Towards Exo-Ego Correspondence: A Technical Review of the State of The Art

Featuring Ego-Exo4D and Object Mask Matching (O-MaMa)

Marco Lomele[1*] Giovanni Mantovani[1*] Filippo Dario Paolucci[1*]

[1]Bocconi University

{marco.lomele,giovanni.mantovani,filippo.dariopaolucci}@studbocconi.com

Abstract

Understanding how to relate objects across egocentric and exocentric viewpoints is a fundamental challenge in multi-camera perception. In this work, we build upon O-MaMa, the state-of-the-art method that reformulates cross-view segmentation as an object mask matching problem, leveraging FastSAM for mask proposals and DINOv2 features for semantic alignment.

Our contribution is a technical exploration of O-MaMa composed of four parts. First, we construct a pipeline that leverages feature pre-extraction, which reduces by 10x training and inference time. Second, we evaluate the impact of finetuning on a subset of Ego-Exo4D's Health scenario dataset and discover the need for a nuanced sampling strategy. Third, we perform a systematic backbone comparison, replacing DINOv2 with two alternatives: DINOv3 and ResNet50+DINOv1. Finally, we study model robustness under a small handmade photo set featuring visual challenges like object variety and partial occlusion.

Overall, our analyses expose important trade-offs between efficiency, finetuning stability, and robustness, offering insights toward developing general cross-view perception systems.

1. Introduction

Egocentric and exocentric visual perspectives provide complementary information about human activities and their surrounding environment. Egocentric (first-person) cameras capture fine-grained details of hand-object interactions, but they suffer from self-occlusion and a restricted field of view. Conversely, exocentric (third-person) cameras provide a global understanding of a scene, but miss subtle interaction details due to distance, object scale variability, and limited resolution.

*Equal contribution. Code available at https://github.com/marcolomele/exo-ego-correspondence.

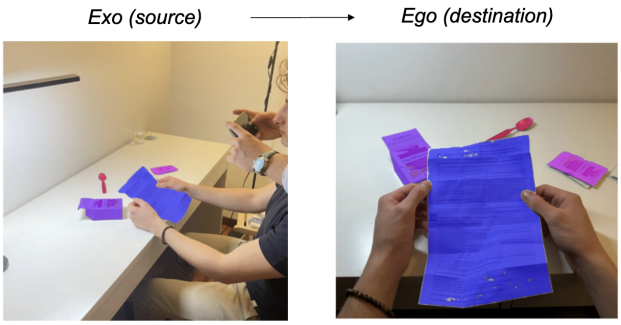


Figure 1. The correspondence direction we are interested in.

Bridging these two perspectives is crucial for applications in augmented reality, robotics, and multi-camera perception systems. The **Ego-Exo4D dataset** [8] introduces several benchmarks for such cross-view understanding, including the **Correspondence task**: given a mask of an object in one view, predict its corresponding mask in the other synchronized view. This setting is significantly more challenging than conventional segmentation or tracking due to drastic viewpoint changes, occlusions, and heterogeneous imaging conditions.

O-MaMa [12] reframes cross-view object correspondence as an object mask matching problem. Instead of generating a segmentation mask directly in the target view, O-MaMa compares a source mask against a set of mask candidates. This formulation achieves state-of-the-art results at the time of writing, while requiring only 1% of the parameters used by competing transformer-based methods. For instance, ObjectRelator [7] learns view-invariant relational embeddings through large multimodal models, but at the cost of having 1.6 billion parameters, compared to the shallower O-MaMa with just 11.6 million for training.

In this project, we inspect O-MaMa with the goal of understanding its practical limitations, robustness, and sensitivity to architectural choices. Specifically, we focus on the **Exo→Ego direction** of the task: given an exocentric object mask, identify the corresponding object in the egocentric frame and produce its mask.

Figure 2. Original O-MaMa architecture performance on three occlusion scenarios, from left to right: absent, partial, complete.

This scenario reflects real-world setups where an external cameras inform the egocentric device (e.g., AR glasses) with information that otherwise is lost due to self-occlusion and limited field of view.

Our technical exploration is composed of four sections. First, we introduce a feature pre-extraction pipeline to remove the computational bottleneck of repeatedly encoding camera views with heavy vision transformers such as DINOv2, achieving 10x improvements in run time. Second, we analyze the effects of finetuning O-MaMa on a subset of the Ego-Exo4D Health scenario, discovering that naïve finetuning leads to strong overfitting. Third, we study the impact of the backbone choice by replacing DINOv2 with DINOv3 and a ResNet50-based representation, providing insights into the trade-offs between efficiency and accuracy. Finally, we evaluate robustness using a custom dataset with visual challenges like partial occlusion and unseen objects.

Through our analyses we reveal several characteristics about O-MaMa, including dependence on the backbone feature extractor and sensitivity to overfitting when finetuning on a non-representative subset of the data.

2. Related Works

2.1. Competing Approaches

The Ego-Exo4D correspondence benchmark formalizes the task as a challenge. Two methods are used as baselines. XSegTx [2] matches objects frame-by-frame, using a ResNet50 [9] backbone pretrained with MoCoV2 [5] and a cross-transformer module to compare image segments between ego and exo views independently.

XView-XMem [3] instead processes entire videos, maintaining a memory of past frames to propagate object masks over time through XMem's attention-based architecture.

Various attempts of the challenge improve localization by combining visual features with text descriptions, leveraging geometry cues, or jointly reasoning over multiple objects. The leading approach in this category is ObjectRelator [7].

A common limitation across these methods is their reliance on predefined object categories or specialized inputs, restricting generalization to open-world scenarios. Additionally, all models suffer of growing parameter size, which limits their application on commercial hardware.

O-MaMa [12] reformulates the correspondence problem as mask matching: the source mask is compared against FastSAM [15]-generated candidates using dense DINOv2 features. This formulation achieves state-of-the-art performance while using only a small number of trainable parameters relative to the competition.

2.2. Contrastive Learning

Contrastive objectives are commonly used in this context. They encourage points of corresponding regions to be close in the embedding space, while pushing dissimilar points apart. Such losses have proven effective in viewpoint-invariant representation learning and in local feature matching when geometric information is unavailable. O-MaMa uses contrastive alignment to bind exocentric and egocentric object-level descriptors. Therefore, performance is highly dependent on the quality and invariance of the features.

