Exo-Ego Correspondence

A Technical Exploration of the State of The Art

20600 – Deep Learning for Computer Vision,
Bocconi University

The ReLUminati

Marco Lomele, Giovanni Mantovani, Filippo Dario Paolucci

Problem Overview

Ego-Exo 4D: world's largest first person to third person video dataset.

Correspondence task: predict the object mask in one viewpoint given a query mask from the other synchronized view.

Difficulties: different PoV, scale variation, occlusion, domain shift.

Exo-centric PoV

Ego-centric PoV

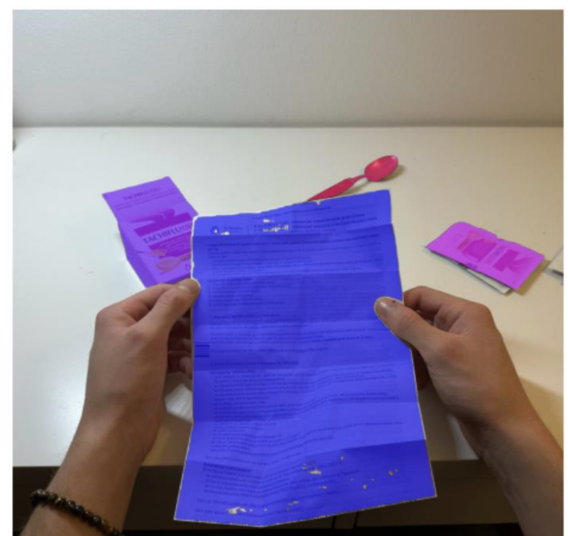
Project Objective

SOTA: Object Masks Matching (O-MaMa), paradigm change, highest accuracy using 1% of parameters w.r.t. competition.

Objective: technical exploration of O-MaMa by experimenting with complementary techniques and analyzing performance impact.

Our focus:

Exo (source) → *Ego (destination)*



Data Overview

- Health scenario
- 20.3 hours of videos
- 299 unique scenes (takes)
- Each take 1 ego camera and 2-6 exo cameras, fully synchronized
- Each camera records 5-20 minutes HD video (500MB - 2GB)

Data Extraction

Step 1: download

For each take from Health scenario:

- download videos + annotations
- extract annotated frames
- apply downscale (ego 2x, exo 4x)
- decode masks from LZString → COCO RLE
- **obtain annotation.json**

Step 2: create pairs

- For each annotated frame of each take define tuple

(ego_rgb, ego_mask, exo_rgb, exo_mask)

- Export pairs to JSON files (train/val/test_exoego_pairs.json)

Note

Focusing on 30% of frames, randomly sampled, **assuming data distribution and object-scene variety represented.**

Resulting frame count per split:

- Train (70%): 9100
- Validation (15%): 1950
- Test (15%): 1950

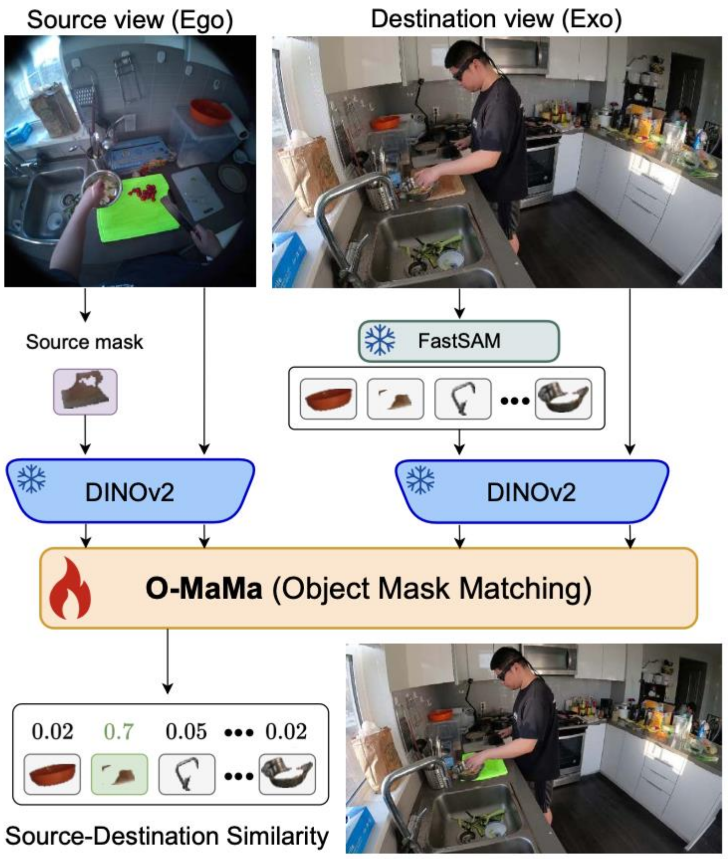

```
"masks": {
  "sterile swab with package_0": {
    "cam04": {
      "4500": {
        "size": [
          2160,
          3840
        ],
        "counts": "ijhV41_S2101N2010002M3N2N3M2M200010З
      }
    },
    "aria02_214-1": {
      "390": {
        "size": [
          1408,
          1408
        ],
        "counts": "[YZQ17h[12N7J6J5K6J6J5K6J6J5K6J6J5K6
      },
      "420": {
        "size": [
          1408,
          1408
        ],
        "counts": "m^nl03h[17I6K5H9L4L3L301N3M2N2O2L3M3
      },
      "450": {
        "size": [
          1408,
          1408
        ],
        "counts": "djbm04g[17QmN9lg0EcPO`1U6WOgh0LWPOk1
      },
```


