top of a frame-level multimodal segmentation model. While our method achieves SOTA results, as demonstrated in Fig. B, there still remains significant room for improvement. This underscores the substantial challenges and rich opportunities in addressing ego-exo object correspondence. Looking ahead, we plan to explore the integration of temporal information to better capture object dynamics.

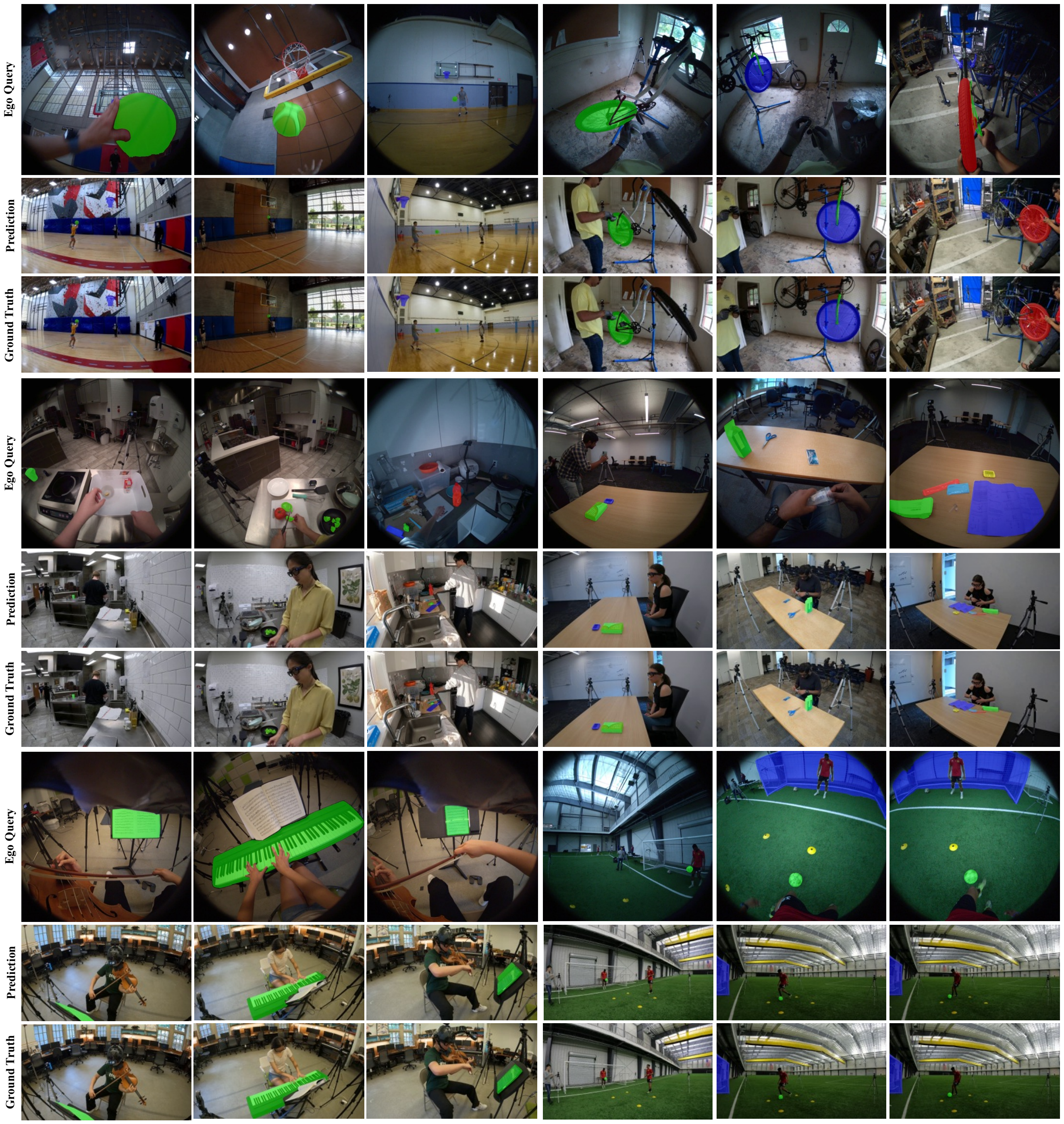


Figure C. Ego2Exo visualization results (ego query, predictions, and ground truth). Ego2Exo model trained on SmallTrain set is used.