2.3. Segmentation and Feature Representation

Cross-view mask matching relies on proposals in the target frame. FastSAM offers fast zero-shot segmentation, generating candidate masks from which the model retrieves

the best match. Unlike class-specific detectors, FastSAM readily produces unsupervised proposals by distilling the SAM [10] into a lightweight CNN. This open-world capability is essential for ego-exo correspondence, where objects of interest are not known in advance and may vary significantly across takes.

Vision Transformers [6] provide dense descriptors with strong view transfer. Self-supervised pretraining on large image corpora allows these models to learn semantically meaningful features that remain consistent under viewpoint and appearance changes, common in exo-ego data.

Motivated by its relevance, part of our analysis will inspect the tradeoff between feature representation quality and computational cost by running O-MaMa on two additional backbones: DINOv3 [14] and ResNet50-DINOv1.

3. Data

3.1. Ego-Exo4D Dataset

We use the Ego-Exo4D dataset [8], which provides synchronized egocentric and exocentric camera streams recorded across a variety of scenarios. Each annotated frame includes RGB imagery for all active cameras, sparse object masks, and metadata such as annotated frame indices and camera identities. We restrict our study to the Health scenario, specifically the "Covid-19 Rapid Antigen Test" task, as it provides 20+ hours of video of consistent scenes with a relatively low variety in object form and high quality annotations on object relations across cameras.

3.2. Processing Pipeline

Download. Following the official processing pipeline, we extract frames only for timestamps with available annotations [1]. Specifically, we download one take at the time, extract each annotated frame I from the feeds of all cameras, and delete the raw take data to manage storage limitations. Ego frames have 1:1 aspect ratio, and exo frames have 16:9 aspect ratio. Masks M are obtained through LZString decompression followed by COCO [11] RLE decoding. This produces aligned pairs of images and masks for each timestamp stored in a take-specific file called `annotation.json`.

Pair Construction. Because our study focuses on the Exo→Ego direction, we construct tuples $(I_{\text{ego}}, M_{\text{ego}}, I_{\text{exo}}, M_{\text{exo}})$ only when the exocentric view contains a valid object mask. This reflects realistic scenarios where the external camera provides more complete context than the egocentric view. Ground-truth egocentric masks are included only when the object is visible either fully or in part. The resulting tuples are the fundamental element over which Mask Matching occurs.

Subset Sampling. We randomly sample 13,000 frames from the Health scenario, 30% of the total, to satisfy storage computing constraints, and then apply a 70-15-15% split. This yields the partition: 9100 for train, 1950 for validation, and 1950 for test. Note that the random sample is executed at the frame-level, implying a core assumption of our technical report: data distribution and object-scene variety are well-represented within the data subset. As we will see in Section 5, this assumption is delicate in complex vision tasks like exo-ego correspondence and requires a nuanced approach.

Custom Dataset. To evaluate the robustness of O-MaMa, we collect a supplementary dataset of 13 exo→ego image pairs. We produce ground truth masks in the ego frames using Segment Anything Model 3, and curate manually the annotations, ensuring object-level pairs exist between ego and exo photos [4]. Extended qualitative examples are provided in the appendix Fig. 12.

4. Methods

In this section, we formalize the Exo→Ego correspondence task, summarize the O-MaMa architecture, and detail the methodological explorations ran during our study: feature pre-extraction, finetuning analysis, and feature extractor backbone substitution. Each modification is designed to investigate O-MaMa's capabilities.

4.1. Problem Formulation

Given a synchronized exocentric image I_{exo} and egocentric image I_{ego}, and a query object mask M_{exo} in the exocentric view, the goal is to predict the corresponding object mask M_{ego} in the egocentric frame:

$$M^*_{\text{ego}} = \arg\max_{m \in \mathcal{C}_{\text{ego}}} \text{sim}(f(M_{\text{exo}}, I_{\text{exo}}),\ f(m, I_{\text{ego}}))\,, \quad (1)$$

where $\mathcal{C}_{\text{ego}}$ is the set of candidate masks in the egocentric frame (generated via FastSAM), $f(\cdot)$ is a mask-context encoder producing object-level embeddings, and sim is a cosine similarity measure.

This formulation reframes the task as finding the best matching mask among ego proposals, instead of directly segmenting the object from scratch.

4.2. O-MaMa Baseline

The Object Mask Matching (O-MaMa) architecture consists of three main components.

1. Mask Proposal Generation. In the target view, FastSAM generates a set of candidate masks at multiple scales. These act as retrieval candidates rather than segmentation outputs.

2. Mask-Context Encoder. For each mask, the encoder pools DINOv2 dense features from (i) the masked region and (ii) the bounding-box context region. The pooled features are concatenated and projected into a shared latent space through a Multi Layer Perceptron (MLP). [13]

3. Contrastive Mask Matching. A contrastive loss encourages embeddings of corresponding masks across views to be similar while enforcing separation from nearby distractors. Formally, for a positive pair $(M_{\text{exo}}, M_{\text{ego}})$,

$$\mathcal{L}_{\text{con}} = -\log \frac{\exp(\text{sim}(z_{\text{exo}}, z_{\text{ego}})/\tau)}{\sum_{m \in \mathcal{C}} \exp(\text{sim}(z_{\text{exo}}, z_m)/\tau)}, \quad (2)$$

with temperature τ and candidate set $\mathcal{C}$ controlling hard negatives selection via adjacency-aware mining. Full O-MaMa architecture presented in Appendix Fig. 13.

4.3. Feature Pre-Extraction

A practical limitation of O-MaMa is the repeated on-the-fly computation of DINOv2 features. Because each tuple is passed through the feature encoder, the backbone is invoked tens of thousands of times per epoch. Additionally, across epochs, features are re-computed for the same frames.

To address this bottleneck we leverage a simple feature pre-extraction strategy. For every frame in the randomly selected subset of the dataset, we compute and store DINO-based feature representations once. Hence, during training, the mask-context encoder retrieves the stored features, and when needed for different feature encoder, performs only lightweight pooling and projections.