annotation.json

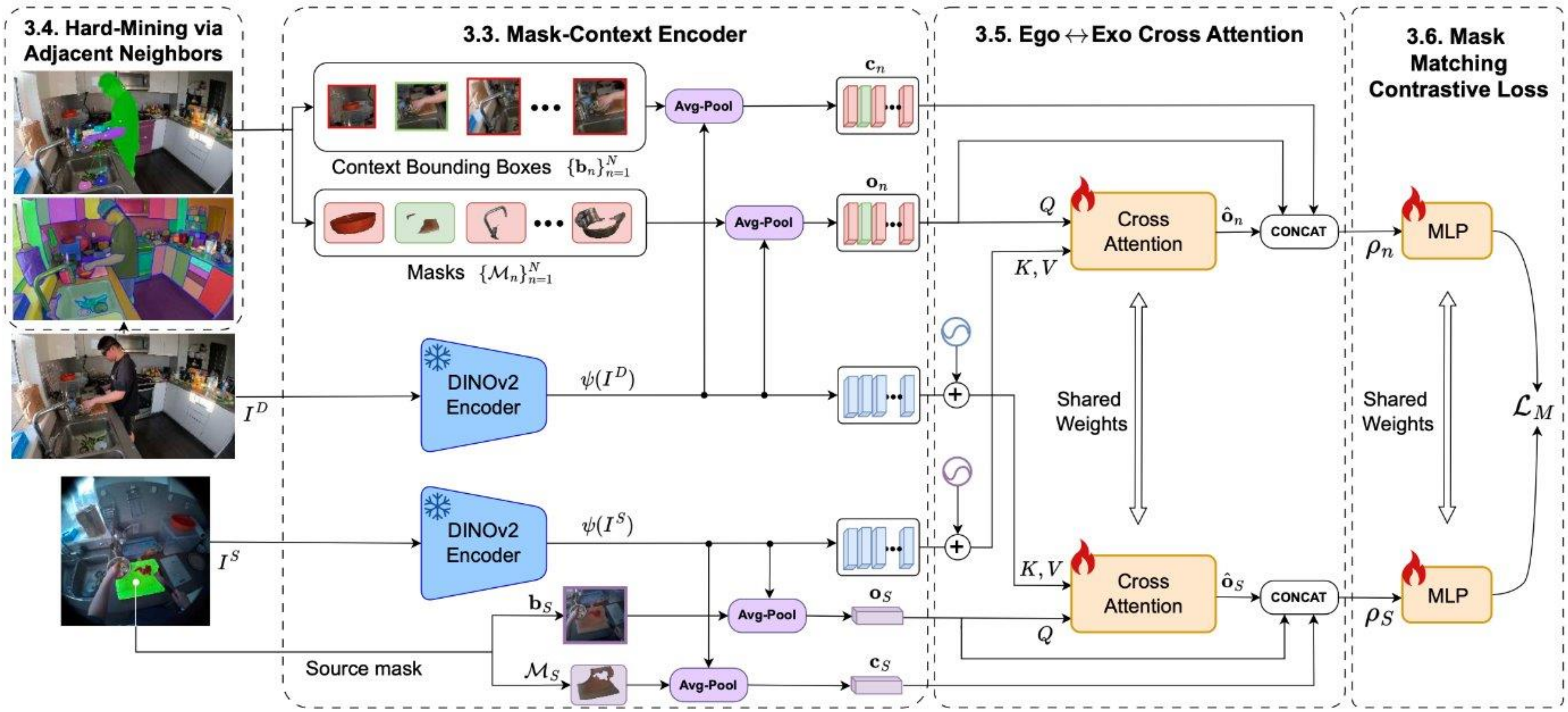
Object Mask Matching (O-MaMa)

Key idea: Reformulate cross-view segmentation as **Object Mask Matching**.

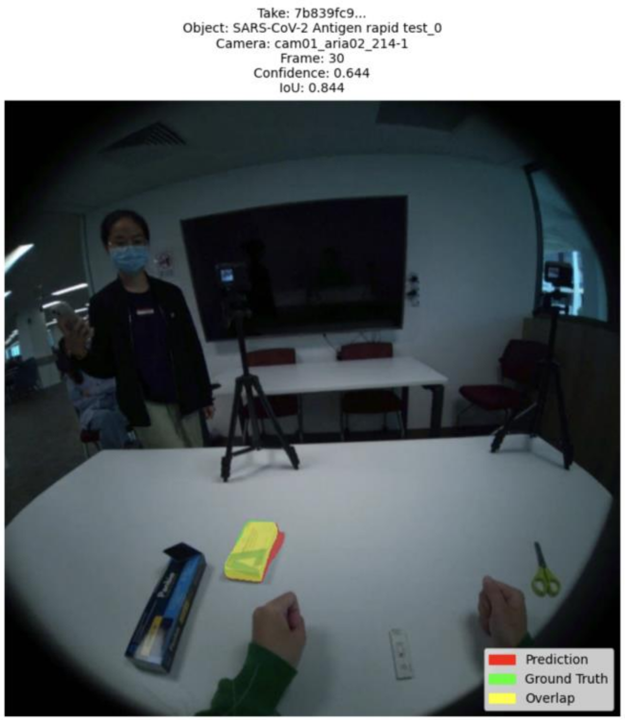
- Use **FastSAM** to generate mask candidates in the destination (Exo).
- Extract semantic features with **DINOv2** from both views.
- Compare source mask embedding with candidate masks.
- Select the **best matching** mask via contrastive similarity.