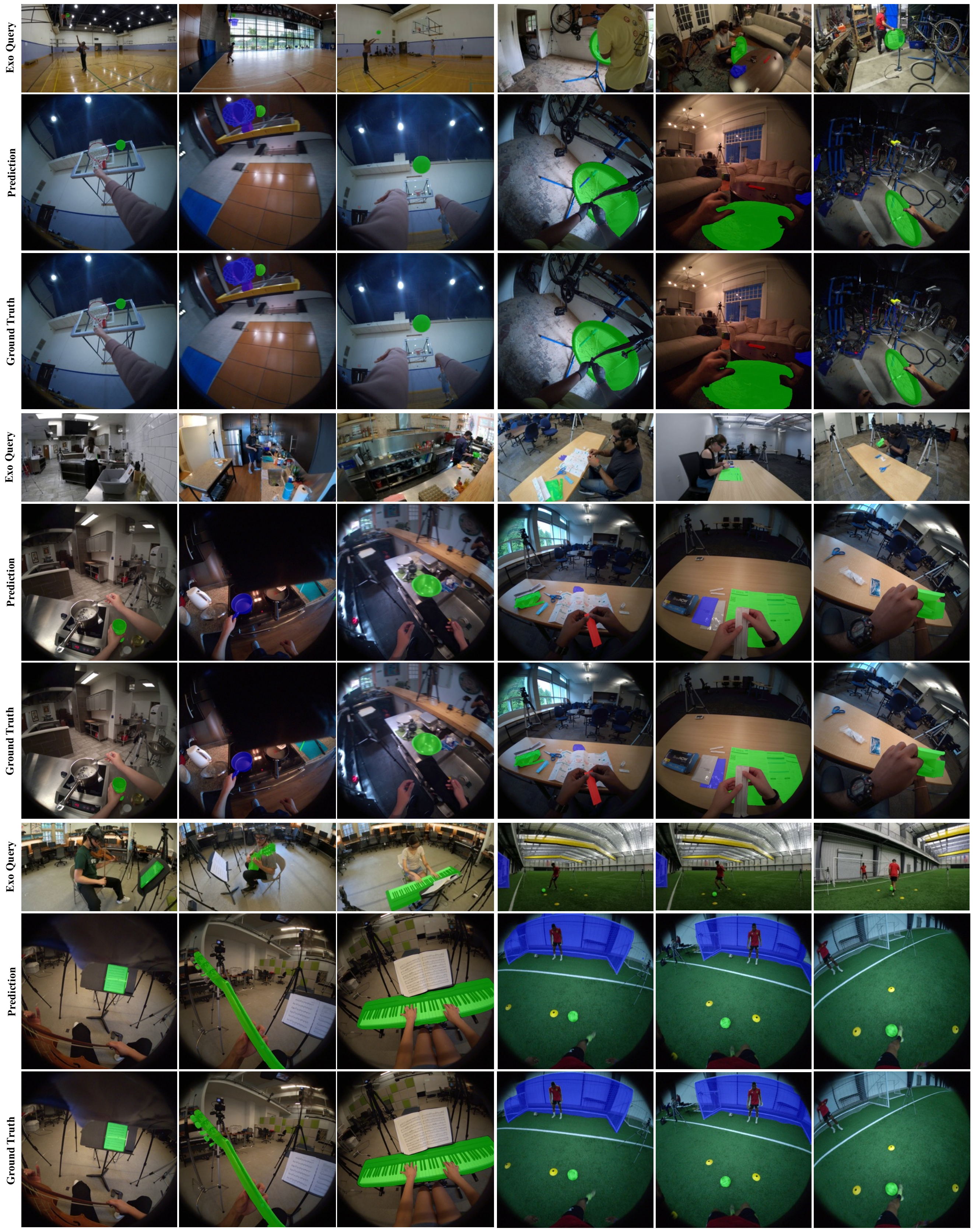


Figure D. Exo2Ego visualization results (exo query, predictions, and ground truth). Exo2Ego model trained on SmallTrain set is used.