Pre-extraction reduces the forward-pass and allows controlled experiments without modifying the training loop, solving the computational bottleneck. Timing profiles when running finetuning on our Azure Virtual Machine (Tesla T4 GPU) reveal how performing DINOv2 feature extraction on-the-fly dominates iteration cost, whereas pre-extracted features reduce the forward-pass time by approximately an order of magnitude.

4.4. Backbone Substitution

To analyze the impact of the feature extractor in the O-MaMa architecture, we replace ViT-B/14 distilled-DINOv2 with two alternatives, then run inference and finetuning under constant regimes, and finally compare metrics.

ViT-S+/16 distilled-DINOv3. A more recent self-supervised transformer with improved visual invariance. We hypothesize improved cross-view consistency due to stronger global semantics. It outputs 384 feature channels per input image.

ResNet50-DINOv1. A lightweight CNN backbone pre-trained with DINOv1 self-supervised learning, removing the final fully-connected layer. This model trades semantic richness for efficiency and acts as a low-cost baseline. It produces 2048 feature channels per input image.

We apply a final linear convolution (Conv2D) that projects each backbone's feature map to the 768 channels required by O-MaMa. In the case of ResNet50-DINOv1, we further apply adaptive pooling to match O-MaMa's spatial grid dimensions (25×25→50×50 for ego, 19×34→38×68 for exo).

All backbones are integrated without altering the O-MaMa architecture: only the feature maps change, while the mask-context encoder and matching head remain identical. This modularity enables controlled comparisons of semantic representation quality, feature granularity and computational efficiency.

4.5. Finetuning

We evaluate whether finetuning improves performance relative to pretrained O-MaMa weights. Specifically, we finetune O-MaMa's cross-view attention blocks, and the Conv2D projection layers when available. All finetuning runs are executed with equal hyperparameters: 10 epochs with learning rate of $8e^{-6}$ and a cosine annealing scheduler.

4.6. Evaluation Metrics

Following the correspondence benchmark, we report the Intersection-over-Union (IoU) between the predicted and ground-truth masks, as it is the most representative measure of correspondence quality. In addition, we report the IoU standard deviation (IoU Std), which quantifies the variability of the model's performance. This metric helps evaluate stability and robustness, especially when comparing different backbones. Further metrics are found in the appendix.

5. Results

Next, we run the experiments and discuss the results. In section 5.1, we validate our implementation of O-MaMa's original architecture with on-the-fly feature computation, and then attempt finetune. From section 5.2 onward, we use pre-extracted features, and abbreviate O-MaMa configurations by the feature encoder, since that is the main changing block across the experiments, with the rest of the architecture kept fixed.

5.1. Reproducing O-MaMa Baseline

We begin by taking the original architecture of O-MaMa with the pre-trained weights provided by the authors and running inference on our health subset. We obtain a IoU of 0.53, 0.09 points above the reported values in the original work. This confirms the correctness of both our data extraction-pairing pipeline and model implementation. Additionally, it validates our assumption: the Health scenario has fairly predictable dynamics relative to other scenarios.

Next, we attempt finetuning of O-MaMa's weights (excluding DINOv2, which is kept frozen) over the randomly selected subset of data. Due to computational limitations, we can run training loops for only 20 hours, which amounts to 3 complete training passes and 2 complete validation passes. Nonetheless, we encounter a striking result: finetuning leads to a decrease in IoU (Tab 1).

Model	IoU ↑	IoU Std ↓
O-MaMa baseline	**0.533**	**0.408**
O-MaMa finetuned	0.526	0.414

Table 1. Comparison of O-MaMa baseline and finetuned variants with on-the-fly feature extraction.

5.2. Feature Pre-Extraction

Pre-extraction prevents repetitive calls to DINOv2 during training. On-the-fly extraction takes $\sim$ 2.5 seconds per iteration, whereas pre-extracted tensors reduce processing to $\sim$ 0.24 seconds (Fig 3). We also observe a dramatic decrease in time spent during the backward propagation of the loss, a topic we leave for future works. For a more detailed timing profile see Appendix Tab 7.

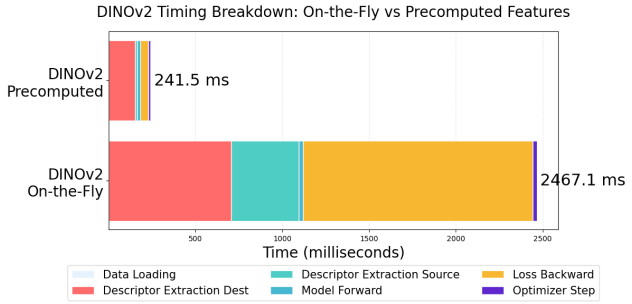


Figure 3. **Iteration runtime with on-the-fly vs. pre-extracted features.** Pre-extraction yields a $\sim 10\times$ speedup.

5.3. Finetuning Analysis

First, we finetune O-MaMa's baseline architecture using the pre-computed features and we compare it with the finetuned on-the-fly counterpart. The pre-extraction results in a notable decrease in the model performance post finetune (Tab 2). Anyhow, we are comparing a model finetuned over 10 epochs vs a model trained over 3 epochs.

Model	IoU ↑	IoU Std ↓
DINOv2 finetuned (precomputed)	0.4108	**0.4044**
DINOv2 finetuned (on-the-fly)	**0.5263**	0.4145

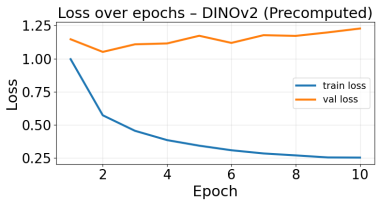
Table 2. **Comparison of precomputed vs. on-the-fly feature extraction.** Feature pre-extraction decreases model performance.

To check the finetune effects on the model performance, we compare O-MaMa with baseline vs. finetuned weights using the DINOv2 pre-extrated features; finetuning significantly decreases IoU (Tab 3).

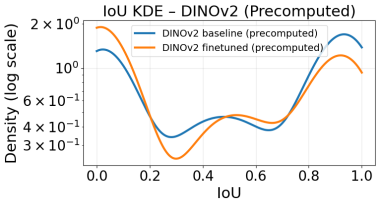
A deeper analysis of the training logs reveals a high degree of overfitting to the training data, with little improvement on the validation (Fig. 4).