Architecture details



What are we trying to achieve?



Easy case, good result

Hard case, mid result

Hard case, bad result

Experiments Roadmap

O-MaMa **baseline weights**

VS

O-MaMa **fine-tuned weights**

(lr=8e-6, 10 epochs)

(1) Time constraint

(**20 hours** for 3 fine-tuning epochs)

Feature pre-extraction

(2) Poor performance

(**Lower metrics** post fine-tuning)

Model	IoU	IoU Std
O-MaMa Baseline	**0.533**	**0.408**
O-MaMa Fine-Tuned	0.526	0.414

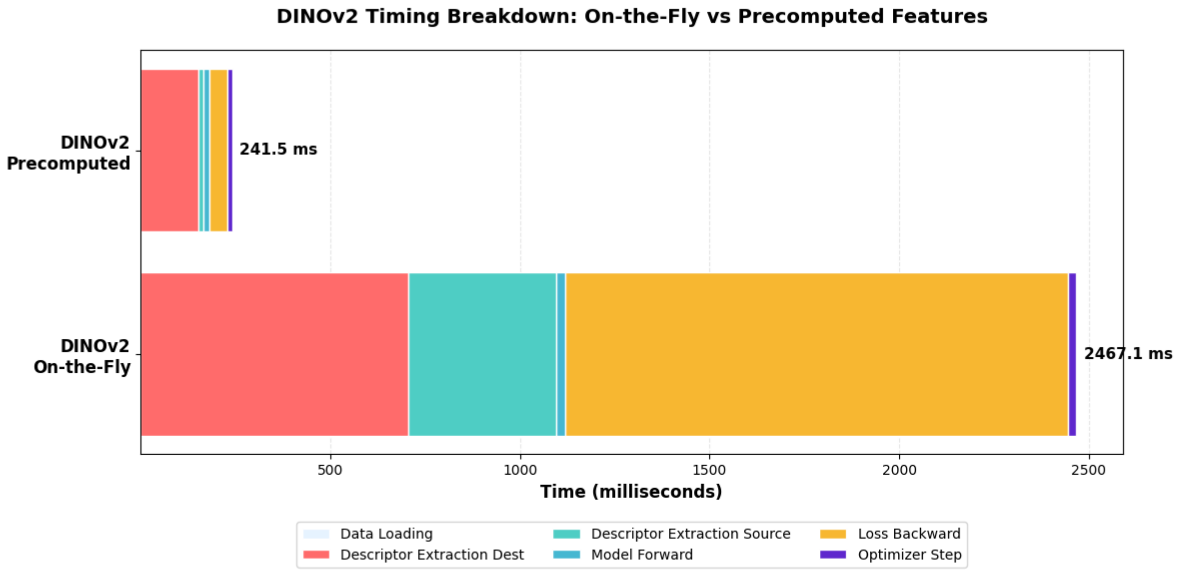
Different representations

(1) Time constraint

Culprit
DinoV2 is called for feature extraction for every camera view of any frame

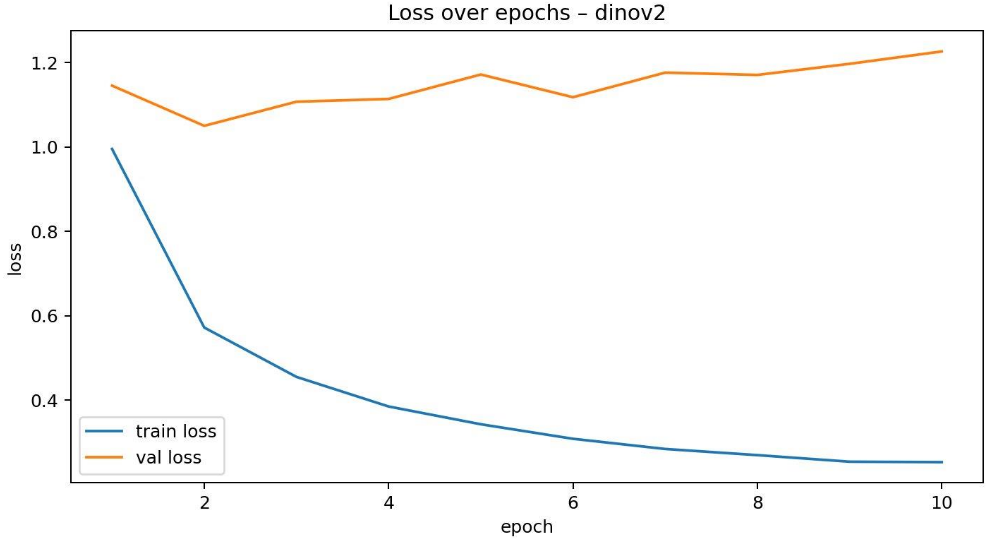
Solution
pre-extract the feature maps with DinoV2 **before** finetuning

Next: finetune the model **without** the feature extraction step.