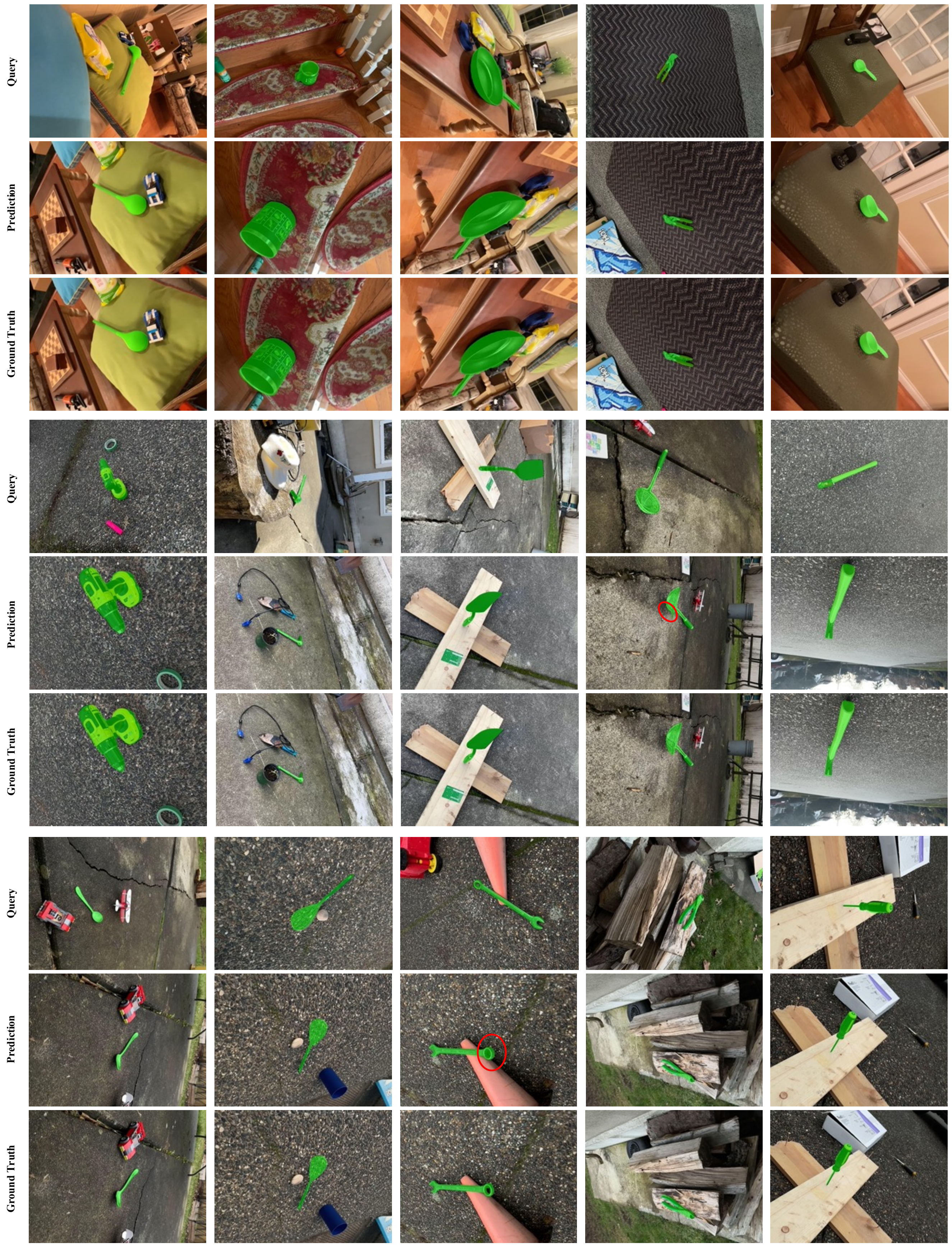


Figure E. HANDAL-X visualization results (query, predictions, and ground truth).