Model	IoU ↑	IoU Std ↓
DINOv2 baseline	**0.5262**	**0.4036**
DINOv2 finetuned	0.4108	0.4044

Table 3. **Comparison of baseline vs. finetuned weights for DINOv2 backbone**. Finetuning decreases model performance.



(a) DINOv2 Loss over epochs



(b) DINOv2 IoU KDE

Figure 4. **Loss over epochs and IoU KDE comparison for DINOv2 backbone.**

We then analyze the effect of changing the feature extractor backbone on the model performance. From Tab 4, it can be seen that both alternative models, after model finetuning, perform poorly compared to DINOv2. In particular, DINOv3 presents an especially low IoU parameter, contradicting our prior expectations. The most striking metric is the performance of both feature extractors when the model uses O-MaMa's baseline weights, both of them falling short of 8%. Training-wise, the logs reveal a similar overfitting picture as DINOv2 (Appendix Fig. 5). For additional metrics on models performance see Appendix Fig. 6, Fig. 8, Fig. 9 and Fig. 10.

Model	IoU ↑	IoU Std ↓
DINOv3 baseline	0.0791	0.2405
DINOv3 finetuned	0.2897	0.3819
ResNet50+DINOv1 baseline	0.0757	**0.2315**
ResNet50+DINOv1 finetuned	**0.3647**	0.4140

Table 4. **Comparison of baseline vs. finetuned weights for DINOv3 and ResNet50 backbones.** Clear under performance of baseline weights models.

Three criticalities emerge from our analysis: strong overfitting to the training set, no improvement (and often degradation) in validation IoU, and critical under performance of the model when using baseline weights and pre-extracted

features from DINOv3 and ResNet50-DINOv1. The first two issues are addressed in the next section. The latter is likely a consequence of our methodology: projection layers are trained together during the finetuning of O-MaMa's parameters. When switching back to the baseline, interactions are decoupled, leading to low IoU.

5.4. Investigating Finetuning Failure

We inspect the worsening of performance induced by finetuning by plotting masks on individual frames. Looking at situations of high divergence in IoU between predictions of baseline weights and predictions of finetuned weighs (see Appendix Fig. 7) we uncover two emergent behaviors: "over-specification" and "easy distraction".

Over-specification refers to O-MaMa's tendency to reduce the area of the predicted masks by focusing on smaller portions of objects. This is analogous to tuning FastSAM such that it segments details rather than follow object contours. Easy distraction refers to O-MaMa exhibiting a distraction mechanism, segmenting the object that is at the center of the frame under manipulation, rather than sticking to the actual object it should predict. See Appendix Fig. 11 for detailed examples.

We provide three conjectures that explain such results. First, our starting assumption of the data distribution and object-scene variety in the subset of the Health scenario dataset failed. The set over which we finetuned O-MaMa was too restrictive, meaning that at inference, the model encountered unseen objects. We confirm this thesis by looking at object label overlap across train and test annotations, which we find to be minimal.

Second, the contrastive loss function corrupted the feature representation by over-emphasizing the minimal object variety of the train set. This lead to space collapse and spurious learning that took the form of the described emergent phenomena.

Finally, we observed catastrophic forgetting. Borrowed from the Natural Language Processing field, it refers to the degradation of previously learned capabilities when a model is fine-tuned on a new task. O-MaMa's baseline weights, initially trained across multiple scenarios, were fine-tuned on a subset of the Health scenario. The domain shift between the full training distribution and a idiosyncratic Health subset led to performance degradation of the original task.

5.5. Robustness to Unseen Data

We test inference of the top two model configurations on our hand-made dataset. We find that finetuned models outperform models with pretrained weights (Tab 5). However, absolute IoUs remain comparable to previous findings, and qualitative results reveal frequent failures under heavy occlusion. See Appendix Fig. 12 for examples. The difference in performance, that counters the other findings of our report, is likely explained by a coincidental resemblance between our dataset and the finetuning dataset.

Model	IoU ↑	IoU Std ↓
DINOv2 baseline	0.4466	0.4307
DINOv2 finetuned	**0.5389**	**0.4275**

Table 5. **DINOv2 baseline vs. finetuned performance on custom data.** The small-scale evaluation on the custom data shows a modest improvement from finetune.

6. Conclusion

In this technical review, we visited the ego-exo correspondence problem by systematically analyzing the O-MaMa framework, focusing on the Exo→Ego setting, using Ego-Exo4D's Health scenario dataset. Through feature pre-extraction, backbone substitutions, and fine-tuning experiments, we exposed key performance determinants and practical limitations. Below, we highlight three main insights.

First, data sampling within exo-ego correspondence datasets requires a nuanced object-level approach, rather than an intuitive frame-level. This ensures that object-scene variances are maintained, providing solid grounds for generalization.

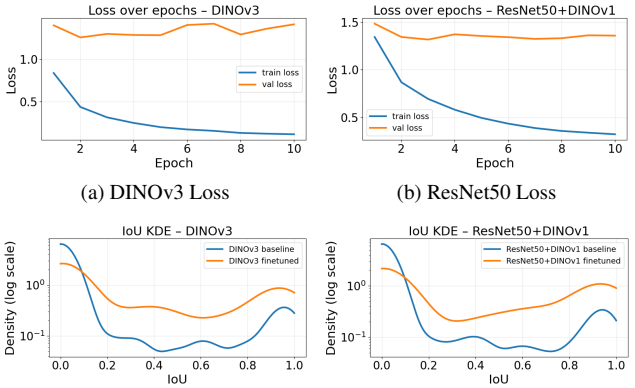
Second, the O-MaMa architecture is prone to overfitting. Finetuning the original architecture with feature extractor DINOv2 on a subset of 30% of the frames for 10 epochs produced a significant degradation in performance. Hence, correspondence accuracy is highly sensitive to backbone design. Given a smarter sampling strategy and more powerful computing resources, finetuning with ResNet50+DINOv1 and DINOv3 remains promising research. As it stands, pretrained DINOv2 features are the most reliable backbone for state of the art exo-ego correspondence.