(2) Poor performance
ViT-B/14 distilled-DinoV2

Run fine-tuning for 10 epochs with learning rate=8e-6.



Finetuning on our training sets results in a high degree of overfitting, **without significant improvement** in validation loss.

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

Decrease in IoU after finetuning on many epochs.

CAVEAT: we are comparing a model trained over 10 epochs vs a model trained over 3 epochs.

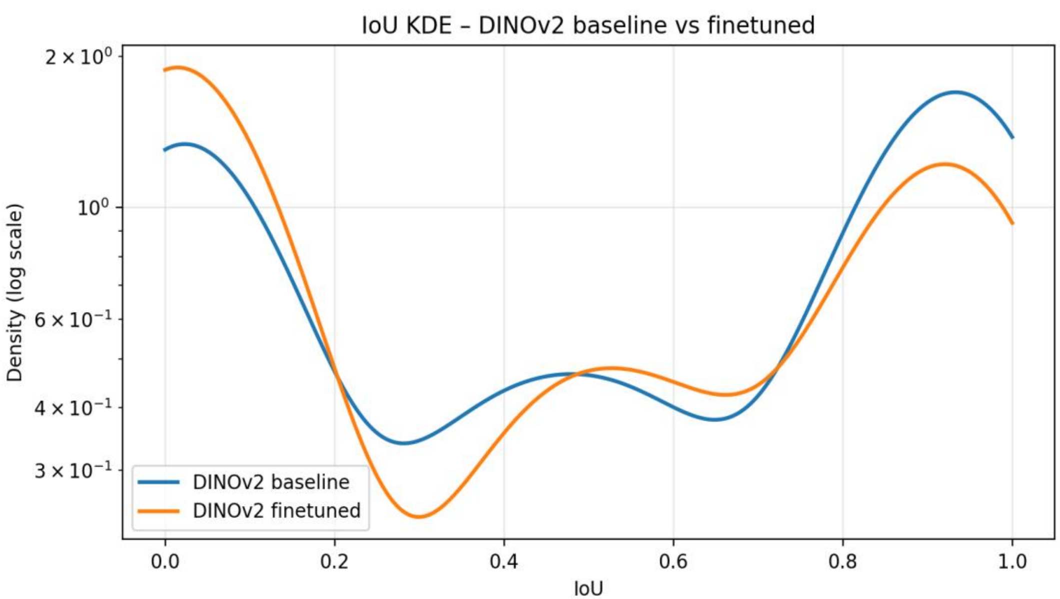
(2) Poor performance
ViT-B/14 distilled-DinoV2

Run fine-tuning for 10 epochs with learning rate=8e-6.

O-MaMa **baseline weights**, pre-extracted features

VS

O-MaMa **finetuned weights**, pre-extracted features



Counterintuitive result: finetuning worsen the overall IoU metric.

Different representations

Poor performance could be explained by the pre-extracted features themselves.

Alternatives:

ViT-B/14 distilled-DinoV2

86 mln parameters

Transformers-based architecture

No feature projection

↓

Current model

ViT-S+/16 distilled-DinoV3

29 mln parameters

Transformers-based architecture

Feature projection: 384 -> 768 with PyTorch's Conv2D

↓

SOTA, we expect more expressive features

ResNet50-DinoV1

23 mln parameters

CNN-based architecture

Feature projection: 2048 -> 768 with PyTorch’s Conv2D

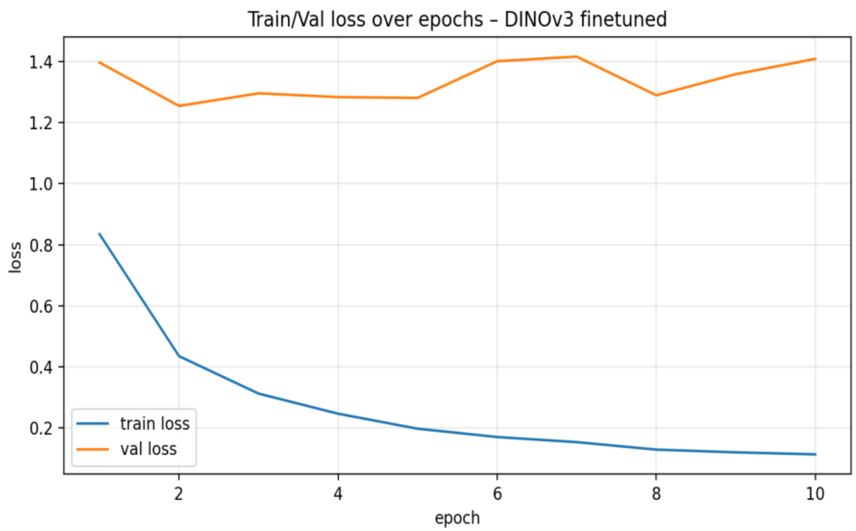
↓

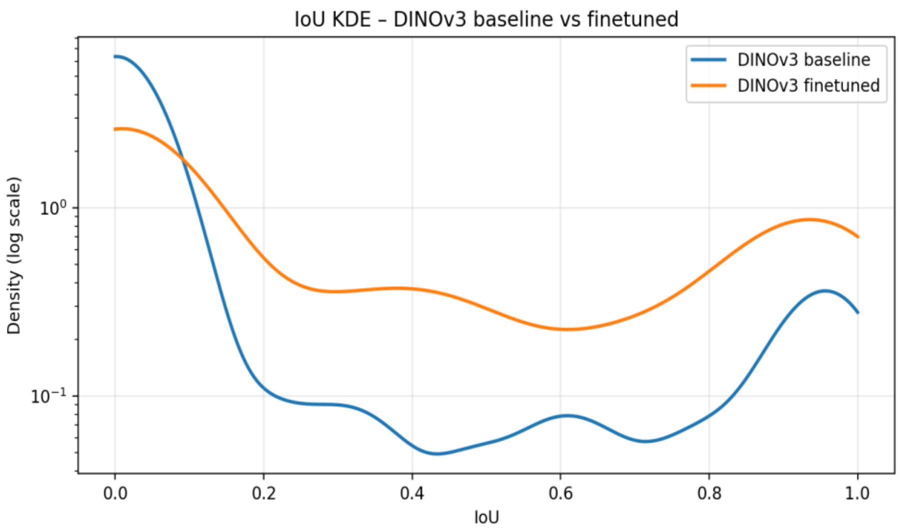
Legacy model and simpler backbone architecture, we expect lower detail features

Comparing performance

ViT-S+/16 distilled-DinoV3

Run fine-tuning for 10 epochs with learning rate=8e-6.



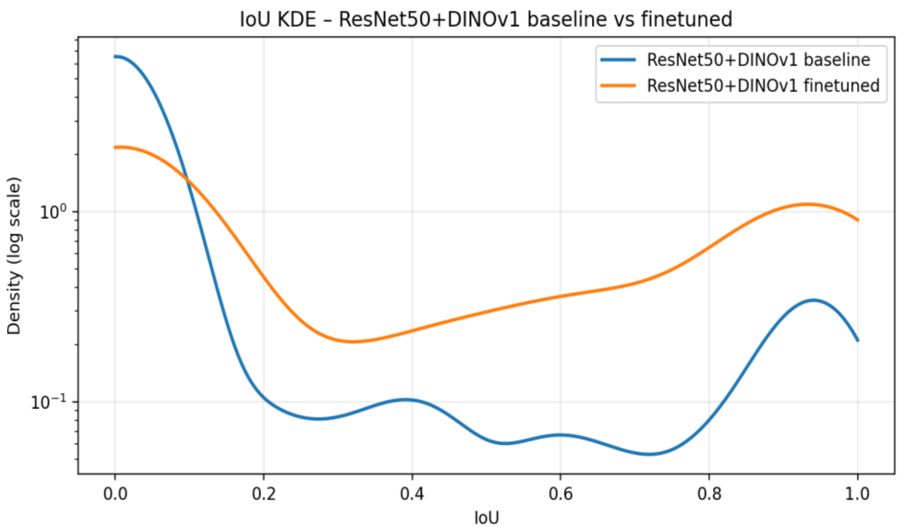


Similar pattern to DinoV2: tendency to overfit & no significant change on the validation loss.

Opposite pattern to DinoV2: O-MaMa with baseline weights clearly underperforms.

Comparing performance
ResNet50-DinoV1

Run fine-tuning for 10 epochs with learning rate=8e-6.



Similar pattern to DinoV2: tendency to overfit & no significant change on the validation loss.

Opposite pattern to DinoV2: O-MaMa with baseline weights clearly underperforms.