Finally, O-MaMa exhibits robustness over unseen data, but struggles significantly with moderate to heavy occlusion. This emphasizes how the current mask-matching pipeline relies on visible cues in both exo and ego point of view. They ignore information about missing object parts, are unable to leverage scene geometry, and fail to reason around nuanced interactions among objects. In contrast, we humans are able to infer object positions even when complete occlusion appear, because of our ability to track space and time conjunctively, while only relying on the ego point of view.

These observations suggest several promising directions for future research: incorporate explicit occlusion modeling, integrate temporal cues by studying mask temporal consistency, and expand train and test sets by combining other multi-view datasets, such as EgoExOR [16].

7. Appendix

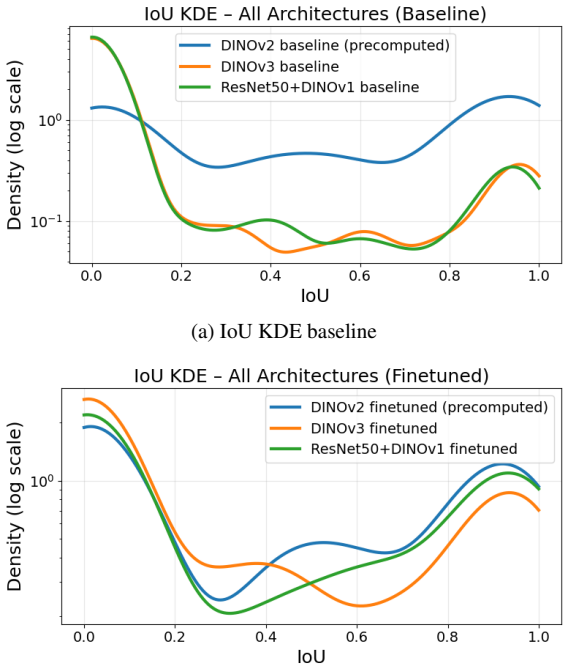
This appendix provides extended quantitative analyses, diagnostics, and qualitative results referenced in the main report.



(a) DINOv3 Loss (b) ResNet50 Loss

(c) DINOv3 IoU KDE (d) ResNet50 IoU KDE

Figure 5. **Loss over epochs and IoU KDE comparison for DINOv3 and ResNet50 backbones.**



(a) IoU KDE baseline

(b) IoU KDE finetune

Figure 6. **IoU distribution between architectures.**

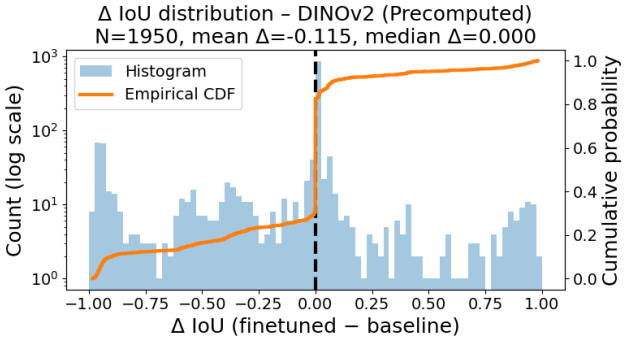
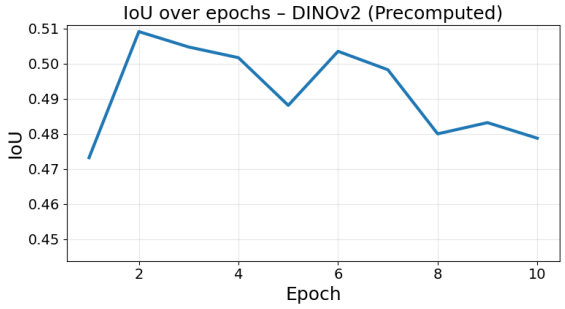
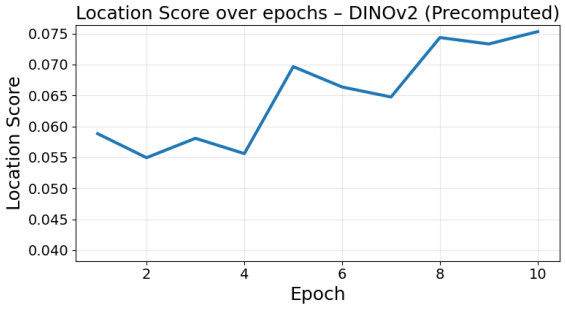


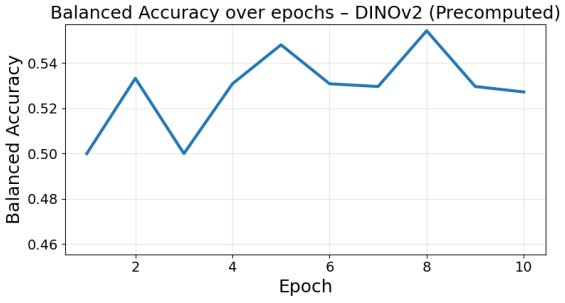
Figure 7. **Δ IoU distribution and Cumulative Distribution.**



(a) IoU over epochs



(b) Location Score over epochs

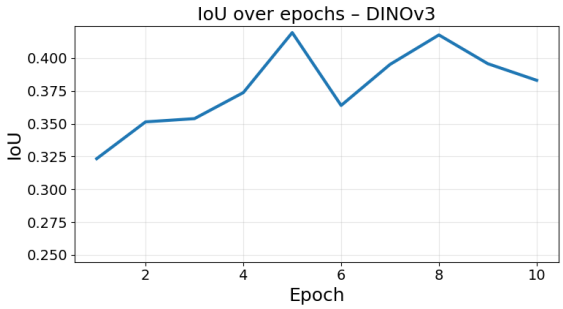


(c) Balanced Accuracy over epochs

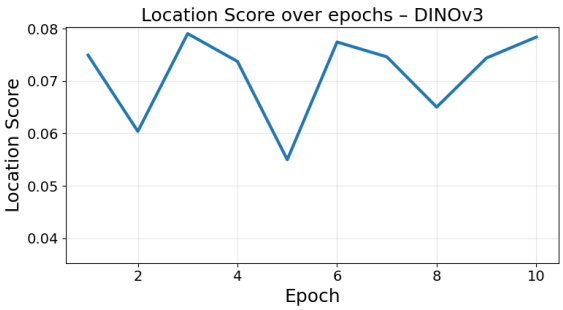
Figure 8. **DINOv2 training metrics.**

Model	IoU ↑	IoU Std ↓	Shape Acc ↑	Location Score ↑	Balanced Acc ↑
DINOv2 baseline (precomputed)	**0.5262**	**0.4036**	**0.6241**	0.0375	0.5193
DINOv2 finetuned (precomputed)	0.4108	0.4044	0.4965	**0.0534**	**0.5386**
DINOv3 baseline	0.0791	**0.2405**	0.1171	**0.1799**	0.5000
DINOv3 finetuned	**0.2897**	0.3819	**0.3869**	0.0836	0.5000
ResNet50+DINOv1 baseline	0.0757	**0.2315**	0.1061	**0.2235**	0.5000
ResNet50+DINOv1 finetuned	**0.3647**	**0.4140**	**0.4225**	0.0775	**0.5287**
DINOv2 on-the-fly baseline	**0.5332**	**0.4084**	**0.5905**	0.0490	0.5224
DINOv2 on-the-fly finetuned	0.5263	0.4145	0.5891	**0.0502**	**0.5944**

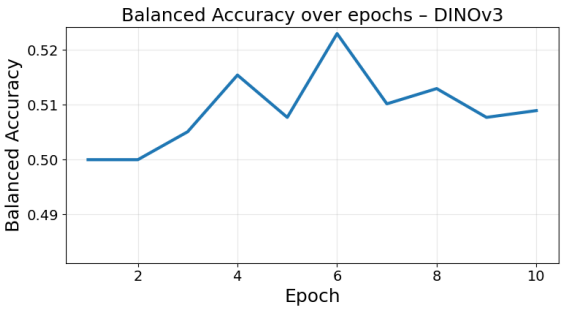
Table 6. **All models evaluation summary.** Comparison of different architectures and training configurations.



(a) IoU over epochs

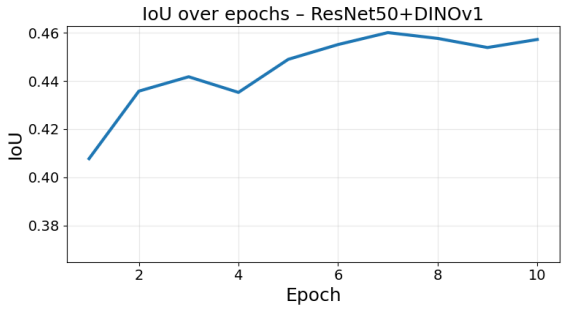


(b) Location Score over epochs

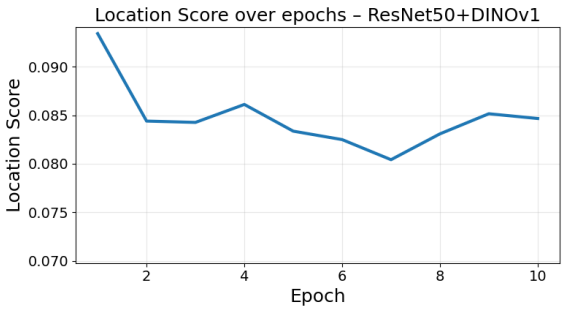


(c) Balanced Accuracy over epochs

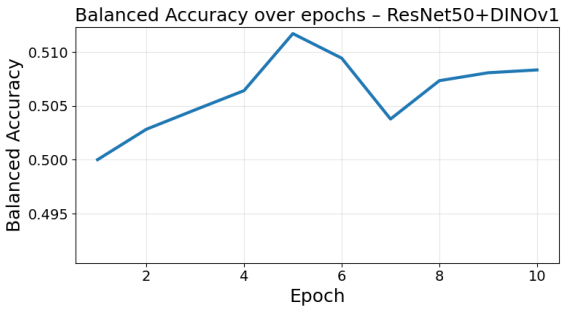
Figure 9. **DINOv3 training metrics.**



(a) IoU over epochs

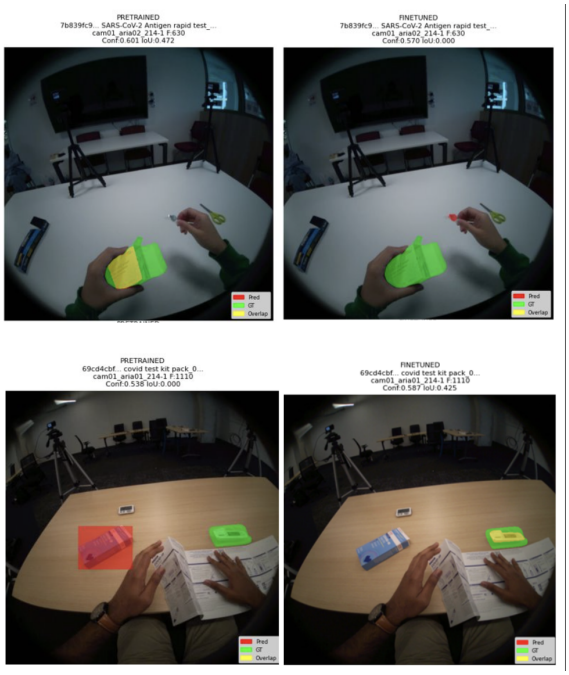


(b) Location Score over epochs

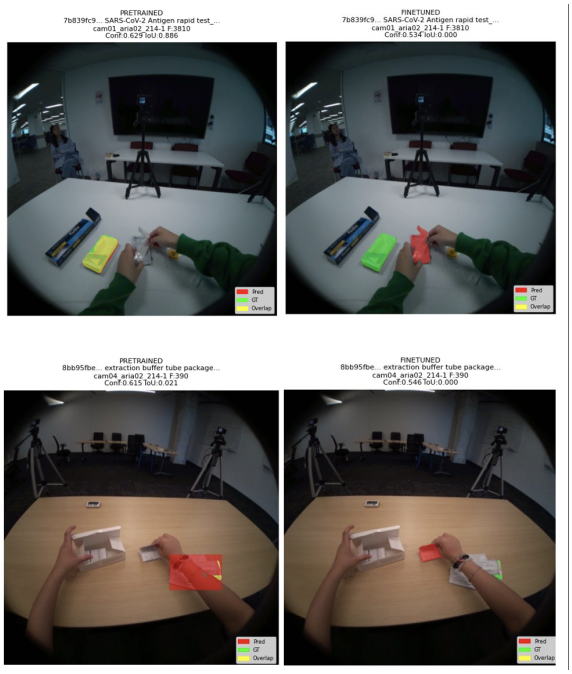


(c) Balanced Accuracy over epochs

Figure 10. **ResNet50-DINOv1 training metrics.**



(a) Over-specificity phenomenon (left are pretrained, right are finetuned)