Comparing performance

Summary

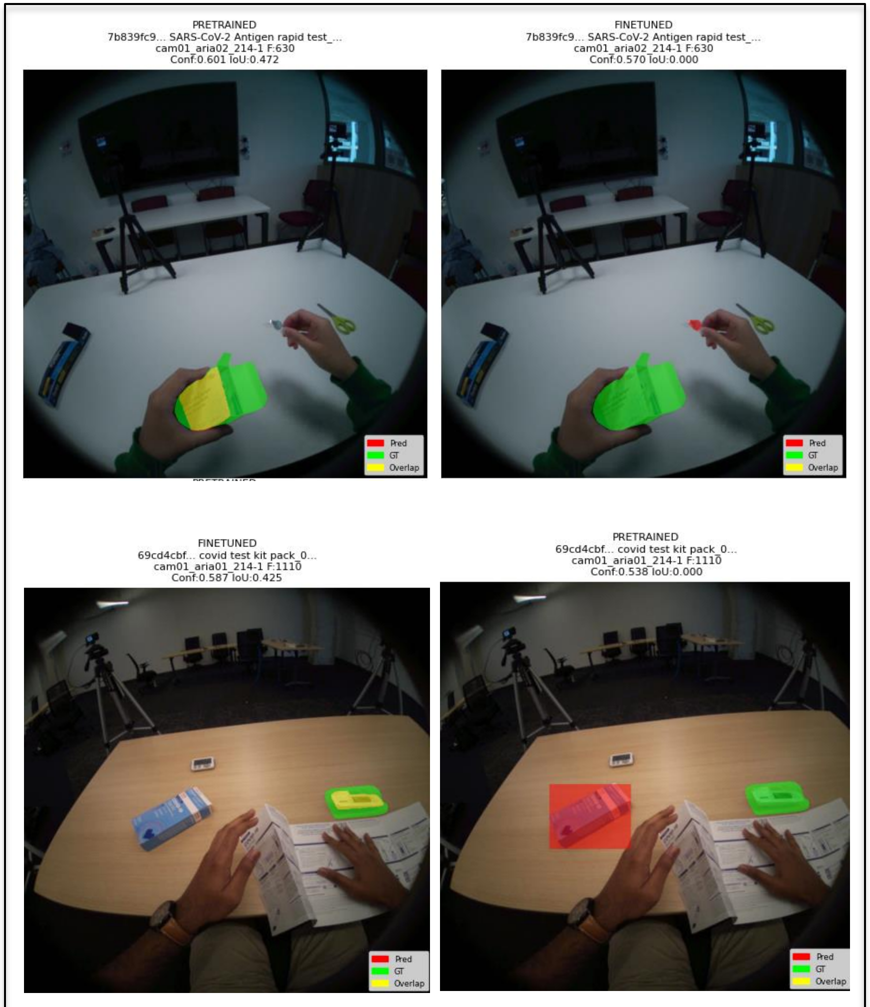
Model	IoU	IoU Std
DINOv2 baseline	**0.5262**	0.4036
DINOv2 finetuned	0.4108	0.4044
DINOv3 baseline	0.0791	0.2405
DINOv3 finetuned	0.2897	0.3819
ResNet50 + DINOv1 baseline	0.0757	**0.2315**
ResNet50 + DINOv1 finetuned	0.3647	0.4140

Conclusions:

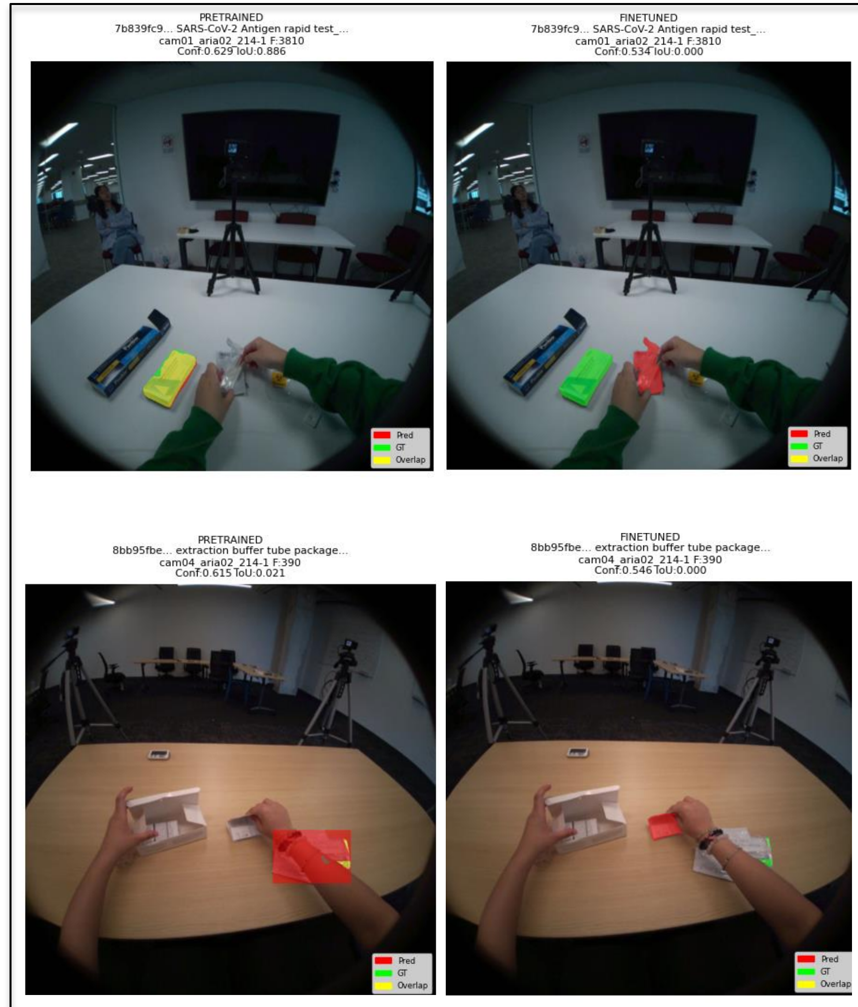
- original set-up with DinoV2 outperforms other representation models.
- scenario-finetuned models outperform baseline models when there is projection.

Failure cases

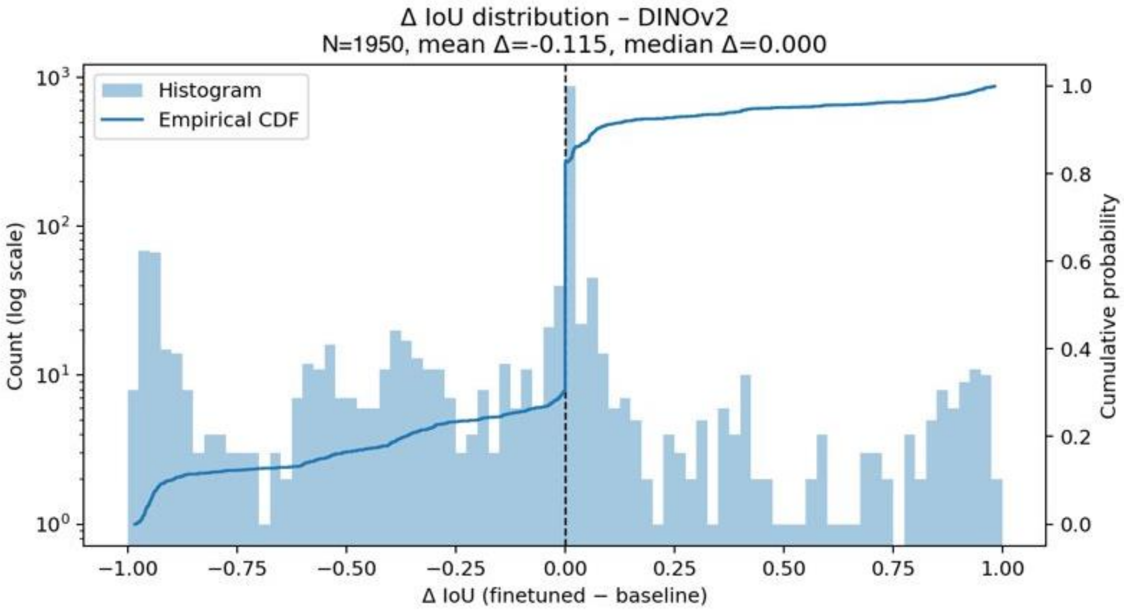
Over-specificity



What's being manipulated



Troubleshooting



Hypotheses:

- Failure of data distribution and object-scene variety assumption in sample => train sample too restrictive, then tested on unseen objects, shortcutting and spurious learning.
- Contrastive loss function with minimal object variety => feature space collapse.
- From generality to specificity on health => catastrophic forgetting.

Bonus: Our data

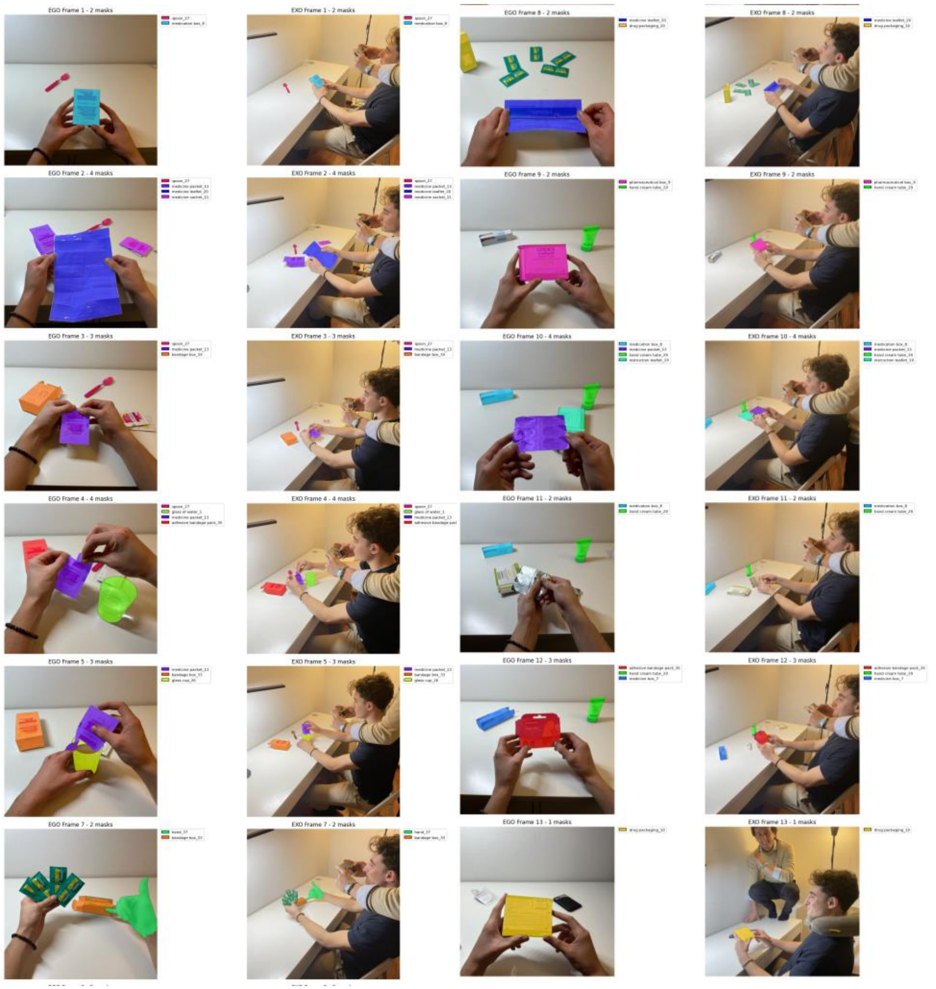
Selected 13 images with 32
1 ego-exo object pairs