(b) Easy distraction phenomenon (left are pretrained, right are finetuned)

Figure 11. **Emergent phenomena post finetuning for DINOv2.**

Operation	Precomputed (ms)	On-the-Fly (ms)
HIGH-LEVEL TIMINGS		
data_loading	0.0004	0.0008
descriptor_extraction_dest	153.01	702.08
descriptor_extraction_source	12.79	390.15
model_forward	16.61	22.09
loss_backward	46.84	1318.96
optimizer_step	12.21	22.62
total_iteration	**242.26**	**2467.10**
MODEL FORWARD PASS BREAKDOWN		
extract_object_descriptors	0.013	0.025
dest_dense_feats_preparation	1.189	0.274
cross_attention_source_to_dest	5.830	7.854
source_dense_feats_preparation	0.323	0.410
cross_attention_dest_to_source	6.380	11.363
mlp_source_descriptors	0.208	0.468
mlp_dest_descriptors	0.433	0.819
descriptor_normalization	0.108	0.216
similarity_computation	0.076	0.170
best_mask_selection	0.035	0.065
topk_selection	0.037	0.068
full_pred_mask_extraction	0.015	0.036
loss_computation	1.751	0.217
sigmoid_activation	0.025	0.048
total_forward_pass	16.43	22.05
CROSS ATTENTION #1 (Source to Dest)		
norm_input	1.362	1.504
query_projection	0.370	0.089
key_projection	2.088	3.222
value_projection	1.303	2.200
attention_matmul_and_scale	0.321	0.299
attention_softmax	0.053	0.069
attention_weighted_sum	0.099	0.159
output_projection	0.059	0.090
residual_add	0.000	0.000
feedforward_mlp	0.147	0.176
total_context_attn	5.811	7.827
CROSS ATTENTION #2 (Dest to Source)		
norm_input	1.049	0.518
query_projection	0.177	0.307
key_projection	1.876	3.943
value_projection	1.720	3.400
attention_matmul_and_scale	0.816	1.728
attention_softmax	0.090	0.234
attention_weighted_sum	0.249	0.507
output_projection	0.170	0.173
residual_add	0.000	0.000
feedforward_mlp	0.216	0.522
total_context_attn	6.370	11.342
MLP MODULE TIMINGS		
MLP call #1	0.189	0.430
MLP call #2	0.390	0.765
SUMMARY		
Total iteration time	242.26	2467.10
Model forward pass	16.61 (6.9%)	22.09 (0.9%)
Total descriptor extraction	165.80 (68.4%)	1092.24 (44.3%)

Table 7. **Timing comparison: Precomputed vs. On-the-Fly DINOv2.** On-the-fly feature extraction adds significant overhead (10.2× slower overall), primarily due to descriptor extraction (4.6×) and backpropagation (28.2×) costs.