SAM 3:
GT masks

FastSAM:
Masks for inference

Same Exo-Ego data
pipeline steps

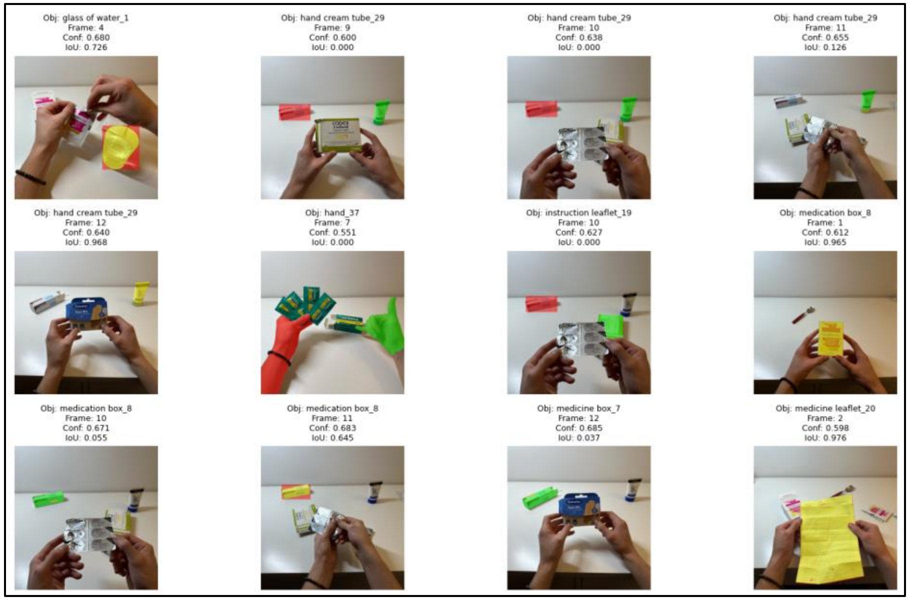
Inference with the best models:
O-MaMa DinoV2 Baseline
O-MaMa DinoV2 Finetuned



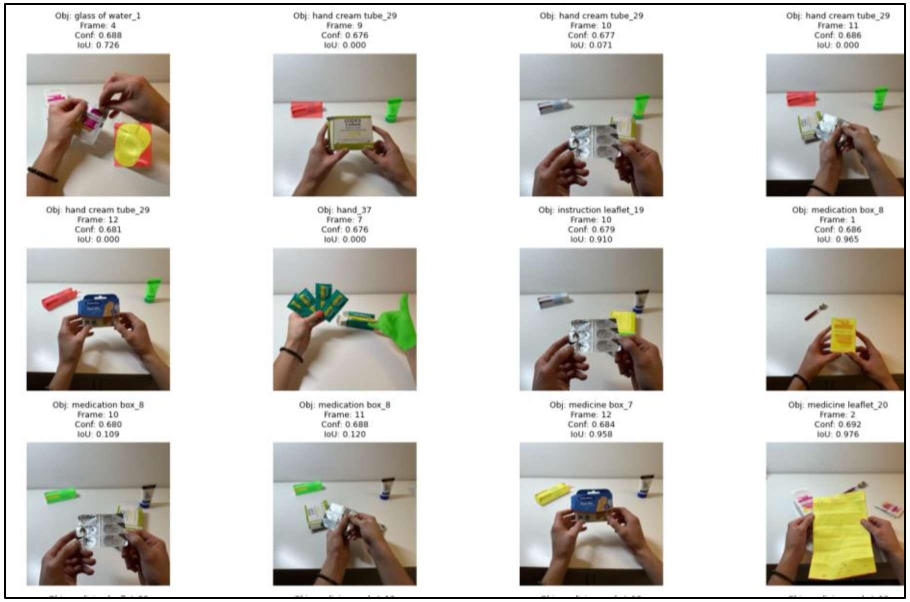
Bonus: Our data

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

DINOv2 baseline:



DINOv2 finetuned:



Conclusion and future work

All finetuning experiments and change of feature extractor defied the assumptions that we formulated.

Finetune worsen the IOU metric

DinoV3 underperforms, ResNet50 overperforms

Deeper analysis on the inherent structure of the model:

- Sampling at the object-level to ensure representative samples
- Mask temporal consistency (IOU for a selected object over sequential frames)
- Test it on novel datasets (EgoExOR)

Thank you for your attention!

EGO Frame 13 - 1 masks

EXO Frame 13 - 1 masks