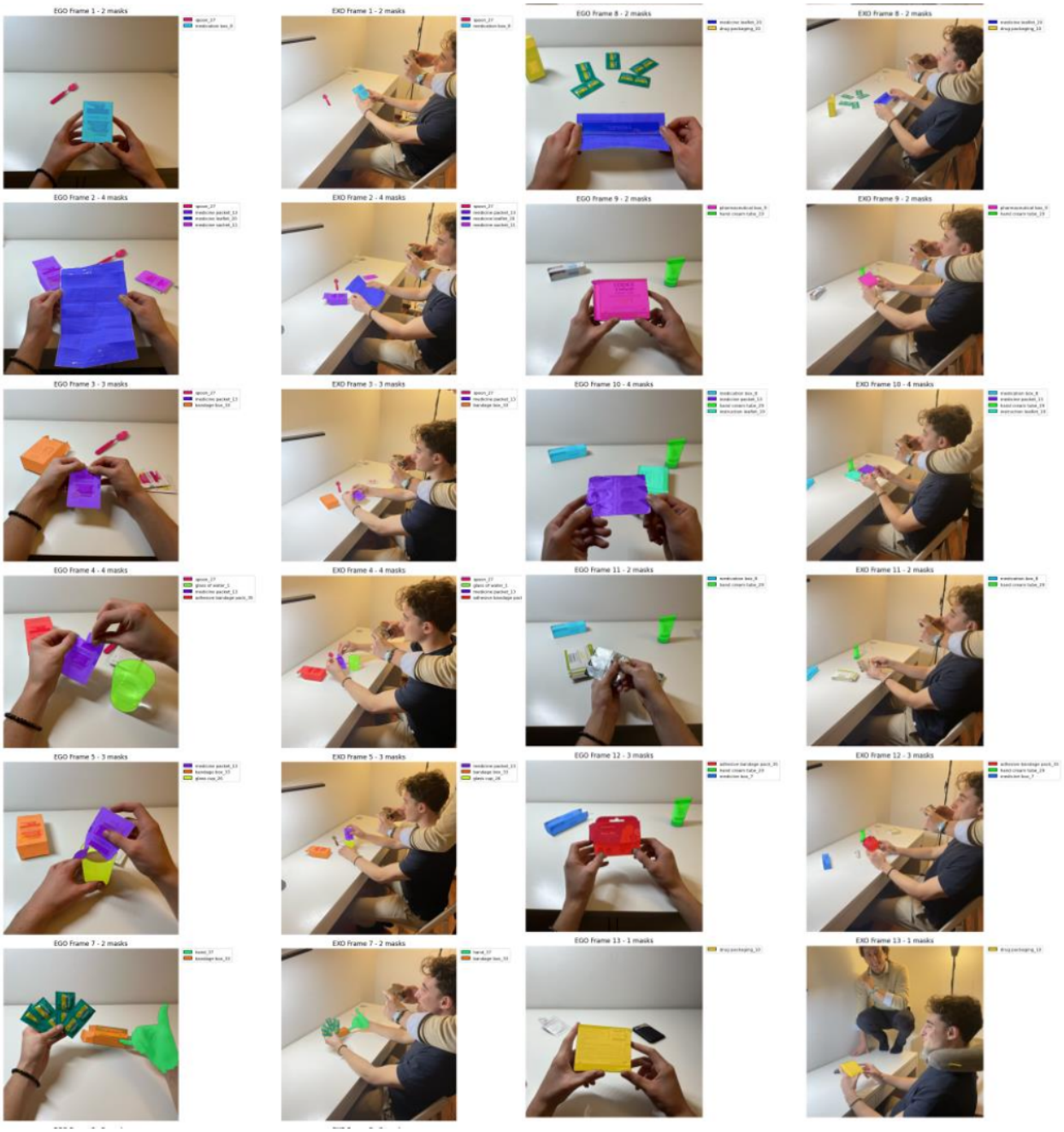


Figure 12. **Samples from our custom occlusion dataset**. Each grid cell shows an egocentric frame with annotated objects under varying occlusion levels (partial, heavy, full). These examples were used to assess robustness.

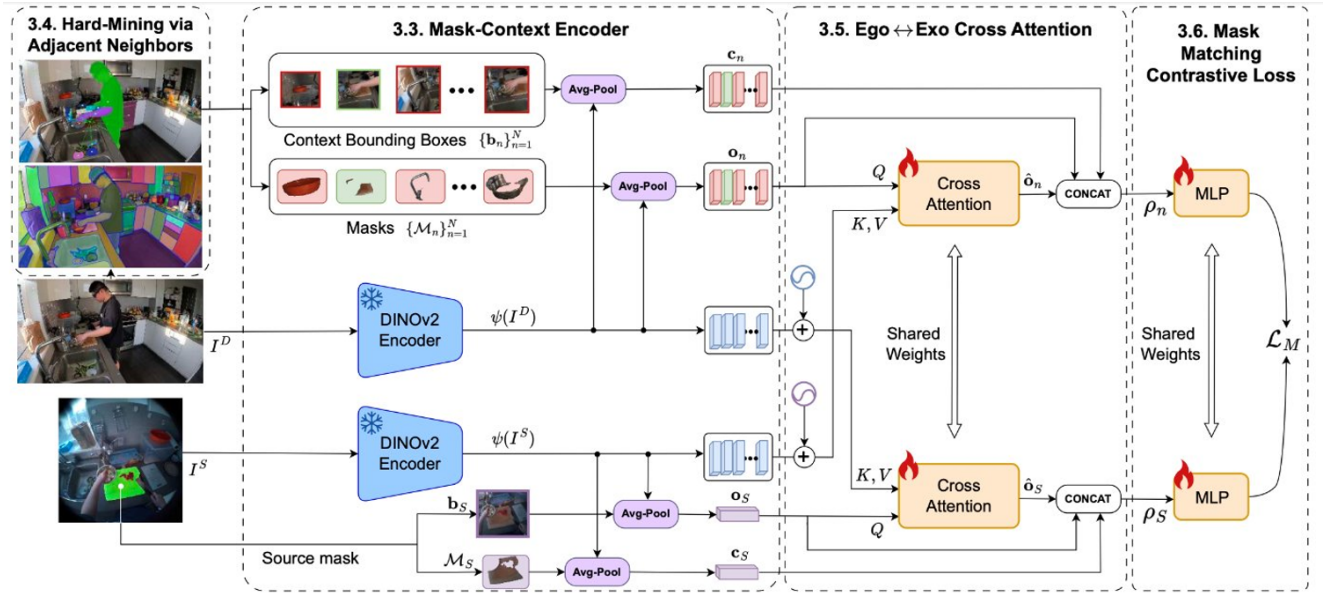


Figure 13. **O-MaMa's model architecture.**

References

[1] Egoexo4d correspondence benchmark processing pipeline. https://github.com/EGO4D/ego-exo4d-relation, 2024.

[2] Xsegtx baseline for ego-exo4d correspondence. https://github.com/EGO4D/ego-exo4d-relation/tree/main/correspondence/SegSwap, 2024.

[3] Xview-xmem baseline for ego-exo4d correspondence. https://github.com/EGO4D/ego-exo4d-relation/tree/main/correspondence/XMem, 2024.

[4] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025.

[5] Xinlei Chen and Kaiming He. Improved baselines with momentum contrastive learning. In *arXiv preprint arXiv:2003.04297*, 2020.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *Int. Conf. Learn. Represent.*, 2021.

[7] Yuqian Fu et al. Objectrelator: Learning object relations between egocentric and exocentric views. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.

[8] Kristen Grauman et al. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[10] Alexander Kirillov, Eric Mintun, Nikhila Ravi, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[11] Tsung-Yi Lin et al. Microsoft coco: Common objects in context. In *Proc. Eur. Conf. Comput. Vis.*, 2014.

[12] Lorenzo Mur-Labadia, Maria Santos-Villafranca, Jesus Bermudez-Cameo, Alejandro Perez-Yus, Ruben Martinez-Cantin, and Jose J. Guerrero. O-mama: Learning object mask matching between egocentric and exocentric views. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2025.

[13] Maxime Oquab et al. Dinov2: Learning robust visual features with self-supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[14] Maxime Oquab et al. Dinov3: Scaling self-supervised learning. *arXiv preprint arXiv:2508.10104*, 2025.

[15] Chenhongyi Zhao et al. Fastsam: High-speed segment anything model. *arXiv preprint arXiv:2306.12156*, 2023.

[16] Ege Özsoy, Arda Mamur, Felix Tristram, Chantal Pellegrini, Magdalena Wysocki, Benjamin Busam, and Nassir Navab. Egoexor: An ego-exo-centric operating room dataset for surgical activity understanding, 2025.